Universität Rostock Traditio et Innovatio

Zur Komplexität der Synthese von Petri-Netzen

Dissertation

zur

Erlangung des akademischen Grades

Doktor-Ingenieur (Dr.-Ing.)

der Fakultät für Informatik und Elektrotechnik

der Universität Rostock

vorgelegt von

Ronny Tredup geboren am 24. April 1983 in Wolfen

aus Rostock

Rostock, 8. November 2021

https://doi.org/10.18453/rosdok_id00003375

Gutachter:
Professor Dr. Karsten Wolf, Institut für Informatik, Universität Rostock
Professor Dr. Rober Lorenz, Institut für Informatik, Universität Augsburg

eingereicht am 10. Mai 2021

verteidigt am 21. Oktober 2021

Danksagung

Zunächst danke ich Karsten Wolf für die Gelegenheit am Lehrstuhl für Theoretische Informatik arbeiten zu dürfen sowie für seine Unterstützung und das außergewöhnliche Ausmaß an gewährter Freiheit bezüglich Themenwahl und Forschung.

Ganz besonders dankbar bin ich meinem Co-Autor Christian Rosenke für seine frühe und wertvolle Unterstützung, seinen nützlichen Rat bezüglich vieler Dinge rund um das Thema Promotion sowie für die konstruktive und ertragreiche Zusammenarbeit, durch die ich viel lernen konnte.

Uli Schlachter danke ich sehr für die vielen Diskussionen, seine Kritik und insbesondere für das Korrekturlesen der Dissertationsschrift und die daraus resultierenden Hinweise und Verbesserungsvoschläge.

Ich danke meinen Co-Autoren Evgeny Erofeev und Sophie Wallner für die freundliche Zusammenarbeit.

Ganz besonders möchte ich Anne Zielske danken: danke für deine unermüdliche Unterstützung auch während der schwierigsten Phasen der Promotion; danke für deinen wertvollen Rat; danke für den Freiraum zum Arbeiten, den du mir zum Beispiel durch die Betreuung unserer Tochter immer wieder verschafft hast. Danke Anne!

Zusammenfassung

Petri-Netz-Synthese für einen Netztyp τ ist die Aufgabe, für ein reguläres Verhalten, das als Transitionssystem A gegeben ist, zu entscheiden, ob es ein Petri-Netz N vom Typ τ (ein τ-Netz) gibt, das A implementiert. Je nach Grad der Genauigkeit wird N einer der folgenden drei *Implementierungen* zugeordnet: eine *Einbettung* bewahrt die Verschiedenheit der Zustände von A; eine *Sprachsimulation* simuliert A bis auf Sprachäquivalenz und eine *Realisierung*, realisiert das durch A spezifizierte Verhalten exakt. Daraus resultieren die *klassischen* Synthese-Probleme τ-*Einbettung*, τ-*Sprachsimulation* und τ-*Realisierung*, die fragen, ob A durch ein τ-Netz entsprechend implementierbar ist.

Diese Dissertation liefert eine vollständige Charakterisierung der Komplexität ebendieser Entscheidungsprobleme für die sogenannten *(puren) b-beschränkten P/T-Netze* sowie deren *Erweiterungen um die Gruppe* $\mathbb{Z}_{b+1}$ sowie von τ-*Einbettung* für die sogenannten *booleschen* Netze. Darüber hinaus wird die Komplexität von τ-*Sprachsimulation* und τ-*Realisierung* für eine erhebliche Teilmenge der booleschen Netztypen charakterisiert, die unter anderem alle bisher in der Literatur explizit diskutierten Netztypen umfasst.

Die Forderung, dass die Anzahl von Transitionen in der Umgebung jedes Platzes des implementierenden Netzes durch eine gegebene Zahl κ begrenzt ist, berücksichtigt, dass Ressourcen des Systems (Plätze) nur von wenigen Akteuren (Transitionen) abhängen sollen. In dieser Arbeit untersuchen wir für Typen der boolesche Netze und für die P/T-Netze die klassische und parametrisierte Komplexität der *Synthese von Netzen mit begrenzten Platz-Umgebungen*, die auf Sprachsimulation und Realisierung abzielt, wobei κ als Parameter gewählt wird.

Die Forderung, dass für jeden Platz die Anzahl der erreichbaren Markierungen des Netzes, in denen der Platz von einer Marke belegt ist, durch eine gegebene Zahl ϱ begrenzt ist, berücksichtigt, dass lokale Eigenschaften (Plätze) nur selten im Gesamtzustand des Systems (Markierung) erfüllt sein sollen. In dieser Dissertation zeigen wir für alle Implementierungen, dass die *Synthese boolescher Netze mit begrenzter Belegungszahl* fest-Parameter-berechenbar ist, wenn ϱ als Parameter gewählt wird.

Unabhängig von Netztyp und Implementierung ist nicht jedes Transitionssystem A implementierbar. Verschiedene Methoden streben an, A zum Zwecke der Implementierbarkeit möglichst geringfügig zu modifizieren: *Neubeschriftung* basiert auf der *Teilung von Ereignissen*, wobei ursprünglich mit demselben Ereignis beschriftete Kanten nun mit aus der Teilung entstandenen Ereignissen neu beschriftet werden; die resultierende Anzahl von Ereignissen des modifizierten Transitionssystems soll dabei eine gegebene Zahl κ nicht übersteigen. *Kanten-* und *Ereignis-* sowie *Zustandslöschung* fragen für A und eine Zahl κ, ob A durch das Löschen von höchstens κ Kanten, Ereignissen oder Zuständen, je nachdem, zu einem implementierbaren Transitionssystem modifiziert werden kann.

Die vorliegende Arbeit präsentiert eine vollständige Charakterisierung der Komplexität von *Neubeschriftung*, *Kantenlöschung*, *Ereignislöschung* und *Zustandslöschung* für jede der Implementierungen Einbettung, Sprachsimulation und Realisierung, die auf (pure) P/T-Netze abzielen.

Abstract

Petri net synthesis for a *type of nets* τ is the task of deciding, for a regular behavior, which is given as a transition system A, whether there exists a Petri net N of type τ (a τ-net) that implements A. Depending on the degree of accuracy, N corresponds to one of the following three *implementations*: an *embedding* preserves the distinctness of states; a *language simulation* simulates A up to language equivalence; and a *realization* realizes exactly the behavior specified by A. This gives rise to the *classical* problems of synthesis *τ-embedding*, *τ-language simulation* and *τ-realization*, which ask whether A can be implemented accordingly by a τ-net.

This thesis presents a complete characterization of the complexity of these same decision problems for the so-called *(pure) b-bounded P/T-nets* as well as their *extensions by the group $\mathbb{Z}_{b+1}$* and of *τ-embedding* for the so-called *Boolean nets*. Moreover, the complexity of *τ-language simulation* and *τ-realization* is characterized for a substantial subset of Boolean net types, which includes, among others, all net types explicitly discussed in the literature so far.

The requirement that the number of transitions in the environment of each place of the implementing net is restricted by a given number κ takes into account that resources of the system (places) should depend on only a few agents (transitions): In this thesis, we study the classical and parameterized complexity of *synthesis of nets with restricted place environments*, which aims at language simulation and realization, for Boolean nets and for the P/T nets, where κ is chosen as the parameter.

The requirement that for each place the number of reachable markings of the net in which the place is occupied by a token should not exceed a given number ϱ takes into account that local properties (places) should rarely be satisfied in the global state of the system (marking): In this work, we show for all implementations that *synthesis of Boolean nets with restricted number of occupancies* is fixed parameter tractable when ϱ is chosen as the parameter.

Regardless of net type and implementation, not every transition system A is implementable. Various methods aim for modifying A as little as possible for the purpose of implementability: *relabeling* is based on *event splitting*, where edges originally labeled with the same event are now relabeled with events resulting from the splitting; the resulting number of events of the modified transition system shall not exceed a given number κ; *edge-* and *event-* as well as *state-deletion* ask for A and a number κ whether A can be modified to an implementable transition system by deleting at most κ edges, events, or states, as the case may be.

This thesis provides a complete characterization of the complexity of *relabeling*, *edge deletion*, *event deletion*, and *state deletion* for each of the implementations embedding, language simulation, and realization, which aim for (pure) P/T-nets.

Inhaltsverzeichnis

1 Einleitung **1**
1.1 Einführung in die Thematik und Motivation 1
1.2 Literaturdiskussion 9
1.3 Struktur der vorliegenden Arbeit 11

2 Grundlagen und Vorbereitungen **15**
2.1 Transitionssysteme, Netztypen und Netze 15
2.2 Synthese: Von Transitionsystemen zu Netzen 19
2.2.1 Implementierungen 19
2.2.2 Regionen 23
2.2.3 Trennungseigenschaften 25
2.3 Kongruenz von Netztypen 27
2.4 Lineare Transitionssysteme 31
2.5 Komplexität 32
2.5.1 Klassische Komplexität 32
2.5.2 Parametrisierte Komplexität 34

I Klassische Syntheseprobleme **37**

3 Drei Entscheidungsprobleme und ein Reduktionsschema **39**
3.1 Einbettung und Sprachsimulation sowie Realisierung 39
3.2 Ein Reduktionsschema 40
3.3 Ein ausführliches Reduktionsbeispiel 42
3.4 Zusammenfassung 49

4 Boolesche Netze **51**
4.1 Sprachsimulation und Realisierung für boolesche Netztypen mit nop 55
4.1.1 Die Netztypen A1-A3 57
4.1.2 Die Netztypen A4 59
4.1.3 Die Netztypen A6 62
4.1.4 Die Netztypen A7-A8 66
4.1.5 Die Netztypen A9 72

4.1.6 Die Netztypen A10 . . . 77
4.1.7 Die Netztypen A11 . . . 82
4.2 Realisierung für Boolesche Netztypen ohne nop und swap . . . 82
4.2.1 Netztypen mit res und set . . . 83
4.2.2 Netztypen mit set aber ohne res und free . . . 86
4.2.3 Netztypen mit set und free aber ohne res . . . 88
4.2.4 Netztypen ohne res und set . . . 92
4.3 Einbettung für Boolesche Netztypen . . . 94
4.3.1 Die Netztypen B1 . . . 96
4.3.2 Die Netztypen B2 - B3 . . . 99
4.3.3 Die Netztypen B4 - B6 . . . 99
4.3.4 Die Netztypen B7 . . . 103
4.3.5 Die Netztypen B8 . . . 103
4.3.6 Die Netztypen B9 - B10 . . . 114
4.4 Zusammenfassung und Ausblick . . . 114

5 A priori beschränkte Platz/Transitions-Netze 117
5.1 b-beschränkte Netztypen . . . 124
5.1.1 τ^b_{PPT}-Sprachsimulation und τ^b_{PPT}-Realisierung . . . 127
5.1.2 τ^b_{PT}-Sprachsimulation und τ^b_{PT}-Realisierung . . . 132
5.1.3 τ^b_{PPT}-Einbettung und τ^b_{PT}-Einbettung . . . 135
5.2 $\mathbb{Z}_{b+1}$-erweiterte b-beschränkte Netztypen . . . 137
5.2.1 Sprachsimulation und Realisierung für gruppenerweiterte b-beschränkte Netze mit $b \geq 2$ . . . 137
5.2.2 Einbettung für gruppenerweiterte b-beschränkte Netze . . . 144
5.2.3 Der Sonderfall $\mathbb{Z}_2$-erweiterter 1-beschränkter Netztypen . . . 149
5.3 Zusammenfassung, Konsequenzen und Ausblick . . . 151

II Parametrisierte Syntheseprobleme 155

6 Boolesche Netze mit begrenzten Platzumgebungen 157
6.1 W[1]-Schwere durch Reduktion von Regular Independent Set . . . 161
6.2 W[1]-Schwere durch Reduktion von Odd Set . . . 174
6.3 W[2]-Schwere durch Reduktion von Hitting Set . . . 189
6.4 Zusammenfassung und Ausblick . . . 203

7 Platz/Transitions-Netze mit begrenzten Platzbereichen 205
7.1 τ_{PT}-SMBPB und τ_{PT}-RMBPB sind in NP . . . 209
7.2 τ_{PT}-SMBPB und τ_{PT}-RMBPB sind NP-schwer . . . 213
7.3 τ_{PT}-SMBPB und τ_{PT}-RMBPB mit Parameter $\varrho + \kappa$ . . . 219
7.4 Zusammenfassung und Ausblick . . . 220

8 Boolesche Netze mit beschränkter Belegungszahl 223
8.1 Ein parametrisierter Algorithmus 225
8.2 Anwendung des Algorithmus an einem Beispiel 234
8.3 Zusammenfassung und Ausblick 237

III Verhaltensänderung zum Zwecke der Implementierbarkeit 239

9 Verhaltensmodifikation durch Neubeschriftung 241
9.1 Neubeschriftung für Sprachsimulation und Realisierung 244
9.2 Neubeschriftung für Einbettung 249
9.3 Zusammenfassung und Ausblick 253

10 Verhaltensmodifikation durch Löschung 255
10.1 Kanten-, Ereignis- und Zustandslöschung: Definitionen und Abgrenzung . . 256
10.2 Kantenlöschung für Sprachsimulation und Realisierung 259
10.3 Ereignislöschung 265
10.3.1 Ereignislöschung für Sprachsimulation und Realisierung 266
10.3.2 Ereignislöschung für Einbettung 268
10.4 Zustandslöschung 269
10.4.1 Zustandslöschung für Einbettung und Realisierung 271
10.4.2 Zustandslöschung für Sprachsimulation 273
10.5 Zusammenfassung und Ausblick 274

11 Fazit 277

Literaturverzeichnis 281

Glossar 297

Index 303

Selbstständigkeitserklärung 306

Lebenslauf 307

Kapitel 1

Einleitung

1.1 Einführung in die Thematik und Motivation

Das Modellieren von Systemen und Prozessen gilt als Kernkompetenz der Informatik. Viele dieser Systeme bestehen aus räumlich verteilten Komponenten, die nebenläufig, dass heißt unabhängig von gemeinsamen Ressourcen, arbeiten und so zur Funktionalität des Gesamtsystems beitragen.

Petri-Netze, im Folgenden auch einfach nur kurz *Netze* genannt, und ihre verschiedenen *Typen* sind ein weitläufig anerkanntes und etabliertes Modell zur Modellierung und Analyse diskreter verteilter Systeme und ihres Verhaltens. Das Konzept der Netze geht zurück auf Carl Adam Petri [Pet73] und wurde anschließend insbesondere durch die Beiträge der Gruppe um Anatol Holt [CHEP71, Hol85] weiterentwickelt.

Die *Synthese* von Netzen (eines bestimmten Typs) umfasst die Aufgabe, für eine gegebene Beschreibung eines Verhaltens algorithmisch zu entscheiden, ob dieses Verhalten durch ein Netz (dieses Typs) implementiert werden kann. Im Fall einer positiven Entscheidung soll ein solches Netz konstruiert werden. Dabei liefert eine gültige Syntheseprozedur ein Netz, welches „per Konstruktion“ korrekt ist, sodass dessen Korrektheit nicht nachträglich verifiziert werden muss.

Netz-Synthese hat viele praktische Anwendungsbereiche wie zum Beispiel Hardware-Synthese [CKK+97a, CKK+02] und Supervisory-Control [HKG97, WWZ11] sowie Process Mining [vdA11, BP09] und das Extrahieren von Nebenläufigkeit und Verteilbarkeit in sequentiell vorliegenden Systemen [BCD02, Dar05, DR12].

Ob wir jedoch überhaupt einen Synthese-Algorithmus finden können und wenn ja, ob so ein Algorithmus effizient sein kann, hängt wesentlich von der Art des Eingabeverhaltens und der Art des gesuchten Netzes ab: Für ein modales Transitionssystem ist es im Allgemeinen unentscheidbar, ob es durch ein sogenanntes Platz/Transitions-Netz implementiert werden kann [Sch16]; entscheidbar ist dagegen, ob ein modales Transitionssystem durch ein b-beschränktes Platz/Transitions-Netz implementiert werden kann

und im Fall einer positiven Entscheidung kann ein solches Netz effektiv konstruiert werden [SW17]. Wenn das Eingabeverhalten als Transitionssystem gegeben ist, dann ist das Problem zu entscheiden, ob ein implementierendes 1-beschränktes Platz/Transitions-Netz existiert, NP-vollständig [BBD97]; ob für eine solche Spezifikation ein (pures) beschränktes Platz/Transitions-Netz existiert ist jedoch in Polynomialzeit entscheidbar [BBD95, BD96a]. Diese Fragen an die Berechenbarkeit und Komplexität des Syntheseproblems, insbesondere in Abhängigkeit von Spezifikation und Netzart, sind seit vielen Jahren Gegenstand der Forschung und bis heute aktuell, wie die genannten Arbeiten bereits zeigen.

In dieser Arbeit untersuchen wir schwerpunktmäßig die Komplexität bestimmter entscheidbarer Instanzen der Netz-Synthese sowie eng damit verwandte Probleme, die darauf abzielen, nicht implementierbare Spezifikationen zum Zwecke der Implementierbarkeit zu modifizieren.

Petri-Netze bestehen aus *Plätzen* und *Transitionen* sowie einem *Fluss* und einer *Anfangsmarkierung*. Diese Bestandteile sind den Komponenten des modellierten Systems zugeordnet: Plätze repräsentieren Speicherzellen oder, ganz allgemein, Ressourcen. Sie können *Marken* enthalten und die jeweilige Anzahl der Marken auf einem Platz modelliert einen konkreten Zustand der Ressource im System. Transitionen entsprechen Ereignissen beziehungsweise Aktionen des Systems, die zu (lokalen) Veränderungen von Ressourcen führen. Dabei entspricht das Auftreten eines Ereignisses im System, dem *Schalten* seiner Transition im Netz. Das Auftreten von Ereignissen verteilter Systeme hängt (nur) lokal von bestimmten Ressourcen ab und hat (nur) lokal Auswirkungen auf den Gesamtzustand des Systems. Der Fluss definiert für jedes Paar von Platz und Transition, das heißt Ressource und Ereignis, ob und in welchem Ausmaß sie einander beeinflussen: Muss der Platz eine Mindestanzahl an Marken beinhalten, um das Schalten der Transition zu erlauben und, wenn ja, welche Anzahl ist das? Kann das Schalten der Transition die Markierung des Platzes beeinflussen und, wenn ja, welcher Art ist dieser Einfluss? Wir betrachten einen Platz und eine Transition als *unabhängig voneinander*, wenn diese Fragen negativ zu beantworten sind. Andernfalls sind sie abhängig voneinander und wir sagen, der Platz befindet sich in der Umgebung der Transition und die Transition befindet sich in der Umgebung des Platzes. Dies reflektiert die Lokalitätsbeziehungen von Ereignissen und Ressourcen im modellierten System. Wir bezeichnen zwei Transitionen als (strukturell) *nebenläufig*, wenn ihre jeweiligen Umgebungen disjunkt sind und folgen damit insbesondere dem Nebenläufigkeitsbegriff von Starke [Sta90]. Die Anfangsmarkierung modelliert schließlich den initialen Zustand jeder einzelnen Ressource des Systems und folglich dessen initialen Gesamtzustand.

Petri Netze werden unter anderem über die Anzahl von Marken klassifiziert, die ein Platz des Netzes in erreichbaren Markierungen höchstens enthalten darf. Im einfachsten Fall darf jeder Platz höchstens eine Marke enthalten. Dies erlaubt Plätze als boolesche Variablen zu interpretieren, die wahr sind, wenn sie eine Marke enthalten und andernfalls falsch sind. Solche Netze gehören daher zur Klasse der sogenannten *booleschen* Netze. In diesem Modell ist für Ressourcen offenbar nur interessant, ob sie vorhanden sind. Die

(vorhandene) Anzahl ihrer Einheiten ist nicht modellierbar. Plätze werden dagegen als Zähler interpretiert, wenn sie mehrere Marken enthalten dürfen. Diese Interpretation führt zur Klasse der sogenannten *Platz/Transitions-Netze*, die wir im Folgenden auch kurz *P/T-Netze* nennen. Innerhalb dieser Klasse werden darüber hinaus jeweils die b-beschränkten, beschränkten und unbeschränkten P/T-Netze voneinander unterschieden: im ersten Fall gilt für eine im Voraus fest gewählte natürliche Zahl b, dass jeder Platz höchstens b Marken enthalten darf, im zweiten Fall gibt es eine (nicht notwendigerweise im Voraus bekannte) Zahl b, welche die maximale Anzahl der Marken auf einem Platz beschränkt und im dritten Fall kann ein Platz unbeschränkt viele Marken enthalten.

In der vorliegenden Arbeit widmen wir uns im Wesentlichen nur der Synthese boolescher Netze, b-beschränkter und beschränkter P/T-Netze sowie einigen ihrer „elementaren“ Erweiterungen. Insbesondere werden gefärbte Netze oder Netze mit Zeit [Sta90], die „nicht-elementare“ Erweiterungen darstellen, nicht diskutiert.

Des Weiteren werden Netze über die Fluss-Beziehungen klassifiziert, die zwischen Transitionen und Plätzen potentiell möglich sind. Diese Beziehungen definieren, inwiefern sich das Schalten von Transitionen und die Markierungen von Plätzen gegenseitig beeinflussen können. Sie umfassen unter anderem das *Konsumieren* und das *Produzieren* von Marken auf Plätzen, das *Testen* auf *Anwesenheit* und auf *Abwesenheit* von Marken auf Plätzen oder auch das *Rechnen modulo einer ganzen Zahl* sowie insbesondere die *Unabhängigkeit* zwischen Transitionen und Plätzen.

Innerhalb der Klasse der booleschen Netze gibt es zum Beispiel genau acht mögliche Fluss-Beziehungen zwischen Plätzen und Transitionen, die *boolesche Interaktionen* genannt werden, nämlich nop (Unabhängigkeit), inp (bedingtes Konsumieren), out (bedingtes Produzieren), res (unbedingtes Konsumieren), set (unbedingtes Produzieren), swap (Rechnen modulo 2), used (Test ob markiert) und free (Test ob unmarkiert). Für eine Menge τ boolescher Interaktionen nennt man ein boolesches Netz ein *τ-Netz*, wenn seine Flusswerte höchstens Elementen aus τ entsprechen. Eine vollständige Diskussion der einzelnen Interaktionen und die Argumentation, dass sie tatsächlich alle relevanten Flussbeziehungen boolescher Netze widerspiegeln, ginge über den Rahmen dieser Einleitung hinaus. Stattdessen werden die booleschen Netze in Kapitel 4 der vorliegenden Arbeit umfassend eingeführt. Wir sehen jedoch an dieser Stelle bereits, dass es genau 256 verschiedene boolesche Netzklassen gibt, da jede Teilmenge der acht booleschen Interaktionen eine solche Klasse definiert.

Unabhängig von der Charakterisierung einer Netzklasse mittels möglicher Markenanzahl oder gestatteter Flussbeziehungen möchte man einheitliche Aussagen bezüglich der Synthese ihrer Netze machen. Um eine Vielzahl der verschiedenen Netzklassen einheitlich behandeln zu können, wurde ursprünglich in [BD96b] der Begriff der *Netztypen* (engl. *types of nets*) eingeführt und die entsprechende Theorie kürzlich in [BBD15] umfassend behandelt. Alle in der vorliegenden Arbeit untersuchten Netzklassen basieren auf diesem universellen Ansatz der Netztypen.

Die (statische) *Struktur* eines Petri-Netzes ist durch die Plätze, die Transitionen, den

Fluss und die Anfangsmarkierung, das heißt die initiale globale Verteilung der Marken, vollständig beschrieben. Wie das *Verhalten* und damit der modellierte Prozess beziehungsweise das System interpretiert wird, hängt davon ab, welche Semantik für diese Interpretation zu Grunde gelegt wird. In der Literatur werden insbesondere die sogenannte *true concurrency semantics* (engl. für echte Nebenläufigkeits-Semantik) und die sogenannte *interleaving semantics* (engl. für Semantik der Verzahnung/Verschränkung) voneinander unterschieden. Im ersten Fall spiegelt die formale Semantik des Netzverhaltens die Nebenläufigkeit von Ereignissen und die Gleichzeitigkeit von Zuständen explizit wider. Im zweiten Fall werden alle möglichen Transitionssequenzen des Netzes, die gemäß der Fluss-Beziehungen von der Anfangsmarkierung aus schaltbar sind, aufgezeichnet und nicht-deterministisch miteinander verschränkt. In dieser Arbeit legen wir, wie überwiegend in der Netz-Synthese [BBD15], diese Semantik für die Interpretation des Verhaltens eines Petri-Netzes zu Grunde, welches dann durch den sogenannten *Erreichbarkeitsgraphen* des Netzes repräsentiert wird. Der Erreichbarkeitsgraph ist ein Transitionssystem. Ein Transitionssystem besteht aus Zuständen, Ereignissen und beschrifteten Kanten, das heißt „Ursprung-Ereignis-Ziel"-Tripeln: das Auftreten des Ereignisses am Ursprung verursacht einen Zustandswechsel zum Ziel. Im Erreichbarkeitsgraphen entsprechen die Zustände den erreichbaren Markierungen des Netzes, die Ereignisse entsprechen den Transitionen und die Kanten entsprechen dem Schalten von Transitionen und den zugehörigen Änderungen der Markierungen. Der Erreichbarkeitsgraph spiegelt die (strukturelle) Nebenläufigkeit von Ereignissen (nur noch) implizit durch „Nicht-Determinismus" beziehungsweise durch „Beliebigkeit der Reihenfolge" wider.

Der Erreichbarkeitsgraph eines Petri-Netzes ist ein wichtiges Werkzeug für die *Analyse* des durch das Netz modellierten Systems. Typische Fragen der Netzanalyse widmen sich der Erreichbarkeit (Ist eine bestimmte Markierung ein Zustand im Erreichbarkeitsgraphen?), der Beschränktheit (Ist der Erreichbarkeitsgraph endlich?), der Deadlockfreiheit (Findet an jedem Zustand des Erreichbarkeitsgaphen ein Ereignis statt?) oder der Lebendigkeit (Ist im Erreichbarkeitsgraphen für jedes Ereignis von jedem Zustand aus ein Zustand erreichbar, an dem das Ereignis stattfindet?). Die Charakterisierung dieser Entscheidungsprobleme aus Sicht der Berechenbarkeit beziehungsweise der Komplexität ist seit vielen Jahren Forschungsgegenstand der Petri-Netz-Theorie [CEP95, EN94, Esp96, Wim08].

Die *Synthese* von Petri-Netzen bildet in natürlicher Weise das (theoretische wie auch praktische) Gegenstück zu dieser Disziplin: Ausgehend von einem regulären Verhalten, das als Transitionssystem A gegeben ist, suchen wir nach einem Petri-Netz N eines bestimmten Typs, welches das gegebene Verhalten implementiert. Ob und, wenn ja, mit welcher „Genauigkeit" ein Netz N das Verhalten A implementiert ist mittels Abbildungen, sogenannten *Simulationen*, zwischen A und dem Erreichbarkeitsgraphen A_N von N beschrieben. Im besten Fall *realisiert* N das Transitionssystem A exakt. Formal bedeutet dies, dass der Erreichbarkeitsgraph A_N des Netzes und das Eingabe-Transitionssystem A isomorph sind und ein Netz, das diese Eigenschaft erfüllt, heißt eine *Realisierung* des Verhaltens. Da eine Realisierung jedoch nicht immer existiert, werden in der Literatur auch Implementierungen „geringerer Genauigkeit" diskutiert, nämlich Sprachsimulationen und

Einbettungen: eine *Einbettung* bewahrt wenigstens die Verschiedenheit von Zuständen des Eingabeverhaltens und eine *Sprachsimulation* simuliert –wie der Name bereits andeutet– das Transitionssystem bis auf Sprachäquivalenz. Jede der Implementierungen Einbettung, Sprachsimulation und Realisierung impliziert in natürlicher Weise eine Instanz des Syntheseproblems, die für ein Transitionssystem A fragt, ob ein Netz N gemäß der jeweiligen Implementierung existiert. Solange nicht explizit anders angegeben, meinen wir deshalb im Folgenden mit *Synthese* immer auch die Interpretation von Einbettung, Sprachsimulation und Realisierung als Entscheidungsprobleme.

Das Syntheseproblem wurde ursprünglich für die boolesche Klasse der sogenannten *elementaren Netz-Systeme* (engl. *elementary net systems*) gelöst, welche genau die booleschen Interaktionen nop, inp und out erlaubt. Die Grundlagen dafür legten Andrzej Ehrenfeucht und Grzegorz Rozenberg: Basierend auf dem Begriff der *Regionen* in sogenannten *partial-2-structures* beziehungsweise (allgemeinen) beschrifteten gerichteten Graphen, entwickelten sie eine axiomatische Charakterisierung der Graphen, die durch elementare Netz-Systeme implementierbar sind [ER90a, ER90b]. Dieser Ansatz wurde anschließend für die Synthese von (puren) P/T-Netzen [BBD95] sowie einer weiteren booleschen Netzklasse, den sogenannten *Flip-Flop-Netzen* (engl. *flip flop nets*) [Sch96], welche genau die Interaktionen nop, inp, out, swap erlaubt, adaptiert. Darüber hinaus haben Badouel, Bernardinello und Darondeau eine einheitliche, auf Regionen basierende Lösung des Syntheseproblems für alle Petri-Netze entwickelt, die sich durch die bereits erwähnten *Netztypen* beschreiben lassen [BD96b, BBD15]. Die Entscheidbarkeit der Synthese ist für diese Netze also gegeben, sodass sich die Frage nach der Existenz *effizienter* Lösungsalgorithmen stellt.

Der erste Teil der vorliegenden Arbeit adressiert die Charakterisierung der Komplexität der Synthese boolescher Netze und b-beschränkter P/T-Netze und deren Erweiterungen um die additive Gruppe $\mathbb{Z}_{b+1}$ ganzer Zahlen modulo $b+1$.

Die Untersuchung der Komplexität boolescher Netz-Synthese ist unter anderem wie folgt motiviert: Nach bestem Wissen des Autors der vorliegenden Arbeit wurde in der Literatur die Komplexität boolescher Netz-Synthese nur für die elementaren Netz-Systeme und die Flip-Flop-Netze diskutiert. Badouel, Bernardinello und Darondeau [BBD97] zeigten, dass Synthese, die auf Realisierung abzielt, für die elementaren Netz-Systeme NP-vollständig ist. Wenig später zeigte Schmitt [Sch96] für die Flip-Flop-Netze, welche die elementaren Netz-Systeme um die Interaktion swap erweitern, dass ihr Syntheseproblem in Polynomialzeit lösbar ist und zwar unabhängig davon, welche der Implementierungen Einbettung, Sprachsimulation oder Realisierung angestrebt wird. Dies wirft die Frage nach der Komplexität des Syntheseproblems für die verbleibenden booleschen Netzklassen unter Berücksichtigung aller Implementierungen auf.

In der vorliegenden Arbeit liefern wir für alle 256 booleschen Netzklassen eine vollständige Charakterisierung der Komplexität der Synthese, die auf Einbettung abzielt. Darüber hinaus zeigen wir für 84 der 128 möglichen booleschen Netzklassen, die nop erlauben und daher Unabhängigkeit zwischen Plätzen und Transitionen ermöglichen, dass Synthese, die auf Sprachsimulation beziehungsweise Realisierung abzielt, NP-vollständig ist. Darunter

sind, neben den elementaren Netzsystemen und außer den Flip-Flop-Netzen, insbesondere alle weiteren in der Literatur explizit diskutierten booleschen Netzklassen: *contextual nets* (nop, inp, out, used, free) [MR95], *event/condition nets* (nop, inp, out, used) [BBD15], *inhibitor nets* (nop, inp, out, free) [Pie97] sowie *set nets* (nop, inp, out, set) [KKPR13] und *trace nets* (nop, inp, out, res, set, used, free) [BD95b].

Die Untersuchung der Synthese (purer) b-beschränkter Netze sowie deren Gruppenerweiterungen ist wie folgt motiviert: Für $b = 1$ fällt die Klasse der puren b-beschränkten P/T-Netze mit der Klasse der elementaren Netz-Systeme zusammen und die Klasse der b-beschränkten P/T-Netze entspricht für $b = 1$ der Klasse der *event/condition nets*. Das Syntheseproblem dieser Netzklassen ist NP-vollständig. Es stellt sich daher natürlich die Frage, ob das Syntheseproblem (purer) b-beschränkter P/T-Netze auch dann NP-vollständig ist, wenn $b \geq 2$ gilt. Darüber hinaus ist diese Frage auch durch eine Verbindung zur Netz-Synthese motiviert, die von *modalen* Transitionssystemen ausgeht: Schlachter [Sch16] zeigte, dass es für ein modales Transitionssystem im Allgemeinen nicht entscheidbar ist, ob es durch ein beschränktes P/T-Netz implementiert werden kann. In [SW17] beweisen Schlachter und Wimmel dagegen, dass es entscheidbar ist, ob ein modales Transitionssystem durch ein (pures) b-beschränktes P/T-Netz implementiert werden kann. Die exakte Komplexität dieser Instanz des Synthese-Problems bleibt ungeklärt. Die Autoren geben jedoch einen ersten Hinweis darauf, dass die Komplexität des Problems, zu einem (nicht-modalen) Transitionssystem ein implementierendes (pures) b-beschränktes P/T-Netz zu finden, eine untere Schranke für das in [SW17] untersuchte Problem definiert. Die Bestimmung der Komplexität der Synthese (purer) b-beschränkter P/T-Netze formulieren Schlachter und Wimmel als offene Frage.

In der vorliegenden Arbeit beantworten wir diese Frage wie folgt: Für alle $b \geq 1$ gilt, dass Synthese, die auf ein (pures) b-beschränktes P/T-Netz abzielt, für alle genannten Implementierungen NP-vollständig ist.

Das Ergebnis von Schmitt [Sch96] impliziert, dass die Erweiterung purer 1-gebundener Netze um die Gruppe $\mathbb{Z}_2$ der ganzen Zahlen modulo 2 für alle Implementierungen zu einem polynomiellen Syntheseproblem führt. Da die Synthese (purer) b-beschränkter P/T-Netze für alle $b \geq 1$ und für alle Implementierungen NP-vollständig ist, wirft dies natürlich die folgende Frage auf: Für welche Implementierungen und für welche $b \geq 2$ führt die Erweiterung (purer) b-beschränkter P/T-Netze um die Gruppe $\mathbb{Z}_{b+1}$ der ganzen Zahlen modulo $b + 1$ ebenso zu polynomieller Synthese?

In der vorliegenden Arbeit beantworten wir diese Frage wie folgt: Die Synthese $\mathbb{Z}_{b+1}$-erweiterter (purer) b-beschränkter P/T-Netze, die eine Einbettung anstrebt, ist für alle $b \geq 2$ in Polynomialzeit lösbar. Im Kontrast dazu ist die Synthese $\mathbb{Z}_{b+1}$-erweiterter (purer) b-beschränkter P/T-Netze, die auf Sprachsimulation oder Realisierung abzielt, für alle $b \geq 2$ NP-vollständig.

Im zweiten Teil der vorliegenden Arbeit untersuchen wir Varianten des Syntheseproblems, welche bestimmte strukturelle oder dynamische Eigenschaften vom gesuchten Netz fordern.

Während Plätze und Transitionen des Netzes Ressourcen und Ereignisse beziehungsweise Aktionen des Systems modellieren, bilden die Flussbeziehungen zwischen Plätzen und Transitionen nach, inwiefern Ereignisse von Ressourcen abhängen oder diese beeinflussen. Je nach System kann es dabei wünschenswert sein, dass für eine beliebige aber feste Ressource gilt, dass nur wenige Ereignisse, das heißt Akteure, von der Ressource abhängen beziehungsweise auf diese zugreifen können. Im Petri-Netz-Modell bedeutet dies, die Anzahl der Transitionen in der Umgebung eines Platzes einzuschränken. Für kleine und im Voraus festgelegte Anzahlen führen solcherlei Einschränkungen unter anderem zu Netzklassen wie *marked graphs* und *(weighted) T-systems* sowie *choice-free nets* die in der Petri-Netz-Theorie zu den am häufigsten untersuchten überhaupt zählen: In der Netz-Analyse weil sie besonders „einfache“ beziehungsweise „schöne“ Analysemethoden erlauben [Bes86, DE95]; in der Netz-Synthese weil sie verbesserte Algorithmen und gänzlich neue beziehungsweise „effizientere“ Methoden ermöglichen [DEH21, BD15a, BDS18].

Wenn die erlaubte Anzahl d von Transitionen in einer Platzumgebung nicht *a priori* fest gewählt ist, sondern ein Teil der Eingabe wird, dann erhalten wir ein entsprechendes Entscheidungsproblem: Die *Synthese von Petri-Netzen mit beschränkten Platzumgebungen* adressiert die Aufgabe, für ein gegebenes Paar aus Transitionssystem und natürlicher Zahl d zu entscheiden, ob es ein implementierendes Netz gibt, in dem jeder Platz von höchstens d Transitionen abhängt. In der vorliegenden Arbeit untersuchen wir die klassische und parametrisierte Komplexität der Synthese von Netzen mit beschränkten Platzumgebungen für verschiedene Implementierungen und unterschiedliche boolesche Netztypen sowie für P/T-Netze. Dabei unterteilen wir für die P/T-Netze die Umgebung eines Platzes noch in einen *Vor-* beziehungsweise *Nachbereich*, das heißt, wir unterscheiden zwischen Markenzahl erhöhenden beziehungsweise verringernden Auswirkungen von Transitionen. Als natürlichen Parameter dieses Syntheseproblems wählen wir d. Einerseits zeigen wir, dass das (zu Grunde liegende nicht-parametrisierte) Problem für die meisten adressierten Netzklassen NP-vollständig ist. Somit können wir gemäß der klassischen Perspektive, welche den Aufwand eines Problems allein an der Eingabelänge misst, keinen effizienten Entscheidungsalgorithmus erwarten. Aus parametrisierter Sicht messen wir die Komplexität jedoch nicht nur in Abhängigkeit von der Eingabelänge, sondern nehmen auch den Parameter d in den Blick. Die Frage ist nun, ob das Problem fest-Parameter-berechenbar ist, das heißt, ob die Schwere des Problems funktional von d abhängt, während die Eingabelänge nur einen polynomiellen Faktor beiträgt. In der vorliegenden Arbeit beantworten wir diese Frage negativ. Insbesondere zeigt sich zwar, dass die Synthese von Netzen mit beschränkter Platzumgebung mit Parameter d zur Komplexitätsklasse XP (engl. *slice-wise polynomial*) gehört. Andererseits ergeben unsere Untersuchungen, dass das Problem nicht fest-Parameter-berechenbar ist, wenn d als Parameter gewählt wird: Wir zeigen, dass diese Synthese-Instanz, je nach Netztyp, W[1]-schwer oder W[2]-schwer ist.

Im Fall der booleschen Netze können Plätze als Bedingungen interpretiert werden, die genau dann wahr sind, wenn sie eine Marke enthalten. Dabei können Plätze auch Bedingungen oder System-Eigenschaften modellieren, die gar nicht oder zumindest nur

selten im Systemverhalten erfüllt sein sollen. Die *Synthese von booleschen Netzen mit beschränkter Belegungszahl* adressiert die Aufgabe, für ein gegebenes Transitionssystem und eine Zahl ϱ zu entscheiden, ob es ein implementierendes Netz gibt, sodass jeder Platz des Netzes in höchstens ϱ erreichbaren Markierungen eine Marke enthält.

In dieser Arbeit wird das Problem für boolesche Netztypen, die Unabhängigkeit erlauben, aus Sicht der klassischen und parametrisierten Komplexität untersucht: Einerseits wird gezeigt, dass das Problem im Allgemeinen NP-vollständig ist. Andererseits zeigen wir, dass es fest-Parameter-berechenbar ist, wenn ϱ als Parameter interpretiert wird.

Unabhängig von der gesuchten Implementierung und der Art des gesuchten Netzes, ist nicht jedes Transitionssystem durch ein Netz implementierbar. In der Literatur sind daher verschiedene Methoden vorgeschlagen worden, die das Eingabeverhalten modifizieren, um Implementierbarkeit zu erreichen. Diese Methoden richten sich in natürlicher Weise an die Bestandteile des Eingabeverhaltens, das heißt an die Zustände, die Ereignisse und die beschrifteten Kanten.

In der vorliegenden Arbeit untersuchen wir die Komplexität von Methoden, die nicht-implementierbares Eingabeverhalten zum Zwecke der Implementierbarkeit modifizieren. Wir werden sehen, dass keine der hier untersuchten Methoden für eine Netzart im Allgemeinen effizient ist, wenn das zu Grunde liegende Syntheseproblem dieser Netzart bereits NP-schwer ist. In dieser Arbeit untersuchen wir deshalb nur Modifikationen für die Synthese (purer) P/T-Netze, da Synthese für diese Netze polynomiell ist [BBD95, BD96a].

Zu den am häufigsten diskutierten Modifikationen gehört die *Teilung von Ereignissen* des Eingabe-Transitionssystems (engl. *label splitting*) [Car12, SW20] mit anschließender *Neubeschriftung* von Kanten: Ursprünglich mit demselben Ereignis beschriftete Kanten werden nun mit den aus der Teilung (des Ereignisses) resultierenden Ereignissen beschriftet. Ziel ist es, die Gesamtanzahl der Ereignisse im modifizierten Transitionssystem so gering wie möglich zu halten. Das zugehörige Entscheidungsproblem fragt für ein Transitionssystem A und eine Zahl κ, ob durch Neubeschriftung ein Transitionssystem mit höchstens κ Ereignissen konstruiert werden kann, das implementierbar ist.

Schlachter und Wimmel [SW20] haben gezeigt, dass das Problem NP-vollständig ist, wenn eine Einbettung gesucht wird. Die Charakterisierung der Komplexität des Problems der Neubeschriftung die auf Sprachsimulation oder Realisierung abzielt, formulieren die Autoren jedoch als offene Forschungsfrage [SW20].

In dieser Arbeit beantworten wir diese Frage und zeigen, dass das Problem für beide Implementierungen NP-vollständig ist.

Die folgenden Probleme adressieren weitere Modifikationen des Eingabeverhaltens: Für ein Transitionssystem A und eine Zahl κ fragen *Kantenlöschung*, *Ereignislöschung* sowie *Zustandslöschung*, ob A durch das Löschen von κ Kanten, Ereignissen beziehungsweise Zuständen zu einem implementierbaren Transitionssystem modifiziert werden kann. Die Methode der Arbeit von Schlachter und Wimmel [SW20] impliziert, dass *Kantenlöschung* NP-vollständig ist, wenn eine Einbettung gesucht wird.

In der vorliegenden Arbeit zeigen wir, dass *Kantenlöschung* auch dann NP-vollständig

ist, wenn eine Sprachsimulation oder eine Realisierung gesucht wird. Darüber hinaus zeigen wir ebenso, dass auch die Probleme *Ereignislöschung* und *Zustandslöschung* NP-vollständig sind, unabhängig davon, welche der Implementierungen Einbettung, Sprachsimulation oder Realisierung angestrebt wird.

1.2 Literaturdiskussion

In [ER90a] wird das Problem, für eine gegebene *labelled partial 2-structure* (LP2S) eine isomorphe *labelled partial set 2-structure* (LPS2S) zu finden, mit Hilfe sogenannter *Regionen* gelöst: Es gibt eine solche LPS2S genau dann, wenn die Eingabe-LP2S genug Regionen besitzt, die nachweisen, dass zwei sogenannte *Trennungsaxiome* erfüllt sind.

In [ER90b] wenden dieselben Autoren diese Theorie [ER90a] an, um das Syntheseproblem für die elementaren Netz-Systeme (engl. *elementary net systems*) zu lösen, das einen Spezialfall des in [ER90a] untersuchten Problems darstellt.

In diesem Kontext ist eine Region eine Teilmenge der Zustandsmenge Z des Transitionssystems, bezüglich der sich Kanten mit gleicher Beschriftung einheitlich verhalten: entweder sind alle eingehend, alle ausgehend oder alle nicht grenzüberschreitend. Regionen liefern die Plätze und den Fluss eines gesuchten Netzes, wenn es ein solches gibt. Auch in [BD95b] werden Regionen als Zustandsmengen interpretiert und –in substantiell modifizierter Form– zur Lösung des Syntheseproblems für *trace nets* verwendet.

Die Arbeiten [ER90a, ER90b] waren Anlass, Beziehungen zwischen Transitionssystemen und Netzen auch aus Sicht der Kategorientheorie zu untersuchen: in [NRT92] für elementare Netz-Systeme, in [BD95b] für *trace nets* und in [Muk92] für P/T-Netze.

Der Ansatz von Ehrenfeucht und Rozenberg [ER90b] für die Synthese elementarer Netz-Systeme verwendet grundsätzlich alle der $2^{|Z|}$ Zustandsmengen, die Regionen des Transitionssystems sind, und liefert daher Netze nicht handhabbarer Größe. Bernardinello [Ber93] zeigte, dass die Menge aller bezüglich Mengen-Inklusion minimalen Regionen eines Transitionssystems ausreichen, wenn ein gesuchtes Netz existiert. Dies führte erstmals zu Netzen praktikabler Größe.

Ein Resultat von Desel und Reisig [DR93, DR96] zeigte schließlich, dass es stets ein Netz polynomieller Größe gibt, wenn es überhaupt ein implementierendes Netz gibt. Dies bewies insbesondere auch, dass das Syntheseproblem elementarer Netz-Systeme in NP ist. Badouel, Bernardinello und Darondeau [BBD97] wiesen darüber hinaus die NP-Schwere –und folglich die NP-Vollständigkeit– des Syntheseproblems dieser Netzklasse nach. Hiraishi [Hir94] hatte zuvor für Teilprobleme der Synthese elementarer Netz-Systeme NP-Vollständigkeit gezeigt, die er aus den erwähnten Trennungsaxiomen ableitete.

In [BBD95] wird der Regionen-Ansatz adaptiert, um das Syntheseproblem für *pure* P/T-Netze zu lösen. Regionen sind hier nun Tripel von Abbildungen anstatt Zustandsmengen. Die Autoren reduzieren P/T-Netz-Synthese auf das Lösen linearer Ungleichungssysteme und zeigen so die polynomielle Lösbarkeit des Problems. Ein polynomieller Synthese-Algorithmus für reguläre Sprachen, die in bestimmter Form vorliegen, wird ebenso vorge-

stellt. Dieselben Autoren erweitern diesen Ansatz in [BD96a], um Synthese für generelle P/T-Netze in Polynomialzeit zu lösen.

Schmitt [Sch96] führt die Flip-Flop-Netze (engl. *flip-flop nets*) als Erweiterung der elementaren Netz-Systeme um die Gruppe $\mathbb{Z}_2$ der ganzen Zahlen modulo 2 ein und zeigt, dass Synthese für diese Netzklasse polynomiell ist. Dazu modifiziert er den Ansatz aus [BBD95], interpretiert Regionen als Abbildungen (anstatt Zustandsmengen) und löst das Problem durch eine Reduktion auf das Lösen von linearen Gleichungssystemen über $\mathbb{Z}_2$.

In [BD96b] wird schließlich eine generelle Theorie der Regionen entwickelt, die Regionen als Morphismen, das heißt spezielle Abbildungen, zwischen Transitionssystemen und sogenannten *Netztypen* (engl. *types of nets*) [BD95a] interpretiert, wobei letztere ebenso Transitionssysteme sind. Dieser Ansatz synthetisiert gewissermaßen eine Menge atomarer Netze, die jeweils nur einen Platz haben, wobei dieser Platz durch eine Region definiert ist. Wenn es ein gesuchtes Netz gibt, dann erhalten wir ein solches durch das „Verschmelzen" solcher atomaren Netze über die Transitionen. Dies ist der Fall, wenn das Transitionssystem, das dem synchronisierten Produkt der Erreichbarkeitsgraphen der atomaren Netze entspricht, die Implementierungs-Bedingung erfüllt.

Viele der bis dahin bekannten Ansätze für die elementaren Netz-Systeme, Flip-Flop-Netze, (pure) P/T-Netze, *trace nets* lassen sich als konkrete Instanzen dieses allgemeinen Ansatzes auffassen. Darüber hinaus regen die Autoren in ihrer Arbeit bereits an, weitere Netztypen über „beliebige" Flussbeziehungen zu definieren.

Der Polynomialzeit-Algorithmus für die Synthese von P/T-Netzen aus [BBD95] basiert auf dem Lösen von Ungleichungssystemen über den rationalen Zahlen. Best und Devillers entwickelten Algorithmen für bestimmte Teilklassen der P/T-Netze wie zum Beispiel *marked graphs* [BD17], *T-systems* [BD15a] oder auch *choice-free nets* [BD15b, BDS18]. Diese Algorithmen vereinfachen die zu lösenden Gleichungssysteme wesentlich oder stellen sogar Synthese-Methoden zur Verfügung, die vollständig auf strukturellen Eigenschaften der Eingabe-Transitionssysteme beruhen, sodass Ungleichungssysteme keine Rolle mehr spielen. Solche Algorithmen wurden auch für spezielle Eingaben entwickelt, nämlich für Transitionssysteme, die als Wörter beziehungsweise Kreise über einem binären Alphabet interpretiert werden können [BESW16, dFKM18, Ero18, EW17].

Eine ganze Reihe von Arbeiten sind Untersuchungen des Syntheseproblem für weitere (von beschränkten P/T-Netzen verschiedene) Netzarten gewidmet wie zum Beispiel unbeschränkte Petri-Netze [Dar03] oder Petri-Netze mit *inhibitior arcs* [LMB07, Pie99] beziehungsweise Petri-Netze mit *localities* [AP20, KP06, KP07].

Statt Transitionssystemen, werden auch andere Spezifikationen für die Synthese von Netzen zu Grunde gelegt wie zum Beispiel *reguläre*, *partielle*, *präfix-abgeschlossene Sprachen* [LBDM07], *trace-Sprachen* beziehungsweise *step-Sprachen* [HKT96], *deterministische kontext-freie Sprachen* beziehungsweise *deterministische Kellerautomaten* [Dar98] sowie *event-logs* [BP06, vdA11, vdAWM04], wobei [LMJ08] eine Übersicht zur Netz-Synthese ausgehend von Sprachen liefert; modale Transitionssysteme [Sch16, SW17], *modal μ-calculus* [Sch18] und unbeschränkte Transitionssysteme [Dar03] oder auch *step-*

Transitionssysteme [Muk92, Pie02], *path-automatic specifications* [BD04] und *deterministische kontext-freie Graphen* [Dar01].

Darüber hinaus sind verschiedene Synthese-Tools verfügbar, darunter *Petrify*, *Synet*, *Genet* und *APT*.

Petrify [CKLY95, CKK+97b] ist eines der ersten Synthese-Tools, das für die Synthese elementarer Netz-Systeme entwickelt wurde und speziell auf die Synthese asynchroner Schaltkreise [CKK+02, CKK+96, CKK+97c] zugeschnitten ist.

Synet [Cai99] basiert auf der in [BBD95, BCD02, BD96a] entwickelten Theorie und generiert beschränkte P/T-Netze.

Das Synthese-Tool *Genet* [CCK09, CCK+08b] ist eine Erweiterung von *Petrify* und kann neben den elementaren Netz-Systemen allgemeine b-beschränkte P/T-Netze konstruieren.

Mit *APT* [BS15, Sch13] ist ein Synthese-Werkzeug implementiert worden, das speziell darauf ausgelegt ist, (allgemeine) beschränkte P/T-Netze mit bestimmten strukturellen Eigenschaften wie zum Beispiel *pure, free-choice, marked-graph* oder *homogeneity* zu synthetisieren.

In [Wol19] wurde ein Algorithmus für sogenannte Δ_1-Petri-Netze entwickelt, der durch die Verwendung einer bestimmten Datenstruktur (*union/find*) auch Eingaben mit mehr als 10^8 Zuständen handhabbar macht. Wie der Autor berichtet, zeigte die vergleichende Anwendung einer Prototyp-Implementierung bezüglich eines Benchmarks mit 105 Transitionssystemen, das dies keinem der bisher existierenden Synthese-Tools gelingt.

Weitere Arbeiten zur Petri-Netz-Synthese werden in den inhaltlich dazu passenden Kapiteln besprochen. Das Buch [BBD15] liefert eine umfängliche und übersichtliche Darstellung der entscheidenden Entwicklungen zur Netz-Synthese.

Der folgende Abschnitt gibt einen kurzen Abriss über die Struktur der vorliegenden Arbeit.

1.3 Struktur der vorliegenden Arbeit

In diesem Abschnitt geben wir einen kurzen Überblick über den Aufbau der vorliegenden Arbeit und einen etwas detaillierteren Einblick in die genannten Beiträge.

In den Abschnitten 2.1, 2.2, 2.3 und 2.4 von Kapitel 2 führen wir zunächst die notwendigen Grundbegriffe zur Netz-Synthese ein und stellen einige grundlegende Ergebnisse aus der Netztheorie vor. In Abschnitt 2.5 wiederholen wir einige Grundbegriffe zur klassischen und parametrisierten Komplexität.

Danach beginnt der Hauptteil der vorliegende Arbeit, der aus drei Teilen besteht.

Teil 1. Dieser Teil basiert auf den Veröffentlichungen [Tre19a, Tre19b, Tre19d, TR19a, TR19b, TE20a, TE20b, Tre21a] und untersucht die Komplexität der *klassischen Syntheseprobleme*, die durch die Implementierungen Einbettung, Sprachsimulation

und Realisierung impliziert werden, für boolesche Netze, b-beschränkte P/T-Netze sowie deren Erweiterungen um die Gruppe $\mathbb{Z}_{b+1}$.

In Kapitel 3 definieren wir die zugehörigen Entscheidungsprobleme zunächst formell. Des Weiteren stellen wir eine generelle Reduktionsstrategie vor, die für alle NP-Vollständigkeits-Beweise in diesem Teil der vorliegenden Arbeit verwendet wird. Daraufhin wird diese Strategie an einem konkreten Beispiel ausführlich durchexerziert.

In Kapitel 4 führen wir die booleschen Netze beziehungsweise Netztypen ein. Daraufhin wird –je nachdem welche Art der Implementierung untersucht wird– die Komplexität des entsprechenden Syntheseproblems bezüglich der Komplexitätsklassen P beziehungsweise NP für eine Vielzahl von oder –im Fall der Einbettung– für alle booleschen Netze charakterisiert.

In Kapitel 5 werden die (puren) b-beschränkten P/T-Netze und ihre Erweiterungen um die Gruppe $\mathbb{Z}_{b+1}$ eingeführt. Eine vollständige Charakterisierung der Komplexität aller klassischen Syntheseprobleme für alle $b \in \mathbb{N}$ folgt.

Teil 2. Dieser Teil basiert auf den Veröffentlichungen [TE21, ET20, TE20c, TE20d, Tre21c] und adressiert die Komplexität von Synthese die auf Netze abzielt, die einerseits alle Unabhängigkeit zwischen Plätzen und Transitionen erlauben, andererseits jedoch bezüglich ihrer Struktur oder ihres Verhaltens eingeschränkt sind.

In Kapitel 6 wird für eine Reihe boolescher Netze, die Unabhängigkeit erlauben, das folgende Problem aus komplexitäts-theoretischer Sicht untersucht: Entscheide für ein Transitionssystem A und eine natürliche Zahl d, ob es ein sprachsimulierendes oder realisierendes Netz gibt, sodass in der Umgebung jedes Platzes höchstens d Transitionen sind. Die Ergebnisse umfassen hier die NP-Vollständigkeit des Problems, seine Zugehörigkeit zur Komplexitäts-Klasse XP sowie den Nachweis, dass das Problem nicht fest-Parameter-berechenbar ist, wenn d als Parameter interpretiert wird.

In Kapitel 7 wird für P/T-Netze das folgende Problem untersucht: Entscheide für ein Transitionssystem A und zwei natürliche Zahlen ϱ und κ, ob es ein sprachsimulierendes oder realisierendes Netz gibt, sodass im Vorbereich jedes Platzes höchstens ϱ und im Nachbereich jedes Platzes höchstens κ Transitionen sind. Die Ergebnisse umfassen hier die NP-Vollständigkeit des Problems, seine Zugehörigkeit zur Komplexitäts-Klasse XP sowie den Nachweis, dass es nicht fest-Parameter-berechenbar ist, wenn $\varrho + \kappa$ als Parameter interpretiert wird.

In Kapitel 8 wird für alle booleschen Netze, die Unabhängikeit erlauben, und alle Implementierungen das folgende Problem untersucht: Entscheide für ein Transitionssystem A und eine natürliche Zahl ϱ, ob es ein implementierendes Netz gibt, sodass jeder Platz in höchstens ϱ Markierungen eine Marke enthält. Die Ergebnisse umfassen hier die NP-Vollständigkeit des Problems sowie den Nachweis seiner fest-Parameter-Berechenbarkeit, wobei ϱ als Parameter interpretiert wird.

Teil 3. Dieser Teil basiert auf den Veröffentlichungen [Tre20b, Tre21b] und adressiert die Komplexität von Methoden zur Modifikation nicht-implementierbarer Transitionssysteme zum Zwecke der Implementierbarkeit für (pure) P/T-Netze. Diese Modifikationen umfassen *Neubeschriftung, Kantenlöschung, Ereignislöschung* sowie *Zustandslöschung*. Insbesondere zeigen wir für alle genannten Modifikationen und alle Implementierungen, dass die entsprechenden Entscheidungsprobleme, die nach der Existenz einer (in konkret definiertem Sinn) „kleinen“ Modifikation fragen, NP-vollständig sind.

Kapitel 2

Grundlagen und Vorbereitungen

In diesem Kapitel führen wir zunächst in den Abschnitten 2.1 und 2.2 die grundlegenden Begriffe zur Netz-Synthese ein, illustrieren diese mit aussagekräftigen Beispielen und stellen einige bekannte Ergebnisse vor, die im weiteren Verlauf der vorliegenden Arbeit verwendet werden. Im Mittelpunkt stehen dabei die fundamentalen Begriffe der *Regionen* und der *Lösbarkeit* von *Zustands-* sowie *Ereignis-Zustands-Trennungsproblemen* von *Transitionssystemen*, auf denen die gesamte in dieser Arbeit diskutierte Theorie der Petri-Netz-Synthese aufgebaut ist. Diese Abschnitte basieren im Wesentlichen auf dem Buch [BBD15].

In Abschnitt 2.3 führen wir für sogenannte *Netztypen* den Begriff der *Kongruenz* ein, der –in genau definiertem Sinne– Ähnlichkeit zwischen Netztypen beschreibt: Einerseits werden wir zeigen, dass es für bestimmte Probleme der Netz-Synthese ausreicht, einen von zwei kongruenten Netztypen zu untersuchen. Andererseits werden wir argumentieren, dass es durchaus kongruente Netztypen gibt, die bezüglich anderer Fragen der Netz-Synthese doch nicht als äquivalent betrachtet werden können, sodass es im Allgemeinen gerechtfertigt ist, zwischen kongruenten Netztypen zu unterscheiden.

Danach führen wir in Abschnitt 2.4 *lineare* Transitionssysteme ein. Solche Transitionssysteme werden häufig das Ergebnis von Reduktionen sein, die wir präsentieren, um NP-Vollständigkeit zu zeigen. Wir werden zeigen, dass sich gewisse Fragestellungen bezüglich der Netz-Synthese für diese Teilklasse von Transitionssystemen erheblich vereinfachen lassen.

Anschließend werden in Abschnitt 2.5 knapp einige Grundlagen zur klassischen sowie parametrisierten Komplexität wiederholt, wobei wir uns hier im Wesentlichen nach dem Buch [CFK+15] richten.

2.1 Transitionssysteme, Netztypen und Netze

Transitionssysteme dienen zur Beschreibung von sequentiellem und verzweigendem globalem Verhalten dynamischer Systeme. Ein Transitionssystem ist ein gerichteter kantenbeschrifteter Graph mit einer Menge von Zuständen, Ereignissen sowie Kanten, die mit

Ereignissen beschriftet sind und als Zustandsübergänge interpretiert werden.

Definition 2.1 (Transitionssystem)**.** Ein *Transitionssystem* $A = (Z, E, \delta)$ besteht aus einer abzählbaren Menge von *Zuständen* Z, einer endlichen Menge von *Ereignissen* E, wobei $Z \cap E = \emptyset$, und einer partiellen *(Zustands-) Übergangsfunktion* $\delta : Z \times E \to Z$.

Bemerkung 2.1. Gemäß Definition 2.1 betrachten wir in der vorliegenden Arbeit nur *deterministische* Transitionssysteme, denn δ ist eine (partielle) Funktion, deshalb gibt es für jeden Zustand z und jedes Ereignis e höchstens einen Zustand z', sodass $\delta(z, e) = z'$.

Definition 2.2 (Ereignis findet statt)**.** Für ein Ereignis $e \in E$ und einen Zustand $z \in Z$ eines Transitionssystems $A = (Z, E, \delta)$ sagen wir e *findet an* z *statt*, wenn $\delta(e, z)$ definiert ist.

Definition 2.3 (Kante, Kantenmenge)**.** Sei $A = (Z, E, \delta)$ ein Transitionssystem. Für Zustände $z, z' \in Z$ und ein Ereignis $e \in E$ nennen wir (z, e, z') eine (e-beschriftete) *Kante* (von z nach z') von A, wenn $\delta(z, e) = z'$ und nennen z einen *Ursprung* und z' ein *Ziel* von e. Mit $\Delta_A = \{(z, e, z') \mid z, z' \in Z, e \in E : \delta(z, e) = z'\}$ bezeichnen wir die *Kantenmenge von* A.

Definition 2.4 (Schleife, schleifenfrei)**.** Eine Kante (z, e, z') eines Transitionssystems A heißt eine *Schleife*, wenn $z = z'$. Wir sagen A ist *schleifenfrei*, wenn keine Kante von A eine Schleife ist.

Notation 2.1. Im weiteren Verlauf notieren wir durch

- $z \xrightarrow{e}$, dass z ein Ursprung von e ist;
- $\xrightarrow{e} z$, dass z ein Ziel von e ist;
- $\neg z \xrightarrow{e}$, dass z kein Ursprung von e ist;
- $z \xrightarrow{e} z'$ eine Kante (z, e, z');
- $z \xrightarrow{e} z' \in A$, dass (z, e, z') eine Kante von (oder auch *in*) A ist;
- $z \xleftrightarrow{e} z'$, dass $z \xrightarrow{e} z'$ und $z' \xrightarrow{e} z$.

Außerdem bezeichnen wir in sinnvoller Weise die graphische Notation $z \xrightarrow{e} z'$ einer Kante (z, e, z') ebenso als Kante und $z \xleftrightarrow{e} z'$ bezeichnen wir auch als *bidirektionale* Kante.

Definition 2.5 (Pfad)**.** Sei $A = (Z, E, \delta)$ ein Transitionssystem. Für Zustände $z, z' \in Z$ ist ein (gerichteter beschrifteter) *Pfad* P von z nach z' (in A) eine Folge von Kanten $z_0 \xrightarrow{e_1} z_1$, $z_1 \xrightarrow{e_2} z_2$, $z_2 \xrightarrow{e_3} z_3, \ldots$, $z_{n-1} \xrightarrow{e_n} z_n$ von A, sodass $z_0 = z$ und $z_n = z'$ und wird durch $z_0 \xrightarrow{e_1} z_1 \xrightarrow{e_2} z_2 \xrightarrow{e_3} \ldots \xrightarrow{e_n} z_n$ notiert. Wir nennen z *Startzustand* und z' *Endzustand* von P. Aus technische Gründen nehmen wir an, dass es für jeden Zustand $z \in Z$ den *leeren Pfad* von z nach z gibt und notieren diesen durch $z \xrightarrow{\varepsilon} z$.

Definition 2.6 (Initialisiertes Transitionssystem)**.** Ein *initialisiertes Transitionssystem* $A = (Z, E, \delta, \iota)$ besteht aus einem Transitionssystem (Z, E, δ) und einem ausgezeichneten *Initialzustand* $\iota \in Z$, sodass es für jeden Zustand $z \in Z$ einen Pfad von ι nach z gibt.

Notation 2.2 (Komponenten eines Transitionssystems)**.** Wenn A ein Transitionssystem ist, dessen Zustandsmenge, Ereignismenge, Übergangsfunktion oder Initialzustand wir zum Beispiel der Übersichtlichkeit wegen nicht explizit definieren, dann adressieren wir die Komponenten von A häufig durch $Z(A)$ (Zustände), $E(A)$ (Ereignisse), $\delta(A)$ (Übergangsfunktion) und $\iota(A)$ (Initialzustand).

Wenn bereits im Kontext zum Beispiel durch Angabe eines Initialzustandes unmissverständlich klar wird, dass ein Transitionssystem initialisiert ist, dann werden wir im Folgenden diese Eigenschaft der Einfachheit halber nicht explizit erwähnen.

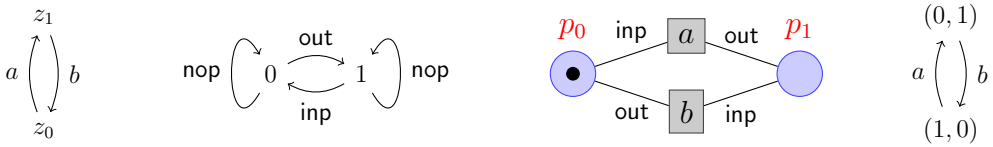
Definition 2.7 (Initialisierter Pfad)**.** Ein Pfad eines Transitionssystems $A = (Z, E, \delta, \iota)$ heißt *initialisierter Pfad*, wenn ι sein Startzustand ist.

Definition 2.8 (Sprache)**.** Die *Sprache* eines Transitionssystems $A = (Z, E, \delta, \iota)$ ist die Menge $L(A)$ von Wörtern $e_1 \dots e_n \in E^*$ für die es Zustände $z_1, \dots, z_n \in Z$ gibt, sodass $\iota \xrightarrow{e_1} z_1 \xrightarrow{e_2} \dots \xrightarrow{e_n} z_n$ ein Pfad in A ist.

Beispiel 2.1. Das Transitionssystem $A' = (Z, E, \delta)$ mit Zustandsmenge $Z = \{z_0, z_1\}$, Ereignismenge $E = \{a, b\}$ und Zustandsübergangsfunktion δ, die definiert ist durch $\delta(z_0, a) = z_1$ und $\delta(z_1, b) = z_0$, lässt sich durch die Auszeichnung von z_0 als Initialzustand zu dem initialisierten Transitionssystem $A = (Z, E, \delta, z_0)$ erweitern. Abbildung 2.1 zeigt die graphische Repräsentation von A. Die Sprache von A ist die Menge $L(A) = \{\varepsilon, a, ab, aba, abab, ababa, \dots\}$.

Netze bestehen aus *Plätzen*, *Transitionen*, einem *Fluss* und einer *Anfangsmarkierung*. Plätze können als Behälter aufgefasst werden, die *Marken* beinhalten können. Eine (globale) *Markierung* des Netzes definiert die (lokalen) Inhalte jedes Platzes, das heißt die Anzahl der Marken auf jedem Platz. Insbesondere definiert die Anfangsmarkierung die initialen Inhalte der Plätze. Bewegung im Netz entsteht durch das *Schalten* von Transitionen, welches (lokale) Veränderungen von Platzinhalten und damit (globale) Veränderungen von Markierungen verursacht. Durch den Fluss steht jedes Paar von Platz und Transition in Beziehung zueinander. Diese Beziehung definiert, welche Inhalte des Platzes das Schalten der Transition erlauben und welche Auswirkungen dieses Schalten auf den Inhalt des Platzes hat.

Netze lassen sich durch die erlaubten Inhalte ihrer Plätze und die möglichen Beziehungen zwischen Plätzen und Transitionen klassifizieren. Diese Art der Klassifizierung lässt über die Wahl der möglichen Platzinhalte, der Wahl möglicher Flussbeziehungen oder beidem eine unendliche Anzahl verschiedener Klassen von Netzen zu. Um dennoch einheitlich über eine Vielzahl verschiedener Klassen sprechen zu können, nutzen wir Transitionssysteme, die *Netztypen* genannt werden, um Netze zu definieren. Die Zustände eines



A τ N A_N

Abbildung 2.1: Das Transitionssystem A (Beispiel 2.1), der elementare Netztyp τ (Beispiel 2.2), das τ-Netz N und sein Erreichbarkeitsgraph A_N (Beispiel 2.3).

Netztyps definieren mögliche Inhalte eines Platzes, seine Ereignisse definieren mögliche Beziehungen zwischen Plätzen und Transitionen und seine Zustandsübergänge beschreiben die Veränderungen des Inhalts eines Platzes durch das Schalten einer Transition.

Definition 2.9 (Netztyp)**.** Ein *Netztyp* ist ein Transitionssystem $\tau = (Z_\tau, E_\tau, \delta_\tau)$ mit $Z_\tau \subseteq \mathbb{N}$.

Definition 2.10 (τ-Netz)**.** Sei τ ein Netztyp. Ein *Netz vom Typ* τ, kurz ein τ*-Netz*, ist ein Tupel $N = (P, T, f, \mathfrak{m}_0)$ bestehend aus endlichen Menge von *Plätzen* P, einer endlichen Menge von *Transitionen* T, wobei $P \cap T = \emptyset$, einem *Fluss* $f : P \times T \to E_\tau$ und einer *Anfangsmarkierung* $\mathfrak{m}_0$, wobei eine *Markierung* $\mathfrak{m}$ von N eine Abbildung $\mathfrak{m} : P \to Z_\tau$ ist.

Definition 2.11 (Aktivierung und Schalten von Transitionen)**.** Eine Transition $t \in T$ ist *aktiviert* beziehungsweise kann *schalten* in einer Markierung $\mathfrak{m}$, wenn $\delta_\tau(\mathfrak{m}(p), f(p,t))$ für alle Plätze $p \in P$ definiert ist. Wir notieren dies durch $\mathfrak{m}[t\rangle$. Ist t in $\mathfrak{m}$ aktiviert, dann führt das *Schalten* von t zur Markierung $\mathfrak{m}'$, wenn $\mathfrak{m}(p) \xrightarrow{f(p,t)} \mathfrak{m}'(p) \in \tau$ für alle Plätze $p \in P$ gilt. Wir notieren dies durch $\mathfrak{m}[t\rangle\mathfrak{m}'$.

Definition 2.12 (Erreichbarkeitsgraph)**.** Die *Menge der Erreichbaren Markierungen* eines τ-Netzes N ist die kleinste Menge $\mathfrak{M}$ von Markierungen, welche die Anfangsmarkierung $\mathfrak{m}_0$ enthält und abgeschlossen ist unter dem Schalten von Transitionen, das heißt, $\mathfrak{m}_0 \in \mathfrak{M}$ und wenn $\mathfrak{m} \in \mathfrak{M}$ und $\mathfrak{m}[t\rangle\mathfrak{m}'$, dann $\mathfrak{m}' \in \mathfrak{M}$. Der *Erreichbarkeitsgraph* von N ist das Transitionssystem $A_N = (\mathfrak{M}, T, \delta, \mathfrak{m}_0)$, wobei für alle $\mathfrak{m}, \mathfrak{m}' \in \mathfrak{M}$ gilt, $\delta(\mathfrak{m}, t) = \mathfrak{m}'$ genau dann, wenn $\mathfrak{m}[t\rangle\mathfrak{m}'$.

Bemerkung 2.2. Viele der in dieser Arbeit betrachteten Netztypen haben ein ausgezeichnetes Ereignis, dass als nop (*no operation* engl. für *keine Verbindung*) oder auch $(0,0)$ beziehungsweise 0 notiert wird, je nachdem in welchem Kontext wir die entsprechenden Netztypen untersuchen. Es entspricht $\delta_\tau(\cdot, \mathsf{nop})$ der Identitätsfunktion auf Z_τ. Durch $f(p,t) = \mathsf{nop}$ wird ausgedrückt, dass der Platz p und die Transition t *unabhängig* sind, dass heißt, weder kann p das Schalten von t verhindern, noch kann das Schalten von t den Inhalt von p verändern.

Beispiel 2.2. Der Netztyp der *elementaren Netz-Systeme* (engl. *elementary net systems*) gehört zur Klasse der booleschen Netztypen, welche in Kapitel 4 umfänglich eingeführt werden und ist definiert durch $\tau = (Z_\tau, E_\tau, \delta_\tau)$ mit $Z_\tau = \{0, 1\}$, $E_\tau = \{\mathsf{nop}, \mathsf{inp}, \mathsf{out}\}$ und $\delta(0, \mathsf{nop}) = 0$, $\delta(1, \mathsf{nop}) = 1$, $\delta(1, \mathsf{inp}) = 0$ und $\delta(0, \mathsf{out}) = 1$. Abbildung 2.1 zeigt die graphische Repräsentation von τ.

Graphische Darstellungen von Netzen zeigen Plätze als Kreise und Transitionen als Rechtecke. Der Flusswert $f(p, t)$ zwischen einem Platz p und einer Transition t wird durch eine ungerichtete mit $f(p, t)$ beschriftete Kante zwischen p und t dargestellt. Gemäß Definition 2.9 enthalten Zustandsmengen von Netztypen stets natürliche Zahlen. Damit entspricht die Markierung eines Platzes einer natürlichen Zahl, welche graphisch durch die zugehörige Anzahl von *Marken* auf dem Platz visualisiert wird.

Beispiel 2.3. Sei τ der elementare Netztyp aus Beispiel 2.2. Abbildung 2.1 zeigt das τ-Netz $N = (P, T, f, \mathfrak{m}_0)$ mit Plätzen $P = \{p_0, p_1\}$, Transitionen $T = \{a, b\}$, Fluss $f(p_0, a) = f(p_1, b) = \mathsf{inp}$, $f(p_1, a) = f(p_0, b) = \mathsf{out}$ und Markierung $\mathfrak{m}_0$, die bestimmt ist durch $(\mathfrak{m}_0(p_0), \mathfrak{m}_0(p_1)) = (1, 0)$. Die Transition b ist nicht aktiviert in $\mathfrak{m}_0$, weil $\delta_\tau(0, \mathsf{inp})$ und damit $\delta_\tau(\mathfrak{m}_0(p_1), f(p_1, b))$ nicht definiert ist. Andererseits gibt es die Kanten $1 \xrightarrow{\mathsf{inp}} 0$ und $0 \xrightarrow{\mathsf{out}} 1$ in τ, sodass a aktiviert ist. Das Schalten von a führt zur Markierung $\mathfrak{m}$, die definiert ist durch $(\mathfrak{m}(p_0), \mathfrak{m}(p_1)) = (0, 1)$. In der Markierung $\mathfrak{m}$ ist nun b statt a aktiviert und das Schalten von b führt zurück zu $\mathfrak{m}_0$. Abbildung 2.1 zeigt den Erreichbarkeitsgraphen A_N von N.

2.2 Synthese: Von Transitionsystemen zu Netzen

Gemäß Definition 2.12 erhalten wir zu jedem Netz durch seinen Erreichbarkeitsgraphen ein Transitionssystem, welches das Verhalten des Netzes beschreibt. Auf natürliche Weise ergibt sich deshalb die Fragestellung, ob es für ein gegebenes Transitionssystem A ein Netz N gibt, welches das durch A beschriebene Verhalten implementiert und, im Fall einer positiven Entscheidung, wie N zu konstruieren ist. Diese Fragestellung ist der Gegenstand von (Petri-Netz-) *Synthese*, um die es in diesem Kapitel gehen soll.

2.2.1 Implementierungen

Das Verhalten eines Netzes N ist durch seinen Erreichbarkeitsgraphen A_N vollständig beschrieben. Es ist deshalb über Beziehungen zwischen A und A_N definiert, ob und mit welcher Genauigkeit N ein gegebenes Transitionssystem A implementiert. Diese Beziehungen werden über Funktionen zwischen (den Zuständen von) A und A_N beschrieben. Die geringste Anforderung an solche Funktionen ist, dass das Bild eines jeden initialen Pfades von A ein initialer Pfad in A_N ist. Intuitiv bedeutet dies, dass N jede Ereignissequenz von A durch eine Schaltsequenz von Transitionen simulieren kann. Solche Funktionen werden deshalb *Simulationen* genannt. Die Frage nach einem simulierenden Netz allein stellt keine

Herausforderung dar: Für jeden Netztyp definieren wir das Netz N mit der leeren Menge von Plätzen, das für jedes Ereignis von A eine Transition enthält. Der Erreichbarkeitsgraph A_N dieses Netzes hat genau einen Zustand z sowie für jedes Ereignis e von A eine Kante $z \xrightarrow{e} z$ und simuliert A. Allerdings simuliert das so definierte Netz *jedes* Transitionssystem, das die gleiche Ereignismenge wie A hat und implementiert A in diesem Sinne mit größtmöglicher Ungenauigkeit. Eine Implementierung höchster Genauigkeit erhalten wir stattdessen durch die Forderung, dass die Simulation ein sogenannter *Isomorphismus* sein soll. In diesem Fall heißt N eine *(exakte) Realisierung* von A, denn dies bedeutet im Wesentlichen, dass A und A_N bis auf die Umbenennung von Zuständen identisch sind. Leider existiert eine solche „optimale Implementierung“ nicht immer. Je nach Anwendung kann jedoch auch ein geringerer Grad an Genauigkeit akzeptabel sein. Dies führt zu weiteren sinnvollen Implementierungsbegriffen wie *Einbettung* beziehungsweise *Sprachsimulation*, die schwächere Forderungen an die Simulation stellen. Die folgenden Definitionen und Beispiele präzisieren und illustrieren die genannten Implementierungsbegriffe.

Definition 2.13 (Simulation)**.** Es seien mit $A = (Z_A, E, \delta_A, \iota_A)$ und $B = (Z_B, E, \delta_B, \iota_B)$ zwei Transitionssysteme mit gleicher Ereignismenge E. Eine *Simulation* φ zwischen A und B ist eine Abbildung $\varphi : Z_A \to Z_B$, sodass $\varphi(\iota_A) = \iota_B$ und wenn $z \xrightarrow{e} z'$ eine Kante in A ist, dann ist das Bild $\varphi(z) \xrightarrow{e} \varphi(z')$ eine Kante in B. Gibt es eine Simulation φ zwischen A und B, sagen wir B *simuliert* A und notieren dies durch $A \leq B$.

Lemma 2.1. *Es kann höchstens eine Simulation zwischen zwei Transitionssystemen $A = (Z_A, E, \delta_A, \iota_A)$ und $B = (Z_B, E, \delta_B, \iota_B)$ geben. Wir können deshalb gegebenenfalls auch von* der *Simulation zwischen A und B reden.*

Beweis. Seien φ_1 und φ_2 Simulationen zwischen A und B. Wir zeigen, dass für alle Zustände z von A gilt, $\varphi_1(z) = \varphi_2(z)$. Sei also z ein beliebiger Zustand von A. Nach Definition 2.6 gibt es einen (möglicherweise leeren) Pfad $z_0 \xrightarrow{e_1} \ldots \xrightarrow{e_n} z_n$, mit $z_0 = \iota_A$ und $z_n = z$. Wir zeigen induktiv über i, dass $\varphi_1(z_i) = \varphi_2(z_i)$ für alle $i \in \{0, \ldots, n\}$. Für $i = 0$ gilt die Aussage, denn nach Definition 2.13 ist $\varphi(\iota_A) = \iota_B$ für jede Simulation. Gelte also $\varphi_1(z_i) = \varphi_2(z_i)$. Weil $\delta_A(z_i, e_{i+1}) = z_{i+1}$ und φ_1 beziehungsweise φ_2 Simulationen sind, gilt sowohl $\delta_B(\varphi_1(z_i), e_{i+1}) = \varphi_1(z_{i+1})$ als auch $\delta_B(\varphi_2(z_i), e_{i+1}) = \varphi_2(z_{i+1})$. Weil nach in Induktionsvoraussetzung $\varphi_1(z_i) = \varphi_2(z_i)$ gilt und δ_B eine Funktion ist, folgt $\varphi_1(z_{i+1}) = \varphi_2(z_{i+1})$. Da z beliebig war, folgt die Behauptung. □

Transitionssysteme beschreiben Verhalten von Systemen. Ein Wort der Sprache eines Transitionssystems A entspricht einer möglichen Sequenz von Ereignissen, welche im durch A beschriebenen System ab einem als initial festgesetzten Moment, dem Initialzustand, stattfinden kann. Simuliert ein Transitionssystem B das Verhalten (des) von A (beschriebenen Systems), erwarten wir deshalb, dass jedes Wort von A ebenso ein Wort von B ist. Das folgende Lemma bestätigt diese Intuition.

Lemma 2.2. *Wenn es eine Simulation zwischen Transitionssystemen $A = (Z_A, E, \delta_A, \iota_A)$ und $B = (Z_B, E, \delta_B, \iota_B)$ gibt, dann gilt $L(A) \subseteq L(B)$.*

Beweis. Sei φ die Simulation zwischen A und B und $w \in L(A)$. Wenn $w = \varepsilon$, dann $w \in L(B)$, nach Definition 2.8. Andernfalls gilt $w = e_1 \dots e_n \in E^+$ und nach Definition 2.8 gibt es einen Pfad $z_0 \xrightarrow{e_1} \dots \xrightarrow{e_n} z_n$ in A, wobei $z_0 = \iota_A$. Weil φ eine Simulation ist, ist $\varphi(z_0) \xrightarrow{e_1} \dots \xrightarrow{e_n} \varphi(z_n)$ ein initialisierter Pfad in B. Dies impliziert $w \in L(B)$. Da w beliebig war folgt die Behauptung. □

Definition 2.14 (Einbettung)**.** Ein Transitionssystem $B = (Z_B, E, \delta_B, \iota_B)$ heißt eine *Einbettung* eines Transitionssystems $A = (Z_A, E, \delta_A, \iota_A)$, notiert durch $A \hookrightarrow B$, wenn es eine injektive Simulation zwischen A und B gibt.

Definition 2.15 (Sprachsimulation)**.** Seien $A = (Z_A, E, \delta_A, \iota_A)$ und $B = (Z_B, E, \delta_B, \iota_B)$ Transitionssysteme. B heißt eine *Sprachsimulation* von A, notiert durch $A \triangleright B$, wenn es eine Simulation φ zwischen A und B gibt, sodass aus $\varphi(z) \xrightarrow{e}$ stets $z \xrightarrow{e}$ folgt.

Bemerkung 2.3. Wenn φ die Simulation ist, welche $A \triangleright B$ rechtfertigt, dann impliziert $\varphi(z) \xrightarrow{e}$ (per Definition) stets $z \xrightarrow{e}$. Es sei an dieser Stelle explizit auf die Kontraposition dieser Implikation hingewiesen: wenn $\neg z \xrightarrow{e}$, dann $\neg \varphi(z) \xrightarrow{e}$. Dieser wird im Zusammenhang mit der sogenannten *Ereignis-Zustands-Trennbarkeits-Eigenschaft*, die später eingeführt wird, noch eine wesentliche Bedeutung bei der Synthese von Netzen aus Transitionssystemen zukommen.

Gemäß des folgenden Lemmas ist der Begriff der *Sprach*simulation gerechtfertigt:

Lemma 2.3. *Wenn $A \triangleright B$ für zwei Transitionssysteme $A = (Z_A, E, \delta_A, \iota_A)$ und $B = (Z_B, E, \delta_B, \iota_B)$ gilt, dann $L(A) = L(B)$.*

Beweis. Nach Lemma 2.2 bleibt $L(B) \subseteq L(A)$ zu zeigen. Sei φ die Simulation zwischen A und B, die $A \triangleright B$ rechtfertigt und $w = e_1 \dots e_n \in L(B)$ beliebig aber fest. Wenn $w = \varepsilon$, dann $w \in L(A)$ nach Definition 2.8. Andernfalls gibt es nach Definition 2.8 einen initialisierten Pfad $q_0 \xrightarrow{e_1} \dots \xrightarrow{e_n} q_n$ in B. Wir zeigen induktiv, dass es für alle $i \in \{0, \dots, n\}$ einen initialisierten Pfad $z_0 \xrightarrow{e_1} \dots \xrightarrow{e_i} z_i$ in A gibt. Für $i = 1$ ist die Aussage wahr, weil $\varphi(\iota_A) = \iota_B = q_0$ und $\iota_B \xrightarrow{e_1}$ sowie Definition 2.15 implizieren, dass $\iota_A \xrightarrow{e_1}$ gilt. Sei also $\iota_A \xrightarrow{e_1} z_1 \xrightarrow{e_2} \dots \xrightarrow{e_i} z_i$ ein Pfad in A. Weil φ eine Sprachsimulation ist und δ_B eine Funktion, gilt $\varphi(z_j) = q_j$ für alle $j \in \{1, \dots, i\}$. Insbesondere gilt $\varphi(z_i) = q_i$ und aus $q_i \xrightarrow{e_{i+1}}$ folgt $z_i \xrightarrow{e_{i+1}}$. Dies impliziert $w \in L(A)$ und aus der Beliebigkeit von w folgt $L(B) \subseteq L(A)$. □

Die folgenden Überlegungen zeigen, dass eine Simulation, die $A \triangleright B$ rechtfertigt, zwar surjektiv ist, jedoch nicht jede surjektive Simulation eine Sprachsimulation ist.

Lemma 2.4. *Eine Simulation φ zwischen Transitionssystemen $A = (Z_A, E, \delta_A, \iota_A)$ und $B = (Z_B, E, \delta_B, \iota_B)$, die $A \triangleright B$ rechtfertigt, ist surjektiv.*

Beweis. Sei $q \in Z_B$. Zu zeigen ist, dass es ein $z \in Z_A$ gibt, sodass $\varphi(z) = q$. Wenn $q = \iota_B$, dann gilt die Aussage mit $z = \iota_A$ nach Definiton 2.15. Andernfalls gibt es nach Definition 2.6 einen Pfad $q_0 \xrightarrow{e_1} \dots \xrightarrow{e_n} q_n$ mit $q_0 = \iota_B$ und $q_n = q$ in B. Insbesondere gilt $e_1 \dots e_n \in L(B)$. Nach Lemma 2.3 gilt $e_1 \dots e_n \in L(A)$. Das heißt, es gibt einen Pfad $z_0 \xrightarrow{e_1} \dots \xrightarrow{e_n} z_n$ mit $z_0 = \iota_A$ in A. Weil φ eine Simulation ist und δ_B eine Funktion, erhalten wir $\varphi(z_i) = q_i$ für alle $i \in \{0, \dots, n\}$. Insbesondere gilt mit $z = z_n$ auch $\varphi(z) = q$. □

Bemerkung 2.4. Im Allgemeinen rechtfertigt eine surjektive Simulation nicht $\triangleright$: Man betrachte das Transitionssystem $A = q_0 \xrightarrow{a} q_1$ und das Transitionssystem A_2 aus Abbildung 2.3. Eine bijektive Simulation φ zwischen A und A_2 ist gegeben durch $\varphi(q_0) = z_0$ und $\varphi(q_1) = z_1$. Jedoch gilt *nicht* $A \triangleright B$, denn $z_1 \xrightarrow{a}$ und $\neg q_1 \xrightarrow{a}$.

Definition 2.16 (Realisierung)**.** Ein Transitionssystem $B = (Z_B, E, \delta_B, \iota_B)$ heißt eine (exakte) *Realisierung* eines Transitionssystems $A = (Z_A, E, \delta_A, \iota_A)$, notiert durch $A \cong B$, wenn es eine bijektive Simulation $\varphi : Z_A \to Z_B$ gibt, sodass $z \xrightarrow{e} z' \in A$ genau dann, wenn $\varphi(z) \xrightarrow{e} \varphi(z') \in B$. Wir nennen φ dann auch *Isomorphismus* und A und B *isomorph*.

Bemerkung 2.5. Ein Isomorphismus φ zwischen A und B rechtfertigt offenbar $A \triangleright B$. Andererseits ist eine Simulation φ, die $A \triangleright B$ rechtfertigt, nach Lemma 2.4 surjektiv. Daher ist φ ein Isomorphismus, wenn φ darüber hinaus injektiv ist.

Bemerkung 2.6. Eine bijektive Simulation ist im Allgemeinen kein Isomorphismus, wie das Beispiel aus Bemerkung 2.4 zeigt.

Die so eingeführten Begriffe Einbettung, Sprachsimulation und Realisierung beschreiben zunächst Beziehungen zwischen Transitionssystemen. Mit Hilfe des Erreichbarkeitsgraphen lassen sie sich jedoch auf natürliche Weise zu (Implementierungs-) Begriffen erweitern, die Beziehungen zwischen Transitionssystemen und Netzen beschreiben.

Definition 2.17 (τ-Implementierungen)**.** Sei τ ein Netztyp. Für ein Transitionssystem $A = (Z, E, \delta, \iota)$ und ein τ-Netz N sagen wir, dass N eine τ-*Einbettung* von A ist, wenn $A \hookrightarrow A_N$, eine τ-*Sprachsimulation* von A ist, wenn $A \triangleright A_N$ und eine (exakte) τ-*Realisierung* von A ist, wenn $A \cong A_N$. Wir nennen N allgemein eine τ-*Implementierung* von A und sagen N *implementiert* A, wenn $A \hookrightarrow A_N$ oder $A \triangleright A_N$ oder $A \cong A_N$.

Zugunsten der besseren Lesbarkeit werden wir im Folgenden statt von τ-Einbettung, τ-Sprachsimulation beziehungsweise τ-Realisierung auch einfach von Einbettung, Sprachsimulation beziehungsweise Realisierung sprechen, wenn aus dem Kontext unmissverständlich hervorgeht, dass es um Implementierungs-Beziehungen geht.

Beispiel 2.4. Sei τ der elementare Netztyp aus Beispiel 2.2 und A_1 und N_1 sowie A_{N_1} definiert wie in Abbildung 2.2. Das τ-Netz N_1 ist eine Einbettung von A_1. Die injektive Simulation φ zwischen A_1 und dem Erreichbarkeitsgraphen A_{N_1} von N_1 ist definiert durch $\varphi(z_0) = (1, 1)$, $\varphi(z_1) = (0, 1)$, $\varphi(z_2) = (1, 0)$ und $\varphi(z_3) = (0, 0)$. N_1 ist keine Sprachsimulation von A_1, da b an $\varphi(z_1) = (0, 1)$ definiert ist, aber nicht an z_1.

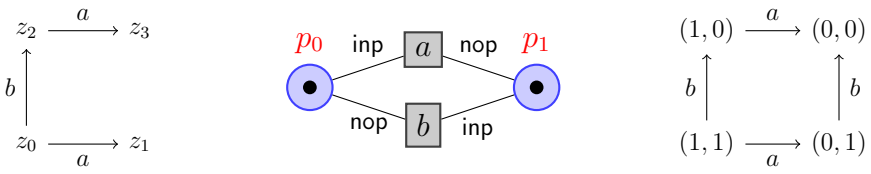


Abbildung 2.2: Sei τ definiert wie in Beispiel 2.2. Von links nach rechts: Transitionssystem A_1, τ-Netz N_1 und sein Erreichbarkeitsgraph A_{N_1}. N_1 ist eine Einbettung von A_1.

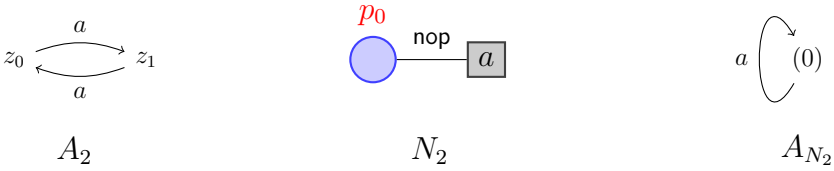


Abbildung 2.3: Sei τ definiert wie in Beispiel 2.2. Von links nach rechts: Transitionssystem A_2, τ-Netz N_2 und sein Erreichbarkeitsgraph A_{N_2}. N_2 ist eine Sprachsimulation von A_2.

Beispiel 2.5. Sei τ der elementare Netztyp aus Beispiel 2.2 und A_2 und N_2 sowie A_{N_2} definiert wie in Abbildung 2.3. Das τ-Netz N_2 ist eine Sprachsimulation von A_2. Die (offensichtlich *nicht* injektive) Sprachsimulation φ zwischen A_2 und dem Erreichbarkeitsgraphen A_{N_2} von N_2 ist definiert durch $\varphi(z_0) = \varphi(z_1) = (0)$. Offenbar impliziert $\varphi(z) \xrightarrow{a}$ stets $z \xrightarrow{a}$ für alle $z \in \{z_0, z_1\}$.

Beispiel 2.6. Sei τ der elementare Netztyp aus Beispiel 2.2 und A und N sowie A_N definiert wie in Abbildung 2.1. Das τ-Netz N ist eine Realisierung von A. Durch $\varphi(z_0) = (1,0)$ und $\varphi(z_1) = (0,1)$ erhalten wir den Isomorphismus φ zwischen A und A_N.

2.2.2 Regionen

Sei τ ein Netztyp. Synthese umfasst den Prozess, ausgehend von einem Transitionssystem $A = (Z, E, \delta, \iota)$ ein implementierendes τ-Netz N zu konstruieren, wenn es ein solches gibt. Da das Netz das Transitionssystem implementieren soll, entsprechen seine Transitionen, das heißt die Ereignisse im Erreichbarkeitsgraphen A_N von N, den Ereignissen von A. Die Plätze, den Fluss und die initiale Markierung von N erhalten wir durch sogenannte τ-Regionen des gegebenen Transitionssystems. Eine *τ-Region* $R = (sup, sig)$ von A besteht aus den Abbildungen *Support* sup und *Signatur* sig. Der Support ordnet jedem Zustand von A einen Zustand von τ zu und die Signatur ordnet jedem Ereignis von A ein Ereignis von τ zu. R modelliert einen (potentiellen) Platz im zu konstruierenden Netz.

Die Signatur $sig(e)$ modelliert den Fluss $f(R, e)$ und $sup(\iota)$ modelliert die initiale Markierung $\mathfrak{m}_0(R)$ dieses Platzes. Über eine Simulation φ stehen die Zustände von A und die Markierungen von N, das heißt die Zustände von A_N, in Beziehung. Insbesondere ordnet φ jedem Zustand z von A eine globale Markierung $\mathfrak{m}$ von N zu. Der Support $sup(z)$ modelliert die lokale Markierung $\mathfrak{m}(R)$ des Platzes R in der globalen Markierung $\mathfrak{m}$ des Netzes, die dem Zustand z durch die Simulation φ zugeordnet wird.

Definition 2.18 (τ-Region)**.** Sei $\tau = (Z_\tau, E_\tau, \delta_\tau)$ ein Netztyp und $A = (Z, E, \delta, \iota)$ ein Transitionssystem. Eine *τ-Region* $R = (sup, sig)$ von A besteht aus dem *Support* $sup : Z \to Z_\tau$ und der *Signatur* $sig : E \to E_\tau$, sodass für jede Kante $z \xrightarrow{e} z'$ von A gilt, dass $sup(z) \xrightarrow{sig(e)} sup(z')$ eine Kante von τ ist.

Definition 2.19 (Bild eines Pfades)**.** Sei A ein Transitionssystem, $P = z_0 \xrightarrow{e_1} \dots \xrightarrow{e_n} z_n$ ein Pfad in A und $R = (sup, sig)$ eine τ-Region von A. Wir nennen den Pfad $P^R = sup(z_0) \xrightarrow{sig(e_1)} \dots \xrightarrow{sig(e_n)} sup(z_n)$ in τ das *Bild von P unter R*.

Die folgende Bemerkung werden wir in der vorliegende Arbeit extensiv nutzen, um Regionen knapp nur durch $sup(\iota)$ und sig darzustellen:

Bemerkung 2.7. Nach Definition 2.6 ist jeder Zustand z von $A = (Z, E, \delta, \iota)$ über einen initialen Pfad $P = z_0 \xrightarrow{e_1} \dots \xrightarrow{e_n} z_n$ mit $\iota = z_0$ und $z_n = z$ erreichbar. Deshalb gilt, wenn $R = (sup, sig)$ eine τ-Region von A ist, dann ist R *implizit* bereits vollständig definiert durch $sup(\iota)$ und sig. Wir erhalten sup explizit durch $sup(z_{i+1}) = \delta_\tau(sup(z_i), sig(e_i))$ für alle $i \in \{0, \dots, n-1\}$. Der Übersichtlichkeit halber werden wir deshalb Regionen häufig nur implizit mittels $sup(\iota)$ und sig präsentieren. Um eine noch kompaktere Darstellung zu erhalten, werden wir die Ereignisse von A häufig gemäß ihrer Signaturen als Menge zusammenfassen, das heißt, für $e_\tau \in E_\tau$ definieren wir $\mathcal{E}^R_{e_\tau} = \{e \in E \mid sig(e) = e_\tau\}$.

Definition 2.20 (Synthetisiertes Netz)**.** Sei τ ein Netztyp. Jede Menge $\mathcal{R}$ von τ-Regionen eines Transitionssystems $A = (Z, E, \delta, \iota)$ definiert das (aus $\mathcal{R}$ und A) *synthetisierte τ-Netz* $N^{\mathcal{R}}_A = (\mathcal{R}, E, f, \mathfrak{m}_0)$, dessen Plätze die Regionen aus $\mathcal{R}$ und dessen Transitionen die Ereignisse von A sind. Der Fluss f und die Anfangsmarkierung $\mathfrak{m}_0$ sind für alle $R = (sup, sig) \in \mathcal{R}$ und alle $e \in E$ definiert durch $f(R, e) = sig(e)$ und $\mathfrak{m}_0(R) = sup(\iota)$.

Beispiel 2.7. Sei τ der elementare Netztyp aus Beispiel 2.2 und A_1 definiert wie in Abbildung 2.2. Das Transitionssystem A_1 hat die τ-Region $R_1 = (sup_1, sig_1)$, welche implizit definiert ist durch $sup_1(z_0) = 1$, $sig_1(a) = \mathsf{inp}$ und $sig(b) = \mathsf{nop}$. Wir erhalten R_1 explizit durch $sup_1(z_1) = \delta_\tau(sup_1(z_0), sig_1(a)) = \delta_\tau(1, \mathsf{inp}) = 0$ sowie $sup_1(z_2) = \delta_\tau(sup_1(z_0), sig_1(b)) = \delta_\tau(1, \mathsf{nop}) = 1$ und $sup_1(z_3) = \delta_\tau(sup_1(z_2), sig_1(a)) = \delta_\tau(1, \mathsf{inp}) = 0$. Das Bild von $P = z_0 \xrightarrow{b} z_2 \xrightarrow{a} z_3$ unter R_1 ist $P^{R_1} = 1 \xrightarrow{\mathsf{nop}} 1 \xrightarrow{\mathsf{inp}} 0$. Eine weitere τ-Region $R_2 = (sup_2, sig_2)$ von A_1 ist definiert durch $sup_2(z_0) = sup_2(z_1) = 1$, $sup_2(z_2) = sup_2(z_3) = 0$ sowie $sig_2(a) = \mathsf{nop}$ und $sig_2(b) = \mathsf{inp}$. Die Menge $\mathcal{R} = \{R_1, R_2\}$ definiert das synthetisierte Netz $N^{\mathcal{R}}_{A_1} = (\{R_1, R_2\}, \{a, b\}, f, \mathfrak{m}_0)$ mit $f(R_1, a) = f(R_2, b) = \mathsf{inp}$,

$f(R_2, a) = f(R_1, b) = \mathsf{nop}$ und $\mathfrak{m}_0 = (\mathfrak{m}_0(R_1), \mathfrak{m}_0(R_2)) = (sup_1(z_0), sup_2(z_0)) = (1, 1)$. Bis auf Umbenennung der Plätze p_0 und p_1 zu R_1 und R_2 entspricht $N_{A_1}^{\mathcal{R}}$ damit genau dem Netz N_1 aus Abbildung 2.2 mit dem dort abgebildeten Erreichbarkeitsgraphen A_{N_1}. Mit den Markierungen $\mathfrak{m}_1 = (0, 1), \mathfrak{m}_2 = (1, 0)$ und $\mathfrak{m}_3 = (0, 0)$ erhalten wir nach Beispiel 2.4 durch $\varphi(z_0) = \mathfrak{m}_0$, $\varphi(z_1) = \mathfrak{m}_1$, $\varphi(z_2) = \mathfrak{m}_2$ und $\varphi(z_3) = \mathfrak{m}_3$ eine eine Einbettung φ zwischen A_1 und A_{N_1}. Für jedes $i \in \{0, \ldots, 3\}$ entsprechen die Werte $sup_1(z_i)$ beziehungsweise $sup_2(z_i)$ den Markierungen $\mathfrak{m}_i(R_1)$ beziehungsweise $\mathfrak{m}_i(R_2)$ der Plätze R_1 und R_2, das heißt, es gilt $\mathfrak{m}_i = (sup_1(z_i), sup_2(z_i))$. In diesem Sinne modellieren die Supporte $sup_1(z_i)$ und $sup_2(z_i)$ die Inhalte von R_1 und R_2 in der Markierung $\mathfrak{m}_i$, die z_i durch die Einbettung φ zugeordnet ist.

2.2.3 Trennungseigenschaften

Unser Ziel ist es, ausgehend von einem Transitionssystem A, ein τ-Netz N zu konstruieren, welches A implementiert. Wir werden sehen, dass N stets einem Netz $N_A^{\mathcal{R}}$ entspricht, das gemäß Definition 2.20 aus einer geeigneten Menge $\mathcal{R}$ von τ-Regionen von A synthetisiert werden kann. In diesem Abschnitt werden wir erläutern, welche Eigenschaften $\mathcal{R}$ erfüllen muss, sodass $N_A^{\mathcal{R}}$ eine gewünschte Implementierung ist.

Betrachten wir zunächst den Fall, dass $N = N_A^{\mathcal{R}}$ eine Einbettung von A ist, das heißt, es gibt eine injektive Simulation φ zwischen A und A_N. Insbesondere gibt es dann für je zwei unterschiedliche Zustände z und z' von A zwei unterschiedliche Markierungen $\mathfrak{m}$ und $\mathfrak{m}'$ von N, sodass $\varphi(z) = \mathfrak{m}$ und $\varphi(z') = \mathfrak{m}'$. Eine globale Markierung $\mathfrak{m}$ von N ist definiert durch die lokalen Inhalte der Plätze P von N. Deshalb unterscheiden sich zwei Markierungen genau dann, wenn sie wenigstens einem Platz von N unterschiedliche Inhalte zuordnen. Im synthetisierten Netz N entsprechen die Plätze den τ-Regionen aus $\mathcal{R}$. Weil N eine Einbettung von A ist, gibt es also eine τ-Region $R = (sup, sig)$ in $\mathcal{R}$, sodass $\mathfrak{m}(R)$ und $\mathfrak{m}'(R)$ voneinander verschieden sind. Es stimmen die Inhalte $\mathfrak{m}(R)$ und $\mathfrak{m}'(R)$ des Platzes R in den Markierungen $\mathfrak{m}$ und $\mathfrak{m}'$ mit den Werten $sup(z)$ und $sup(z')$ überein, deshalb gilt $sup(z) \neq sup(z')$. Diese Überlegungen zeigen das Folgende: Wenn $N = N_A^{\mathcal{R}}$ eine Einbettung von A ist, dann gibt es für je zwei (verschiedene) Zustände z und z' von A (notwendigerweise) eine τ-Region in $\mathcal{R}$, deren Support diesen Zuständen unterschiedliche Werte zuordnet. Wir sagen auch, diese Region *löst* das durch z und z' definierte *Zustands-Trennungs-Problem*. Die folgende Definition präzisiert die Begrifflichkeiten.

Definition 2.21 (Zustands-Trennbarkeits-Eigenschaft)**.** Sei τ ein Netztyp. Zwei voneinander verschiedene Zustände z und z' eines Transitionssystems $A = (Z, E, \delta, \iota)$ definieren das *Zustands-Trennungs-Problem* (kurz ZTP) $\alpha = (z, z')$ von A; α ist *lösbar durch eine τ-Region*, kurz τ-lösbar, wenn es eine τ-Region $R = (sup, sig)$ von A gibt, sodass $sup(z) \neq sup(z')$. Ein Zustand z heißt τ-lösbar, wenn für alle $z \neq z' \in Z$ das ZTP (z, z') τ-lösbar ist. A hat die *τ-Zustands-Trennbarkeits-Eigenschaft*, kurz die τ-ZTE, falls jeder Zustand von A τ-lösbar ist.

Betrachten wir nun den Fall, dass $N = N_A^{\mathcal{R}}$ eine Sprachsimulation von A ist. Nach

Definition 2.15 und Bemerkung 2.3 gibt es also eine Simulation φ zwischen A und A_N, sodass $\neg z \xrightarrow{e}$ stets $\neg\varphi(z) \xrightarrow{e}$ impliziert. Es entspricht $\varphi(z)$ einer erreichbaren Markierung $\mathfrak{m}$ von N, in der die Transition e nicht aktiviert ist. Nach Definition 2.11 bedeutet dies, dass es einen Platz von N gibt, dessen Inhalt in dieser Markierung die Aktivierung von e verhindert. Da die τ-Regionen aus $\mathcal{R}$ die Plätze von N sind, gibt es somit eine τ-Region $R = (sup, sig)$ in $\mathcal{R}$, sodass $\delta_\tau(\mathfrak{m}(R), f(R, e))$ beziehungsweise $\delta_\tau(sup(z), sig(e))$ nicht definiert ist, denn es gilt $\mathfrak{m}(R) = sup(z)$ und $f(R, e) = sig(e)$. Insbesondere definieren e und z ein sogenanntes *Ereignis-Zustands-Trennungsproblem*, welches von R gelöst wird. Die folgende Definition präzisiert die zughörigen Begriffe.

Definition 2.22 (Ereignis-Zustands-Trennbarkeits-Eigenschaft)**.** Sie τ ein Netztyp. Ein Ereignis e und ein Zustand z eines Transitionssystems $A = (Z, E, \delta, \iota)$ für die $\neg z \xrightarrow{e}$ gilt, definieren das *Ereignis-Zustands-Trennungs-Problem* (kurz EZTP) $\alpha = (e, z)$; α ist *lösbar durch eine τ-Region*, kurz τ-lösbar, wenn es eine τ-Region $R = (sup, sig)$ von A gibt, sodass $\neg sup(z) \xrightarrow{sig(e)}$. Wenn für alle $z \in Z$ mit $\neg z \xrightarrow{e}$ das EZTP (e, z) τ-lösbar ist, dann heißt e *τ-lösbar*. Das Transitionssystem A hat die *τ-Ereignis-Zustands-Trennbarkeits-Eigenschaft*, kurz die τ-EZTE, falls jedes seiner Ereignisse τ-lösbar ist.

Definition 2.23 (Zeuge)**.** Sei τ ein Netztyp und $A = (Z, E, \delta, \iota)$ ein Transitionssystem. Wir nennen eine Menge $\mathcal{R}$ von τ-Region von A einen *Zeugen* für die τ-ZTE beziehungsweise für die τ-EZTE von A, falls es für jedes ZTP beziehungsweise EZTP α von A eine Region in $\mathcal{R}$ gibt, die α löst. Wir sagen auch $\mathcal{R}$ *bezeugt* die τ-ZTE beziehungsweise die τ-EZTE von A.

Die bisherigen Überlegungen zeigen bereits: Wenn es ein τ-Netz gibt, welches eine Einbettung beziehungsweise eine Sprachsimulation von A ist, dann muss A notwendigerweise die τ-ZTE beziehungsweise die τ-EZTE haben. Ist N sogar eine Realisierung von A, dann gibt es nach Definition 2.16 einen Isomorphismus φ zwischen A und A_N. Da φ dann insbesondere $A \hookrightarrow A_N$ und $A \triangleright A_N$ rechtfertigt, folgt, dass A notwendigerweise die τ-ZTE und τ-EZTE hat. Der folgende Satz zeigt zunächst, dass jeweils auch die Umkehrungen dieser Implikationen gelten. Wenn A die τ-ZTE, die τ-EZTE beziehungsweise die τ-ZTE und die τ-EZTE hat, dann gibt es ein τ-Netz, das eine Einbettung, eine Sprachsimulation beziehungsweise eine Realisierung von A ist. Darüber hinaus ist eine fundamentale und in ihrer Bedeutung für die Synthese nicht überschätzbare Aussage des Satzes, dass jede Menge $\mathcal{R}$ von τ-Regionen, welche die jeweilige Eigenschaft bezeugt, bereits ein implementierendes Netz $N_A^{\mathcal{R}}$ definiert.

Satz 2.1 ([BBD15])**.** *Sei τ ein Netztyp, A ein Transitionssystem und N ein τ-Netz.*

1. *Es gilt genau dann $A \hookrightarrow A_N$, wenn es einen Zeugen $\mathcal{R}$ für die τ-ZTE von A gibt und $N = N_A^{\mathcal{R}}$.*
2. *Es gilt genau dann $A \triangleright A_N$, wenn es einen Zeugen $\mathcal{R}$ für die τ-EZTE von A gibt und $N = N_A^{\mathcal{R}}$.*

3. *Es gilt genau dann $A \cong A_N$, wenn es einen Zeugen $\mathcal{R}$ für die τ-ZTE und die τ-EZTE von A gibt und $N = N_A^{\mathcal{R}}$.*

Beispiel 2.8. Sei τ der elementare Netztyp aus Beispiel 2.2 und seien A_1 sowie R_1 und R_2 definiert wie in Beispiel 2.7. Das Transitionssystem A_1 hat die ZTP $\alpha_1 = (z_0, z_1)$, $\alpha_2 = (z_0, z_2)$, $\alpha_3 = (z_0, z_3)$, $\alpha_4 = (z_1, z_2)$, $\alpha_5 = (z_1, z_3)$ und $\alpha_6 = (z_2, z_3)$ und die ETZP $\beta_1 = (a, z_1)$, $\beta_2 = (a, z_3)$, $\beta_3 = (b, z_2)$, $\beta_4 = (b, z_3)$ und $\beta_5 = (b, z_1)$. Die τ-Region R_1 löst $\alpha_1, \alpha_3, \alpha_4, \alpha_6$ und β_1 sowie β_2. Die τ-Region R_2 löst α_2, α_5 und β_3 sowie β_4. Insbesondere ist $\mathcal{R} = \{R_1, R_2\}$ eine Zeuge für die τ-ZTE von A_1 und $N_{A_1}^{\mathcal{R}}$ ist eine Einbettung von A_1, wie bereits diskutiert wurde.

Wir zeigen, dass β_5 nicht τ-lösbar ist. Nehmen wir an, das $R = (sup, sig)$ eine τ-Region ist, die das EZTP β_5 löst. Nach Definition 2.22 gilt also $\neg sup(z_1) \xrightarrow{sig(b)}$. Da nop sowohl an 0 und an 1 definiert ist, folgt $sig(b) \in \{\mathsf{inp}, \mathsf{out}\}$. Wenn $sig(b) = \mathsf{inp}$, dann $sup(z_0) = 1$ und $sup(z_1) = 0$, weil $sup(z_0) \xrightarrow{sig(b)}$ und $\neg sup(z_1) \xrightarrow{sig(b)}$. Einerseits folgt aus $sig(b) = \mathsf{inp}$, dass $sup(z_2) = 0$. Andererseits folgt aus $sup(z_0) = 1$, $sup(z_1) = 0$ und $sup(z_0) \xrightarrow{sig(a)} sup(z_1) \in \tau$, dass $sig(a) = \mathsf{inp}$. Dies impliziert mit $sup(z_2) \xrightarrow{sig(a)}$ den Widerspruch $sup(z_2) = 1$. Völlig analog führt die Annahme $sig(b) = \mathsf{out}$ zum Widerspruch. Daher kann es keine τ-Region geben, die β_5 löst und A_1 hat nicht die τ-EZTE.

Beispiel 2.9. Sei τ der elementare Netztyp aus Beispiel 2.2 und seien A_2 sowie N_2 und A_{N_2} definiert wie in Beispiel 2.5. Das Transitionssystem A_2 hat kein EZTP und die Menge $\mathcal{R} = \{R\}$ mit der τ-Region $R = (sup, sig)$, die definiert ist durch $sup(z_0) = sup(z_1) = 0$ und $sig(a) = \mathsf{nop}$, bezeugt die τ-EZTE von A_2. Das synthetisierte Netz $N_{A_2}^{\mathcal{R}}$ entspricht genau dem Netz N_2 welches eine Sprachsimulation von A_2 ist. Weiterhin hat A_2 das ZTP $\alpha = (z_0, z_1)$, welches nicht τ-lösbar ist: Wenn $R = (sup, sig)$ eine τ-Region ist, die α löst, dann gilt entweder $sup(z_0) = 1$ und $sup(z_1) = 0$ oder $sup(z_0) = 0$ und $sup(z_1) = 1$. Wegen $sup(z_0) \xrightarrow{sig(a)} sup(z_1) \in \tau$ und $sup(z_1) \xrightarrow{sig(a)} sup(z_0) \in \tau$, implizieren beide Fälle den Widerspruch, dass sowohl $sig(a) = \mathsf{inp}$ als auch $sig(a) = \mathsf{out}$ gilt. Insbesondere hat A_2 nicht die τ-ZTE.

Beispiel 2.10. Sei τ der elementare Netztyp aus Beispiel 2.2 und seien A und N sowie A_N definiert wie in Beispiel 2.6. Die τ-Regionen $R_1 = (sup_1, sig_1)$ und $R_2 = (sup_2, sig_2)$ von A, die definiert sind durch $sup_1(z_0) = sup_2(z_1) = 1$, $sup_1(z_1) = sup_2(z_0) = 0$ und $sig_1(a) = sig_2(b) = \mathsf{inp}$ sowie $sig_1(b) = sig_2(a) = \mathsf{out}$, lösen alle ZTP und alle EZTP von A. Daher bezeugt $\mathcal{R} = \{R_1, R_2\}$ die τ-ZTE und die τ-EZTE von A. Das synthetisierte τ-Netz $N_A^{\mathcal{R}}$ entspricht bis auf die Umbenennung von R_1 zu p_0 und R_2 zu p_1 dem Netz N.

2.3 Kongruenz von Netztypen

In dieser Arbeit untersuchen wir die Komplexität von Synthese-Problemen für Netze, die über Netztypen definiert werden. Dabei können sich verschiedene Netztypen bis auf *Kongruenz* ähnlich sein:

Definition 2.24 (Kongruenz)**.** Zwei Netztypen $\tau_0 = (Z_{\tau_0}, E_{\tau_0}, \delta_{\tau_0})$ und $\tau_1 = (Z_{\tau_1}, E_{\tau_1}, \delta_{\tau_1})$ heißen *kongruent*, notiert durch $\tau_0 \equiv \tau_1$, wenn es bijektive Funktionen $\varphi : Z_{\tau_0} \to Z_{\tau_1}$ und $\psi : E_{\tau_0} \to E_{\tau_1}$ gibt, sodass ein Kante $z \xrightarrow{e} z'$ genau dann in τ_0 ist, wenn das Bild $\varphi(z) \xrightarrow{\psi(e)} \varphi(z')$ dieser Kante unter φ und ψ in τ_1 ist.

Beispiel 2.11. Abbildung 2.4 zeigt zwei kongruente Netztypen τ_0 und τ_1. Die Kongruenz $\tau_0 \equiv \tau_1$ ist gerechtfertigt durch die Funktionen φ und ψ, wobei $\varphi(0) = 1$ und $\varphi(1) = 0$ sowie $\psi(\mathsf{nop}) = \mathsf{nop}$, $\psi(\mathsf{inp}) = \mathsf{out}$ und $\psi(\mathsf{used}) = \mathsf{free}$.

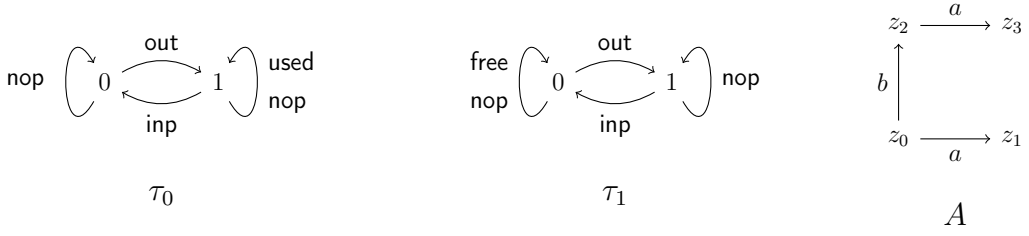


Abbildung 2.4: Die kongruenten Netztypen τ_0 und τ_1 sowie das Transitionssystem A.

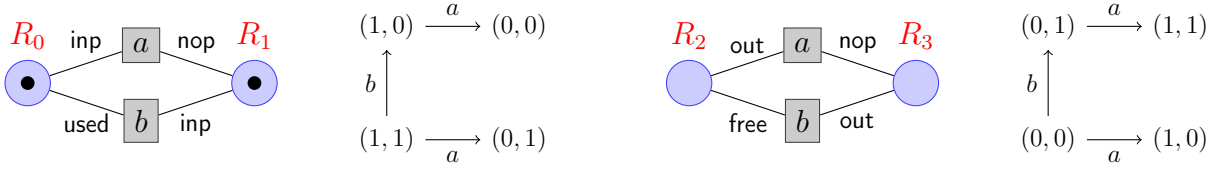
Um Aussagen über die Implementierbarkeit eines Transitionssystems A durch ein τ-Netz treffen zu können, untersuchen wir die τ-Trennbarkeit bestimmter Trennungsprobleme von A nach Satz 2.1. Gemäß des folgenden Satzes und seines Korollars kann sich im Fall kongruenter Netztypen dabei auf einen der Netztypen beschränkt werden:

Satz 2.2. *Es seien τ_0 und τ_1 zwei kongruente Netztypen, $A = (Z, E, \delta, \iota)$ ein Transitionssystem und $\alpha = (z, z')$ ein ZTP und $\beta = (e, z)$ ein EZTP von A. Es sind α beziehungsweise β genau dann τ_0-lösbar, wenn sie τ_1-lösbar sind. Insbesondere hat A genau dann die τ_0-ZTE beziehungsweise die τ_0-EZTE, wenn es die τ_1-ZTE beziehungsweise die τ_1-EZTE hat.*

Beweis. Seien φ und ψ die Funktionen, welche $\tau_0 \equiv \tau_1$ rechtfertigen. Wir beginnen mit dem Beweis der ersten Aussage, welche die Lösbarkeit von Trennungsproblemen adressiert:

„$\Rightarrow$“: Wir zeigen, dass eine τ_0-Region R von A, welche α beziehungsweise β löst, eine τ_1-Region R' von A impliziert, die ebenso α beziehungsweise β löst. Sei also $R = (sup, sig)$ eine τ_0-Region R von A. Wir definieren $R' = (sup', sig')$ mit $sup'(z) = \varphi(sup(z))$ für alle $z \in Z$ und $sig'(e) = \psi(sig(e))$ für alle $e \in E$. Es ist R' eine τ_1-Region von A, denn für alle Kanten $z \xrightarrow{e} z'$ in A gilt: $sup(z) \xrightarrow{sig(e)} sup(z') \in \tau_0 \Leftrightarrow \varphi(sup(z)) \xrightarrow{\psi(sig(e))} \varphi(sup(z')) \in \tau_1 \Leftrightarrow sup'(z) \xrightarrow{sig'(e)} sup'(z') \in \tau_1$.

Es löse R das ZTP α, dass heißt $sup(z) \neq sup(z')$. Aus der Bijektivität von φ folgt, dass $sup'(z) = \varphi(sup(z)) \neq \varphi(sup(z')) = sup'(z')$. Insbesondere löst R' auch α.



N_0 A_{N_0} N_1 A_{N_1}

Abbildung 2.5: Mit den Netztypen τ_0 und τ_1 aus Abbildung 2.4 von links nach rechts: das τ_0-Netz N_0 mit dem Erreichbarkeitsgraph A_{N_0} und das τ_1-Netz N_1 mit dem Erreichbarkeitsgraph A_{N_1}.

Es löse R das EZTP β, dass heißt, es gilt $\neg sup(z) \xrightarrow{sig(e)}$. Angenommen, es gäbe ein $x \in Z_{\tau_1}$, sodass $sup'(z) \xrightarrow{sig'(e)} x \in \tau_1$. Weil φ surjektiv ist, gibt es ein $y \in Z_{\tau_0}$, sodass $\varphi(y) = x$. Dies impliziert $\varphi(sup(z)) \xrightarrow{\psi(sig(e))} \varphi(y) \in \tau_1$ und somit $sup(z) \xrightarrow{sig(e)} y \in \tau_0$. Dies widerspricht der Voraussetzung, dass R das EZTP β löst, das heißt $\neg sup(z) \xrightarrow{sig(e)}$. Somit gilt $\neg sup'(z) \xrightarrow{sig'(e)}$ und R' löst ebenso β.

„$\Leftarrow$“: Der Beweis dieser Implikation ist analog zur Hinrichtung, da die Umkehrfunktionen φ^{-1} und ψ^{-1} die Beziehung $\tau_1 \equiv \tau_0$ rechtfertigen.

Die zweite Aussage des Satzes ist nun eine einfache Folgerung der gerade bewiesenen Äquivalenz bezüglich der Lösbarkeit von Trennungsproblemen. □

Beispiel 2.12. Seien τ_0, τ_1 die Netztypen aus Abbildung 2.4 und φ sowie ψ wie in Beispiel 2.11. Das Transitionsstem A aus Abbildung 2.4 ist τ_0-realisierbar. Die Menge $\mathcal{R}_0 = \{R_0, R_1\}$ von τ_0-Regionen bezeugt die τ_0-ZTE und die τ_0-EZTE von A, wobei

- $R_0 = (sup_0, sig_0)$ mit $sup_0(z_0) = sup_0(z_2) = 1$ und $sup_0(z_1) = sup_0(z_3) = 0$ und $sig_0(a) = \mathsf{inp}$ und $sig_0(b) = \mathsf{used}$;
- $R_1 = (sup_1, sig_1)$ mit $sup_1(z_0) = sup_1(z_1) = 1$ und $sup_1(z_2) = sup_1(z_3) = 0$ und $sig_1(a) = \mathsf{nop}$ und $sig_1(b) = \mathsf{inp}$.

Abbildung 2.5 zeigt das τ_0-Netz $N_0 = N_A^{\mathcal{R}_0}$ und seinen Erreichbarkeitsgraphen A_{N_0}. Wie im Beweis von Satz 2.2 können wir mit φ und ψ aus R_0 beziehungsweise R_1 die folgenden τ_1-Regionen R_2 und R_3 und damit einen Zeugen $\mathcal{R}_1 = \{R_2, R_3\}$ für die τ_1-ZTE und die τ_1-EZTE von A ableiten:

- $R_2 = (sup_2, sig_2)$ mit $sup_2(z_0) = sup_2(z_2) = \varphi(1) = 0$ und $sup_2(z_1) = sup_2(z_3) = \varphi(0) = 1$ und $sig_2(a) = \psi(\mathsf{inp}) = \mathsf{out}$ und $sig_2(b) = \psi(\mathsf{used}) = \mathsf{free}$;
- $R_3 = (sup_3, sig_3)$ mit $sup_3(z_0) = sup_3(z_1) = \varphi(1) = 0$ und $sup_3(z_2) = sup_3(z_3) = \varphi(0) = 1$ und $sig_1(a) = \psi(\mathsf{nop}) = \mathsf{nop}$ und $sig_1(b) = \psi(\mathsf{inp}) = \mathsf{out}$.

Abbildung 2.5 zeigt das τ_1-Netz $N_1 = N_A^{\mathcal{R}_1}$ und seinen Erreichbarkeitsgraphen A_{N_1}. Es ist leicht zu sehen, dass A_{N_0} und A_{N_1} beide isomorph zu dem Transitionssystem A aus Abbildung 2.4 sind.

Mit der Aussage von Satz 2.2 kommt natürlich die Frage auf, ob es aus Sicht der Netz-Synthese überhaupt sinnvoll ist, zwischen bis auf Kongruenz gleichen und bis auf Umbenennung identischen Netztypen zu unterscheiden. Wir wollen im Folgenden knapp argumentieren, dass diese Unterscheidung durchaus berechtigt ist.

Seien τ_0, τ_1, N_0 und N_1 wie in Beispiel 2.12. Beide Netze realisieren dasselbe Verhalten. In beiden Netzen gilt, dass im Initialzustand beide Transitionen a und b aktiviert sind. Während das Schalten von b in N_0 verifiziert, dass der Platz für die Ressource R_0 besetzt ist (used), beweist das Schalten von b in N_1, dass der Platz für die Ressource R_2 frei ist (free). In beiden Netzen deaktiviert das Schalten von a im Initialzustand die Transition b. In N_0 ist der Grund für die Deaktivierung von b, dass der Platz R_0 *keine* Marke enthält und in N_1 ist der Grund dafür, dass R_2 *eine* Marke enthält. Aus Sicht der Modellierung bedeutet dies: während das Schalten von b in N_0 durch die *Abwesenheit* einer Ressource verhindert wird, wird das Schalten von b in N_1 durch die *Anwesenheit* einer Ressource verhindert. Offenbar können R_0 und R_2 bei gleicher Interpretation von b (in beiden Netzen) *nicht* dieselbe Ressource modellieren. Insbesondere erlaubt nur τ_0 den Test auf Anwesenheit, während nur τ_1 den Test auf Abwesenheit erlaubt, sodass die Netztypen aus Perspektive der Modellierung im Allgemeinen *nicht* gleich sind.

In der vorliegenden Arbeit betrachten wir auch Syntheseprobleme, die, neben den Simulationsbeziehungen zwischen Eingabe-Transitionssystem und Erreichbarkeitsgraph, weitere Anforderungen an das synthetisierte Netz stellen. So sollen zum Beispiel Plätze nur in einer begrenzten Anzahl von erreichbaren Markierungen des Netzes eine Marke enthalten. Dies modelliert, dass Ressourcen mit nur begrenzter Häufigkeit im (diskreten) Gesamtverhalten verfügbar sind. Betrachten wir zum Beispiel die Netztypen τ_2 und τ_3 aus Abbildung 2.6, die offenbar kongruent sind: mit $\varphi(0) = 1$ und $\varphi(1) = 0$ sowie $\psi(\mathsf{swap}) = \mathsf{swap}$, $\psi(\mathsf{nop}) = \mathsf{nop}$ und $\psi(\mathsf{set}) = \mathsf{res}$ erhalten wir Funktionen, welche die Kongruenz rechtfertigen.

Abbildung 2.7 zeigt das τ_2-Netz N_2 mit Erreichbarkeitsgraph A_{N_2} und das τ_3-Netz N_3 mit Erreichbarkeitsgraph A_{N_3}. Es ist leicht zu sehen, dass N_2 und N_3 das Transitionssys-

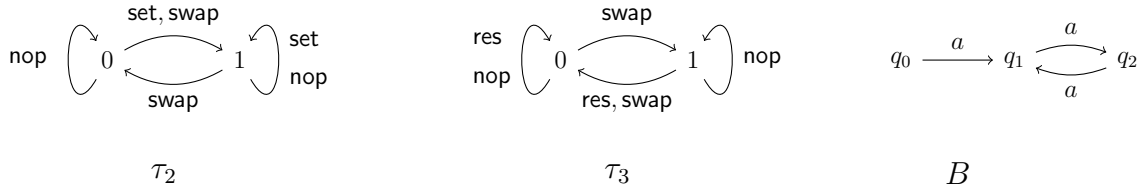


Abbildung 2.6: Die kongruenten Netztypen τ_2 und τ_3 sowie das Transitionssystem B.

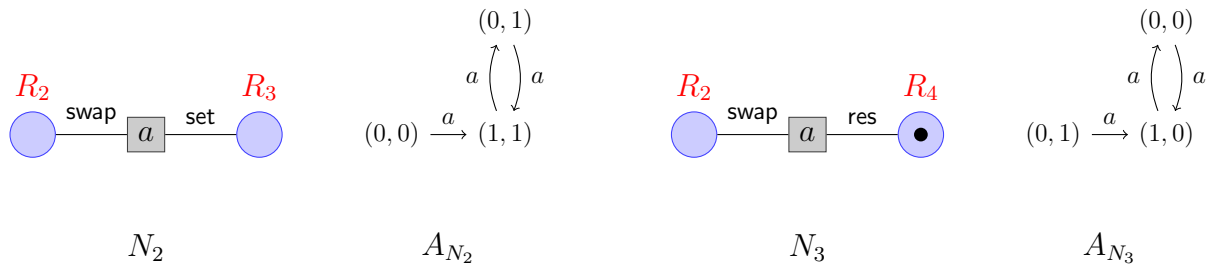


Abbildung 2.7: Mit den Netztypen τ_2 und τ_3 aus Abbildung 2.6 von links nach rechts: das τ_2-Netz N_2 mit Erreichbarkeitsgraph A_{N_2} und das τ_3-Netz N_3 mit Erreichbarkeitsgraph A_{N_3}.

tem B aus Abbildung 2.6 realisieren. Dabei basiert N_2 auf den τ_2-Regionen R_2 und R_3 von B, wobei

- $R_2 = (sup_2, sig_2)$ mit $sup_2(q_0) = sup_2(q_2) = 0$, $sup_2(q_1) = 1$, $sig_2(a) = \mathsf{swap}$ und
- $R_3 = (sup_3, sig_3)$ mit $sup_3(q_0) = 0$, $sup_3(q_1) = sup_3(q_2) = 1$, $sig_3(a) = \mathsf{set}$.

Das τ_3-Netz N_3 basiert auf den τ_3-Regionen R_2 und R_4 von B, wobei $R_4 = (sup_4, sig_4)$ mit $sup_4(q_0) = 1$, $sup_4(q_1) = sup_4(q_2) = 0$, $sig_4(a) = \mathsf{res}$.

Obwohl beide Netze das Verhalten B realisieren, sieht man den folgenden Unterschied: Während alle Plätze von N_3 in höchstens einer erreichbaren Markierung eine Marke enthalten, ist dies für die Plätze von N_2 nicht der Fall: der Platz R_3 ist in zwei erreichbaren Markierungen markiert. Darüber hinaus überlegt man sich, dass es kein τ_2-Netz N gibt, dass B realisiert, sodass jeder Platz von N in höchstens einer Markierung eine Marke enthält. Dieses Minimalbeispiel zeigt: Bezüglich der Frage, ob ein Transitionssystem durch ein Netz implementiert werden kann, sodass jeder Platz von N in höchstens ϱ erreichbaren Markierungen eine Marke enthält, sind kongruente Netztypen im Allgemeinen *nicht* gleich.

2.4 Lineare Transitionssysteme

In der vorliegenden Arbeit präsentieren wir eine Reihe von NP-Vollständigkeits-Ergebnissen, welche durch Polynomialzeit-Reduktionen gewonnen werden. Viele dieser Reduktionen resultieren in einem sogenannten *linearen* Transitionssystem:

Definition 2.25 (Lineares Transitionssystem, Terminalzustand)**.** Ein Transitionssystem A heißt *linear*, wenn es ein kreis- und schleifenfreier Pfad ist, das heißt, es hat die Gestalt $A = z_0 \xrightarrow{e_1} z_1 \xrightarrow{e_2} \dots \xrightarrow{e_n} z_n$, wobei die Zustände $z_0, z_1, \dots, z_{n-1}, z_n$, aber nicht zwingend die Ereignisse $e_1, \dots, e_n$, paarweise voneinander verschieden sind. Es heißt z_n der *Terminalzustand* von A. Ein Ereignis e_i findet *vor* einem Zustand z_j beziehungsweise *vor* einem

Ereignis e_ℓ statt, wenn $i \leq j$ beziehungsweise $i < \ell$. Es findet e_i *nach* z_j beziehungsweise e_ℓ statt, wenn $j, \ell < i$.

Für *beliebige* Netztypen haben lineare Transitionssysteme die überraschende Eigenschaft, dass für sie die τ-EZTE stets die τ-ZTE impliziert. Dies führt zum Beispiel dazu, dass wir uns bei der Untersuchung der Frage, ob ein lineares Transitionssystem A durch ein τ-Netz realisierbar ist, auf die Frage beschränken können, ob A die τ-EZTE hat. Das folgende Lemma adressiert diese Behauptung.

Lemma 2.5. *Sei τ ein Netztyp und sei $A = z_0 \xrightarrow{e_1} \dots \xrightarrow{e_n} z_n$ ein lineares Transitionssystem. Wenn $\mathcal{R}$ ein Zeuge für die τ-EZTE von A ist, dann bezeugt $\mathcal{R}$ auch die τ-ZTE von A.*

Beweis. Nehmen wir an, dass $\mathcal{R}$ kein Zeuge für die τ-ZTE von A ist. Dann hat A ein ZTP, welches nicht durch Regionen aus $\mathcal{R}$ gelöst werden kann. Deshalb gibt es ein ZTP $\alpha = (z_{i_j}, z_{i_k})$ mit $i_j, i_k \in \{0, \dots, n\}$ von A welches den Zustand z_{i_k} mit dem größten Index i_k unter allen Zuständen enthält, die an einem nicht (von Regionen aus $\mathcal{R}$) lösbaren ZTP beteiligt sind: wenn (z_{i_ℓ}, z_{i_m}) ein (durch Regionen aus $\mathcal{R}$) unlösbares ZTP von A ist, dann gelten $i_\ell \leq i_k$ und $i_m \leq i_k$. Insbesondere folgt $i_j < i_k$. Wegen $i_j < i_k$ gibt es die Kante $z_{i_j} \xrightarrow{e_{i_j+1}} z_{i_j+1}$ in A. Weil α nicht durch Regionen aus $\mathcal{R}$ gelöst werden kann, gilt $sup(z_{i_j}) = sup(z_{i_k})$ für alle $(sup, sig) \in \mathcal{R}$. Daraus folgt, dass das Ereignis e_{i_j+1} an z_{i_k} stattfindet, denn andernfalls wäre das EZTP (e_{i_j+1}, z_{i_k}) nicht lösbar: $sup(z_{i_j}) \xrightarrow{sig(e_{i_j+1})}$ und $\neg sup(z_{i_k}) \xrightarrow{sig(e_{i_j+1})}$ impliziert den Widerspruch $sup(z_{i_j}) \neq sup(z_{i_k})$. Also ist $z_{i_k} \xrightarrow{e_{i_j+1}} z_{i_k+1}$ eine Kante in A. Weil nun $sup(z_{i_j}) = sup(z_{i_k})$ für alle $(sup, sig) \in \mathcal{R}$ gilt und δ_τ eine Funktion ist, erhalten wir mit $z_{i_j} \xrightarrow{e_{i_j+1}} z_{i_j+1}$ und $z_{i_k} \xrightarrow{e_{i_j+1}} z_{i_k+1}$ auch $sup(z_{i_j+1}) = sup(z_{i_k+1})$ für alle $(sup, sig) \in \mathcal{R}$. Insbesondere ist das ZTP (z_{i_j+1}, z_{i_k+1}) nicht durch Regionen aus $\mathcal{R}$ lösbar. Wegen $i_k < i_k + 1$ erhalten wir einen Widerspruch zur Wahl von α. Mithin war unsere Annahme falsch und $\mathcal{R}$ ist ein Zeuge für die τ-ZTE von A. □

2.5 Komplexität

Da eine umfassende Einführung in die Komplexitätstheorie weit über den Rahmen der vorliegenden Arbeit hinausgeht, werden wir in diesem Abschnitt nur kurz einige grundlegende Begrifflichkeiten der klassischen Komplexität in Abschnitt 2.5.1 sowie der parametrisierten Komplexität in Abschnitt 2.5.2 wiederholen.

2.5.1 Klassische Komplexität

In diesem Abschnitt wiederholen wir knapp einige grundlegende Begrifflichkeiten aus der klassischen Komplexitätstheorie. Für eine umfassende Einführung verweisen wir auf die Lehrbücher [Sip97, AB09, GJ79, Pap94].

In der klassischen Komplexitätstheorie wird die Komplexität von algorithmisch lösbaren *Entscheidungsproblemen* untersucht:

Definition 2.26. Ein *Entscheidungsproblem* ist eine Sprache $L \subseteq \Sigma^*$ über einem Alphabet Σ.

Obwohl als Alphabet grundsätzlich jede endliche nicht-leere Menge gewählt werden kann, wird üblicherweise $\Sigma = \{0, 1\}$ angenommen.

Ein Algorithmus $\mathcal{A}$ *löst* beziehungsweise *entscheidet* L, wenn er die *charakteristische Funktion* $\chi_L : \Sigma^* \to \{0, 1\}$ mit $\chi_L(\omega) = 1$, falls $\omega \in L$ und andernfalls $\chi_L(\omega) = 0$, berechnet.

In der vorliegenden Arbeit präsentieren wir Entscheidungsprobleme in der gängigen „Frage-Antwort"-Form, wobei das zu Grunde liegende Alphabet üblicherweise nicht explizit genannt wird:

L	
Eingabe:	ω
Frage:	$\omega \in L$?

Die Komplexität des Entscheidungsproblems L ist beschränkt durch die Laufzeit des effizientesten bekannten Algorithmus, der L entscheidet. Die Laufzeit eines Algorithmus $\mathcal{A}$ wird üblicherweise durch eine Funktion $f_\mathcal{A} : \mathbb{N} \to \mathbb{N}$ in Abhängigkeit von der *(Eingabe)-Länge* n eines Eingabeworts ω beschrieben. Eine Zeiteinheit entspricht dabei einem Konfigurationsübergang des zu Grunde gelegten Rechnermodells wie zum Beispiel Turing- oder Registermaschinen. Dabei betrachten wir stets das Szenario des „schlimmsten Falls", das heißt, $f_\mathcal{A}(n)$ entspricht der maximalen Laufzeit von $\mathcal{A}$ über alle Wörter ω der Länge n. Vereinfacht wird das (asymptotische) Laufzeitverhalten mit Hilfe der „Groß-O-Notation" angegeben: Für eine Funktion $g : \mathbb{N} \to \mathbb{N}$ ist $f_\mathcal{A} \in \mathcal{O}(g)$, wenn es eine Konstante $c > 0$ und ein $n_0 \in \mathbb{N}$ gibt, sodass $f_\mathcal{A}(n) \leq c \cdot g(n)$ für alle $n \geq n_0$.

Die Komplexitätsklasse P ist die Menge der *effizient* lösbaren Entscheidungsprobleme: ein Entscheidungsproblem L ist in P, wenn es eine deterministische Turing-Maschine $\mathcal{A}$ mit polynomieller Laufzeit gibt die L entscheidet, das heißt, $f_\mathcal{A} \in \mathcal{O}(n^c)$ für eine Konstante c.

Analog ist ein Entscheidungsproblem L in der Komplexitätsklasse NP, wenn es eine nicht-deterministische Turing-Maschine $\mathcal{A}$ mit polynomieller Laufzeit gibt, die L entscheidet.

Es ist klar, dass $\mathsf{P} \subseteq \mathsf{NP}$, aber die Frage, ob $\mathsf{P} \neq \mathsf{NP}$, ist eines der sogenannten *Millenium-Probleme* [mil00]. Jedoch wird gemeinhin die Annahme akzeptiert, dass P und NP voneinander verschieden sind. Unter dieser Annahme sind die sogenannten NP-schweren Probleme in NP sicherlich nicht in P. Der Begriff der NP-Schwere ist mit Hilfe des folgenden Reduktionsbegriffs definiert:

Definition 2.27 (Polynomielle Reduktion)**.** Seien $L_0, L_1 \subseteq \Sigma^*$ zwei Entscheidungsprobleme. Es heißt L_0 *polynomiell (many-one) reduzierbar* auf L_1, wenn es eine in Polynomialzeit

berechenbare Funktion $f_m^p : \Sigma^* \to \Sigma^*$ gibt, sodass für alle Wörter $\omega \in \Sigma^*$ genau dann $\omega \in L_1$, wenn $f_m^p(x) \in L_2$. Wir nennen f_m^p eine *polynomielle (many-one) Reduktion* (von L_1 auf L_2).

Ein Problem L ist NP-schwer, wenn sich jedes Problem L' aus NP gemäß Definition 2.27 auf L reduzieren lässt. Gilt zudem noch, dass L in NP ist, dann heißt L NP-*vollständig*. Wir bezeichnen mit NPV die Klasse der NP-vollständigen Probleme. Die NP-vollständigen Probleme werden als nicht effizient lösbar angesehen, da sie –unter der Annahme $\mathsf{P} \neq \mathsf{NP}$– offenbar nicht in Polynomialzeit entscheidbar sind.

In der vorliegenden Arbeit beinhalten Eingaben der untersuchten Entscheidungsprobleme häufig ein Transitionssystem $A = (Z, E, \delta, \iota)$. Wir definieren $|A|$ als die *Eingabelänge* (einer „vernünftigen“ Kodierung) von A. Es ist üblich anzunehmen, dass die Bestandteile von A explizit durch die Eingabe (gemäß der gewählten Kodierung) gegeben sind. Deshalb dürfen wir $|A| \geq |Z| + |E|$ annehmen.

Wir präsentieren Algorithmen in Pseudocode, wie es in den gängigen Lehrbüchern üblich ist, siehe zum Beispiel Sipser [Sip97] oder Niedermeyer [Nie06]. Ebenso wie Niedermeyer legen wir zur Abschätzung der Laufzeit von Algorithmen *Registermaschinen* (engl. *random acces machine*) als Rechnermodell mit dem *uniformen Kostenmaß* zu Grunde. Gemäß dieses Modells sind „einfache“ grundlegende Operationen wie Addition, Multiplikation oder auch *if*-Aufrufe jeweils in konstanter Zeit durchführbar. Dabei beachten wir, diese Eigenschaften nicht in einem Sinne falsch zu verwenden, dass die entsprechenden Algorithmen nicht auch effizient durch eine Turing-Maschine implementiert werden könnten. In diesem Fall ist ein Entscheidungsproblem genau dann in polynomieller Zeit von einer Registermaschine lösbar, wenn es in polynomieller Zeit von einer deterministischen Turing-Maschine gelöst werden kann.

2.5.2 Parametrisierte Komplexität

In diesem Abschnitt wiederholen wir knapp einige grundlegende Begriffe der parametrisierten Komplexität. Für eine umfassende Einführung verweisen wir auf die Lehrbücher [CFK+15, DF13, FG06, Nie06], wobei sich die folgenden Darstellungen insbesondere an [CFK+15] orientieren.

In der Disziplin der parametrisierten Komplexität werden *parametrisierte Probleme* untersucht:

Definition 2.28 (Parametrisiertes Problem)**.** Ein *parametrisiertes Problem* ist eine Sprache $L \subseteq \Sigma^* \times \mathbb{N}$, wobei Σ ein Alphabet ist. Für eine Instanz $(\omega, \kappa) \in \Sigma^* \times \mathbb{N}$ nennen wir κ den *Parameter*.

Wie in der Literatur üblich präsentieren wir ein parametrisiertes Problem $L \subseteq \Sigma^* \times \mathbb{N}$ in der Form:

L	
Eingabe:	ω
Parameter:	κ
Frage:	$\omega \in L$?

Der allgemeinen Konvention folgend, definieren wir die Größe einer Eingabe $(\omega, \kappa) \in \Sigma^* \times \mathbb{N}$ als $|\omega| + \kappa$. Dies entspricht der Interpretation, dass der Parameter κ unär kodiert ist.

Definition 2.29 (fest-Parameter-berechenbar, FPT)**.** Ein parametrisiertes Problem $L \subseteq \Sigma^* \times \mathbb{N}$ heißt *fest-Parameter-berechenbar*, wenn es eine Konstante c und eine berechenbare Funktion $f : \mathbb{N} \to \mathbb{N}$ sowie einen (parametrisierten) Algorithmus gibt der für alle (ω, κ) aus $\Sigma^* \times \mathbb{N}$ mit einer Laufzeit von höchstens $f(\kappa) \cdot |\omega|^c$ korrekt entscheidet, ob (ω, κ) in L ist. Die Klasse aller fest-Parameter-berechenbarer Probleme wird FPT(engl. für *fixed parameter tractable*) genannt.

Bemerkung 2.8. Für die Funktion f aus Definition 2.29 werden wir im Folgenden aus technischen Gründen annehmen, dass sie monoton wachsend ist: wenn $n < m$, dann gilt $f(n) \leq f(m)$. Diese Annahme ist ohne Beeinträchtigung der Allgemeinheit gerechtfertigt, weil es zu jeder berechenbaren Funktion f eine monoton wachsende Funktion $\overline{f}$ gibt, die an keiner Stelle kleiner als f ist nämlich $\overline{f}(n) = \max\{f(i) \mid i \in \{0, \ldots, n\}\}$.

Die Klasse FPT ist eine Teilmenge der Komplexitätsklasse XP:

Definition 2.30 (XP)**.** Ein parametrisiertes Problem $L \subseteq \Sigma^* \times \mathbb{N}$ heißt *slice-wise*-polynomiell, wenn es berechenbare Funktionen $f, g : \mathbb{N} \to \mathbb{N}$ und einen (parametrisierten) Algorithmus gibt, der für alle (ω, κ) aus $\Sigma^* \times \mathbb{N}$ mit einer Laufzeit von höchstens $f(\kappa) \cdot |\omega|^{g(\kappa)}$ korrekt entscheidet, ob (ω, κ) in L ist. Die Klasse aller *slice-wise*-polynomieller Probleme wird *XP* genannt.

In der klassischen Komplexität beweisen wir, üblicherweise mittels einer polynomiellen Reduktion, dass ein Problem NP-schwer ist, um zu zeigen, dass es (höchstwahrscheinlich) keinen effizienten Lösungsalgorithmus besitzt. Ähnlich gehen wir in der parametrisierten Komplexitätstheorie vor, um zu argumentieren, dass ein parametrisiertes Problem nicht fest-Parameter-berechenbar ist. Für jedes $i \in \mathbb{N}^+$ definiert $\mathsf{W}[i]$ eine Klasse parametrisierter Probleme und es gelten die folgenden Teilmengen-Beziehungen:

$$\mathsf{FPT} \subseteq \mathsf{W}[1] \subseteq \mathsf{W}[2] \subseteq \mathsf{W}[3] \cdots \subseteq \mathsf{XP}$$

Die genaue Definition der Klasse $\mathsf{W}[i]$, $i \in \mathbb{N}^+$, geht über den Rahmen dieser knappen Einführung hinaus. Wir verweisen deshalb auf [CFK+15]. Es lässt sich beweisen, dass $\mathsf{FPT} \neq \mathsf{XP}$. Gemäß der allgemein akzeptierten Arbeitshypothese gilt darüber hinaus sogar $\mathsf{FPT} \neq \mathsf{W}[1]$ und $\mathsf{W}[i] \neq \mathsf{W}[i+1]$ für alle $i \in \mathbb{N}^+$. Folglich wird für alle $i \geq 1$ angenommen, dass jedes $\mathsf{W}[i]$-schwere Problem nicht fest-Parameter-berechenbar ist. Dabei

bezeichnen wir ein parametrisiertes Problem L_2 als $\mathsf{W}[i]$-schwer, wenn sich jedes parametrisierte Problem L_1 aus $\mathsf{W}[i]$ mittels einer *parametrisierten Reduktion* auf L_2 reduzieren lässt:

Definition 2.31 (Parametrisierte Reduktion)**.** Seien $L_1, L_2 \subseteq \Sigma^* \times \mathbb{N}$ zwei parametrisierte Probleme. Es heißt L_1 durch eine *parametrisierte (many-one) Reduktion* auf L_2 *reduzierbar*, wenn es eine Konstante c und berechenbare Funktionen $g, h : \mathbb{N} \to \mathbb{N}$ sowie eine Funktion $f_m^{\mathsf{para}} : \Sigma^* \times \mathbb{N} \to \Sigma^* \times \mathbb{N}$ gibt, sodass für alle $(\omega, \kappa) \in \Sigma^* \times \mathbb{N}$ die folgenden Eigenschaften erfüllt sind:

1. es gilt genau dann $(\omega, \kappa) \in L_1$, wenn $f_m^{\mathsf{para}}(\omega, \kappa) = (\omega', \kappa') \in L_2$;
2. f_m^{para} ist mit Zeitaufwand $g(\kappa) \cdot |\omega|^c$ berechenbar;
3. $\kappa' \leq h(\kappa)$.

Um zeigen, dass ein parametrisiertes Problem L_2 nicht fest-Parameter-berechenbar ist, zeigen wir für ein bekanntes $W[i]$-schweres Problem L_1, wobei $i \in \mathbb{N}^+$, dass es durch eine parametrisierte Reduktion auf L_2 reduziert werden kann. Weil L_1 $W[i]$-schwer ist und demnach (höchstwahrscheinlich) nicht fest-Parameter-berechenbar ist, gilt gemäß des folgenden Lemmas, dass auch L_2 nicht fest-Parameter-berechenbar ist:

Lemma 2.6 ([CFK+15])**.** *Wenn L_1 und L_2 zwei parametrisierte Probleme sind, sodass L_1 durch eine parametrisierte Reduktion auf L_2 reduzierbar ist und $L_2 \in \mathsf{FPT}$, dann $L_1 \in \mathsf{FPT}$.*

Beweis. Sei L_1 durch eine parametrisierte Reduktion auf L_2 reduzierbar und $L_2 \in \mathsf{FPT}$, das heißt, zu einer Eingabe (ω, κ) für L_1 liefert die Reduktion in Zeit $g(\kappa) \cdot |\omega|^{c_1}$ eine äquivalente Eingabe (ω', κ') für L_2, es gilt $\kappa' \leq h(\kappa)$ und es gibt eine Algorithmus, der in Zeit $f(\kappa')|\omega'|^{c_2}$ entscheidet, ob (ω', κ') in L_2 ist. Die Laufzeit (des Algorithmus) der Reduktion beschränkt die Länge von ω', das heißt $|\omega'| \leq g(\kappa) \cdot |\omega|^{c_1}$. Außerdem gilt $\kappa' \leq h(\kappa)$ und gemäß Bemerkung 2.8 dürfen wir annehmen, dass f monoton wachsend ist. Damit gilt die folgende Ungleichung:

$$f(\kappa') \cdot |\omega'|^{c_2} \leq f(h(\kappa)) \cdot (g(\kappa)|\omega|^{c_1})^{c_2}$$

Unter Berücksichtigung der Laufzeit für die Reduktion erhalten wir eine Gesamtlaufzeit von höchstens $f'(\kappa) \cdot |\omega|^{c_1 \cdot c_2}$, wobei $f'(\kappa) = g(\kappa) + f(h(\kappa)) \cdot g(\kappa)^{c_2}$. Insbesondere ist L_1 fest-Parameter-berechenbar. □

Teil I

Klassische Syntheseprobleme

Kapitel 3

Drei Entscheidungsprobleme und ein Reduktionsschema

Synthese bezüglich eines bestimmten Netztyps τ ist die Aufgabe, für ein gegebenes Transitionssystem A zu entscheiden, ob es ein τ-Netz N gibt, welches A implementiert. Im Fall einer positiven Entscheidung soll N darüber hinaus konstruiert werden. Unabhängig von Netztyp oder gesuchter Implementierung gibt es nicht für jedes Transitionssystem ein implementierendes Netz. *Synthese* umfasst damit sowohl ein Entscheidungsproblem als auch ein Suchproblem. Der Schwerpunkt dieser Arbeit liegt darauf, die Komplexität von Netz-Synthese zu klassifizieren. Weil die entsprechenden Komplexitätsklassen für Entscheidungsprobleme definiert sind, werden wir uns auf den Aspekt der Entscheidung konzentrieren. Dabei ist klar, dass die Komplexität des Entscheidungsproblems eine untere Schranke für das entsprechende Suchproblem definiert. Der folgende Abschnitt 3.1 definiert die sogenannten *klassischen* Entscheidungsprobleme der Netz-Synthese, die für alle in dieser Arbeit untersuchten Netztypen in der Komplexitätsklasse NP sind. Daraufhin wird in Abschnitt 3.2 ein Reduktionsschema vorgestellt, nach dem im Wesentlichen alle NP-Vollständigkeits-Resultate in diesem Teil der Arbeit bewiesen werden. Anschließend wird dieses Reduktionsschema in Abschnitt 3.3 beispielhaft für zwei einfache Netztypen ausführlich diskutiert.

3.1 Einbettung und Sprachsimulation sowie Realisierung

Für einen Netztyp τ und ein Transitionssystem A ist Frage nach der Existenz eines implementierenden τ-Netzes für A äquivalent zu der Frage nach der Existenz eines Zeugen für die (τ-) Trennbarkeits-Eigenschaft von A, welche der gesuchten Implementierung entspricht. Dies führt zur Definition der folgenden *klassischen* Syntheseprobleme, deren Komplexität für eine Reihe von Netztypen in diesem Teil der Arbeit diskutiert wird.

Gemäß Satz 2.1.1, gibt es genau dann eine τ-Einbettung für A, wenn A die τ-ZTE

hat. Dies führt zu folgendem Entscheidungsproblem:

τ-Einbettung	
Eingabe:	Ein endliches Transitionssystem $A = (Z, E, \delta, \iota)$.
Frage:	Gibt es eine Menge $\mathcal{R}$ von τ-Regionen, welche die τ-ZTE von A bezeugt?

Darüber hinaus ist A nach Satz 2.1.2 genau dann durch eine τ-Sprachsimulation implementierbar, wenn es die τ-EZTE hat. Das folgende Entscheidungsproblem adressiert diese Art der Implementation:

τ-Sprachsimulation	
Eingabe:	Ein endliches Transitionssystem $A = (Z, E, \delta, \iota)$.
Frage:	Gibt es eine Menge $\mathcal{R}$ von τ-Regionen, welche die τ-EZTE von A bezeugt?

Schließlich ist A gemäß Satz 2.1.3 genau dann durch ein τ-Netz realisierbar, wenn es sowohl die τ-ZTE als auch die τ-EZTE hat:

τ-Realisierung	
Eingabe:	Ein endliches Transitionssystem $A = (Z, E, \delta, \iota)$.
Frage:	Gibt es eine Menge $\mathcal{R}$ von τ-Regionen, welche die τ-ZTE und die τ-EZTE von A bezeugt?

3.2 Ein Reduktionsschema

Für jeden Netztyp τ, der in dieser Arbeit diskutiert wird, sind τ-Einbettung und τ-Sprachsimulation sowie τ-Realisierung in NP. Deshalb kann die Komplexitäts-Klassifikation für ein bestimmtes Paar von Netztyp und Entscheidungsproblem im Wesentlichen zwei Ergebnisse liefern: Im „negativen“ Fall, stellen wir die NP-Schwere (und folglich die NP-Vollständigkeit) des Problems fest und im „positiven“ Fall stellt sich das Problem als in Polynomialzeit lösbar heraus.

Eine Vielzahl der NP-Vollständigkeits-Resultate in diesem Teil der vorliegenden Arbeit basieren auf *einem* Reduktionsschema, das im Folgenden vorgestellt werden soll. Das folgende Problem ist die Grundlage für das angekündigte Schema:

Cubic Monotone 1 in 3 3Sat (CM 1 in 3 3Sat)	
Eingabe:	Ein Paar $(\mathfrak{U}, M)$ bestehend aus einer Menge $\mathfrak{U}$ boolescher Variablen und einer Menge $M = \{M_0, \ldots, M_{m-1}\}$ mit 3-Klauseln, sodass $M_i = \{X_{i_0}, X_{i_1}, X_{i_2}\} \subseteq \mathfrak{U}$ und $i_0 < i_1 < i_2$ für alle $i \in \{0, \ldots, m-1\}$. Jede Variable $X \in \mathfrak{U}$ kommt in genau drei verschiedenen Klauseln vor.
Frage:	Gibt es ein Eins-in-drei-Modell für $(\mathfrak{U}, M)$, das heißt eine Menge $S \subseteq \mathfrak{U}$, sodass $\lvert S \cap M_i \rvert = 1$ für alle $i \in \{0, \ldots, m-1\}$?

Satz 3.1 ([MR01]). CUBIC MONOTONE 1 IN 3 3SAT *ist NP-vollständig.*

Für eine gültige Eingabe $(\mathfrak{U}, M)$ von CM 1 IN 3 3SAT gilt stets $|\mathfrak{U}| = |M|$: Insgesamt enthalten die Klauseln von M genau $3|M|$ Literale, wobei jedes Literal einer (unnegierten) Variable entspricht. Da jede Variable in genau drei verschiedenen Klauseln vorkommt, erhalten wir somit $|\mathfrak{U}| = \frac{1}{3} \cdot 3|M| = |M|$. Die folgende Instanz erlaubt eine positive Entscheidung und wird in der gesamten Arbeit als Standardbeispiel für CM 1 IN 3 3SAT verwendet.

Beispiel 3.1. Das Paar $(\mathfrak{U}, M)$ mit Variablenmenge $\mathfrak{U} = \{X_0, \ldots, X_5\}$ und Klauselmenge $M = \{M_0, \ldots, M_5\}$ mit den Klauseln $M_0 = \{X_0, X_1, X_2\}$, $M_1 = \{X_0, X_2, X_3\}$, $M_2 = \{X_0, X_1, X_3\}$, $M_3 = \{X_2, X_4, X_5\}$ sowie $M_4 = \{X_1, X_4, X_5\}$ und $M_5 = \{X_3, X_4, X_5\}$, hat das Eins-in-drei-Modell $S = \{X_0, X_4\}$.

Jedes vorgestellte Syntheseproblem fragt für ein Transitionssystem A, ob dieses eine Menge $\mathcal{R}$ von τ-Regionen hat, welche die τ-ZTE oder die τ-EZTE von A oder beide bezeugt. Solange nicht ausdrücklich anders angegeben sei im weiteren Verlauf dieses Teils der vorliegenden Arbeit $(\mathfrak{U}, M)$ eine beliebige aber feste Eingabe von CM 1 IN 3 3SAT mit $\mathfrak{U} = \{X_0, \ldots, X_{m-1}\}$ und $M = \{M_0, \ldots, M_{m-1}\}$, wobei $M_i = \{X_{i_0}, X_{i_1}, X_{i_2}\}$ für alle $i \in \{0, \ldots, m-1\}$. Das Reduktionsschema funktioniert wie folgt.

Wir reduzieren $(\mathfrak{U}, M)$ zu einem Transitionssystem A_τ, sodass es genau dann ein Eins-in-drei-Modell S für $(\mathfrak{U}, M)$ gibt, wenn es einen Zeugen $\mathcal{R}$ für die erfragte Eigenschaft von A_τ gibt. Das Transitionssystem A_τ ist zusammengesetzt aus mehreren Komponenten, wobei jede Komponente für sich als Transitionssystem betrachtet werden kann und im Folgenden –wie für Komponenten bei Reduktionen üblich [GJ79]– auch als Gadget bezeichnet wird. Durch das Hinzufügen weiterer Kanten werden die Gadgets schließlich zu *einem* Transitionssystem A_τ verbunden.

In A_τ werden die Elemente von $\mathfrak{U}$, die Variablen, durch Ereignisse repräsentiert. Die Elemente von M, die Klauseln, werden durch Pfade repräsentiert auf denen die Variablen der Klauseln vorkommen. Genauer enthält A_τ für jedes $i \in \{0, \ldots, m-1\}$ und Klausel $M_i = \{X_{i_0}, X_{i_1}, X_{i_2}\}$ einen Pfad $P_i = \ldots \xrightarrow{X_{i_0}} \ldots \xrightarrow{X_{i_1}} \ldots \xrightarrow{X_{i_2}} \ldots$ auf dem die Variablen von M_i als Ereignisse vorkommen. Weiterhin gibt es per Konstruktion ein Trennungsproblem α dessen τ-Lösbarkeit an die Existenz eines Eins-in-drei-Modells S für $(\mathfrak{U}, M)$ geknüpft ist. Sinnvollerweise wählen wir α entsprechend der erfragten Trennungseigenschaft, das heißt, α ist ein ZTP, falls wir nach der τ-ZTE fragen und ein EZTP, falls wir *wenigstens* nach der τ-EZTE fragen. Die Konstruktion von A_τ ist derart, dass es eine Menge $\mathcal{E} \subseteq E_\tau$ von möglichen Signaturwerten gibt, sodass Folgendes gilt: Wenn eine τ-Region $R = (sup, sig)$ von A_τ das Trennungsproblem α löst, dann definiert die Menge der Variablen-Ereignisse, die eine Signatur aus $\mathcal{E}$ haben, ein Eins-in-drei-Modell S von $(\mathfrak{U}, M)$, das heißt, für $S = \{X \in \mathfrak{U} \mid sig(X) \in \mathcal{E}\}$ gilt $|S \cap M_i| = 1$ für alle $i \in \{0, \ldots, m-1\}$. Wenn nun A_τ die erfragte Eigenschaft hat, dann gibt es eine Menge $\mathcal{R}$ von τ-Regionen, die alle dieser Eigenschaft entsprechenden Trennungsprobleme von A_τ löst. Insbesondere enthält $\mathcal{R}$ eine Region, die α löst und impliziert damit ein Eins-in-drei-Modell von $(\mathfrak{U}, M)$.

Anderseits ist α und jedes weitere dem Zielproblem entsprechende Trennungsproblem von A_τ nach Konstruktion lösbar, wenn $(\mathfrak{U}, M)$ ein gesuchtes Eins-in-drei-Modell S hat. Somit erlaubt $(\mathfrak{U}, M)$ genau dann eine positive Entscheidung, wenn A_τ diese erlaubt.

Im folgenden Abschnitt werden wir das eben skizzierte Reduktionsschema an einem konkreten Beispiel detailliert durchexerzieren.

3.3 Ein ausführliches Reduktionsbeispiel

In diesem Abschnitt wird das vorgestellte Reduktionsschema für die Netztypen τ_0 und τ_1 aus Abbildung 3.1 ausführlich diskutiert. Beide Netztypen gehören zu den sogenannten *booleschen* Netztypen, welche in Kapitel 4 umfänglich eingeführt werden. An dieser Stelle ist diese Klassifizierung jedoch unerheblich, da es allein um die Funktionsweise des Schemas geht.

Abbildung 3.1: Die Netztypen τ_0 und τ_1 für das ausführliche Reduktionsbeispiel.

Solange nicht ausdrücklich anders vereinbart, sei also $\tau \in \{\tau_0, \tau_1\}$ im Folgenden beliebig aber fest, wobei τ_0 und τ_1 wie in Abbildung 3.1 definiert sind. Mit Hilfe des vorgestellten Reduktionsschemas wird nun der folgende Satz bewiesen:

Satz 3.2. τ-Sprachsimulation *und* τ-Realisierung *sind* NP*-vollständig.*

Zunächst überlegt man sich, dass τ-Sprachsimulation und τ-Realisierung in NP sind: Ein Transitionssystem $A = (Z, E, \delta, \iota)$ hat gemäß Definition 2.21 und Definition 2.22 höchstens $|Z|^2$ ZTP und höchstens $|E| \cdot |Z|$ EZTP. Im Fall einer positiven Entscheidung gibt es daher einen Zeugen $\mathcal{R}$ mit höchstens $|E| \cdot |Z| + |Z|^2$ τ-Regionen für die erfragte Eigenschaft von A. Offenbar ist die Größe jeder τ-Region $R = (sup, sig)$ polynomiell durch $|E| + |Z|$ beschränkt. Daher kann eine nicht-deterministische Turingmaschine einen entsprechenden Zeugen $\mathcal{R}$ mit polynomiellen Zeitaufwand raten und anschließend deterministisch verifizieren.

Bemerkung 3.1. Unabhängig vom aktuell untersuchten Netztyp hat A höchstens $|E| \cdot |Z| + |Z|^2$ Trennungsprobleme, die untersucht werden müssen. Dennoch ist die Zugehörigkeit der entsprechenden Entscheidungsprobleme zu NP nicht für jeden Netztyp so leicht zu sehen. Genauer gesagt werden in dieser Arbeit auch Netztypen diskutiert, deren Zustands- beziehungsweise Ereignismenge den natürlichen Zahlen entspricht. Für diese erfordert der Beweis, dass die Existenz eines Zeugen (für die erfragte Eigenschaft) stets die Existenz eines Zeugen mit Regionen polynomiell beschränkter Größe impliziert, größeren Aufwand.

Wir reduzieren $(\mathfrak{U}, M)$ nun mittels des in Abschnitt 3.2 vorgestellten Reduktionsansatzes und beweisen so, dass sowohl τ-SPRACHSIMULATION als auch τ-REALISIERUNG NP-schwer sind. Insbesondere werden wir für beide Probleme mit derselben (konkreten) Reduktion auskommen.

Das angekündigte Transitionssystem A_τ hat das folgende Gadget H und für jedes $i \in \{0, \ldots, m-1\}$ das folgende Gadget T_i, auf dem die Elemente von $M_i = \{X_{i_0}, X_{i_1}, X_{i_2}\}$ als Ereignisse stattfinden:

$$H = \quad h_0 \xrightarrow{k_0} h_1 \xrightarrow{k_1} h_2 \qquad\qquad T_i = \quad t_{i,0} \xrightarrow{X_{i_0}} t_{i,1} \xrightarrow{X_{i_1}} t_{i,2} \xrightarrow{X_{i_2}} t_{i,3} \xrightarrow{k_1} t_{i,4}$$
$$\Big\downarrow k_0$$
$$t_{i,5}$$

Das Gadget H liefert das angekündigte EZTP $\alpha = (k_1, h_0)$. Das Gadget T_i liefert den angekündigten Pfad $P_i = t_{i,0} \xrightarrow{X_{i_0}} t_{i,1} \xrightarrow{X_{i_1}} t_{i,2} \xrightarrow{X_{i_2}} t_{i,3}$ auf dem die Variablen der Klausel $M_i = \{X_{i_0}, X_{i_1}, X_{i_2}\}$ als Ereignisse vorkommen, um M_i zu repräsentieren. Weiterhin verwenden wir die frischen Zustände der Menge $\bot = \{\bot_0, \ldots, \bot_m\}$, Ereignisse der Mengen $\ominus = \{\ominus_1, \ldots, \ominus_m\}$ und $\oplus = \{\oplus_0, \ldots, \oplus_m\}$ und Kanten $\bot_i \xrightarrow{\ominus_{i+1}} \bot_{i+1}$ und $\bot_i \xrightarrow{\oplus_i} t_{i,0}$ für alle $i \in \{0, \ldots, m-1\}$ sowie $\bot_m \xrightarrow{\oplus_m} h_0$, um die Komponenten H und $T_0, \ldots, T_{m-1}$ zu dem Transitionssystem $A_\tau = (Z(A_\tau), E(A_\tau), \delta(A_\tau), \bot_0)$ mit Initialzustand $\bot_0$ zusammenzufügen. Abbildung 3.2 zeigt A_τ, für die konkrete Instanz $(\mathfrak{U}, M)$ aus Beispiel 3.1.

Lemma 3.1. *Wenn es einen Zeugen für die τ-EZTE von A_τ gibt, dann hat $(\mathfrak{U}, M)$ ein Eins-in-drei-Modell.*

Beweis. Wenn es einen Zeugen für die τ-EZTE von A_τ gibt, dann gibt es eine τ-Region, die α löst. Sei $R = (sup, sig)$ eine solche Region. Es wird nun gezeigt, dass die Menge $S = \{X \in \mathfrak{U} \mid sig(X) = \mathsf{inp}\}$ ein Eins-in-drei-Modell von $(\mathfrak{U}, M)$ definiert.

Weil R das EZTP α löst, gilt $\neg sup(h_0) \xrightarrow{sig(k_1)}$ nach Definition 2.22. Weil nop an 0 und 1 stattfindet, muss $sig(k_1) \in \{\mathsf{inp}, \mathsf{used}, \mathsf{free}\}$ gelten.

Nehmen wir zunächst $sig(k_1) \in \{\mathsf{inp}, \mathsf{used}\}$ an. Dies impliziert $sup(h_1) = 1$ und $sup(h_0) = 0$, weil inp und used an 1 definiert sind, aber $sig(k_1)$ nicht an $sup(h_0)$ definiert ist. Nach Definition 2.18 gilt, dass $sup(h_0) \xrightarrow{sig(k_0)} sup(h_1)$ und somit $0 \xrightarrow{sig(k_0)} 1$ eine Kante in τ ist. Dies ist ein Widerspruch, denn es gibt keine Kante von 0 nach 1 in τ. Folglich gilt $sig(k_1) \notin \{\mathsf{inp}, \mathsf{used}\}$.

Diese Betrachtungen zeigen, dass $sig(k_1) = \mathsf{free}$. Somit gilt $sup(z) = 0$ für alle Zustände z von A_τ, an denen k_1 stattfindet. Weil free an $sup(h_0)$ nicht stattfindet, erhalten wir $sup(h_0) = 1$. Dies impliziert mit $sup(h_1) = 0$ und $sup(h_0) \xrightarrow{sig(k_0)} sup(h_1) \in \tau$, dass $sig(k_0) = \mathsf{inp}$. Letzteres bedeutet für alle Zustände z von A_τ, an denen k_0 stattfindet, dass

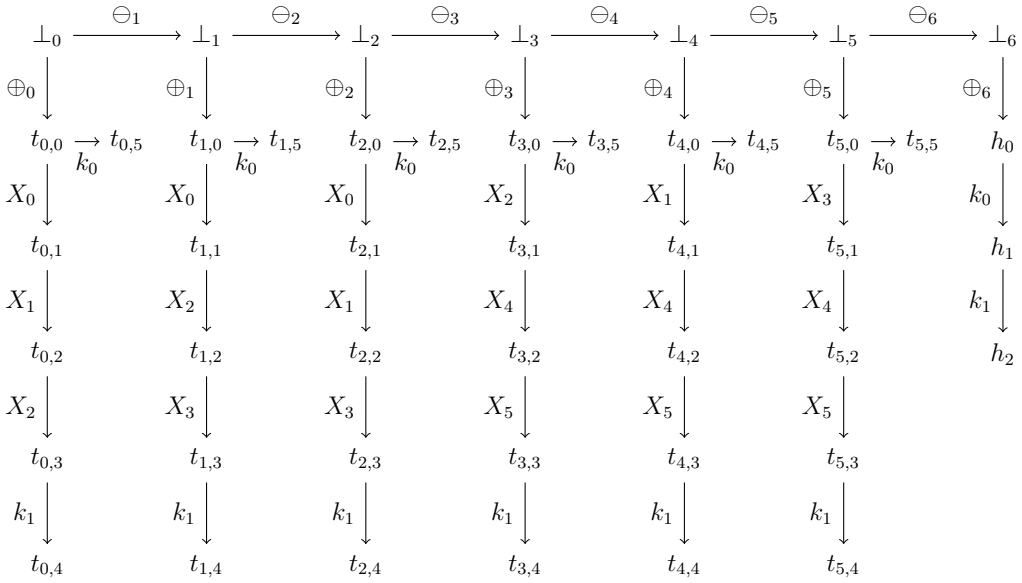


Abbildung 3.2: Das Transitionssystem A_τ, welches aus Beispiel 3.1 resultiert.

$sup(z) = 1$. Da für jedes $i \in \{0, \ldots, m-1\}$ das Ereignis k_0 an $t_{i,0}$ und das Ereignis k_1 an $t_{i,3}$ stattfindet, erhalten wir $sup(t_{i,0}) = 1$ und $sup(t_{i,3}) = 0$ für alle $i \in \{0, \ldots, m-1\}$. Insbesondere ist das Bild P_i^R von P_i unter R ein Pfad von 1 in 0 in τ, sodass es offensichtlich ein Ereignis $X \in \{X_{i_0}, X_{i_1}, X_{i_2}\}$ geben muss, für das $sig(X) = \mathsf{inp}$ gilt. Außerdem gibt es keinen Pfad in τ, auf dem inp zweimal stattfindet, da τ keine Kante von 0 nach 1 enthält. Somit gibt es für alle $i \in \{0, \ldots, m-1\}$ genau ein $X \in \{X_{i_0}, X_{i_1}, X_{i_2}\}$, sodass $sig(X) = \mathsf{inp}$ gilt. Insbesondere gilt mit $S = \{X \in \mathfrak{U} \mid sig(X) = \mathsf{inp}\}$, dass $|M_i \cap S| = 1$ für alle $i \in \{0, \ldots, m-1\}$. Wie behauptet, ist S somit ein Eins-in-drei-Modell von $(\mathfrak{U}, M)$. □

Wenn A_τ eine Ja-Instanz von τ-Sprachsimulation oder τ-Realisierung ist, dann hat A_τ wenigstens die τ-EZTE. Somit folgt aus Lemma 3.1, dass $(\mathfrak{U}, M)$ eine positive Entscheidung erlaubt, wenn A_τ diese erlaubt. Für die andere Richtung zeigen wir, dass es einen Zeugen für die τ-EZTE *und* die τ-ZTE von A_τ gibt, wenn es ein Eins-in-drei-Modell für $(\mathfrak{U}, M)$ gibt. Die Behandlung der einzelnen Trennungsprobleme von A_τ und die Präsentation der lösenden Regionen ist technisch aufwendig. Der Einfachheit und der Übersichtlichkeit halber werden die lösenden Regionen $R = (sup, sig)$ deshalb gemäß Bemerkung 2.7 implizit definiert, das heißt, über $sup(\bot_0)$ und sig, wobei wir Ereignisse mit der gleichen Signatur $e_\tau \in \tau$ häufig in der Menge $\mathcal{E}_{e_\tau}^R$ zusammenfassen. Weil jeder von $\bot_0$ verschiedene Zustand z_i von A_τ über einen Pfad $z_0 \xrightarrow{e_1} \ldots \xrightarrow{e_i} z_i$ von $z_0 = \bot_0$ nach z_i erreichbar ist, kann $sup(z_i)$ dann induktiv explizit bestimmt werden: $sup(z_j) = \delta_\tau(sup(z_{j-1}), sig(e_j))$ für alle $j \in \{1, \ldots, i\}$.

Die folgenden drei Fakten adressieren sukzessive die Lösbarkeit der Ereignisse aus $\{k_0, k_1\}$ und $\mathfrak{U}$ sowie $\{\ominus_1, \ldots, \ominus_m\}$ und $\{\oplus_0, \ldots, \oplus_m\}$. Um dem Leser die Nachvollzieh-

barkeit der impliziert definierten Regionen zu erleichtern, werden für (fast) alle Regionen konkrete Beispiele am Transitionssystem A_τ aus Abbildung 3.2 graphisch skizziert. Dabei werden für eine entsprechende Region $R = (sup, sig)$ genau die Zustände von A_τ farblich markiert, deren Support-Wert 1 ist. Es ist für jede Kante $z \xrightarrow{e} z'$ von A_τ dann leicht nachvollziehbar, dass $sup(z) \xrightarrow{sig(e)} sup(z')$ eine Kante in τ ist: Wenn z und z' beide markiert sind, gilt $sig(e) \in \{\mathsf{nop}, \mathsf{used}\}$, wenn z und z' beide unmarkiert sind, gilt $sig(e) \in \{\mathsf{nop}, \mathsf{free}\}$ und wenn z markiert und z' unmarkiert ist, dann muss $sig(e) = \mathsf{inp}$ gelten. Weil es keinen Zustandsübergang von 0 nach 1 in τ gibt, tritt der Fall, dass z unmarkiert und z' markiert ist, nicht ein.

Fakt 3.1. *Wenn $(\mathfrak{U}, M)$ ein Eins-in-drei-Modell hat, dann sind k_0 und k_1 τ-lösbar.*

Beweis. Sei S ein Eins-in-drei Modell von $(\mathfrak{U}, M)$.

Die folgende Region $R_1 = (sup_1, sig_1)$ löst (k_0, z) und (k_1, z) für alle $z \in \{\bot_0, \ldots, \bot_m\}$: $sup_1(\bot_0) = 1$ und $\mathcal{E}^{R_1}_{\mathsf{inp}} = \{\oplus_0, \ldots, \oplus_m\}$ und $\mathcal{E}^{R_1}_{\mathsf{nop}} = \{\ominus_0, \ldots, \ominus_m\}$ und $\mathcal{E}^{R_1}_{\mathsf{free}} = E(A_\tau) \setminus (\mathcal{E}^{R_1}_{\mathsf{inp}} \cup \mathcal{E}^{R_1}_{\mathsf{nop}})$.

Sei $i \in \{0, \ldots, m-1\}$ beliebig aber fest. Die folgende Region $R_2 = (sup_2, sig_2)$ löst (k_0, z) für alle $z \in \{t_{i,1}, \ldots, t_{i,5}\} \cup \{h_1, h_2\}$: $sup_2(\bot_0) = 1$ und $\mathcal{E}^{R_2}_{\mathsf{inp}} = \{k_0, X_{i_0}\}$ und $\mathcal{E}^{R_2}_{\mathsf{nop}} = E(A_\tau) \setminus \mathcal{E}^{R_2}_{\mathsf{inp}}$. (Siehe Abbildung 3.3.) Aus der Beliebigkeit von i folgt die τ-Lösbarkeit von k_0.

Die folgende Region $R_3 = (sup_3, sig_3)$ löst α, das heißt (k_1, h_0): $sup_3(\bot_0) = 1$ und $\mathcal{E}^{R_3}_{\mathsf{inp}} = \{k_0\} \cup S$ und $\mathcal{E}^{R_3}_{\mathsf{free}} = \{k_1\}$ und $\mathcal{E}^{R_3}_{\mathsf{nop}} = E(A_\tau) \setminus (\mathcal{E}^{R_3}_{\mathsf{inp}} \cup \mathcal{E}^{R_3}_{\mathsf{free}})$. Diese Region ist (nur deshalb) wohldefiniert, weil es in jedem Gadget $T_0, \ldots, T_{m-1}$ genau ein Variablen-Ereignis mit einer inp-Signatur gibt. Dies ist der Fall, weil S ein Eins-in-drei-Modell ist. (Siehe Abbildung 3.4.)

Die folgende Region $R_4 = (sup_4, sig_4)$ löst (k_1, z) für alle $z \in \{t_{0,4}, \ldots, t_{m-1,4}\} \cup \{h_2\}$: $sup_4(\bot_0) = 1$ und $\mathcal{E}^{R_4}_{\mathsf{inp}} = \{k_1\}$ und $\mathcal{E}^{R_4}_{\mathsf{nop}} = E(A_\tau) \setminus \mathcal{E}^{R_4}_{\mathsf{inp}}$. (Siehe Abbildung 3.3.)

Sei $i \in \{0, \ldots, m-1\}$ beliebig aber fest. Die folgende Region $R_5 = (sup_5, sig_5)$ löst (k_1, z) für alle $z \in \{t_{i,0}, t_{i,1}, t_{i,2}, t_{i,5}\}$: $sup_5(\bot_0) = 1$ und für alle $e \in E(A_\tau)$, wenn $e = k_1$, dann $sig_5(e) = \mathsf{free}$; wenn $e \in \{X_{i_2}, \oplus_m\}$, dann $sig_5(e) = \mathsf{inp}$; wenn $e = \oplus_j$, sodass $j \in \{0, \ldots, m-1\}$ und Variable X_{i_2} kommt nicht in Klausel M_j vor, dann $sig_5(e) = \mathsf{inp}$, andernfalls $sig_5(e) = \mathsf{nop}$. (Siehe Abbildung 3.5.) Weil i beliebig war, folgt die τ-Lösbarkeit von k_1. Dies beweist den Fakt. □

Fakt 3.2. *Wenn $e \in \mathfrak{U}$, dann ist e τ-lösbar.*

Beweis. Sei $i \in \{0, \ldots, m-1\}$ beliebig aber fest. Wir zeigen explizit die τ-Lösbarkeit von X_i. Aus der Beliebigkeit von i folgt dann die Behauptung. Seien $j_0, j_1, j_2 \in \{0, \ldots, m-1\}$ die drei paarweise verschiedenen Indizes, sodass $X_i \in M_\ell$ für alle $\ell \in \{j_0, j_1, j_2\}$.

Die Region R_1 aus Fakt 3.1 löst (X_i, z) für alle $z \in \{\bot_0, \ldots, \bot_m\}$.

Die folgende Region $R_6 = (sup_6, sig_6)$ löst (X_i, z) für alle Zustände z, die Gadgets angehören, die X_i nicht enthalten, das heißt für alle $z \in \{h_0, h_1, h_2\}$ und alle

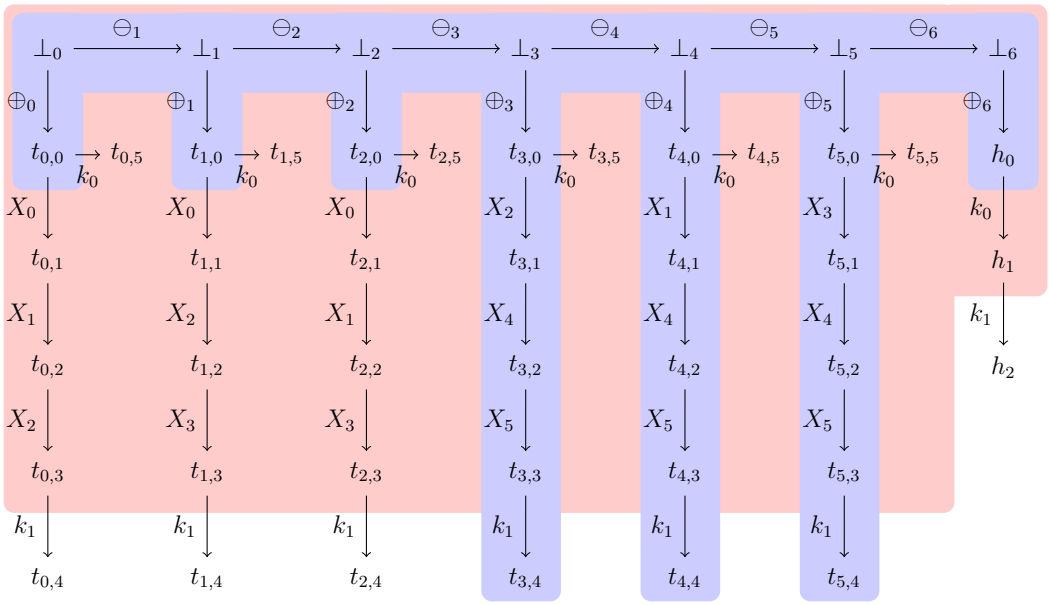


Abbildung 3.3: Der blaue Bereich skizziert die Region R_2 für jedes beliebige aber feste $i \in \{0, 1, 2\}$ und der rote Bereich die Region R_4, die beide für Fakt 3.1 definiert sind.

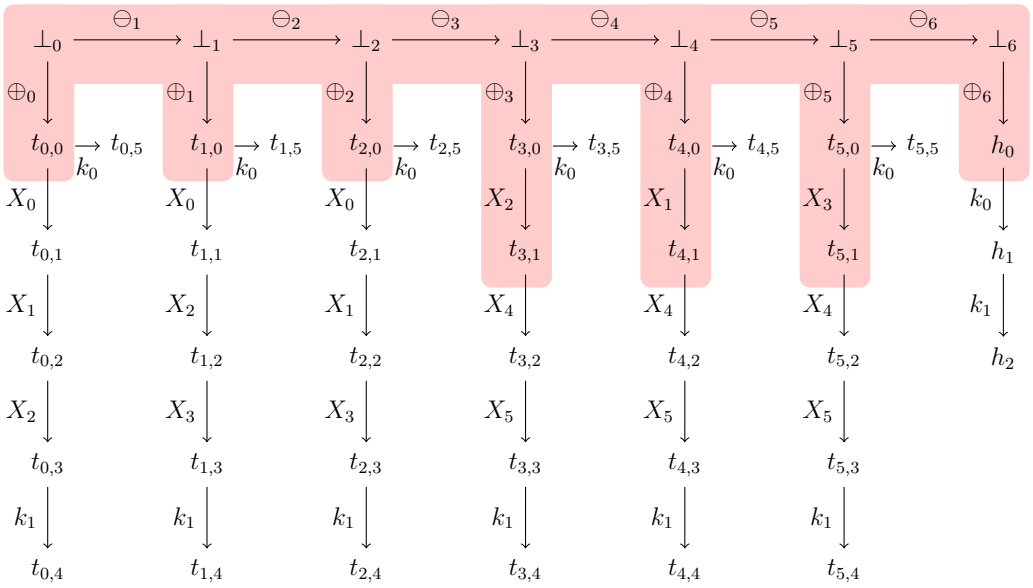


Abbildung 3.4: Eine Skizze der Region R_3 aus Fakt 3.1, die gemäß Beispiel 3.1 aus dem Eins-in-drei-Modell $\{X_0, X_4\}$ resultiert und $\alpha = (k_1, h_0)$ löst.

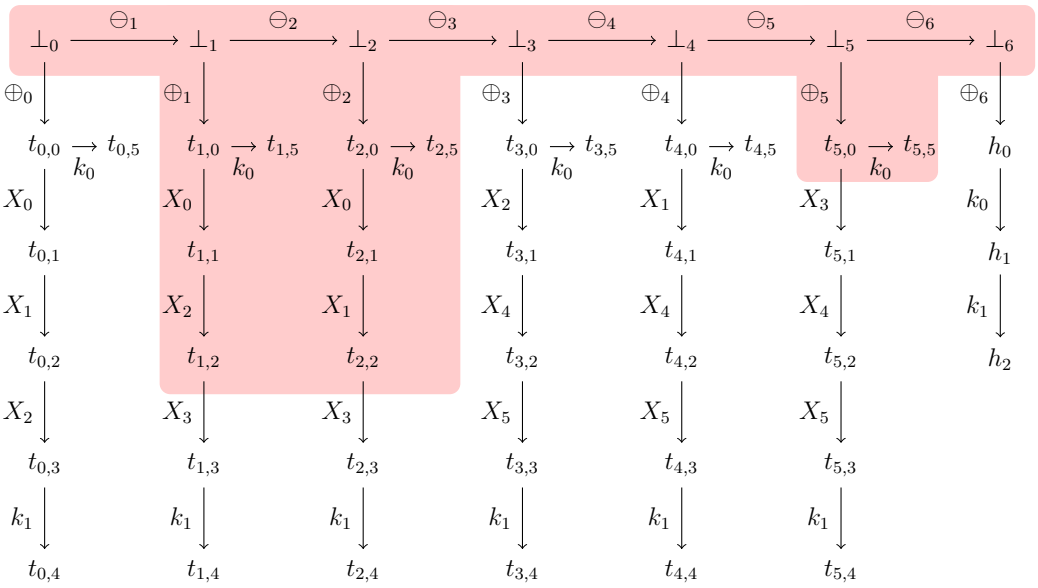


Abbildung 3.5: Der rote Bereich skizziert die Region R_5 aus Fakt 3.1 für den Fall $i = 2$. Der skizzierte Support entspricht auch der Region R_6 aus Fakt 3.2 für den Fall $i = 3$.

$z \in \{t_{\ell,0}, \ldots, t_{\ell,5} \mid \ell \in \{0, \ldots, m-1\} \setminus \{j_0, j_1, j_2\}\}$: $sup_6(\perp_0) = 1$ und $\mathcal{E}^{R_6}_{\mathsf{inp}} = \{X_i\} \cup (\{\oplus_0, \ldots, \oplus_m\} \setminus \{\oplus_{j_0}, \oplus_{j_1}, \oplus_{j_2}\})$ und $\mathcal{E}^{R_6}_{\mathsf{nop}} = E_{A_\tau} \setminus \mathcal{E}^{R_6}_{\mathsf{inp}}$. (Siehe Abbildung 3.5.)

Es bleibt noch die Lösbarkeit von (X_i, z) für alle relevanten Zustände z der Gadgets $T_{j_0}, T_{j_1}, T_{j_2}$ zu zeigen. Sei dazu $\ell \in \{j_0, j_1, j_2\}$ beliebig aber fest und sei $n \in \{0, 1, 2\}$ der eindeutige Index, sodass $X_i = X_{\ell_n}$. Wir zeigen, dass (X_i, z) für alle relevanten Zustände z aus T_ℓ τ-lösbar ist. Aus der Beliebigkeit von ℓ folgt dann die Lösbarkeit von X_i und damit schließlich auch der Fakt.

Die Region $R_7 = (sup_7, sig_7)$ löst (X_i, z) für alle $z \in \{t_{\ell,n+1}, \ldots, t_{\ell,5}\}$: $sup_7(\perp_0) = 1$ und $\mathcal{E}^{R_7}_{\mathsf{inp}} = \{k_0, X_i\}$ und $\mathcal{E}^{R_7}_{\mathsf{nop}} = E(A_\tau) \setminus \mathcal{E}^{R_7}_{\mathsf{inp}}$. (Siehe Abbildung 3.6.)

Falls $n = 0$, dann sind wir fertig. Andernfalls löst die folgende Region $R_8 = (sup_8, sig_8)$ das EZTP (X_i, z) für alle $z \in \{t_{\ell,0}, \ldots, t_{\ell,n-1}\}$: $sup_8(\perp_0) = 1$ und für alle $e \in E(A_\tau)$, wenn $e = X_i$, dann $sig_8(e) = \mathsf{free}$; wenn $e = X_{\ell_{n-1}}$, dann $sig_8(e) = \mathsf{inp}$; falls $e = \oplus_j$ mit $j \in \{j_0, j_1, j_2\} \setminus \{\ell\}$ und $X_{\ell_{n-1}} \notin M_j$, dann $sig_8(e) = \mathsf{inp}$, andernfalls $sig_8(e) = \mathsf{nop}$. (Siehe Abbildung 3.7.) Für diese Region ist wesentlich, dass die Variablen in den Klauseln ihrem Index nach aufsteigend „sortiert“ sind: Wenn $X_{\ell_{n-1}}$ und X_i in demselben Gadget stattfinden, dann findet $X_{\ell_{n-1}}$ stets vor X_i statt. Somit sichert $sig_8(X_{\ell_{n-1}}) = \mathsf{inp}$ dann zu, dass der eindeutige Ursprung von X_i im jeweiligen Gadget auf 0 abgebildet wird. □

Fakt 3.3. *Wenn $e \in \{\ominus_1, \ldots, \ominus_m\} \cup \{\oplus_0, \ldots, \oplus_m\}$, dann ist e τ-lösbar.*

Beweis. Sei $i \in \{0, \ldots, m\}$ beliebig aber fest. Wir beachten, dass $\ominus_0$ und $\ominus_{m+1}$ nicht existieren. Außerdem teilen sich $\oplus_i$ und $\ominus_{i+1}$ den gemeinsamen Ursprung $\perp_i$, wenn $i < m$.

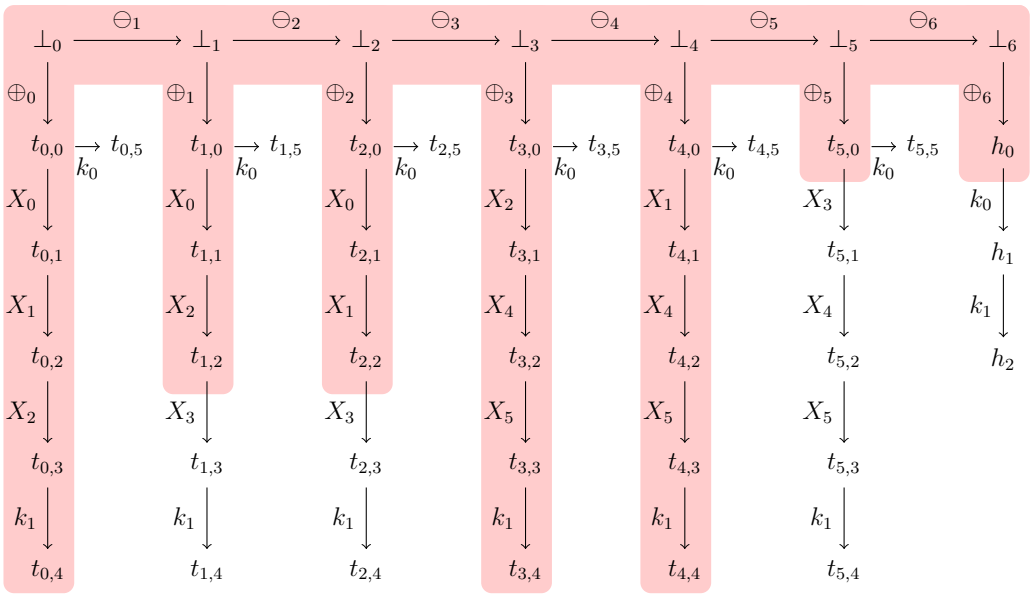


Abbildung 3.6: Der rote Bereich skizziert die Region R_7 für den Fall $i = 3$, $\ell = 2$ und $n = 2$, das heißt, $X_{2_2} = X_3$.

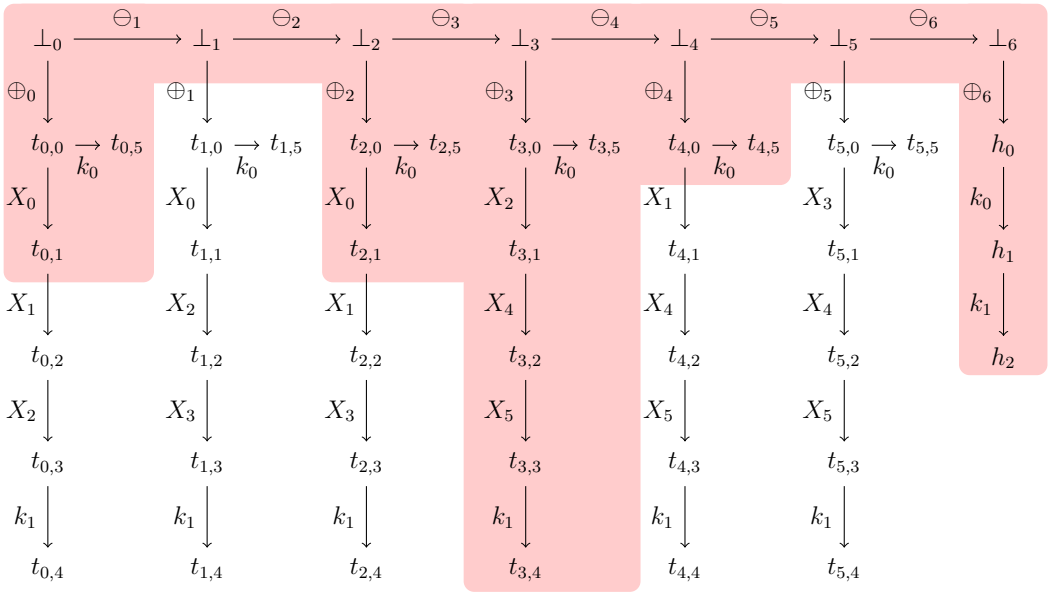


Abbildung 3.7: Der rote Bereich skizziert die Region R_8 für den Fall $i = 3$, $\ell = 2$ und $n = 2$ und löst $(X_3, t_{2,0})$ sowie $(X_3, t_{2,1})$.

Die folgende Region $R_9 = (sup_9, sig_9)$ löst $(\oplus_i, z)$ und wenn $i < m$, dann auch $(\ominus_{i+1}, z)$ für alle $z \in Z(A_\tau) \setminus \{\perp_0, \ldots, \perp_i\}$: $sup_9(\perp_0) = 1$ und wenn $i < m$, dann $\mathcal{E}^{R_9}_{\mathsf{inp}} = \{\ominus_{i+1}\} \cup$

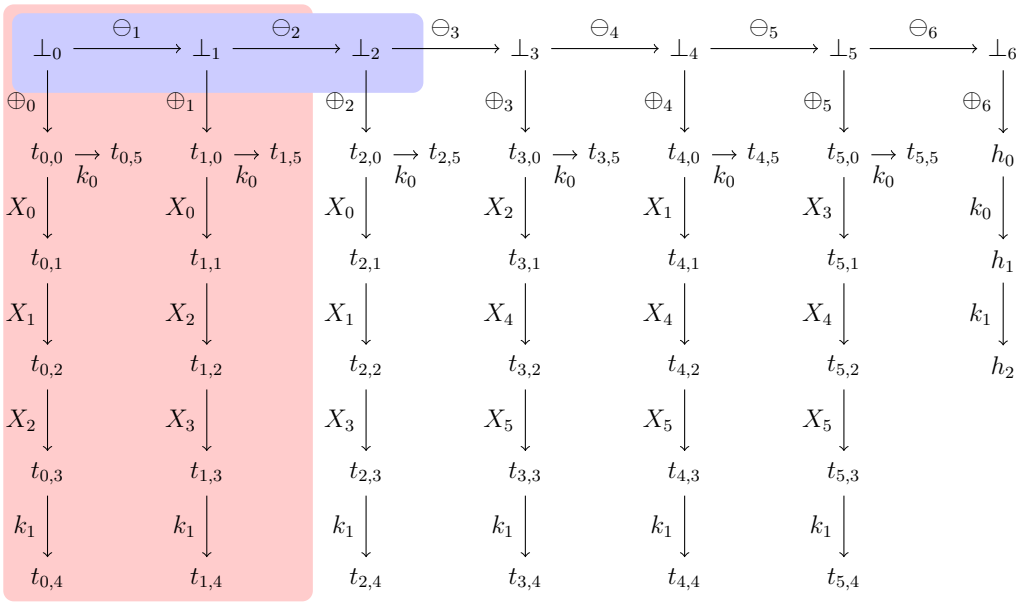


Abbildung 3.8: Der blaue Bereich skizziert die Region R_9 und der rote die Region R_{10} aus Fakt 3.3 für den Fall $i = 2$.

$\{\oplus_0, \ldots, \oplus_i\}$, sonst $\mathcal{E}^{R_9}_{\mathsf{inp}} = \{\oplus_0, \ldots, \oplus_i\}$, und $\mathcal{E}^{R_9}_{\mathsf{nop}} = E(A_\tau) \setminus \mathcal{E}^{R_9}_{\mathsf{inp}}$. (Siehe Abbildung 3.8.)

Weil $\oplus_0$ und $\ominus_1$ an $\bot_0$ stattfinden, bleibt noch die Zustände $\bot_1, \ldots, \bot_{i-1}$ für $i \geq 1$ zu diskutieren. Die folgende Region $R_{10} = (sup_{10}, sig_{10})$ löst $(\oplus_i, z)$ und wenn $i < m$, dann auch $(\ominus_{i+1}, z)$ für alle $z \in \{\bot_0, \ldots, \bot_{i-1}\}$: $sup_{10}(\bot_0) = 1$ und wenn $i < m$, dann $\mathcal{E}^{R_{10}}_{\mathsf{free}} = \{\ominus_{i+1}, \oplus_i\}$, sonst $\mathcal{E}^{R_{10}}_{\mathsf{free}} = \{\oplus_i\}$, und $\mathcal{E}^{R_{10}}_{\mathsf{inp}} = \{\ominus_i\}$ und $\mathcal{E}^{R_{10}}_{\mathsf{nop}} = E(A_\tau) \setminus (\mathcal{E}^{R_{10}}_{\mathsf{free}} \cup \mathcal{E}^{R_{10}}_{\mathsf{inp}})$. (Siehe Abbildung 3.8.) Aus der Beliebigkeit von i folgt der Fakt. □

Gemäß Fakt 3.1 und Fakt 3.2 sowie Fakt 3.3 hat A_τ die EZTE, wenn $(\mathfrak{U}, M)$ ein Eins-in-drei-Modell hat. Darüber hinaus sieht man leicht, dass die Regionen, welche die Fakten beweisen, bereits ebenso die τ-ZTE von A_τ bezeugen. Insgesamt haben wir damit das folgende Lemma bewiesen:

Lemma 3.2. *Wenn $(\mathfrak{U}, M)$ ein Eins-in-drei-Modell hat, dann hat A_τ die τ-EZTE und die τ-ZTE.*

Offensichtlich ist die präsentierte Reduktion polynomiell, sodass wir mit Satz 3.1, Lemma 3.1 und Lemma 3.2 den Satz 3.2 als bewiesen betrachten dürfen.

3.4 Zusammenfassung

In diesem Kapitel haben wir die klassischen Entscheidungsprobleme τ-Einbettung, τ-Sprachsimulation und τ-Realisierung definiert, die für einen Netztypen τ und ein

gegebenes Transitionssystem A fragen, ob es ein τ-Netz N gibt, welches A gemäß der namensgebenden Bezeichnung implementiert. Die Frage nach dem implementierenden Netz ist gemäß Satz 2.1 in die Frage nach einem Zeugen übersetzt, der die Existenz der jeweiligen Implementierung nachweist. Diese Entscheidungsprobleme und die Charakterisierung ihrer Komplexität sind der Hauptgegenstand der restlichen Kapitel dieses Teils der vorliegenden Arbeit. Dazu wurde in diesem Kapitel eine sehr allgemeine Reduktionsstrategie zum Nachweis der NP-Schwere vorgestellt, die auf der Reduktion des Problems CM 1 IN 3 3SAT basiert. Darüber hinaus wurde diese Strategie an zwei konkreten Netztypen beispielhaft durchexerziert und die NP-Vollständigkeit der zugehörigen Probleme τ-SPRACHSIMULATION und τ-REALISIERUNG bewiesen. Alle Reduktionen der folgenden zwei Kapitel orientieren sich bezüglich der zu Grunde liegenden Strategie, der Struktur und der Darstellung an diesem Beispiel.

Kapitel 4

Boolesche Netze

In diesem Kapitel wird die Komplexität der klassischen Syntheseprobleme für boolesche Netztypen untersucht. Dazu führen wir zunächst den Begriff der booleschen Netztypen und der booleschen Interaktionen ein und argumentieren, dass es genau 256 verschiedene boolesche Netztypen gibt. Darüber hinaus zeigen wir, dass die Entscheidungsprobleme τ-Sprachsimulation und τ-Realisierung sowie τ-Einbettung für alle booleschen Netztypen in NP sind. Danach erörtern wir, unter welchen Bedingungen eine Teilmenge von Zuständen eines Transitionsystems als Support einer τ-Region aufgefasst werden kann.

In Abschnitt 4.1 wird dann die Komplexität der Probleme τ-Sprachsimulation und τ-Realisierung für boolesche Netztypen τ untersucht, die Unabhängigkeit zwischen Plätzen und Transitionen erlauben, das heißt, welche die boolesche Interaktion nop enthalten. Wir zeigen unter anderem NP-Vollständigkeit der Probleme für 84 von 128 möglichen Netztypen und polynomielle Entscheidbarkeit für 36 (τ-Sprachsimulation) beziehungsweise 32 (τ-Realisierung) von ihnen. Dieser Abschnitt basiert auf den Publikationen [TR19a, TR19b, Tre19d, Tre21a].

Im darauffolgenden Abschnitt 4.2 wird gezeigt, dass τ-Realisierung für die 64 booleschen Netztypen, welche die Interaktionen nop und swap nicht erlauben, in Polynomialzeit entscheidbar ist. Dieser Abschnitt basiert auf der Publikation [TE20b].

Schließlich charakterisieren wir in Abschnitt 4.3 die Komplexität des Problems τ-Einbettung für alle 256 booleschen Netztypen vollständig. Dieser Abschnitt basiert auf dem Konferenzbeitrag [TE20a].

Gegenstand unserer Untersuchungen ist die Synthese boolescher Netze, deren Definition auf dem Begriff boolescher Netztypen basiert:

Definition 4.1 (Boolescher Netztyp)**.** Ein Netztyp $\tau = (Z_\tau, E_\tau, \delta_\tau)$ heißt *boolescher Netztyp*, wenn $Z_\tau = \{0, 1\}$.

Definition 4.2 (Boolesches Netz)**.** Wenn τ ein boolescher Netztyp ist, dann nennen wir ein τ-Netz N *boolesches Netz*.

x	nop(x)	inp(x)	out(x)	res(x)	set(x)	swap(x)	used(x)	free(x)
0	0		1	0	1	1		0
1	1	0		0	1	0	1	

Abbildung 4.1: Alle booleschen Interaktionen $e \in I$. Eine leere Zelle $e(x)$ bedeutet, dass e für x nicht definiert ist.

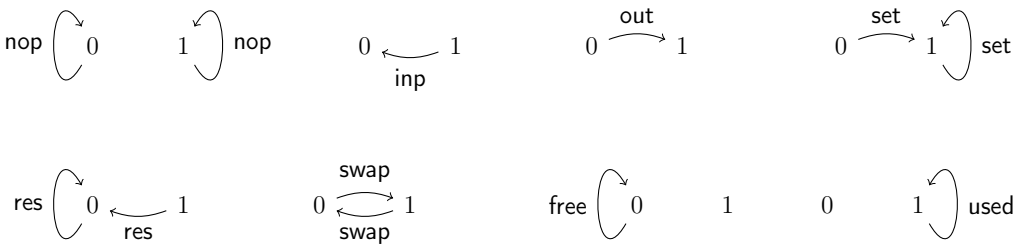


Abbildung 4.2: Für jede boolesche Interaktion $e \in I$ die zugehörigen e-beschrifteten Kanten, die jeder boolesche Netztyp τ mit $e \in E_\tau$ enthält.

Gemäß Definition 2.10 ist jede erreichbare Markierung M eines booleschen Netzes $N = (P, T, f, M_0)$ eine Abbildung $M : P \to \{0, 1\}$. Deshalb wird ein Platz p von N als boolesche Bedingung interpretiert, die wahr ist, wenn $M(p) = 1$ und andernfalls falsch ist.

Einerseits kann jedes Ereignis $e \in E_\tau$ eines booleschen Netztyps $\tau = (\{0, 1\}, E_\tau, \delta_\tau)$ als partielle Funktion $e : \{0, 1\} \to \{0, 1\}$ aufgefasst werden, wobei für alle $z \in \{0, 1\}$ gilt $\delta_\tau(z, e) = e(z)$. Andererseits definiert jede Menge $F = \{f_0, \ldots, f_n\}$ partieller Funktionen von $\{0, 1\}$ nach $\{0, 1\}$ einen booleschen Netztyp $\tau = (\{0, 1\}, F, \delta_\tau)$ mit $\delta_\tau(z, f_i) = f_i(z)$ für alle $i \in \{0, \ldots, n\}$ und alle $z \in \{0, 1\}$. Deshalb werden wir im Folgenden häufig τ mit E_τ identifizieren und $\tau = E_\tau$ für $\tau = (\{0, 1\}, E_\tau, \delta_\tau)$ notieren. Abbildung 4.1 zeigt, außer der nirgends definierten Funktion ndf, die Menge $I = \{\mathsf{nop}, \mathsf{inp}, \mathsf{out}, \mathsf{res}, \mathsf{set}, \mathsf{swap}, \mathsf{used}, \mathsf{free}\}$ aller partieller Funktionen von $\{0, 1\}$ nach $\{0, 1\}$, die wir im Folgenden auch *boolesche Interaktionen* nennen werden. Das Auslassen der nirgends definierten Funktion ndf ist keine wesentliche Einschränkung, da die booleschen Interaktionen (Ereignisse) eines Netztyps dazu dienen, die Beziehung $f(p, t)$ zwischen Platz p und Transition t eines τ-Netzes zu charakterisieren. Ist $f(p, t) = \mathsf{ndf}$, dann kann die Transition t in keiner Markierung schalten und trägt damit nichts zu dem Verhalten des Netzes bei. Folglich können die Transition t als auch der Flusswert $f(p, t)$ entfallen und wir werden im Folgenden annehmen, dass ndf niemals dem Flusswert $f(p, t)$ eines booleschen Netzes entspricht.

Wir identifizieren τ mit E_τ und es gibt insgesamt acht verschiedene boolesche Interaktionen. Jede Teilmenge der Interaktionen definiert einen Netztyp. Deshalb gibt es insgesamt genau $2^8 = 256$ verschiedene boolesche Netztypen.

Boolesche Interaktionen klassifizieren Beziehungen zwischen Plätzen und Transitionen boolescher Netze. Es ist deshalb sinnvoll, für alle booleschen Interaktionen $e \in I$ die

Bedeutung $f(p,t) = e$ auf der Netzebene zu interpretieren.

Die Interaktion nop (engl. *no operation* für *keine Verbindung*) modelliert Unabhängigkeit zwischen p und t, das heißt, weder kann die Markierung von p das Schalten von t verhindern, noch ändert das Schalten von t die Markierung von p.

Es bedeutet inp (engl. *input* für *eingeben*), dass p vor dem Schalten von t markiert und nach dem Schalten von t unmarkiert sein muss. Daher wird p in diesem Fall als *Eingabe*platz für t interpretiert. Umgekehrt bedeutet out (engl. *output* für *ausgeben*), dass p vor dem Schalten von t unmarkiert und nach dem Schalten von t markiert sein muss. Deshalb wird p hier als *Ausgabe*platz für t gedeutet.

Für die Interaktionen res (engl. *reset* für *zurücksetzen*), set (engl. für *setzen*) und swap (engl. für *tauschen*) gilt, dass die Aktivierung von t unabhängig von der aktuellen Markierung von p ist. Jedoch ist p nach dem Schalten von t im ersten Fall unmarkiert, im zweiten markiert und im letzten Fall wird der boolesche Wert (die Markierung) von p invertiert.

Die Interaktionen used (engl. für *belegt*) und free (engl. für *unbelegt*) bedeuten, dass p belegt beziehungsweise frei ist, wenn t schaltet und das Schalten von t ändert die Markierung von p nicht. In diesem Sinne entspricht das Schalten von t dann dem positiven Test auf *wahr* beziehungsweise *falsch* der durch p modellierten booleschen Bedingung.

Gemäß des folgenden Lemmas sind die klassischen Syntheseprobleme für boolesche Netztypen in NP:

Lemma 4.1. *Wenn τ ein boolescher Netztyp ist, dann sind τ-*Einbettung *und τ-*Sprachsimulation *sowie τ-*Realisierung *in NP.*

Beweis. Ein Transitionssystem $A = (Z, E, \delta, \iota)$ hat gemäß Definition 2.21 und Definition 2.22 höchstens $|Z|^2$ ZTP und höchstens $|E| \cdot |Z|$ EZTP. Wenn A die τ-EZTE oder die τ-ZTE hat, dann gibt es demnach einen entsprechende Zeugen $\mathcal{R}$ mit höchstens $|Z|^2 + |E| \cdot |Z|$ τ-Regionen. Weil $|\tau| \leq 8$, ist die Größe jeder dieser Regionen offenbar polynomiell durch die Größe von A beschränkt. Somit kann $\mathcal{R}$ im Fall seiner Existenz von einer nicht-deterministischen Turing-Maschine geraten werden und anschließend in deterministischer Polynomialzeit verifiziert werden. □

In der vorliegenden Arbeit präsentieren wir für eine Reihe boolescher Netztypen polynomielle Synthese-Algorithmen. Diese Algorithmen basieren häufig auf dem Prinzip, zunächst eine Teilmenge von Zuständen des Transitionssystems zu bestimmen, die als Support interpretiert werden kann und anschließend eine passende Signatur zu definieren: Die definierende Besonderheit boolescher Netztypen ist die Zustandsmenge $\{0, 1\}$. Wenn τ ein boolescher Netztyp ist und $A = (Z, E, \delta, \iota)$ ein Transitionssystem und $R = (sup, sig)$ eine τ-Region, dann kann der Support sup daher als charakteristische Funktion der Menge $Q = \{z \in Z \mid sup(z) = 1\}$ aufgefasst werden. Andererseits definiert jede Teilmenge von Zuständen $Q \subseteq Z$ ihre eindeutige charakteristische Funktion $f_Q : Z \to \{0, 1\}$ mit $f_Q(z) = 1$ wenn $z \in Q$, sonst $f_Q(z) = 0$.

Der Bequemlichkeit halber werden wir deshalb im Folgenden solche Funktionen f_Q und Mengen Q miteinander identifizieren. Insbesondere identifizieren wir einen Support $sup : Z \to \{0,1\}$ stets auch mit der Menge $sup = \{z \in Z \mid sup(z) = 1\}$.

Unter Umständen erlaubt eine Teilmenge $Q \subseteq Z$ eine τ-Region $R = (Q, sig)$ mit Support Q. Dies ist nach Definition 2.18 genau dann der Fall, wenn für alle $e \in E$ und alle $z \xrightarrow{e} z' \in A$ gilt, dass $Q(z) \xrightarrow{sig(e)} Q'(z') \in \tau$. Dabei kann eine Menge Q durchaus mehrere Signaturen erlauben. Das wie folgt definierte *Signaturenpotential von* Q kann Aufschluss darüber geben, ob eine entsprechende Signatur für Q existiert:

Definition 4.3 (Signaturpotential)**.** Sei $A = (Z, E, \delta, \iota)$ ein Transitionssystem und $Q \subseteq Z$. Für jedes Ereignis $e \in E$ nennen wir die Menge $S_e^Q = \{(Q(z), Q(z')) \mid z \xrightarrow{e} z' \in A\}$ das *Signaturpotential* von e (bezüglich Q).

Das folgende Lemma zeigt, in welchen Fällen wir von den Signaturpotentialen der Ereignisse eines Transitionssystems auf die Existenz einer passenden Signatur schließen können:

Lemma 4.2. *Sei τ ein boolescher Netztyp und $A = (Z, E, \delta, \iota)$ ein Transitionssystem und $Q \subseteq Z$.*

1. *Wenn es eine Signatur $sig : E \to \tau$ gibt, sodass $R = (Q, sig)$ eine τ-Region ist, dann enthält das Signaturpotential S_e^Q jedes Elements $e \in E$ höchstens zwei Elemente.*

2. *Es gibt genau dann eine Signatur $sig : E \to \tau$, sodass $R = (Q, sig)$ eine τ-Region von A ist, wenn für alle $e \in E$ die folgende Eigenschaft erfüllt ist: Es gibt eine boolesche Interaktion $i \in \tau$, sodass für alle $(x, y) \in S_e^Q$ gilt, dass $x \xrightarrow{i} y \in \tau$.*

Beweis. (1): Sei $R = (Q, sig)$ eine τ-Region von A. Angenommen, es gibt $e \in E$, sodass $|S_e^Q| \geq 3$. Nach Definition 4.3 gibt es also paarweise verschiedene Kanten $p \xrightarrow{e} p'$, $q \xrightarrow{e} q'$ und $z \xrightarrow{e} z'$, sodass die Elemente $(Q(p), Q(p'))$ und $(Q(q), Q(q'))$ sowie $(Q(z), Q(z'))$ paarweise verschieden sind. Weil $Q(s) \in \{0,1\}$ für alle $s \in Z$, muss es zwei Paare geben, deren erster Eintrag übereinstimmt. Nehmen wir ohne Beeinträchtigung der Allgemeinheit an, dass $Q(p) = Q(q)$. Mit $(Q(p), Q(p')) \neq (Q(q), Q(q'))$ impliziert dies $Q(p') \neq Q(q')$. Weil R eine Region ist, gilt $Q(p) \xrightarrow{sig(e)} Q(p') \in \tau$ und $Q(q) \xrightarrow{sig(e)} Q(q') \in \tau$, das heißt, $sig(e)$ entspricht keiner booleschen Funktion. Dies ist ein Widerspruch. Mithin war unsere Annahme falsch und die Behauptung folgt.

(2): Nach Definition einer Region folgt, dass wenn $R = (Q, sig)$ eine Region ist, dann gilt für alle $e \in E$: wenn $z \xrightarrow{e} z'$, dann $Q(z) \xrightarrow{sig(e)} Q'(z) \in \tau$. Mit $i = sig(e)$ haben wir daher eine Interaktion aus τ, welche die Forderung erfüllt.

Für die andere Richtung bezeichne für jedes Ereignis $e \in E$ die Interaktion i_e ein beliebiges aber festes Ereignis von τ, sodass $x \xrightarrow{i_e} y \in \tau$ für alle $(x, y) \in S_e^Q$ und für alle

$e \in E$ sei $sig(e) = i_e$. Nun gilt für alle $e \in E$ und alle $z \xrightarrow{e} z' \in A$: Wenn $z \xrightarrow{e} z' \in A$, dann $(Q(z), Q(z')) \in S_e^Q$ und somit $Q(z) \xrightarrow{sig(e)} Q(z') \in \tau$. Somit ist $R = (Q, sig)$ eine τ-Region. □

Neben der Komplexitäts-Klassifikation der genannten Entscheidungsprobleme in Abhängigkeit von τ ist ein wesentliches Ziel dieser Arbeit verborgene Strukturen beziehungsweise strukturelle Eigenheiten ganzer (Unter-) Klassen von Netztypen soweit als möglich aufzuzeigen. Dafür ist es sinnvoll Interaktionen gemäß ihrer potentiellen Möglichkeit (auf Netzebene) Marken zu konsumieren, zu produzieren, zu erhalten oder ihrer Fähigkeit das Schalten von Transitionen zu verhindern zusammenzufassen:

Notation 4.1. Wir bezeichnen mit

- $\mathfrak{consume}$ (engl. für *konsumieren*) die Menge $\{\mathsf{inp}, \mathsf{res}, \mathsf{swap}\}$,
- $\mathfrak{produce}$ (engl. für *produzieren*) die Menge $\{\mathsf{out}, \mathsf{set}, \mathsf{swap}\}$,
- $\mathfrak{save}_1$ (engl. für *bewahre 1*) die Menge $\{\mathsf{nop}, \mathsf{set}, \mathsf{used}\}$,
- $\mathfrak{save}_0$ (engl. für *bewahre 0*) die Menge $\{\mathsf{nop}, \mathsf{res}, \mathsf{free}\}$,
- $\mathfrak{save}$ die Menge $\mathfrak{save}_1 \cup \mathfrak{save}_0$,
- $\mathfrak{inhibit}$ (engl. für *verhindern*) die Menge $\{\mathsf{inp}, \mathsf{out}, \mathsf{used}, \mathsf{free}\}$.

Wir werden zum Beispiel Folgendes sehen: Wenn τ ein boolescher Netztyp ist, der $\{\mathsf{nop}, \mathsf{res}\} \subseteq \tau$ erfüllt, dann ist τ-SPRACHSIMULATION genau dann NP-vollständig, wenn $\tau \cap \mathfrak{produce} \neq \emptyset$ und $\tau \cap \mathfrak{inhibit} \neq \emptyset$. Andernfalls ist das Problem in Polynomialzeit entscheidbar. Diese Klassifikation umfasst 64 boolesche Netztypen, deren strukturelle Gemeinsamkeit die Präsenz von nop und res ist. Ebenso werden wir für die 64 booleschen Typen mit $\mathsf{nop} \notin \tau$ und $\mathsf{swap} \in \tau$ zeigen, dass τ-EINBETTUNG genau dann NP-vollständig ist, wenn $\tau \cap \mathfrak{save} \neq \emptyset$.

4.1 SPRACHSIMULATION und REALISIERUNG für boolesche Netztypen mit nop

In diesem Abschnitt untersuchen wir die Komplexität von τ-SPRACHSIMULATION und von τ-REALISIERUNG für boolesche Netztypen τ mit nop untersucht. Abbildung 4.3 gibt eine Übersicht über die (soweit bekannten) Ergebnisse in Abhängigkeit des Netztyps. Der Vollständigkeit halber sind auch jene Netztypen ohne nop mit in die Tabelle aufgenommen (Abbildung 4.3 A12-A13). Diese Netztypen werden für REALISIERUNG in Kapitel 4.2 gesondert besprochen.

A	Netztypen τ	Aufwand SM	Aufwand RG	Anz.	Satz
1	$\{\mathsf{nop}, \mathsf{inp}, \mathsf{free}\}$, $\{\mathsf{nop}, \mathsf{inp}, \mathsf{used}, \mathsf{free}\}$, $\{\mathsf{nop}, \mathsf{out}, \mathsf{used}\}$, $\{\mathsf{nop}, \mathsf{out}, \mathsf{used}, \mathsf{free}\}$	NPV		4	3.2, 4.1
2	$\{\mathsf{nop}, \mathsf{inp}, \mathsf{out}\} \cup \omega$ mit $\omega \subseteq \{\mathsf{used}, \mathsf{free}\}$	NPV		4	4.2, 4.3
3	$\{\mathsf{nop}\} \cup \omega$ mit $\omega \subseteq \{\mathsf{used}, \mathsf{free}\}$	P		4	4.4
4	$\{\mathsf{nop}, \mathsf{swap}\} \cup \omega$ mit $\omega \subseteq \{\mathsf{inp}, \mathsf{out}, \mathsf{used}, \mathsf{free}\}$	P		16	4.5
5	$\{\mathsf{nop}, \mathsf{inp}\}$, $\{\mathsf{nop}, \mathsf{inp}, \mathsf{used}\}$, $\{\mathsf{nop}, \mathsf{out}\}$, $\{\mathsf{nop}, \mathsf{out}, \mathsf{free}\}$	offen		4	
6	$\{\mathsf{nop}, \mathsf{set}, \mathsf{res}\} \cup \omega$ mit $\omega \subseteq \{\mathsf{swap}, \mathsf{used}, \mathsf{free}\}$ und $\omega \cap \{\mathsf{used}, \mathsf{free}\} \neq \emptyset$	NPV		6	4.6
7	$\{\mathsf{nop}, \mathsf{inp}, \mathsf{set}\} \cup \omega$ mit $\omega \subseteq \{\mathsf{out}, \mathsf{res}, \mathsf{swap}, \mathsf{used}, \mathsf{free}\}$ $\{\mathsf{nop}, \mathsf{out}, \mathsf{res}\} \cup \omega$ mit $\omega \subseteq \{\mathsf{inp}, \mathsf{set}, \mathsf{swap}, \mathsf{used}, \mathsf{free}\}$	NPV		56	4.7.1
8	$\{\mathsf{nop}, \mathsf{inp}, \mathsf{res}, \mathsf{swap}\} \cup \omega$ oder $\{\mathsf{nop}, \mathsf{out}, \mathsf{set}, \mathsf{swap}\} \cup \omega$ mit $\omega \subseteq \{\mathsf{used}, \mathsf{free}\}$	NPV		8	4.7.2
9	$\{\mathsf{nop}, \mathsf{set}, \mathsf{swap}\} \cup \omega$, $\{\mathsf{nop}, \mathsf{res}, \mathsf{swap}\} \cup \omega$ mit $\emptyset \neq \omega \subseteq \{\mathsf{used}, \mathsf{free}\}$	NPV		6	4.8
10	$\{\mathsf{nop}, \mathsf{res}\} \cup \omega$ mit $\omega \subseteq \{\mathsf{inp}, \mathsf{used}, \mathsf{free}\}$, $\{\mathsf{nop}, \mathsf{set}\} \cup \omega$ mit $\omega \subseteq \{\mathsf{out}, \mathsf{used}, \mathsf{free}\}$	P		16	4.9
11	$\{\mathsf{nop}, \mathsf{set}, \mathsf{swap}\}$, $\{\mathsf{nop}, \mathsf{res}, \mathsf{swap}\}$, $\{\mathsf{nop}, \mathsf{set}, \mathsf{res}\}$, $\{\mathsf{nop}, \mathsf{set}, \mathsf{res}, \mathsf{swap}\}$	P	offen	4	4.10
12	ω mit $\omega \subseteq \{\mathsf{inp}, \mathsf{out}, \mathsf{res}, \mathsf{set}, \mathsf{used}, \mathsf{free}\}$	offen	P	64	4.11
13	$\{\mathsf{swap}\} \cup \omega$ mit $\omega \subseteq \{\mathsf{inp}, \mathsf{out}, \mathsf{res}, \mathsf{set}, \mathsf{used}, \mathsf{free}\}$	offen		64	

Abbildung 4.3: Eine Übersicht über die Komplexität von τ-Sprachsimulation (SM) und τ-Realisierung (RG) in Abhängigkeit vom booleschen Netztyp τ. Kongruente Typen sind jeweils in einer Zeile zusammengefasst, die vierte Spalte entspricht der Anzahl (Anz.) der Netztypen, die fünfte Spalte referenziert auf den Satz mit dem Ergebnis.

Die Tabelle aus Abbildung 4.3 fasst, mittels ihrer Zeilen, verschiedene Netztypen grob gesagt nach wenigstens zwei Kategorien zusammen: Einerseits werden kongruente Netztypen gruppiert, was besonders deutlich wird in Abbildung 4.3 A1, A7, A8, A10, andererseits erlauben die zusammengefassten Netztypen im Wesentlichen den gleichen Beweisansatz –im NP-schweren Fall bedeutet dies insbesondere die gleiche konkrete Reduktion und im polynomiellen Fall den gleichen algorithmischen Ansatz. Darüber hinaus folgen die Darstellungen dem Prinzip $\tau = \tau' \cup \omega$ mit „Basis-Interaktionen“ τ und „Beiwerk-Interaktionen“ ω. Dies Art der Präsentation offenbart wesentliche strukturelle Gemeinsamkeiten beziehungsweise Eigenschaften der jeweiligen Netztypen. So zeigt die Darstellung aus Abbildung 4.3 A7, dass sowohl $\{\mathsf{nop}, \mathsf{inp}, \mathsf{set}\}$ als auch $\{\mathsf{nop}, \mathsf{out}, \mathsf{res}\}$ einen „harten Kern“ bilden: Wenn ein Netztyp τ eine der beiden Mengen enthält, dann sind die zugehörigen Entscheidungsprobleme NP-vollständig, welche anderen Interaktionen τ auch immer enthalten

mag. Es ist aus komplexitäts-theoretischer Sicht beachtenswert, dass dies für den Netztyp der sogenannten *elementaren Netzsysteme*, das heißt $\{\mathsf{nop}, \mathsf{inp}, \mathsf{out}\}$ (Abbildung 4.3 A2), nicht der Fall ist: fügt man zum Beispiel swap zu diesem Netztyp hinzu (Abbildung 4.3 A4), lassen sich SPRACHSIMULATION und REALISIERUNG nun in Polynomialzeit entscheiden. Aus Sicht der Modellierung mag der Netztyp $\{\mathsf{nop}, \mathsf{inp}, \mathsf{out}\}$ also als elementar betrachtet werden können, aus komplexitäts-theoretischer Perspektive scheinen aufgrund dieser Ergebnisse eher der Netztyp $\{\mathsf{nop}, \mathsf{inp}, \mathsf{set}\}$ und sein kongruenter Widerpart $\{\mathsf{nop}, \mathsf{out}, \mathsf{res}\}$ elementar, das heißt bestimmend, für die Komplexität zu sein.

Obwohl die Gruppierungen der Netztypen in Abbildung 4.3 bereits wesentlich von einzelnen Netztypen abstrahieren, wollen wir uns im Folgenden noch weiter von den konkreten Ausprägungen bestimmter Netztypen lösen und durch weitere Verallgemeinerungen zusätzliche Strukturen aufdecken. Wir unterscheiden dabei jene Netztypen ohne res und set von jenen, mit wenigstens einer von diesen Interaktionen. Für die erste Gruppe lässt sich Folgendes feststellen: Wenn τ ein boolescher Netztyp mit nop aber ohne res und set ist, dann gelten die folgenden Aussagen:

1. Wenn $\mathsf{inp} \in \tau$ und $\tau \cap \{\mathsf{out}, \mathsf{free}\} \neq \emptyset$ oder $\mathsf{out} \in \tau$ und $\tau \cap \{\mathsf{inp}, \mathsf{used}\} \neq \emptyset$, dann sind τ-SPRACHSIMULATION und τ-REALISIERUNG NP-vollständig.

2. Wenn $\mathsf{swap} \in \tau$, dann sind τ-SPRACHSIMULATION und τ-REALISIERUNG in Polynomialzeit entscheidbar.

Für die Netztypen τ mit nop und res oder set (oder beiden) lassen sich folgende Zusammenhänge entdecken, wobei man beachte, dass Item 1 alle 76 Netztypen aus Abbildung 4.3 A6-A9 charakterisiert:

1. Wenn $\tau \cap \mathfrak{inhibit} \neq \emptyset$ und $\mathsf{res} \in \tau$ und $\tau \cap \mathfrak{produce} \neq \emptyset$ beziehungsweise $\tau \cap \mathfrak{inhibit} \neq \emptyset$ und $\mathsf{set} \in \tau$ und $\tau \cap \mathfrak{consume} \neq \emptyset$, dann sind τ-SPRACHSIMULATION und τ-REALISIERUNG NP-vollständig.

2. Wenn $\mathsf{res} \in \tau$ und $\tau \cap \mathfrak{produce} = \emptyset$ beziehungsweise wenn $\mathsf{set} \in \tau$ und $\tau \cap \mathfrak{consume} = \emptyset$, dann sind τ-SPRACHSIMULATION und τ-REALISIERUNG in Polynomialzeit entscheidbar.

Der Rest dieses Abschnitts ist dem Beweis der bereits angedeuteten Komplexitäts-Resultate gewidmet.

4.1.1 Die Netztypen A1-A3

Mit Satz 3.2 und dem folgenden Satz folgt bereits die NP-Vollständigkeit der Probleme τ-SPRACHSIMULATION und τ-REALISIERUNG für die Netztypen A1 aus Abbildung 4.3:

Satz 4.1. *Wenn* $\tau = \{\mathsf{nop}, \mathsf{out}, \mathsf{used}\}$ *oder* $\tau = \{\mathsf{nop}, \mathsf{out}, \mathsf{used}, \mathsf{free}\}$*, dann sind sowohl* τ-SPRACHSIMULATION *als auch* τ-REALISIERUNG *NP-vollständig.*

Beweis. Die Behauptung folgt mittels Kongruenz unmittelbar aus Satz 2.2 und Satz 3.2. □

Die folgenden Sätze umfassen die verbleibenden Typen A2 - A3 aus Abbildung 4.3.

Satz 4.2. *Wenn* $\tau = \{\mathsf{nop}, \mathsf{inp}, \mathsf{out}\}$ *oder* $\tau = \{\mathsf{nop}, \mathsf{inp}, \mathsf{out}, \mathsf{used}\}$ *dann sind sowohl* τ-SPRACHSIMULATION *als auch* τ-REALISIERUNG *NP-vollständig.*

Für die elementaren Netzsysteme $\tau = \{\mathsf{nop}, \mathsf{inp}, \mathsf{out}\}$ und deren τ-REALISIERUNG ist das Ergebnis ursprünglich in [BBD97] gezeigt worden. Den Beweis der restlichen Aussagen von Satz 4.2 folgt in Abschnitt 5.1 aus unseren Untersuchungen der puren b-beschränkten P/T-Netztypen, welche dort umfänglich eingeführt werden: Wir zeigen für diese Netztypen, dass die Probleme τ-SPRACHSIMULATION und τ-REALISIERUNG NP-vollständig sind. Die Netztypen $\{\mathsf{nop}, \mathsf{inp}, \mathsf{out}\}$ beziehungsweise $\{\mathsf{nop}, \mathsf{inp}, \mathsf{out}, \mathsf{used}\}$ sind zu den Netztypen der puren 1-beschränkten P/T-Netze beziehungsweise 1-beschränkten P/T-Netze kongruent. Der Beweis von Satz 4.2 folgt deshalb in Abschnitt 5.1 aus dem genannten Ergebnis und wir werden an entsprechender Stelle nochmal ausdrücklich darauf eingehen.

Satz 4.3. *Wenn* $\tau = \{\mathsf{nop}, \mathsf{inp}, \mathsf{out}, \mathsf{free}\}$ *oder* $\tau = \{\mathsf{nop}, \mathsf{inp}, \mathsf{out}, \mathsf{used}, \mathsf{free}\}$ *dann sind* τ-SPRACHSIMULATION *und* τ-REALISIERUNG *NP-vollständig.*

Beweis. Für $\tau = \{\mathsf{nop}, \mathsf{inp}, \mathsf{out}, \mathsf{free}\}$ folgt die Aussage aus Satz 2.2 und Satz 4.2, denn es gilt $\{\mathsf{nop}, \mathsf{inp}, \mathsf{out}, \mathsf{used}\} \equiv \{\mathsf{nop}, \mathsf{inp}, \mathsf{out}, \mathsf{free}\}$.

Sei also $\tau = \{\mathsf{nop}, \mathsf{inp}, \mathsf{out}, \mathsf{used}, \mathsf{free}\}$. Die Aussage des Satzes folgt auch in diesem Fall aus Satz 4.2: Sei $\tau' = \{\mathsf{nop}, \mathsf{inp}, \mathsf{out}, \mathsf{used}\}$ und $A = (Z, E, \delta, \iota)$ ein Transitionssystem. Zum Beweis genügt es zu zeigen, dass ein ZTP α beziehungsweise ein EZTP $\beta = (z, e)$ von A genau dann τ-lösbar ist, wenn es τ' lösbar ist. Dies impliziert, dass sich τ'-SPRACHSIMULATION und τ'-REALISIERUNG trivial auf τ-SPRACHSIMULATION und τ-REALISIERUNG reduzieren lassen.

„⇐“: Wegen $\tau' \subseteq \tau$, ist jede τ'-Region eine τ-Region.

„⇒“: Sei $R = (sup, sig)$ eine τ-Region von A. Wenn R das ZTP α löst, dann ist die folgende τ'-Region $R' = (sup', sig')$ wohldefiniert und löst α, weil für alle $e \in E$ und alle $z \xrightarrow{e} z' \in A$ gilt, wenn $sig(e) = \mathsf{free}$, dann $sup(z) = sup(z')$: Für alle $z \in Z$ und alle $e \in E$ sei $sup'(z) = sup(z)$ und wenn $sig(e) = \mathsf{free}$, dann sei $sig'(e) = \mathsf{nop}$, andernfalls sei $sig'(e) = sig(e)$.

Wenn R das EZTP β löst und $sig(e) \in \{\mathsf{inp}, \mathsf{out}, \mathsf{used}\}$, dann löst die eben definierte τ'-Region R' ebenso β. Somit bleibt nur noch der Fall, dass R das EZTP β löst, sodass $sig(e) = \mathsf{free}$ und $sup(z) = 1$. Die folgende „komplementäre“ τ'-Region $R' = (sup', sig')$ ist wohldefiniert und löst β: Für alle $z \in Z$ und alle $e \in E$ sei $sup'(z) = 1 - sup(z)$; wenn $sig(e) = \mathsf{free}$, dann sei $sig'(e) = \mathsf{used}$; wenn $sig(e) = \mathsf{inp}$, dann sei $sig'(e) = \mathsf{out}$; wenn $sig(e) = \mathsf{out}$, dann sei $sig'(e) = \mathsf{inp}$; andernfalls sei $sig'(e) = \mathsf{nop}$. □

Satz 4.4. *Wenn* $\tau = \{\mathsf{nop}\} \cup \omega$ *mit* $\omega \subseteq \{\mathsf{used}, \mathsf{free}\}$*, dann sind* τ-SPRACHSIMULATION *und* τ-REALISIERBARKEIT *in Polynomialzeit entscheidbar.*

Beweis. Für alle $e \in \tau$ gilt $0 \xrightarrow{e} 1 \notin \tau$ und $1 \xrightarrow{e} 0 \notin \tau$. Deshalb kann ein ZTP nicht τ-lösbar sein. Ein Transitionssystem A hat deshalb genau dann die τ-ZTE beziehungsweise die τ-EZTE, wenn es höchstes einen Zustand besitzt. Dies lässt sich offensichtlich in Polynomialzeit verifizieren. □

4.1.2 Die Netztypen A4

Satz 4.5. *Wenn* $\tau = \{\mathsf{nop}, \mathsf{swap}\} \cup \omega$ *mit* $\omega \subseteq \{\mathsf{inp}, \mathsf{out}, \mathsf{used}, \mathsf{free}\}$*, dann sind sowohl* τ-SPRACHSIMULATION *als auch* τ-REALISIERUNG *in Polynomialzeit entscheidbar.*

Solange nicht ausdrücklich anders vereinbart, sei im weiteren Verlauf dieses Abschnitts $\tau = \{\mathsf{nop}, \mathsf{swap}\} \cup \omega$ mit $\omega \subseteq \{\mathsf{inp}, \mathsf{out}, \mathsf{used}, \mathsf{free}\}$ beliebig aber fest. In diesem Abschnitt werden wir argumentieren, dass τ-SPRACHSIMULATION und τ-REALISIERUNG in Polynomialzeit entscheidbar sind. Dabei sei an dieser Stelle noch einmal ausdrücklich hervorgehoben, dass diese Aussage für den Netztypen $\{\mathsf{nop}, \mathsf{inp}, \mathsf{out}, \mathsf{swap}\}$ (*flip-flop nets*) ursprünglich bereits von Schmitt [Sch96] gezeigt wurde. Der Vollständigkeit halber und um mühsame Fallunterscheidungen zu vermeiden, werden wir in der vorliegenden Arbeit die entsprechenden Beweise dennoch für alle Netztypen A4 präsentieren.

Im Weiteren gehen wir wie folgt vor. Zunächst widmen wir uns der Frage, ob ein Transitionssystem A die τ-ZTE hat. Wir werden zeigen, dass dies genau dann der Fall ist, wenn A die $\{\mathsf{nop}, \mathsf{swap}\}$-ZTE hat. Aus der Beliebigkeit von τ folgt damit, dass sich die Frage nach der τ-ZTE auf die Frage nach der $\{\mathsf{nop}, \mathsf{inp}, \mathsf{out}, \mathsf{swap}\}$-ZTE reduzieren lässt:

$$\begin{aligned} A \text{ hat die } \tau\text{-ZTE} &\iff A \text{ hat die } \{\mathsf{nop}, \mathsf{swap}\}\text{-ZTE} \\ &\iff A \text{ hat die } \{\mathsf{nop}, \mathsf{inp}, \mathsf{out}, \mathsf{swap}\}\text{-ZTE} \end{aligned}$$

Da Letztere nach Schmitt [Sch96] in Polynomialzeit entscheidbar ist, ist auch die Frage nach der τ-ZTE effizient entscheidbar.

Anschließend untersuchen wir die Frage nach der τ-EZTE. Dazu werden wir zunächst zeigen, dass sich die Frage, ob ein EZTP α von A τ-lösbar ist, auf die Frage reduzieren lässt, ob es $\{\mathsf{nop}, \mathsf{inp}, \mathsf{swap}\}$-lösbar beziehungsweise ob es $\{\mathsf{nop}, \mathsf{swap}, \mathsf{used}\}$-lösbar ist. Wir werden des Weiteren argumentieren, dass die $\{\mathsf{nop}, \mathsf{inp}, \mathsf{swap}\}$-Lösbarkeit beziehungsweise die $\{\mathsf{nop}, \mathsf{swap}, \mathsf{used}\}$-Lösbarkeit jeweils in Polynomialzeit entscheidbar ist. Die Polynomialität der τ-EZTE ist dann eine Folge dieses Resultats.

Insgesamt folgen dann aus der polynomiellen Entscheidbarkeit der τ-ZTE und der τ-EZTE die entsprechenden Behauptungen für die τ-SPRACHSIMULATION und die τ-REALISIERUNG. Die Beweise zweier Aussagen dieses Kapitels, nämlich die von Lemma 4.4 und Lemma 4.6, benötigen Begrifflichkeiten die uns an dieser Stelle noch nicht zur Verfügung stehen. Diese Begriffe werden erst in Abschnitt 5.2 für die Untersuchungen der allgemeineren gruppenerweiterten b-beschränkten P/T-Netze entwickelt. Wir werden die Beweise deshalb hier auslassen und in Abschnitt 5.2 nachliefern.

Lemma 4.3. *Sei $\tau = \{\mathsf{nop}, \mathsf{swap}\} \cup \omega$ und $\omega \subseteq \{\mathsf{inp}, \mathsf{out}, \mathsf{used}, \mathsf{free}\}$. Ein Transitionssystem $A = (Z, E, \delta, \iota)$ hat genau dann die τ-ZTE, wenn es die $\{\mathsf{nop}, \mathsf{swap}\}$-ZTE hat.*

Beweis. Sei β ein beliebiges aber festes ZTP von A. Wegen $\{\mathsf{nop}, \mathsf{swap}\} \subseteq \tau$ ist jede $\{\mathsf{nop}, \mathsf{swap}\}$-Region, die β löst, eine τ-Region.

Sei im Gegenzug $R = (sup, sig)$ eine τ-Region, die β löst. Für alle $e \in E$ und alle $z \xrightarrow{e} z' \in A$ gilt, dass wenn $sig(e) \in \{\mathsf{nop}, \mathsf{used}, \mathsf{free}\}$, dann $sup(z) = sup(z')$ und wenn $sig(e) \in \{\mathsf{inp}, \mathsf{out}, \mathsf{swap}\}$, dann $sup(z) \neq sup(z')$. Die folgende $\{\mathsf{nop}, \mathsf{swap}\}$-Region $R' = (sup, sig')$ (mit dem gleichen Support wie R) ist daher wohldefiniert: für alle $e \in E$, wenn $sig(e) \in \{\mathsf{nop}, \mathsf{used}, \mathsf{free}\}$, dann $sig'(e) = \mathsf{nop}$, andernfalls $sig(e) = \mathsf{swap}$. Dies zeigt, dass β genau dann τ-lösbar ist, wenn β $\{\mathsf{nop}, \mathsf{swap}\}$-lösbar ist. Da β beliebig war, folgt die Behauptung. □

Lemma 4.4. *Es gibt einen Algorithmus, der mit Zeitaufwand $\mathcal{O}(|Z|^6 \cdot |E|^3)$ entscheidet, ob A die $\{\mathsf{nop}, \mathsf{inp}, \mathsf{out}, \mathsf{swap}\}$-ZTE hat.*

Die Werkzeuge, um Lemma 4.4 zu beweisen, stehen uns an dieser Stelle noch nicht zur Verfügung. Wir werden jedoch sehen: Für $b = 1$ ist der Netztyp $\{\mathsf{nop}, \mathsf{inp}, \mathsf{out}, \mathsf{swap}\}$ kongruent zu dem Netztypen der $\mathbb{Z}_{b+1}$-erweiterten b-beschränkten P/T-Netze, die in Kapitel 5 eingeführt werden. In Abschnitt 5.2, Lemma 5.18, wird bewiesen, dass für diese Netztypen mit Zeitaufwand $\mathcal{O}(|Z|^6 \cdot |E|^3)$ entschieden werden kann, ob ein Transitionssystem die τ-ZTE hat. Lemma 4.4 ist deshalb eine Folgerung aus Lemma 5.18.

Das folgende Lemma zeigt, dass sich die Frage, ob ein EZTP τ-lösbar ist, auf die Frage reduzieren lässt, ob es $\{\mathsf{nop}, \mathsf{inp}, \mathsf{swap}\}$-lösbar beziehungsweise $\{\mathsf{nop}, \mathsf{swap}, \mathsf{used}\}$-lösbar ist:

Lemma 4.5. *Es sei $A = (Z, E, \delta, \iota)$ ein Transitionssystem und (a, q) ein EZTP von A. Es ist α genau dann τ-lösbar, wenn eine der folgenden Bedingungen erfüllt ist:*

1. *$\tau \cap \{\mathsf{inp}, \mathsf{out}\} \neq \emptyset$ und α ist $\{\mathsf{nop}, \mathsf{inp}, \mathsf{swap}\}$-lösbar.*
2. *$\tau \cap \{\mathsf{used}, \mathsf{free}\} \neq \emptyset$ und α ist $\{\mathsf{nop}, \mathsf{swap}, \mathsf{used}\}$-lösbar.*

Beweis. Sei $R = (sup, sig)$ eine τ-Region von A und sei $z \xrightarrow{e} z' \in A$. Wir überlegen uns, dass nach Definition der booleschen Interaktionen Folgendes gilt: Wenn $sig(e) \in \{\mathsf{used}, \mathsf{free}\}$, dann $sup(z) \xrightarrow{\mathsf{nop}} sup(z')$; wenn $sig(e) \in \{\mathsf{inp}, \mathsf{out}\}$, dann $sup(z) \xrightarrow{\mathsf{swap}} sup(z')$; wenn $sup'(z) = 1 - sup(z)$ und $sup'(z') = 1 - sup(z')$, dann $sup'(z) \xrightarrow{\mathsf{out}} sup'(z')$, falls $sig(e) = \mathsf{inp}$, und $sup'(z) \xrightarrow{\mathsf{inp}} sup'(z')$, falls $sig(e) = \mathsf{out}$; wenn R das EZTP α löst, dann $\neg sup(z) \xrightarrow{sig(e)}$. Diese Beobachtungen implizieren die folgenden Aussagen:

1. Wenn $R = (sup, sig)$ eine Region ist, die α löst, dann löst auch die folgende $\{sig(a), \mathsf{nop}, \mathsf{swap}\}$-Region $R' = (sup', sig')$ α: $sup' = sup$ und für alle $e \in E$, $e = a$, dann $sig'(e) = sig(a)$ und wenn $e \neq a$ und $sig(e) \in \{\mathsf{nop}, \mathsf{used}, \mathsf{free}\}$, dann $sig'(e) = \mathsf{nop}$, sonst $sig'(e) = \mathsf{swap}$

2. Es ist $R = (sup, sig)$ genau dann eine $\{\mathsf{nop}, \mathsf{inp}, \mathsf{swap}\}$-Region von A, wenn $R' = (sup', sig')$ eine $\{\mathsf{nop}, \mathsf{out}, \mathsf{swap}\}$-Region ist, wobei $sup' = Z \setminus sup$ und für alle $e \in E$: $sig(e) = \mathsf{inp} \Leftrightarrow sig'(e) = \mathsf{out}$ und $sig(e) = \mathsf{swap} \Leftrightarrow sig'(e) = \mathsf{swap}$ und $sig(e) = \mathsf{nop} \Leftrightarrow sig'(e) = \mathsf{nop}$ für alle $e \in E$.

3. Es ist $R = (sup, sig)$ genau dann eine $\{\mathsf{nop}, \mathsf{swap}, \mathsf{used}\}$-Region von A, wenn $R' = (sup', sig')$ eine $\{\mathsf{nop}, \mathsf{swap}, \mathsf{free}\}$-Region ist, wobei $sup' = Z \setminus sup$ und für alle $e \in E$: $sig(e) = \mathsf{used} \Leftrightarrow sig'(e) = \mathsf{free}$ und $sig(e) = \mathsf{swap} \Leftrightarrow sig'(e) = \mathsf{swap}$ und $sig(e) = \mathsf{nop} \Leftrightarrow sig'(e) = \mathsf{nop}$.

4. Wenn R eine τ-Region ist, die α löst, dann ist $sig(a) \in \{\mathsf{inp}, \mathsf{out}, \mathsf{used}, \mathsf{free}\}$.

$\Rightarrow$: Löse R nun α. Mit (4) erhalten wir $sig(a) \in \{\mathsf{inp}, \mathsf{out}\}$ oder $sig(a) \in \{\mathsf{used}, \mathsf{free}\}$. Nehmen wir zunächst $sig(a) \in \{\mathsf{inp}, \mathsf{out}\}$ an. Wenn $sig(a) = \mathsf{inp}$, dann erhalten wir mit (1) die $\{\mathsf{nop}, \mathsf{inp}, \mathsf{swap}\}$-Lösbarkeit von α. Andernfalls ist $sig(a) = \mathsf{out}$ und wir erhalten mit (1) eine $\{\mathsf{nop}, \mathsf{out}, \mathsf{swap}\}$-Region, die α löst und daraus anschließend mit (2) eine $\{\mathsf{nop}, \mathsf{inp}, \mathsf{swap}\}$-Region, die α löst.

Die Argumentation für den Fall $sig(a) \in \{\mathsf{used}, \mathsf{free}\}$ ist analog.

$\Leftarrow$: Nehmen wir zunächst an, die erste Bedingung sei erfüllt. Wenn $\{\mathsf{nop}, \mathsf{inp}, \mathsf{swap}\} \subseteq \tau$, dann sind wir fertig. Andernfalls gilt mit $\tau \cap \{\mathsf{inp}, \mathsf{out}\} \neq \emptyset$, dass $\{\mathsf{nop}, \mathsf{out}, \mathsf{swap}\} \subseteq \tau$. Weil α $\{\mathsf{nop}, \mathsf{inp}, \mathsf{swap}\}$-lösbar ist, gibt es eine $\{\mathsf{nop}, \mathsf{inp}, \mathsf{swap}\}$-Region $R = (sup, sig)$, die α löst. Mit (2) erhalten wir daraus eine τ-Region, die α löst. Analog argumentiert man für den Fall, dass die zweite Bedingung erfüllt ist. Die Behauptung folgt. □

Gemäß des folgenden Lemmas lässt sich für ein EZTP in Polynomialzeit enscheiden, ob es $\{\mathsf{nop}, \mathsf{inp}, \mathsf{swap}\}$-lösbar oder $\{\mathsf{nop}, \mathsf{swap}, \mathsf{used}\}$-lösbar ist:

Lemma 4.6. *Es sei $\tau \in \{\{\mathsf{nop}, \mathsf{inp}, \mathsf{swap}\}, \{\mathsf{nop}, \mathsf{swap}, \mathsf{used}\}\}$ beliebig aber fest und $A = (Z, E, \delta, \iota)$ ein Transitionssystem und $\alpha = (a, q)$ ein EZTP von A. Es gibt einen Algorithmus, der mit Zeitaufwand $\mathcal{O}(|Z|^4 \cdot |E|^3)$ entscheidet, ob es eine τ-Region gibt, die α löst.*

Wie bereits eingangs angekündigt, stehen uns die Werkzeuge, um Lemma 4.6 zu beweisen, an dieser Stelle noch nicht zur Verfügung. Wir liefern den entsprechenden Beweis in Abschnitt 5.2.3 (als eine Folgerung von Lemma 5.20) nach.

Das folgende Lemma umfasst nun alle Netztypen A4 aus Abbildung 4.3 und beweist Satz 4.5:

Lemma 4.7. *Sei $\tau = \{\mathsf{nop}, \mathsf{swap}\} \cup \omega$ mit $\omega \subseteq \{\mathsf{inp}, \mathsf{out}, \mathsf{used}, \mathsf{free}\}$. Es gibt einen Algorithmus, der mit Zeitaufwand $\mathcal{O}(|Z|^6 \cdot |E|^4)$ entscheidet, ob A die τ-ZTE oder die τ-EZTE (oder beide) hat.*

Beweis. Die Aussage für die τ-ZTE folgt offenbar aus Lemma 4.3 und Lemma 4.4. Weil A höchstens $|Z| \cdot |E|$ EZTP hat, folgt die Aussage für die τ-EZTE darüber hinaus aus Lemma 4.5 und Lemma 4.6. □

4.1.3 Die Netztypen A6

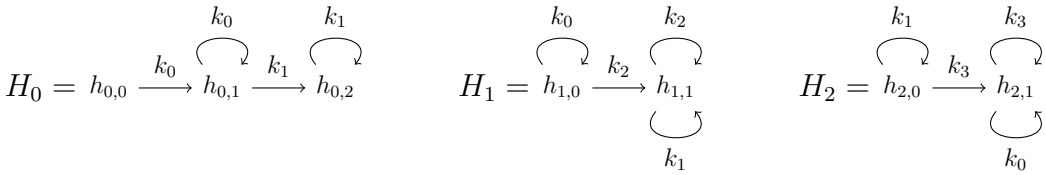
Satz 4.6. *Wenn* $\tau = \{\mathsf{nop}, \mathsf{res}, \mathsf{set}\} \cup \omega$ *mit* $\omega \subseteq \{\mathsf{swap}, \mathsf{used}, \mathsf{free}\}$ *und* $\omega \cap \{\mathsf{used}, \mathsf{free}\} \neq \emptyset$*, dann sind* τ-SPRACHSIMULATION *und* τ-REALISIERUNG *NP-vollständig.*

Der Rest dieses Abschnitts behandelt den Beweis von Satz 4.6, der gemäß der in Kapitel 3 vorgestellten Reduktions-Strategie geführt wird. Bevor wir starten, wollen wir kurz skizzieren, warum wir für die hier behandelten Netztypen eine „neue“ Reduktion brauchen und nicht einfach die verwenden, die wir für das Beispiel in Kapitel 3 verwendet haben: Für die aktuellen Netztypen gilt, dass wenn $R = (sup, sig)$ eine τ-Region ist, die ein EZTP (a, q) löst, dann gilt $sig(a) \in \{\mathsf{used}, \mathsf{free}\}$, denn alle anderen Interaktionen von τ finden sowohl an 0 als auch an 1 statt. Somit gilt für jede Kante $z \xrightarrow{e} z'$ eines Transitionssystems, dass ein EZTP (e, z') nicht τ-lösbar ist, denn $sig(e) \in \{\mathsf{used}, \mathsf{free}\}$ impliziert $sup(z) = sup(z')$. Daraus folgt sofort, dass für Transitionssysteme mit der τ-EZTE gilt, dass für jede Kante $z \xrightarrow{e} z'$ das Ereignis e auch an z' stattfindet. Offenbar erfüllt das Transitionssystem der Reduktion aus Kapitel 3 diese Eigenschaft nicht, sodass diese Reduktion –bezüglich τ– stets Nein-Instanzen produziert. Andererseits sieht man leicht, dass ein (nicht-triviales) Transitionssystem, in welchem für jede Kante $z \xrightarrow{e} z'$ auch $z' \xrightarrow{e}$ gilt, stets eine Nein-Instanz für die Netztypen $\{\mathsf{nop}, \mathsf{inp}, \mathsf{free}\}$ und $\{\mathsf{nop}, \mathsf{inp}, \mathsf{free}, \mathsf{used}\}$ ist. Mit anderen Worten, eine konkrete Reduktion für alle diese Netztypen zu finden ist sicherlich keine leichte Aufgabe.

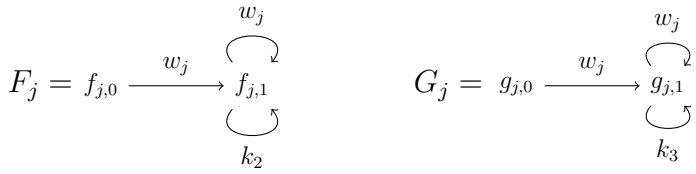
Wir beweisen nun die NP-Schwere der Probleme aus Satz 4.6, wobei wir beachten, dass die Zugehörigkeit zu NP bereits aus Lemma 4.1 folgt. Das angekündigte Transitionssystem A_τ hat die folgende Eigenschaft, die uns ermöglichen wird, die Interaktion swap im Argumentationsverlauf vollständig zu vernachlässigen: Für jedes Ereignis e von A_τ gibt es eine Schleife $z \xrightarrow{e} z$ in A_τ. Tatsächlich impliziert dies mit der Definition von swap unverzüglich $sig(e) \neq \mathsf{swap}$ für alle Ereignisse von A_τ. R ist also genau dann eine τ-Region von A_τ, wenn R eine $(\tau \setminus \{\mathsf{swap}\})$-Region von A_τ ist. Darüber hinaus folgt ähnlich zu den Argumenten im Beweis von Lemma 4.3, dass ein ZTP beziehungsweise ein EZTP eines Transitionssystems genau dann $\{\mathsf{nop}, \mathsf{res}, \mathsf{set}, \mathsf{used}\}$-lösbar ist, wenn es $\{\mathsf{nop}, \mathsf{res}, \mathsf{set}, \mathsf{free}\}$-lösbar beziehungsweise wenn es $\{\mathsf{nop}, \mathsf{res}, \mathsf{set}, \mathsf{used}, \mathsf{free}\}$-lösbar ist. Ohne Beschränkung der Allgemeinheit nehmen wir deshalb im weiteren Verlauf dieses Abschnitts $\tau = \{\mathsf{nop}, \mathsf{res}, \mathsf{set}, \mathsf{used}\}$ an.

Wir stellen nun zunächst die Gadgets des zu konstruierenden Transitionssystems A_τ vor. Im Anschluss werden die Kanten eingeführt, welche die Komponenten zu A_τ verbinden. Dann zeigen wir, dass die τ-Lösbarkeit des angekündigten EZTP α ein Eins-in-drei-Modell von $(\mathfrak{U}, M)$ impliziert. Schließlich zeigen wir für die Rückrichtung, dass A_τ die τ-ZTE und die τ-EZTE hat, wenn $(\mathfrak{U}, M)$ ein Eins-in-drei-Modell hat.

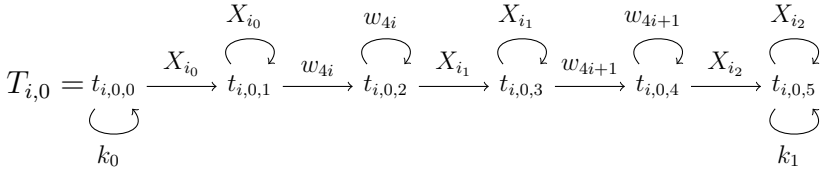
Das Transitionssystem A_τ hat die folgenden Gadgets H_0, H_1 und H_2, wobei H_0 das EZTP $\alpha = (k_0, h_{0,2})$ liefert:

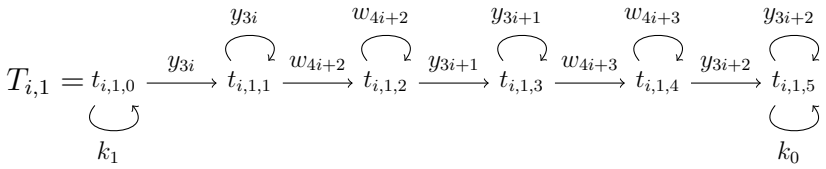


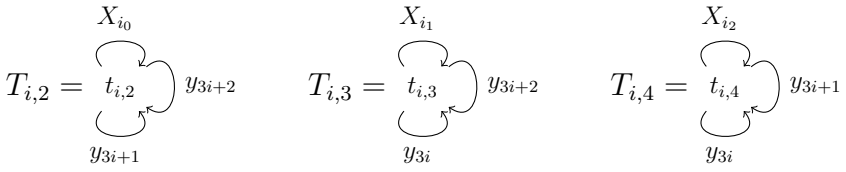
Des Weiteren hat A_τ für jedes $j \in \{0, \ldots, 4m-1\}$ die folgenden Gadgets F_j und G_j:



Es hat A_τ außerdem für jedes $i \in \{0, \ldots, m-1\}$ und Klausel $M_i = \{X_{i_0}, X_{i_1}, X_{i_2}\}$ die folgenden fünf Gadgets $T_{i,0}, T_{i,1} T_{i,2}, T_{i,3}$ und $T_{i,4}$:







Wir nutzen die Zustände der Menge $\perp = \{\perp_0, \ldots, \perp_{13m+2}\}$ und die Ereignisse der Mengen $\ominus = \{\ominus_1, \ldots, \ominus_{13m+2}\}$ sowie $\oplus = \{\oplus_0, \ldots, \oplus_{13m+2}\}$, um die Komponenten von A_τ zu verbinden: Für alle $j \in \{0, \ldots, 13m+1\}$ fügen wir die Kante $\perp_j \xrightarrow{\ominus_{j+1}} \perp_{j+1}$ hinzu. Für alle $j \in \{0, \ldots, 4m-1\}$ werden die Kanten $\perp_j \xrightarrow{\oplus_j} f_{j,0}$ sowie $\perp_{4m+j} \xrightarrow{\oplus_{4m+j}} g_{j,0}$ hinzugefügt. Für alle $i \in \{0, \ldots, m-1\}$, alle $j \in \{0,1\}$ und alle $\ell \in \{2,3,4\}$ fügen wir die Kanten $\perp_{8m+5i+j} \xrightarrow{\oplus_{8m+5i+j}} t_{i,j,0}$ und $\perp_{8m+5i+\ell} \xrightarrow{\oplus_{8m+5i+\ell}} t_{i,\ell}$ hinzu. Außerdem wird für alle $i \in \{0,1,2\}$ die Kante $\perp_{13m+i} \xrightarrow{\oplus_{13m+i}} h_{i,0}$ hinzugefügt. Schließlich fügen wir für alle $e \in \ominus \cup \oplus$ und alle eingeführten (Verbindungs-) Kanten $z \xrightarrow{e} z'$ die Schleife $z' \xrightarrow{e} z'$ hinzu. Das resultierende Transitionssystem mit Initialzustand $\perp_0$ ist A_τ.

Lemma 4.8. *Wenn A_τ die τ-EZTE hat, dann hat $(\mathfrak{U}, M)$ ein Eins-in-drei-Modell.*

Beweis. Das Gadget H_0 liefert das EZTP $\alpha = (k_0, h_{0,2})$. Weil A_τ die τ-EZTE hat, gibt es eine τ-Region von A_τ, die α löst. Sei $R = (sup, sig)$ so eine Region, dass heißt $sig(k_0) = \mathsf{used}$ und $sup(h_{0,2}) = 0$. Es sichern H_0, H_1, H_2 zu, dass $sig(k_2) = \mathsf{res}$ und $sig(k_3) = \mathsf{set}$ gilt: Aus $sig(k_0) = \mathsf{used}$ folgt $sup(h_{0,1}) = sup(h_{1,0}) = sup(h_{2,1}) = 1$ und impliziert mit $sup(h_{0,2}) = 0$, dass $sig(k_1) = \mathsf{res}$. Mit $sig(k_1) = \mathsf{res}$ erhalten wir $sup(h_{1,1}) = sup(h_{2,0}) = 0$. Aus $sup(h_{1,0}) = sup(h_{2,1}) = 1$ und $sup(h_{1,1}) = sup(h_{2,0}) = 0$ folgt schließlich, dass $sig(k_2) = \mathsf{res}$ und $sig(k_3) = \mathsf{set}$.

Für jedes $j \in \{0, \ldots, 4m-1\}$ stellen F_j und G_j mit Hilfe von k_2 und k_3 sicher, dass $sig(w_j) = \mathsf{nop}$: Aus $sig(k_2) = \mathsf{res}$ und $sig(k_3) = \mathsf{set}$ folgt, dass $sup(f_{j,1}) = 0$ und $sup(g_{j,1}) = 1$. Dies impliziert $\xrightarrow{sig(w_j)}0$ sowie $\xrightarrow{sig(w_j)}1$ und somit $sig(w_j) = \mathsf{nop}$.

Sei $i \in \{0, \ldots, m-1\}$ beliebig aber fest. Die Gadgets $T_{i,0}, \ldots, T_{i,4}$ stellen sicher, dass genau ein Variablen-Ereignis aus Klausel M_i eine res-Signatur hat. Gadget $T_{i,0}$ enthält den Pfad

$$P_i = t_{i,0,0} \xrightarrow{X_{i_0}} t_{i,0,1} \xrightarrow{w_{4i}} t_{i,0,2} \xrightarrow{X_{i_1}} t_{i,0,3} \xrightarrow{w_{4i+1}} t_{i,0,4} \xrightarrow{X_{i_2}} t_{i,0,5}.$$

Mit $sig(k_0) = \mathsf{used}$ und $sig(k_1) = \mathsf{res}$ erhalten wir $sup(t_{i,0,0}) = 1$ und $sup(t_{i,0,5}) = 0$. Insbesondere ist das Bild P_i^R von P_i ein Pfad von 1 nach 0 in τ. Deshalb findet ein Ereignis auf P_i statt, dessen Signatur res ist. Weil $sig(w_{4i}) = sig(w_{4i+1}) = \mathsf{nop}$ gilt, gibt es also ein $X \in \{X_{i_0}, X_{i_1}, X_{i_2}\}$ mit $sig(X) = \mathsf{res}$. Analog überlegt man sich für $T_{i,1}$: Aus $sig(k_0) = \mathsf{used}$ und $sig(k_1) = \mathsf{res}$ folgt $sup(t_{i,1,0}) = 0$ und $sup(t_{i,1,5}) = 1$ und daher gibt es ein $y \in \{y_{3i}, y_{3i+1}, y_{3i+2}\}$ mit $sig(y) = \mathsf{set}$. Wir werden nun zeigen, dass X eindeutig ist.

Wenn $X = X_{i_0}$, dass heißt $sig(X_{i_0}) = \mathsf{res}$, dann $sup(t_{i,2}) = 0$. Folglich gilt $sig(y_{3i+1}) \neq \mathsf{set}$ und $sig(y_{3i+2}) \neq \mathsf{set}$. Da es ein $y \in \{y_{3i}, y_{3i+1}, y_{3i+2}\}$ mit $sig(y) = \mathsf{set}$ gibt, schließen wir auf $sig(y_{3i}) = \mathsf{set}$. Dies impliziert $sup(t_{i,3}) = sup(t_{i,4}) = 1$ und daher $sig(X_{i_1}) \neq \mathsf{res}$ und $sig(X_{i_2}) \neq \mathsf{res}$.

Analog folgt aus $sig(X_{i_1}) = \mathsf{res}$, dass $sup(t_{i,3}) = 0$, $sig(y_{3i}) \neq \mathsf{set}$ und $sig(y_{3i+2}) \neq \mathsf{set}$ und somit $sig(y_{3i+1}) = \mathsf{set}$. Daraus folgt $sup(t_{i,2}) = sup(t_{i,4}) = 1$ und daher $sig(X_{i_0}) \neq \mathsf{res}$ und $sig(X_{i_2}) \neq \mathsf{res}$.

Da $sig(X_{i_0}) = \mathsf{res}$ und $sig(X_{i_1}) = \mathsf{res}$ jeweils implizieren, dass $sig(X_{i_2}) \neq \mathsf{res}$, erhalten wir aus der Annahme $sig(X_{i_2}) = \mathsf{res}$ sofort $sig(X_{i_0}) \neq \mathsf{res}$ und $sig(X_{i_1}) \neq \mathsf{res}$.

Insgesamt folgt, dass es genau ein $X \in M_i$ mit $sig(X) = \mathsf{res}$ gibt. Aus der Beliebigkeit von i schließen wir, dass $S = \{X \in \mathfrak{U} \mid sig(X) = \mathsf{res}\}$ ein Eins-in-drei-Modell von $(\mathfrak{U}, M)$ ist. □

Für die andere Richtung zeigen wir, dass ein Eins-in-drei-Modell $(\mathfrak{U}, M)$ die τ-EZTE und die τ-ZTE für A_τ impliziert. Solange nicht ausdrücklich anders vereinbart, sei S im Folgenden also ein solches Modell. Wir argumentieren zunächst, dass A_τ die τ-EZTP hat. Der Beweis ist technisch aufwendig. Um den Aufwand zu verringern, werden wir einige Vereinbarungen treffen beziehungsweise Eigenschaften von A_τ ausnutzen: Sei Z_ι definiert als die Menge der Startzustände der Gadgets, dass heißt $Z_\iota = \{h_{0,0}, h_{1,0}, h_{2,0}\} \cup \{f_{j,0}, g_{j,0} \mid j \in \{0, \ldots, 4m-1\} \cup \{t_{i,0,0}, t_{i,1,0}, t_{i,2}, t_{i,3}, t_{i,4} \mid i \in \{0, \ldots, m-1\}\}$, und sei $\underline{Z} = Z(A_\tau) \setminus \perp$

und sei $\underline{E} = E(A_\tau) \setminus (\ominus \cup \oplus)$. Es ist leicht zu sehen, dass alle EZTP (e, z) von A_τ mit $e \in \ominus \cup \oplus$ τ-lösbar sind. Wir werden diese hier deshalb nicht explizit diskutieren. Des Weiteren werden wir eine Region $R = (sup, sig)$, die ein EZTP (a, q) löst, stets implizit mittels des Supports der Startzustände der Gadgets und der Signatur der Ereignisse aus $\underline{E}$ definieren. Für solche Regionen wird immer $sig(a) = \mathsf{used}$ und $sup(q) = 0$ gelten. Deshalb kann R auf naheliegende Weise auf $\bot$ und $\ominus \cup \oplus$ fortgesetzt werden: Definiere $sup(z) = 0$ und $sig(e) = \mathsf{nop}$ für alle $z \in \bot$ und alle $e \in \ominus$. Für alle $e \in \oplus$ und die eindeutige (Verbindungs-) Kante $z \xrightarrow{e} z'$ mit $z \neq z'$ in A_τ definiere $sig(e) = \mathsf{set}$, wenn $sup(z') = 1$, sonst $sig(e) = \mathsf{nop}$. Man bemerke, dass diese Fortsetzung gleichzeitig (a, z) für alle $z \in \bot$ löst. Da für jedes Ereignis $e \in \underline{E}$ wenigstens ein EZTP (e, z) mit $z \in \underline{Z}$ gelöst wird, impliziert dieses Vorgehen, dass das EZTP (e, z) für alle $e \in \underline{E}$ und alle $z \in \bot$ lösbar ist. Ebenso erkennt man leicht, dass ein beliebiges aber fest gewähltes $e \in \underline{E}$ von den Zuständen jener Gadgets trennbar ist, in denen es an keinem Zustand stattfindet: Eine entsprechende Region von A_τ bildet alle Zustände von Gadgets, in denen e stattfindet, auf 1 ab und alle anderen auf 0; außerdem wird e auf used abgebildet und alle anderen Ereignisse aus $\underline{E} \setminus \{e\}$ auf nop. Diese Konstruktion lässt sich, wie eben beschrieben, auf $\bot$ und $\ominus \cup \oplus$ fortsetzen. Wir werden uns deshalb im Folgenden auf EZTP (e, z) beschränken, für die gilt, dass e und z im gleichen Gadget stattfinden. Neben den bereits genannten Restriktionen bezüglich der Darstellung einer τ-Region $R = (sup, sig)$, werden wir für eine noch kompaktere und gleichzeitig übersichtlichere Darstellung der Signatur die Ereignisse aus $\underline{E}$ mit gleichem Signaturwert wenn möglich wieder zusammenfassen: $\mathcal{E}^R_{e_\tau} = \{e \in \underline{E} \mid sig(e) = e_\tau\}$.

Fakt 4.1. *Die Ereignisse k_0, k_1, k_2 und k_3 sind τ-lösbar.*

Beweis. Die folgende Region $R_1 = (sup_1, sig_1)$ nutzt das Eins-in-drei-Modell S und löst $\alpha = (k_0, h_{0,2})$ sowie (k_0, z) für alle $z \in \{h_{1,1}, h_{2,0}\}$: $sup_1(z) = 1$ für alle $z \in \{h_{0,0}, h_{1,0}\} \cup \{g_{j,0} \mid j \in \{0 \ldots, 4m-1\}\}$ und $sup_1(z) = 0$ für alle $z \in \{f_{j,0} \mid j \in \{0 \ldots, 4m-1\}\}$; für alle $i \in \{0, \ldots, m-1\}$, $sup_1(t_{i,0,0}) = 1$ und $sup_1(t_{i,1,0}) = 0$ und falls $X_{i_0} \in S$, dann $sup_1(t_{i,3}) = sup_1(t_{i,4}) = 1$ und $sup_1(t_{i,2}) = 0$, falls $X_{i_1} \in S$, dann $sup_1(t_{i,2}) = sup_1(t_{i,4}) = 1$ und $sup_1(t_{i,3}) = 0$ und falls $X_{i_2} \in S$, dann $sup_1(t_{i,2}) = sup_1(t_{i,3}) = 1$ und $sup_1(t_{i,4}) = 0$; außerdem $\mathcal{E}^{R_1}_{\mathsf{used}} = \{k_0\}$ und $\mathcal{E}^{R_1}_{\mathsf{res}} = \{k_1, k_2\} \cup S$ und $\mathcal{E}^{R_1}_{\mathsf{set}} = \{k_3\} \cup \{y_{3i+j} \mid i \in \{0, \ldots, m-1\}, j \in \{0,1,2\} : X_{i_j} \in S\}$ und $\mathcal{E}^{R_1}_{\mathsf{nop}} = \underline{E} \setminus (\mathcal{E}^{R_1}_{\mathsf{used}} \cup \mathcal{E}^{R_1}_{\mathsf{res}} \cup \mathcal{E}^{R_1}_{\mathsf{set}})$.

Sei $i \in \{0, \ldots, m-1\}$ beliebig aber fest. Die folgende Region $R_2 = (sup_2, sig_2)$ löst (k_0, z) für alle $z \in \{t_{i,0,1}, \ldots, t_{i,0,5}\}$: Für alle $z \in Z_\iota$, falls es $j \in \{0, \ldots, m-1\}$ und $\ell \in \{2,3,4\}$ gibt, sodass $z = t_{j,\ell}$ und $z \xrightarrow{X_{i_0}}$, dann $sup_2(z) = 0$, sonst $sup_2(z) = 1$; außerdem $\mathcal{E}^{R_2}_{\mathsf{used}} = \{k_0\}$ und $\mathcal{E}^{R_2}_{\mathsf{res}} = \{X_{i_0}\}$ und $\mathcal{E}^{R_2}_{\mathsf{nop}} = \underline{E} \setminus (\mathcal{E}^{R_2}_{\mathsf{used}} \cup \mathcal{E}^{R_2}_{\mathsf{res}})$.

Die folgende Region $R_3 = (sup_3, sig_3)$ löst (k_0, z) für alle $z \in \{t_{i,1,0}, \ldots, t_{i,1,4}\}$: Für alle $z \in Z_\iota$, falls $z = t_{i,1,0}$, dann $sup(z) = 0$, sonst $sup(z) = 1$; außerdem $\mathcal{E}^{R_3}_{\mathsf{used}} = \{k_0\}$ und $\mathcal{E}^{R_3}_{\mathsf{set}} = \{y_{3i+2}\}$ und $\mathcal{E}^{R_3}_{\mathsf{nop}} = \underline{E} \setminus (\mathcal{E}^{R_3}_{\mathsf{used}} \cup \mathcal{E}^{R_3}_{\mathsf{set}})$.

Weil i beliebig war, folgt die Lösbarkeit von k_0. Die Lösbarkeit von k_1 wird ähnlich zur Lösbarkeit von k_0 gezeigt und es ist leicht zu sehen, dass sowohl k_2 als auch k_3 lösbar

sind. Wir verzichten deshalb auf eine explizite Diskussion dieser Ereignisse. □

Fakt 4.2. *Wenn $e \in \mathfrak{U} \cup \{w_0, \ldots, w_{4m-1}\} \cup \{y_0, \ldots, y_{3m-1}\}$, dann ist e τ-lösbar.*

Beweis. Wir diskutieren zunächst explizit den Fall, dass $e \in \mathfrak{U}$. Seien $i \in \{0, \ldots, m-1\}$ und $j \in \{0, 1, 2\}$ beliebig aber fest.

Die folgende Region $R_1 = (sup_1, sig_1)$ ist nur für $j < 2$ notwendig (und wohldefiniert) und löst $\{X_{i_j}, z\}$ für alle $z \in \{t_{i,0,2j+2}, \ldots, t_{i,0,5}\}$: $sup_1(z) = 1$ für alle $z \in Z_\iota$ und $\mathcal{E}^{R_1}_{\mathsf{used}} = \{X_{i_j}\}$ und $\mathcal{E}^{R_1}_{\mathsf{res}} = \{w_{4i+j}\}$ und $\mathcal{E}^{R_1}_{\mathsf{nop}} = \underline{E} \setminus (\mathcal{E}^{R_1}_{\mathsf{used}} \cup \mathcal{E}^{R_1}_{\mathsf{res}})$.

Die folgende Region $R_2 = (sup_2, sig_2)$ ist nur für $j \geq 1$ notwendig (und wohldefiniert) und löst $\{X_{i_j}, z\}$ für alle $z \in \{t_{i,0,0}, \ldots, t_{i,0,2j-1}\}$: Für alle $z \in Z_\iota$, falls $z = t_{i,0,0}$, dann $sup_2(z) = 0$, sonst $sup_2(z) = 1$; außerdem $\mathcal{E}^{R_2}_{\mathsf{used}} = \{X_{i_j}\}$ und $\mathcal{E}^{R_2}_{\mathsf{set}} = \{w_{4i+j-1}\}$ und $\mathcal{E}^{R_2}_{\mathsf{nop}} = \underline{E} \setminus (\mathcal{E}^{R_2}_{\mathsf{used}} \cup \mathcal{E}^{R_2}_{\mathsf{set}})$.

Aus der Beliebigkeit von i und j folgt die Lösbarkeit der Ereignisse aus $\mathfrak{U}$. Analog zeigt man für beliebige aber feste $i \in \{0, \ldots, m-1\}$ und $j \in \{0, 1, 2\}$ beziehungsweise $\ell \in \{0, \ldots, 3\}$ die Lösbarkeit von y_{3i+j} sowie $w_{4i+\ell}$ bezüglich der entsprechenden Zustände aus $T_{i,0}, T_{i,1}$, je nachdem. Dabei beachten wir, dass die Lösbarkeit der Ereignisse an Zuständen außerhalb dieser Gadgets (in denen sie nicht stattfinden), bereits oben begründet wurde. □

Insgesamt zeigen die bisherigen Überlegungen, dass A_τ τ-EZTE hat. Man überlegt sich außerdem, dass die vorgestellten Regionen insgesamt bereits jedes ZTP (z, z') von A_τ lösen, für das z und z' in demselben Gadget von A_τ stattfinden. Es ist leicht zu sehen, dass ein ZTP (z, z') von A_τ lösbar ist, wenn z und z' in verschiedenen Gadgets stattfinden. Mithin erhalten wir das folgende Lemma, das den Beweis von Satz 4.6 vervollständigt:

Lemma 4.9. *Wenn es ein Eins-in-drei-Modell für $(\mathfrak{U}, M)$ gibt, dann gibt es einen Zeugen für die τ-EZTE und die τ-ZTE von A.*

4.1.4 Die Netztypen A7-A8

Satz 4.7. τ-Sprachsimulation *und* τ-Realisierung *sind NP-vollständig, wenn τ ein boolescher Netztyp ist, der eine der folgenden Bedingungen erfüllt:*

1. $\tau \supseteq \{\mathsf{nop}, \mathsf{inp}, \mathsf{set}\}$ *oder* $\tau \supseteq \{\mathsf{nop}, \mathsf{out}, \mathsf{res}\}$*;*
2. $\tau = \{\mathsf{nop}, \mathsf{inp}, \mathsf{res}, \mathsf{swap}\} \cup \omega$ *oder* $\tau = \{\mathsf{nop}, \mathsf{out}, \mathsf{set}, \mathsf{swap}\} \cup \omega$ *mit jeweils* $\omega \subseteq \{\mathsf{used}, \mathsf{free}\}$

Im weiteren Verlauf dieses Abschnitts wird Satz 4.7 bewiesen. Dabei folgen wir der in Kapitel 3 vorgestellten Strategie. Es seien im Folgenden τ_0, τ_1 und τ_2 boolesche Netztypen, sodass $\{\mathsf{nop}, \mathsf{inp}, \mathsf{set}\} \subseteq \tau_0$ und $\mathsf{swap} \notin \tau_0$ sowie $\{\mathsf{nop}, \mathsf{inp}, \mathsf{set}, \mathsf{swap}\} \subseteq \tau_1$ und $\tau_2 = \{\mathsf{nop}, \mathsf{inp}, \mathsf{res}, \mathsf{swap}\} \cup \omega$ mit $\omega \subseteq \{\mathsf{used}, \mathsf{free}\}$. Wir zeigen die NP-Vollständigkeit von τ-Sprachsimulation und τ-Realisierung für alle $\tau \in \{\tau_0, \tau_1, \tau_2\}$. Jeder von Satz 4.7

umfasste Netztyp ist kongruent zu einem Netztyp, der durch τ_0, τ_1 oder τ_2 definiert ist. Der Beweis des Satzes wird daher durch die Aussage von Satz 2.2 komplettiert.

Im Folgenden führen wir sukzessive die Komponenten von A_{τ_0}, A_{τ_1} sowie A_{τ_2} ein. Wir starten mit den Komponenten von A_{τ_0}, welche auch in A_{τ_1} und A_{τ_2} enthalten sind. Darüber hinaus enthalten A_{τ_1} und A_{τ_2} jeweils weitere Komponenten, die den Besonderheiten von τ_1 beziehungsweise τ_2 geschuldet sind. Danach werden die Kanten eingeführt, welche die jeweiligen Gadgets von A_τ miteinander verbinden. Anschließend argumentieren wir (gleichzeitig) für alle $\tau \in \{\tau_0, \tau_1, \tau_2\}$, dass eine τ-Region von A_τ, die α löst, ein Eins-in-drei-Modell von $(\mathfrak{U}, M)$ impliziert. Schließlich argumentieren wir, dass A_τ sowohl die τ-EZTE als auch die τ-ZTE hat.

Solange nicht ausdrücklich anders vereinbart, sei im Folgenden $\tau \in \{\tau_0, \tau_1, \tau_2\}$ beliebig aber fest. Das Transitionssystem A_τ hat das folgende Gadget H, welches das EZTP $\alpha = (k, h_2)$ liefert:

$$H = \; h_0 \xrightarrow{\;k\;} h_1 \xrightarrow{\;w\;} h_2 \xrightarrow{\;y\;} h_3 \xrightarrow{\;k\;} h_4$$

Des Weiteren enthält die Ereignismenge von A_τ für jedes $i \in \{0, \ldots, m-1\}$ außer dem Ereignis X_i aus $\mathfrak{U} = \{X_0, \ldots, X_{m-1}\}$ auch das Ereignis x_i aus der Menge $\mathfrak{u} = \{x_0, \ldots, x_{m-1}\}$, das mit X_i korrespondiert. Genauer gesagt enthält A_τ für jede Kante $z \xrightarrow{X_i} z'$ die korrespondierende Kante $z' \xrightarrow{x_i} z$ mit entgegengesetzter Richtung. Die Umkehrung gilt im Allgemeinen nicht. Für jedes $i \in \{0, \ldots, m-1\}$ hat das Transitionssystem A_τ die folgenden drei Gadgets $T_{i,0}, T_{i,1}$ und $T_{i,2}$, welche die Variablen der Klausel $M_i = \{X_{i_0}, X_{i_1}, X_{i_2}\}$ und deren korrespondierende Ereignisse verwenden:

$$T_{i,0} = \; t_{i,0,0} \xrightarrow{\;k\;} t_{i,0,1} \underset{x_{i_0}}{\overset{X_{i_0}}{\rightleftarrows}} t_{i,0,2} \underset{x_{i_1}}{\overset{X_{i_1}}{\rightleftarrows}} t_{i,0,3} \underset{x_{i_2}}{\overset{X_{i_2}}{\rightleftarrows}} t_{i,0,4} \xrightarrow{\;w\;} t_{i,0,5} \xrightarrow{\;k\;} t_{i,0,6}$$

$$T_{i,1} = \; t_{i,1,0} \xrightarrow{\;k\;} t_{i,1,1} \underset{x_{i_1}}{\overset{X_{i_1}}{\rightleftarrows}} t_{i,1,2} \underset{x_{i_2}}{\overset{X_{i_2}}{\rightleftarrows}} t_{i,1,3} \underset{x_{i_0}}{\overset{X_{i_0}}{\rightleftarrows}} t_{i,1,4} \xrightarrow{\;w\;} t_{i,1,5} \xrightarrow{\;k\;} t_{i,1,6}$$

$$T_{i,2} = \; t_{i,2,0} \xrightarrow{\;k\;} t_{i,2,1} \underset{x_{i_2}}{\overset{X_{i_2}}{\rightleftarrows}} t_{i,2,2} \underset{x_{i_0}}{\overset{X_{i_0}}{\rightleftarrows}} t_{i,2,3} \underset{x_{i_1}}{\overset{X_{i_1}}{\rightleftarrows}} t_{i,2,4} \xrightarrow{\;w\;} t_{i,2,5} \xrightarrow{\;k\;} t_{i,2,6}$$

Es enthält A_{τ_0} keine weiteren Gadgets. Für jedes $i \in \{0, \ldots, m-1\}$ enthält A_{τ_1} das folgende Gadget F_i mit dem Variablen-Ereignis X_i und A_{τ_2} das Gadget G_i mit dem Ereignis x_i, das mit X_i korrespondiert:

$$F_i = \quad \begin{array}{ccc} f_{i,0} & \xrightarrow{X_i} & f_{i,1} \\ {\scriptstyle k}\downarrow & & \downarrow{\scriptstyle k} \\ f_{i,3} & & f_{i,2} \end{array} \qquad\qquad G_i = \quad \begin{array}{ccc} g_{i,0} & \xrightarrow{u_i} & g_{i,1} \\ {\scriptstyle k}\downarrow & & \downarrow{\scriptstyle k} \\ g_{i,3} & \xleftarrow[x_i]{} & g_{i,2} \end{array}$$

Es enthalten A_{τ_1} und A_{τ_2} keine weiteren Gadgets. Wir nutzen die Zustände aus $\bot = \{\bot_0, \ldots, \bot_{4m}\}$ und die Ereignisse aus $\ominus = \{\ominus_1, \ldots, \ominus_{4m}\}$ sowie $\oplus = \{\oplus_0, \ldots, \oplus_{4m}\}$ um die Gadgets von A_τ zu verbinden: Für jedes $i \in \{0, \ldots, 3m-1\}$, falls $\tau = \tau_0$, beziehungsweise für alle $i \in \{0, \ldots, 4m-1\}$, falls $\tau \in \{\tau_1, \tau_2\}$, fügen wir die Kante $\bot_i \xrightarrow{\ominus_{i+1}} \bot_{i+1}$ hinzu. Darüber hinaus fügen wir für jedes $i \in \{0, \ldots, m-1\}$ und jedes $\ell \in \{0,1,2\}$ die Kante $\bot_{3i+\ell} \xrightarrow{\oplus_{3i+\ell}} t_{i,\ell}$ hinzu. Schließlich wird noch die Kante $\bot_{3m} \xrightarrow{\oplus_{3m}} h_0$ hinzugefügt. Für A_{τ_0} sind wir fertig. Für jedes $i \in \{0, \ldots, m-1\}$ fügen wir des Weiteren die Kante $\bot_{3m+i+1} \xrightarrow{\oplus_{3m+i+1}} f_{i,0}$ hinzu und erhalten A_{τ_1}. Analog fügen wir für jedes $i \in \{0, \ldots, m-1\}$ die Kante $\bot_{3m+i+1} \xrightarrow{\oplus_{3m+i+1}} g_{i,0}$ hinzu, um A_{τ_2} zu erhalten.

Lemma 4.10. *Wenn A_τ die τ-EZTE hat, dann hat $(\mathfrak{U}, M)$ ein Eins-in-drei-Modell.*

Beweis. Weil A_τ die τ-EZTE hat, gibt es eine τ-Region, die α löst. Sei $R = (sup, sig)$ eine solche Region. Der Argumentationsverlauf ist wie folgt: (1) Wir zeigen, es gilt $sig(X_i) \neq \mathsf{swap}$ für alle $i \in \{0, \ldots, m-1\}$, falls $\tau \in \{\tau_0, \tau_1\}$, und $sig(x_i) \neq \mathsf{swap}$ für alle $i \in \{0, \ldots, m-1\}$, falls $\tau = \tau_2$. (2) Mittels dieser Eigenschaft zeigen wir dann: Wenn $sup(t_{i,j,1}) = 1$ und $sup(t_{i,j,4}) = 0$ für alle $i \in \{0, \ldots, m-1\}$ und alle $j \in \{0,1,2\}$ gilt oder $sup(t_{i,j,1}) = 0$ und $sup(t_{i,j,4}) = 1$ für alle $i \in \{0, \ldots, m-1\}$ und alle $j \in \{0,1,2\}$ gilt, dann definiert die Menge $S = \{X \in \mathfrak{U} \mid sig(X) \neq \mathsf{nop}\}$ ein Eins-in-drei-Modell von $(\mathfrak{U}, M)$. (3) Schließlich zeigen wir, dass R die Voraussetzung dieser Implikation erfüllt.

zu (1): Wenn $\tau = \tau_0$, dann gilt $\mathsf{swap} \notin \tau$ und daher haben wir $sig(X_i) \neq \mathsf{swap}$ für alle $i \in \{0, \ldots, m-1\}$. Wir argumentieren für $\tau \in \{\tau_1, \tau_2\}$: Weil R das EZTP α löst, gilt $sig(k) \in \mathfrak{inhibit}$. Folglich gilt $sup(z) = sup(q)$ und $sup(z') = sup(q')$ für je zwei k-beschriftete Kanten $z \xrightarrow{k} z'$ und $q \xrightarrow{k} q'$ von A_τ. Für alle $i \in \{0, \ldots, m-1\}$ sind aber $f_{i,0}, f_{i,1}$ beziehungsweise $g_{i,3}, g_{i,2}$ Ursprünge beziehungsweise Ziele von k. Daher erhalten wir mit den Kanten $f_{i,0} \xrightarrow{X_i} f_{i,1}$ von A_{τ_1} beziehungsweise $g_{i,2} \xrightarrow{x_i} g_{i,3}$ von A_{τ_2}, dass $sig(X_i) \neq \mathsf{swap}$, falls $\tau = \tau_1$, und $sig(x_i) \neq \mathsf{swap}$, falls $\tau = \tau_2$.

zu (2): Gelte zunächst $sup(t_{i,j,1}) = 0$ und $sup(t_{i,j,4}) = 1$ für alle $i \in \{0, \ldots, m-1\}$ und $j \in \{0,1,2\}$. Sei $i \in \{0, \ldots, m-1\}$ beliebig aber fest. Wegen $sup(t_{i,j,1}) = 0$ und $sup(t_{i,j,4}) = 1$ für alle $j \in \{0,1,2\}$, gibt es für jedes $X \in \{X_{i_0}, X_{i_1}, X_{i_2}\}$ ein Ziel z mit $sup(z) = 1$ und für jedes $x \in \{x_{i_0}, x_{i_1}, x_{i_2}\}$ ein Ziel mit $sup(z) = 0$. Daraus folgen $sig(X) \in \mathfrak{produce} \cup \{\mathsf{nop}\}$ für alle $X \in \{X_{i_0}, X_{i_1}, X_{i_2}\}$ beziehungsweise $sig(x) \in \mathfrak{consume} \cup \{\mathsf{nop}\}$ für alle $x \in \{x_{i_0}, x_{i_1}, x_{i_2}\}$. Darüber hinaus gilt wegen $sup(t_{i,0,1}) = 0$ und $sup(t_{i,0,4}) = 1$, dass das Bild P_i^R des Pfades $P_i = t_{i,0,1} \xrightarrow{X_{i_0}} t_{i,0,2} \xrightarrow{X_{i_1}} t_{i,0,3} \xrightarrow{X_{i_2}} t_{i,0,4}$ ein Pfad von 0 nach 1 in τ ist. Daher muss es ein $X \in \{X_{i_0}, X_{i_1}, X_{i_2}\}$ geben, sodass $sig(X) \in \mathfrak{produce}$. Wir argumentieren, dass X eindeutig ist.

Sei $X = X_{i_0}$, das heißt $sig(X_{i_0}) \in \mathfrak{produce}$. Wir zeigen, dass dies $sig(Y) = \mathsf{nop}$ für alle $Y \in \{X_{i_0}, X_{i_1}, X_{i_2}\} \setminus \{X\}$ impliziert. Es gilt $sup(t_{i,0,2}) = 1$, weil $sig(X_{i_0}) \in \mathfrak{produce}$. Dies impliziert $sig(x_{i_0}) \in \mathfrak{consume}$, weil $sup(t_{i,0,1}) = 0$. Damit folgt aus $sup(t_{i,1,4}) = 1$ und $t_{i,1,4} \xrightarrow{x_{i_0}} t_{i,1,3}$, dass $sup(t_{i,1,3}) = 0$. Mit $\xrightarrow{X_{i_2}} t_{i,1,3}$ und $sig(X_{i_2}) \in \mathfrak{produce} \cup \{\mathsf{nop}\}$

folgt des Weiteren $sig(X_{i_2}) \in \{\mathsf{swap}, \mathsf{nop}\}$. Für $\tau \in \{\tau_0, \tau_1\}$ erhalten wir wegen der eingangs diskutierten Eigenschaft sofort $sig(X_{i_2}) = \mathsf{nop}$. Nehmen wir für $\tau = \tau_2$ zunächst $sig(X_{i_2}) = \mathsf{swap}$ an. Mit $sup(t_{i,2,1}) = 0$ impliziert dies $sup(t_{i,2,2}) = 1$. Aus $\xrightarrow{x_{i_0}} t_{i,2,2}$ und $sup(t_{i,2,2}) = 1$ sowie $sig(x_{i_0}) \in \mathfrak{consume}$ folgt nun der Widerspruch $sig(x_{i_0}) = \mathsf{swap}$. Es war also die Annahme $sig(X_{i_2}) = \mathsf{swap}$ falsch und es gilt $sig(X_{i_2}) = \mathsf{nop}$ für alle $\tau \in \{\tau_0, \tau_1, \tau_2\}$. Mit $sup(t_{i,0,4}) = 1$ und $sup(t_{i,1,3}) = 0$ impliziert dies $sup(t_{i,0,3}) = 1$ und $sup(t_{i,1,2}) = 0$. Es folgt mit $sup(t_{i,0,2}) = 1$ und $sup(t_{i,1,1}) = 0$ also $1 \xrightarrow{sig(X_1)} 1$ und $0 \xrightarrow{sig(X_1)} 0$. Daraus schließen wir $sig(X_{i_1}) = \mathsf{nop}$.

Die Symmetrie der Gadgets $T_{i,0}, T_{i,1}, T_{i,2}$ lässt uns leicht erkennen, dass die Annahme $sig(X_{i_1}) \in \mathfrak{produce}$ beziehungsweise $sig(X_{i_2}) \in \mathfrak{produce}$ mit analogen Argumenten zu $sig(X_{i_0}) = sig(X_{i_2}) = \mathsf{nop}$ beziehungsweise zu $sig(X_{i_0}) = sig(X_{i_1}) = \mathsf{nop}$ führt. Weil i beliebig war, gibt es somit für jedes $i \in \{0, \ldots, m-1\}$ genau ein $X \in M_i$ mit $sig(X) \neq \mathsf{nop}$. Daher ist $S = \{X \in \mathfrak{U} \mid sig(X) \neq \mathsf{nop}\}$ ein gesuchtes Eins-in-drei-Modell. Des Weiteren führt die Annahme $sup(t_{i,j,1}) = 1$ und $sup(t_{i,j,4}) = 0$ für alle $i \in \{0, \ldots, m-1\}$ und $j \in \{0, 1, 2\}$ mit ähnlichen Argumenten zu diesem Resultat.

zu (3) Es bleibt zu zeigen, dass $sup(t_{i,j,1}) = 1$ und $sup(t_{i,j,4}) = 0$ für alle $i \in \{0, \ldots, m-1\}$ und $j \in \{0, 1, 2\}$ oder $sup(t_{i,j,1}) = 0$ und $sup(t_{i,j,4}) = 1$ für alle $i \in \{0, \ldots, m-1\}$ und $j \in \{0, 1, 2\}$ gilt.

Weil $sig(k)$ nicht an $sup(h_2)$ definiert ist, gilt entweder $sig(k) \in \{\mathsf{inp}, \mathsf{used}\}$ und $sup(h_2) = 0$ oder $sig(k) \in \{\mathsf{out}, \mathsf{free}\}$ und $sup(h_2) = 1$. Nachfolgend diskutieren wir die einzelnen Fälle.

Sei zunächst $sig(k) = \mathsf{inp}$ und $sup(h_2) = 0$. Wegen $sig(k) = \mathsf{inp}$ gilt $sup(h_1) = 0$ sowie $sup(t_{0,0,5}) = 1$. Aus $sup(h_1) = sup(h_2) = 0$ folgt $sig(w) \in \mathfrak{save}_0$ und mit $sup(t_{0,0,5}) = 1$ impliziert dies $sig(w) = \mathsf{nop}$. Mit $sig(k) = \mathsf{inp}$ und $sig(w) = \mathsf{nop}$ erhalten wir schließlich $sup(t_{i,j,1}) = 0$ und $sup(t_{i,j,4}) = 1$ für alle $i \in \{0, \ldots, m-1\}$ und $j \in \{0, 1, 2\}$. Analog führt die Annahme $sig(k) = \mathsf{out}$ und $sup(h_2) = 1$ zu $sup(t_{i,j,1}) = 1$ und $sup(t_{i,j,4}) = 0$ für alle $i \in \{0, \ldots, m-1\}$.

Gelte nun $sig(k) = \mathsf{used}$ und $sup(h_2) = 0$. Wegen $sig(k) = \mathsf{used}$ erhalten wir $sup(h_1) = sup(t_{i,0,5}) = 1$. Aus $sup(h_1) = 1$ und $sup(h_2) = 0$ folgt $sig(w) \in \mathfrak{consume}$. Mit $sup(t_{i,0,5}) = 1$ impliziert dies $sig(w) = \mathsf{swap}$. Aus $sig(k) = \mathsf{used}$ und $sig(w) = \mathsf{swap}$ erhalten wir endlich $sup(t_{i,j,1}) = 1$ und $sup(t_{i,j,4}) = 0$ für alle $i \in \{0, \ldots, m-1\}$ und $j \in \{0, 1, 2\}$. Ebenso führt die Annahme $sig(k) = \mathsf{free}$ und $sup(h_2) = 1$ mit analoger Argumentation zu $sup(t_{i,j,1}) = 0$ und $sup(t_{i,j,4}) = 1$ für alle $i \in \{0, \ldots, m-1\}$ und $j \in \{0, 1, 2\}$.

Insgesamt folgt für alle $\tau \in \{\tau_0, \tau_1, \tau_2\}$, dass eine τ-Region $R = (sup, sig)$, die α löst, mittels $S = \{X \in \mathfrak{U} \mid sig(X) \neq \mathsf{nop}\}$ ein Eins-in-drei-Modell für $(\mathfrak{U}, M)$ definiert. □

Im weiteren Verlauf des Beweises zeigen wir für alle $\tau \in \{\tau_0, \tau_1, \tau_2\}$, dass A_τ die τ-EZTE und die τ-ZTE hat. Sei $\underline{E} = E(A_\tau) \setminus (\ominus \cup \oplus)$ und $\underline{Z} = Z(A_\tau) \setminus \bot$ und $Z_T = \{t_{i,0,0}, t_{i,1,0}, t_{i,2,0} \mid i \in \{0, \ldots, m-1\}\}$ und $Z_F = \{f_{i,0} \mid i \in \{0, \ldots, m-1\}\}$ und $Z_G = \{g_{i,0} \mid i \in \{0, \ldots, m-1\}\}$ und Z_ι die Menge der Startzustände aller Gadgets, das heißt $Z_\iota = \{h_0\} \cup Z_T \cup Z_F \cup Z_G$. Argumentiert wird zunächst für die τ-EZTE. Um diesen technischen Teil des Beweises so kurz wie möglich zu halten, werden wir erneut einige

Vereinfachungen vereinbaren: Zunächst ist klar, dass die Ereignisse aus $\ominus \cup \oplus$ lösbar sind. Wir beschränken uns deshalb darauf, die Lösbarkeit der Ereignisse aus $\underline{E}$ explizit zu diskutieren. Insbesondere konzentrieren wir uns auf EZTP (a, q) mit $a \in \underline{E}$ und $q \in \underline{Z}$. Entsprechende lösende Regionen $R = (sup, sig)$ werden implizit mittels der Startzustände Z_ι der Gadgets und der Signatur der Ereignisse aus $\underline{E}$ definiert. Dabei wird immer $sig(a) = \mathsf{inp}$ und $sup(q) = 0$ gelten. Das Paar $R = (sup, sig)$ kann stets auf naheliegende Weise auf $\bot$ und $\ominus \cup \oplus$ und damit zu einer Region von A_τ fortgesetzt werden: Definiere $sup(z) = 0$ für alle $z \in \bot$ und $sig(e) = \mathsf{nop}$ für alle $e \in \ominus$; für alle $e \in \oplus$ und die eindeutige (Verbindungs-) Kante $z \xrightarrow{e} z'$ in A_τ definiere (passend) $sig(e) \in \mathfrak{produce}$, falls $sup(z') = 1$, sonst $sig(e) = \mathsf{nop}$. Man bemerke, dass diese Definition die Lösbarkeit von (a, z) für alle $e \in \underline{E}$ und $z \in \bot$ impliziert, denn für jedes Ereignis $a \in \underline{E}$ ist ein EZTP (a, q) mit $q \in \underline{Z}$ zu lösen. Um ermüdende Fallunterscheidungen zu vermeiden, definieren wir die Supporte und Signaturen stets für die Startzustände beziehungsweise Ereignisse aller Gadgets. Für ein konkretes $A \in \{A_{\tau_0}, A_{\tau_1}, A_{\tau_2}\}$ sind dann natürlich nur die Werte der Zustände interessant, die in A_τ tatsächlich vorkommen. Wir erinnern daran, dass $\tau \in \{\tau_0, \tau_1, \tau_2\}$ beliebig aber fest gewählt ist.

Fakt 4.3. *Wenn es ein Eins-in-drei-Modell für $(\mathfrak{U}, M)$ gibt, dann ist das Ereignis k τ-lösbar.*

Beweis. Sei S ein Eins-in-drei-Modell von $(\mathfrak{U}, M)$. Die folgende Region $R_1 = (sup_1, sig_1)$ löst α: Für alle $z \in Z_\iota$ sei $sup_1(z) = 1$; für alle $e \in \underline{E}$, wenn $e = k$, dann $sig_1(e) = \mathsf{inp}$; wenn $e \in \{y\} \cup S$, dann $sig_1(e) = \mathsf{set}$, falls $\mathsf{set} \in \tau$, sonst $sig_1(e) = \mathsf{swap}$; wenn $e \in \{x_i \mid i \in \{0, \ldots, m-1\}$ und $X_i \in S\}$, dann $sig_1(e) = \mathsf{inp}$, falls $\tau \neq \tau_2$, sonst $sig_1(e) = \mathsf{res}$; andernfalls $sig_1(e) = \mathsf{nop}$.

Die folgende Region $R_2 = (sup_2, sig_2)$ löst (k, z) für alle verbleibenden $z \in \underline{Z}$ an denen k nicht stattfindet: Für alle $z \in Z_\iota$ sei $sup_2(z) = 1$; für alle $e \in \underline{E}$, wenn $e = k$, dann $sig_2(e) = \mathsf{inp}$; wenn $e = w$, dann $sig_2(e) = \mathsf{set}$, falls $\mathsf{set} \in \tau$, sonst $sig_2(e) = \mathsf{swap}$; andernfalls $sig_2(e) = \mathsf{nop}$. □

Fakt 4.4. *Wenn $e \in \mathfrak{U} \cup \mathfrak{u}$, dann ist e τ-lösbar.*

Beweis. Nehmen wir zunächst an, dass $e \in \mathfrak{U}$. Es ist leicht zu sehen, dass (e, z) für alle Zustände $z \in \underline{Z}$ lösbar ist, die aus Gadgets herrühren, in denen e nicht stattfindet. Wir konzentrieren uns deshalb nur auf den Fall, dass e und z im gleichen Gadget stattfinden. Sei $i \in \{0, \ldots, m-1\}$ beliebig aber fest. Wir zeigen, dass (X_{i_0}, z) für alle relevanten $z \in Z(T_{i,0}) \cup Z(T_{i,1}) \cup Z(T_{i,2}) \cup Z(F_{i_0}) \cup Z(G_{i_0})$ lösbar ist. Wegen der Symmetrie der Gadgets erkennt man leicht, dass die entsprechende Argumentation für (X_{i_1}, z) beziehungsweise (X_{i_2}, z) und alle relevanten Zustände z aus $T_{i,0}, T_{i,1}, T_{i,2}, F_j, G_j$ mit $j = i_1$ beziehungsweise $j = i_2$ analog zu führen ist. Aus der Beliebigkeit von i folgt damit die Lösbarkeit aller Ereignisse $e \in \mathfrak{U}$.

Die folgende Region $R_1 = (sup_1, sig_1)$ löst (X_{i_0}, z) für alle $z \in \{t_{i,0,0}, t_{i,1,0}, t_{i,2,0}, f_{i_0,3}\}$: Für alle $z \in \{h_0\} \cup Z_T$ sie $sup_1(z) = 0$ und für alle $z \in Z_F \cup Z_G$ sei $sup_1(z) = 1$; für

alle $e \in \underline{E}$, wenn $e = X_{i_0}$, dann $sig_1(e) = \mathsf{inp}$; wenn $e \in \{k, x_{i_0}\}$, dann $sig_1(e) = \mathsf{set}$, falls $\tau \notin \{\tau_1, \tau_2\}$, sonst $sig_1(e) = \mathsf{swap}$; wenn $e = u_i$ (das heißt, $\tau = \tau_2$), dann $sig_1(e) = \mathsf{swap}$; andernfalls $sig_1(e) = \mathsf{nop}$.

Seien $j, \ell \in \{0, \ldots, m-1\} \setminus \{i\}$ die voneinander verschiedenen Indizes, sodass $X_{i_0} \in M_i \cap M_j \cap M_\ell$. Die folgende Region $R_2 = (sup_2, sig_2)$ löst (X_{i_0}, z) für alle $z \in \{t_{i,0,2}, \ldots, t_{i,0,6}\} \cup \{t_{i,1,4}, t_{i,1,5}, t_{i,1,6}\} \cup \{t_{i,2,3}, \ldots, t_{i,2,6}\} \cup \{f_{i_0,1}, f_{i_0,2}\}$: Es sei $sup_2(z) = 1$ für alle $z \in Z_\iota$ und für alle $e \in \underline{E}$, wenn $e = X_{i_0}$, dann $sig_2(e) = \mathsf{inp}$, sonst $sig_2(e) = \mathsf{nop}$.

Es bleibt noch, die Lösbarkeit von (X_{i_0}, z) für alle $z \in \{t_{i,1,0}, t_{i,1,1}, t_{i,1,2}, t_{i,2,0}, t_{i,2,1}\}$ zu zeigen. Wir nutzen eine Region $R_3 = (sup_3, sig_3)$ deren Signatur wie folgt definiert ist: Für alle $e \in \underline{E}$, wenn $e \in \{X_{i_0}, x_{i_2}\}$, dann $sig_3(e) = \mathsf{inp}$; wenn $e \in \{X_{i_2}, x_{i_0}\}$, dann $sig_3(e) = \mathsf{set}$, falls $\mathsf{set} \in \tau$, sonst $sig_3(e) = \mathsf{swap}$; falls $e \in \{u_{i_0}, u_{i_2}\}$ und $\tau = \tau_2$, dann $sig_3(e)$=swap; andernfalls $sig_3(e) = \mathsf{nop}$. Mit Blick auf $T_{i,0}, T_{i,1}, T_{i,2}$ und F_i sowie G_i lässt sich (die angestrebte Region) R_3 wie in Abbildung 4.4 skizzieren.

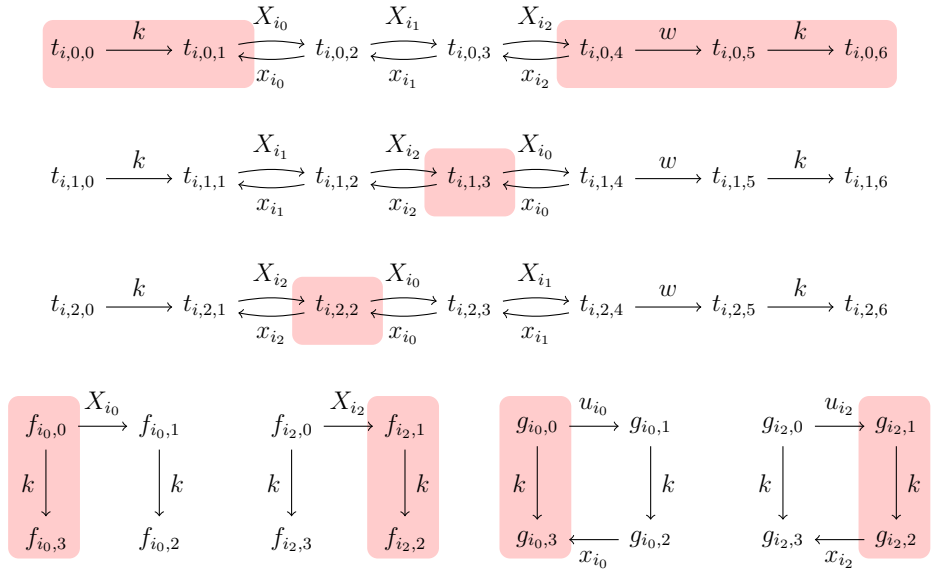


Abbildung 4.4: Skizze der Region R_3 aus Fakt 4.4. Der rote Bereich markiert die Zustände mit positivem Support.

Es bleibt noch den Support passend zu definieren. Für alle Gadgets von A_τ, die für alle $e \in \{X_{i_0}, x_{i_0}, X_{i_2}, x_{i_2}\}$ das Ereignis e nicht enthalten bildet sup_3 die entsprechenden Startzustände der Gadgets auf 0 ab. Außerdem gilt $sup_3(f_{i_0}, 0) = sup_3(g_{i_0,0}) = 1$ und $sup_3(f_{i_2}, 0) = sup_3(g_{i_2,0}) = 0$. Es verbleibt die „T-Gadgets“ zu betrachten, in denen X_{i_0}, x_{i_0} oder X_{i_2}, x_{i_2} vorkommen. Genauer gesagt definieren wir für alle $j \in \{0, \ldots, m-1\}$ und alle $\ell \in \{0, 1, 2\}$ mit $X_{i_0} \in M_j$ oder $X_{i_2} \in M_j$ den Support der Initialzustände wie folgt: Falls $X_{i_0} \in M_j$ und $X_{i_2} \notin M_j$, dann $sup_3(t_{j,\ell,0}) = 1$ und falls $X_{i_2} \in M_j$ und $X_{i_0} \notin M_j$, dann $sup_3(t_{j,\ell,0}) = 0$. Es bleibt der Fall, dass $X_{i_0} \in M_j$ und $X_{i_2} \in M_j$: Es hat $T_{j,\ell}$ den

eindeutigen Pfad $P_{j,\ell} = t_{j,\ell,1} \xrightarrow{e_0} t_{j,\ell,2} \xrightarrow{e_1} t_{j,\ell,3} \xrightarrow{e_2} t_{j,\ell,4}$, mit $X_{i_0}, X_{i_2} \in \{e_0, e_1, e_2\}$. Wenn $e_0 = X_{i_0}$ oder wenn $e_1 = X_{i_0}$ und $e_2 = X_{i_2}$ (siehe zum Beispiel $T_{i,0}$ für $(j, \ell) = (i, 0)$), dann definieren wir $sup_3(t_{j,\ell,0}) = 1$, sonst $sup_3(t_{j,\ell,0}) = 0$ (siehe zum Beispiel $T_{i,1}$ für $(j, \ell) = (i, 1)$). Es resultiert R_3.

Ähnlich zur Lösbarkeit der Ereignisse aus $\mathfrak{U}$ erhält man –wiederum mittels Symmetrie– die Lösbarkeit der Ereignisse aus $\mathfrak{u}$. Darüber hinaus ist es leicht zu sehen, dass die Ereignisse $u_0, \ldots, u_{m-1}$ in A_{τ_2} lösbar sind. Auf eine explizite Diskussion dieser Ereignisse wird deshalb verzichtet. □

Fakt 4.5. *Wenn $e \in \{w, y\}$, dann ist e τ-lösbar.*

Beweis. Beginnen wir mit dem Ereignis w. Wie man leicht sieht, ist (w, z) für alle Zustände $z \in Z(H)$ τ-lösbar, an denen w nicht definiert ist. Seien $i \in \{0, \ldots, m-1\}$ und $j \in \{0, 1, 2\}$ jeweils beliebig aber fest. Die folgende Region $R = (sup, sig)$ löst (w, z) für alle $z \in Z(T_{i,j}) \setminus \{t_{i,j,4}\}$: für alle $z \in Z_\iota$, wenn $z = t_{i,0,0}$, dann $sup(z) = 0$, sonst $sup(z) = 1$; für alle $e \in \underline{E}$, wenn $e = w$, dann $sig(e) = \mathsf{inp}$; wenn $\xrightarrow{e} t_{i,j,4}$, das heißt, e ist das Variablen-Ereignis, welches der direkte Vorgänger von w in $T_{i,j}$ ist, dann $sig(e) = \mathsf{set}$, sonst $sig(e) = \mathsf{nop}$. Aus der Beliebigkeit von i und j, sowie der Beobachtung, dass w von den Zuständen der Gadgets trennbar ist, in denen es nicht stattfindet, folgt die τ-Lösbarkeit von w.

Wie man leicht sieht, ist das Ereignis y, welches genau einmal in A_τ stattfindet, ebenso τ-lösbar. □

Schließlich sieht man leicht, dass die Regionen, welche die τ-EZTE bezeugen, ebenso die τ-ZTE implizieren. Dies beweist das folgende Lemma und komplettiert den Beweis von damit Satz 4.7:

Lemma 4.11. *Wenn $(\mathfrak{U}, M)$ ein Eins-in-drei-Modell hat, dann hat A_τ die τ-EZTE und die τ-ZTE.*

4.1.5 Die Netztypen A9

Satz 4.8. *Es seien $\tau_0 = \{\mathsf{nop}, \mathsf{set}, \mathsf{swap}\} \cup \omega$ und $\tau_1 = \{\mathsf{nop}, \mathsf{res}, \mathsf{swap}\} \cup \omega$, sodass $\emptyset \neq \omega \subseteq \{\mathsf{used}, \mathsf{free}\}$. Wenn $\tau \in \{\tau_0, \tau_1\}$, dann sind τ-*Sprachsimulation *und τ-*Realisierung *NP-vollständig.*

Im weiteren Verlauf dieses Abschnitts beweisen wir Satz 4.8 gemäß der in Kapitel 3 vorgestellten Reduktionsstrategie. Folgende Kongruenzen zwischen Netztypen werden den Argumentationsverlauf vereinfachen: Für alle $\tau' = \{\mathsf{nop}, \mathsf{res}, \mathsf{swap}\} \cup \omega'$ mit $\emptyset \neq \omega' \subseteq \{\mathsf{used}, \mathsf{free}\}$ gibt es $\emptyset \neq \omega \subseteq \{\mathsf{used}, \mathsf{free}\}$, sodass $\{\mathsf{nop}, \mathsf{set}, \mathsf{swap}\} \cup \omega \equiv \tau'$. Wir werden deshalb im Folgenden $\tau = \{\mathsf{nop}, \mathsf{set}, \mathsf{swap}\} \cup \omega$ mit $\emptyset \neq \omega \subseteq \{\mathsf{used}, \mathsf{free}\}$ annehmen und die NP-Schwere der zugehörigen Entscheidungsprobleme beweisen. Die Aussage des Satzes folgt dann mit Satz 2.2 und Lemma 4.1.

In den nachfolgenden Ausführungen definieren wir zunächst die einzelnen Komponenten von A_τ mit dem angekündigten EZTP α. Danach führen wir zusätzliche Kanten ein, welche die Gadgets von A_τ miteinander verbinden. Dann beweisen wir, dass die τ-Lösbarkeit von α ein Eins-in-drei-Modell impliziert. Schließlich argumentieren wir, dass ein Eins-in-drei-Modell von $(\mathfrak{U}, M)$ die τ-EZTE und die τ-ZTE von A_τ impliziert.

Fangen wir mit der Definition der Komponenten an: Das Transitionssystem A_τ hat zunächst die folgenden beiden Gadgets H_0 und H_1, welche das EZTP $\alpha = (k, h_{0,2})$ zur Verfügung stellen:

$$H_0 = \; h_{0,0} \xleftrightarrow{k} h_{0,1} \xleftrightarrow{o_0} h_{0,2} \xleftrightarrow{o_1} h_{0,3} \xleftrightarrow{k} h_{0,4}$$

$$H_1 = \; h_{1,0} \xleftrightarrow{k} h_{1,1} \xleftrightarrow{w_0} h_{1,2} \xleftrightarrow{o_0} h_{1,3} \xleftrightarrow{w_1} h_{1,4} \xleftrightarrow{k} h_{1,5}$$

Darüber hinaus hat A_τ für jedes $i \in \{0, \ldots, m-1\}$ das folgende Gadget $T_{i,0}$, das die Variablen der Klausel $M_i = \{X_{i_0}, X_{i_1}, X_{i_2}\}$ als Ereignisse nutzt:

$$T_{i,0} \;=\; t_{i,0,0} \xleftrightarrow{k} t_{i,0,1} \xleftrightarrow{w_0} t_{i,0,2} \xleftrightarrow{a_{3i}} t_{i,0,3} \xrightarrow{X_{i_0}} t_{i,0,4} \xleftrightarrow{X_{i_0}} t_{i,0,5} \xleftrightarrow{a_{3i}} t_{i,0,6} \xleftrightarrow{a_{3i+1}} t_{i,0,7} \downarrow X_{i_1}$$

$$t_{i,0,16} \xleftrightarrow{k} t_{i,0,15} \xleftrightarrow{w_1} t_{i,0,14} \xleftrightarrow{a_{3i+2}} t_{i,0,13} \xleftrightarrow{X_{i_2}} t_{i,0,12} \xleftarrow{X_{i_2}} t_{i,0,11} \xleftrightarrow{a_{3i+2}} t_{i,0,10} \xleftrightarrow{a_{3i+1}} t_{i,0,9} \xleftrightarrow{X_{i_1}} t_{i,0,8}$$

Ebenso hat A_τ für jedes $i \in \{0, \ldots, m-1\}$ die folgenden drei Gadgets $T_{i,1}$ und $T_{i,2}$ sowie $T_{i,3}$ auf denen die Elemente von M_i als Ereignisse stattfinden:

$$T_{i,1} = \; t_{i,1,0} \xleftrightarrow{X_{i_0}} t_{i,1,1} \xleftrightarrow{o_1} t_{i,1,2} \xleftrightarrow{X_{i_1}} t_{i,1,3}$$

$$T_{i,2} = \; t_{i,2,0} \xleftrightarrow{X_{i_0}} t_{i,2,1} \xleftrightarrow{o_1} t_{i,2,2} \xleftrightarrow{X_{i_2}} t_{i,2,3}$$

$$T_{i,3} = \; t_{i,3,0} \xleftrightarrow{X_{i_1}} t_{i,3,1} \xleftrightarrow{o_1} t_{i,3,2} \xleftrightarrow{X_{i_2}} t_{i,3,3}$$

Die Zustände aus $\perp = \{\perp_0, \ldots, \perp_{4m+1}\}$ und Ereignisse aus $\ominus = \{\ominus_1, \ldots, \ominus_{4m+1}\}$ sowie $\oplus = \{\oplus_0, \ldots, \oplus_{4m+1}\}$ nutzen wir, um die eingeführten Gadgets zu verbinden: Für jedes $i \in \{0, \ldots, 4m\}$ fügen wir die bidirektionale Kante $\perp_i \xleftrightarrow{\ominus_{i+1}} \perp_{i+1}$ hinzu; für jedes $i \in \{0, \ldots, m-1\}$ und jedes $j \in \{0, 1, 2, 3\}$ fügen wir die bidirektionale Kante $\perp_{4i+j} \xleftrightarrow{\oplus_{4i+j}} t_{i,j,0}$ hinzu. Abschließend fügen wir die bidirektionalen Kanten $\perp_{4m} \xleftrightarrow{\oplus_{4m}} h_{0,0}$ und $\perp_{4m+1} \xleftrightarrow{\oplus_{4m+1}} h_{1,0}$ hinzu. Das resultierende Transitionssystem mit Initialzustand $\perp_0$ ist A_τ.

Lemma 4.12. *Wenn A_τ die τ-EZTE hat, dann hat $(\mathfrak{U}, M)$ ein Eins-in-drei-Modell.*

Beweis. Sei $R = (sup, sig)$ eine τ-Region, die α löst. Weil $sig(k)$ nicht an $sup(h_{0,2})$ definiert ist, gilt entweder $sig(k) = \mathsf{used}$ und $sup(h_{2,0}) = 0$ oder $sig(k) = \mathsf{free}$ und $sup(h_{2,0}) = 1$. Wir werden nur den ersten Fall explizit diskutieren, die Argumentation für den zweiten ist vollkommen analog. Sei also $sig(k) = \mathsf{used}$ und $sup(h_{2,0}) = 0$. Wegen $sig(k) = \mathsf{used}$ erhalten wir $sup(z) = sup(z') = 1$ für alle Kanten $z \xrightarrow{k} z'$ in A_τ. Außerdem impliziert jede bidirektionale Kante $z \xleftrightarrow{e} z'$ in A_τ stets $sig(e) = \mathsf{swap}$, wenn $sup(z) \neq sup(z')$. Aus $sup(h_{0,1}) = sup(h_{0,3}) = 1$ und $sup(h_{0,2}) = 0$ erhalten wir damit $sig(o_0) = sig(o_1) = \mathsf{swap}$. Weiterhin erhalten wir mit $sup(h_{1,1}) = sup(h_{1,4}) = 1$, dass das Bild des Pfades $h_{1,1} \xrightarrow{w_0} \dots \xrightarrow{w_1} h_{1,4}$ ein Pfad von 1 nach 1 in τ ist. Deshalb muss die Anzahl der Zustandswechsel zwischen 1 und 0 entlang diese Pfades gerade sein. Weil $sig(o_0) = \mathsf{swap}$ gilt, gibt es also genau ein Ereignis $e \in \{w_0, w_1\}$ mit $sig(e) = \mathsf{swap}$. Wir nehmen im Folgenden $e = w_0$ an. Die Argumentation für den Fall $e = w_1$ verläuft analog. Insbesondere gilt mit $sup(h_{1,1}) = 1$ und $sig(w_0) = sig(o_0) = \mathsf{swap}$, dass $sup(h_{1,3}) = 1$. Mit $sup(h_{1,4}) = 1$ impliziert dies $sig(w_1) \in \mathfrak{save}_1$.

Sei $i \in \{0, \dots, m-1\}$ beliebig aber fest. Mit $sup(t_{i,0,1}) = sup(t_{i,0,15}) = 1$ und $sig(w_0) = \mathsf{swap}$ und $sig(w_1) \in \mathfrak{save}_1$ erhalten wir, dass das Bild P_i^R des Pfades $P_i = t_{i,0,2} \xrightarrow{a_{3_i}} \dots \xrightarrow{a_{3i+2}} t_{i,0,14}$ ein Pfad von 0 nach 1 in τ ist. Wir werden zeigen, dass es deshalb ein eindeutiges $X \in \{X_{i_0}, X_{i_1}, X_{i_2}\}$ mit $sig(X) = \mathsf{set}$ gibt. Nehmen wir zunächst $sig(X) \neq \mathsf{set}$ für alle $X \in \{X_{i_0}, X_{i_1}, X_{i_2}\}$, dass heißt $sig(X) \in \{\mathsf{nop}, \mathsf{swap}, \mathsf{used}, \mathsf{free}\}$. Man überlegt sich, dass dies $sup(z) = sup(z'')$ impliziert, falls $z \xrightarrow{X} z'$ und $z' \xrightarrow{X} z''$ Kanten in A_τ sind: Für $sig(X) \in \{\mathsf{nop}, \mathsf{used}, \mathsf{free}\}$ ist das sofort klar. Falls $sig(X) = \mathsf{swap}$, dann $sup(z) \neq sup(z') \neq sup(z'')$, was offenbar ebenso $sup(z) = sup(z'')$ impliziert. Daraus folgt sofort $sup(t_{i,0,3}) = sup(t_{i,0,5})$ und $sup(t_{i,0,7}) = sup(t_{i,0,9})$ sowie $sup(t_{i,0,11}) = sup(t_{i,0,13})$. Wir zeigen nun, dass dies der Voraussetzung $sup(t_{i,0,2}) = 1$ und $sup(t_{i,0,14}) = 0$ widerspricht: Wenn $sig(a_{3_i}) \in \{\mathsf{nop}, \mathsf{set}, \mathsf{used}, \mathsf{free}\}$, dann gilt $sup(t_{i,0,2}) = sup(t_{i,0,3}) = sup(t_{i,0,5}) = sup(t_{i,0,6})$ und wenn $sig(a_{3i}) = \mathsf{swap}$, dann gilt $sup(t_{i,0,2}) \neq sup(t_{i,0,3}) = sup(t_{i,0,5}) \neq sup(t_{i,0,6})$. Insbesondere gilt also stets $sup(t_{i,0,2}) = sup(t_{i,0,6})$. Analog zeigt man $sup(t_{i,0,6}) = sup(t_{i,0,10})$ sowie $sup(t_{i,0,10}) = sup(t_{i,0,14})$. Insgesamt erhalten wir den Widerspruch $sup(t_{i,0,2}) = sup(t_{i,0,14})$. Daher war unsere Annahme falsch und es gibt ein $X \in \{X_{i_0}, X_{i_1}, X_{i_2}\}$ mit $sig(X) = \mathsf{set}$. Wir zeigen, dass X eindeutig ist.

Wir nehmen zunächst $sig(X_{i_0}) = \mathsf{set}$ an und zeigen $sig(Y) \neq \mathsf{set}$ für alle $Y \in \{X_{i_1}, X_{i_2}\}$. Aus $sig(X_{i_0}) = \mathsf{set}$ folgt $sup(t_{i,1,1}) = sup(t_{i,2,1}) = 1$. Wegen $sig(o_1) = \mathsf{swap}$ folgt damit $sup(t_{i,1,2}) = sup(t_{i,2,2}) = 0$ und ebenso $sig(Y) \neq \mathsf{set}$ für alle $Y \in \{X_{i_1}, X_{i_2}\}$. Die Argumente für die Fall $sig(X_{i_1}) = \mathsf{set}$ beziehungsweise $sig(X_{i_2}) = \mathsf{set}$ verlaufen wegen der Symmetrie der Gadgets $T_{i,1}, T_{i,2}, T_{i,3}$ analog.

Aus der Beliebigkeit von i folgt, dass die Menge $S = \{X \in \mathfrak{U} \mid sig(X) = \mathsf{set}\}$ ein Eins-in-drei-Modell von $(\mathfrak{U}, M)$ definiert. Die τ-EZTE von A_τ hat die Lösbarkeit von α zu Folge. Damit haben wir bewiesen, dass die τ-EZTE von A_τ ein Eins-in-drei-Modell von $(\mathfrak{U}, M)$ impliziert. □

Für die andere Richtung gehen wir wieder von der Existenz eines Eins-in-drei Modells

S aus und zeigen, dass dies die τ-EZTE und die τ-ZTE von A_τ impliziert. Sei S ein Eins-in-drei-Modell von $(\mathfrak{U}, M)$ und $\underline{E} = E(A_\tau) \setminus (\ominus \cup \oplus)$ und $\underline{Z} = Z(A_\tau) \setminus \bot$ und Z_ι die Menge der Startzustände der Gadgets, dass heißt $Z_\iota = \{h_{0,0}, h_{1,0}\} \cup \{t_{i,0,0}, t_{i,1,0}, t_{i,2,0}, t_{i,3,0} \mid i \in \{0, \ldots, m-1\}\}$. Wir nehmen den Fall $\mathsf{used} \in \tau$ an und zeigen zunächst, dass A_τ die τ-EZTE hat. Der Fall $\mathsf{used} \notin \tau$, welcher $\mathsf{free} \in \tau$ impliziert, ist ähnlich zu behandeln. Wieder werden wir uns einiger Erleichterungen bedienen, um diesen technischen und mitunter kraftraubenden Teil des Beweises klein zu halten: Erneut ist klar, dass die Ereignisse aus $\ominus \cup \oplus$ lösbar sind, sodass wir diese nicht explizit diskutieren. Darüber hinaus ist ein EZTP (a, q) mit $a \in \underline{E}$ und $q \in \underline{Z}$ stets lösbar, wenn a und q nicht im gleichen Gadget stattfinden. Genauer gesagt erhalten wir dann wie folgt eine τ-Region $R = (sup, sig)$, die (a, q) löst: sup bildet alle Zustände der Gadgets, die e enthalten, auf 1 ab und alle anderen Zustände auf 0; sig bildet e auf used ab; wenn $z \in Z_\iota$ mit $sup(z) = 1$, dann wird das eindeutige Ereignis $e \in \oplus$ mit $\xrightarrow{e} z$ auf swap abgebildet; alle anderen Ereignisse erhalten den Wert nop. Die resultierende Region $R = (sup, sig)$ tut es. Deshalb werden wir im weiteren Verlauf explizit nur solche EZTP (e, z) diskutieren, für die e und z im gleichen Gadget vorkommen. Wie gehabt, werden lösende Regionen $R = (sup, sig)$ implizit mittels des Supports der Zustände aus Z_ι und der Signatur der Ereignisse aus $\underline{E}$ definiert. Dabei gilt stets $sig(e) = \mathsf{used}$ und $sup(z) = 0$. Auf natürliche Weise kann R dann auf $\bot$ und $\ominus \cup \oplus$ fortgesetzt werden: für alle $z \in \bot$ definiere $sup(\bot) = 0$; für alle $e \in \ominus$ definiere $sig(e) = \mathsf{nop}$; für alle $e \in \oplus$ und die eindeutige Kante $z \xrightarrow{e} z'$ definiere $sig(e) = \mathsf{swap}$, falls $sup(z) \neq sup(z')$ und andernfalls $sig(e) = \mathsf{nop}$. Man bemerke, dass diese Definition von R die Lösbarkeit von (e, q) für alle $q \in \bot$ zur Folge hat. Für eine kompaktere und übersichtlichere Darstellung fassen wir Ereignisse gleicher Signatur häufig zusammen: $\mathcal{E}^R_{e_\tau} = \{e \in \underline{E} \mid sig(e) = e_\tau\}$.

Fakt 4.6. *Das Ereignis k ist τ-lösbar.*

Beweis. Die folgende Region $R_1 = (sup_1, sig_1)$ löst α: Für alle $z \in Z_\iota$ definiere $sup_1(z)$ wie folgt: falls $z \in \{h_{0,0}, h_{1,0}, t_{i,0,0} \mid i \in \{0, \ldots, m-1\}\}$, dann $sup_1(z) = 1$; für alle $i \in \{0, \ldots, m-1\}$, falls $z \in \{t_{i,1,0}, t_{i,2,0} \mid i \in \{0, \ldots, m-1\}\}$ und $X_{i_0} \in S$, dann $sup_1(z) = 1$; falls $z \in \{t_{i,3,0} \mid i \in \{0, \ldots, m-1\}\}$ und $X_{i_1} \in S$, dann $sup_1(z) = 1$, sonst $sup_1(z) = 0$; außerdem $\mathcal{E}^{R_1}_{\mathsf{used}} = \{k\}$ und $\mathcal{E}^{R_1}_{\mathsf{swap}} = \{o_0, o_1, w_0\}$ und $\mathcal{E}^{R_1}_{\mathsf{set}} = S$ und $\mathcal{E}^{R_1}_{\mathsf{nop}} = \underline{E} \setminus (\mathcal{E}^{R_1}_{\mathsf{used}} \cup \mathcal{E}^{R_1}_{\mathsf{swap}} \cup \mathcal{E}^{R_1}_{\mathsf{set}})$.

Die folgende Region $R_2 = (sup_2, sig_2)$ löst (k, z) für alle verbleibenden relevanten Zustände: für alle $z \in Z_\iota$, wenn $z \in \{h_{0,0}, h_{1,0}, t_{i,0,0} \mid i \in \{0, \ldots, m-1\}\}$, dann $sup_2(z) = 1$, sonst $sup_2(z) = 0$; darüber hinaus $\mathcal{E}^{R_2}_{\mathsf{used}} = \{k\}$ und $\mathcal{E}^{R_2}_{\mathsf{swap}} = \{w_0, w_1\}$ und $\mathcal{E}^{R_2}_{\mathsf{nop}} = \underline{E} \setminus (\mathcal{E}^{R_2}_{\mathsf{used}} \cup \mathcal{E}^{R_2}_{\mathsf{swap}})$. □

Fakt 4.7. *Wenn $e \in \{o_0, o_1\}$, dann ist e τ-lösbar.*

Beweis. Die folgende Region $R_1 = (sup_1, sig_1)$ löst (o_0, z) für alle $z \in \{h_{0,0}, h_{0,4}, h_{1,0}, h_{1,5}\}$: $sup_1(z) = 0$ für alle $z \in Z_\iota$; außerdem $\mathcal{E}^{R_2}_{\mathsf{used}} = \{o_0\}$ und $\mathcal{E}^{R_1}_{\mathsf{swap}} = \{k\}$ und $\mathcal{E}^{R_1}_{\mathsf{nop}} = \underline{E} \setminus \{o_0, k\}$.

Die folgende Region $R_2 = (sup_2, sig_2)$ löst (o_0, z) für alle $z \in \{h_{0,3}, h_{1,1}, h_{1,4}\}$: für alle $z \in Z_\iota$, falls $z = h_{0,0}$, dann $sup_2(z) = 1$, sonst $sup_2(z) = 0$; darüber hinaus $\mathcal{E}^{R_2}_{\mathsf{used}} = \{o_0\}$ und $\mathcal{E}^{R_2}_{\mathsf{swap}} = \{o_1, w_0, w_1\}$ und $\mathcal{E}^{R_2}_{\mathsf{nop}} = \underline{E} \setminus (\mathcal{E}^{R_2}_{\mathsf{used}} \cup \mathcal{E}^{R_2}_{\mathsf{swap}})$.

Dies zeigt die Lösbarkeit von o_0 und die Lösbarkeit von (o_1, z) kann für alle relevanten $z \in Z(H_0) \cup Z(H_1)$ analog bewiesen werden, wobei die Zustände der „T-Gadgets“ die o_1 enthalten auf 1 abgebildet werden.

Die folgende Region $R_3 = (sup_3, sig_3)$ löst (o_1, z) für alle $z \in \{t_{i,\ell,0}, t_{i,\ell,3} \mid i \in \{0, \ldots, m-1\}, \ell \in \{1,2,3\}\}$: für alle $z \in Z_\iota$, falls $z \in \{h_{0,0}, h_{1,0}\}$, dann $sup_3(z) = 1$, sonst $sup_3(z) = 0$; außerdem $\mathcal{E}^{R_3}_{\mathsf{used}} = \{o_0\}$ und $\mathcal{E}^{R_3}_{\mathsf{swap}} = \mathfrak{U}$ und $\mathcal{E}^{R_3}_{\mathsf{nop}} = \underline{E} \setminus (\{o_0\} \cup \mathfrak{U})$. □

Fakt 4.8. *Wenn $e \in \{w_0, w_1\}$, dann ist e τ-lösbar.*

Beweis. Wir argumentieren für w_0, die Argumente für w_1 sind analog. Sei $i \in \{0, \ldots, m-1\}$ beliebig aber fest.

Die nächste Region $R_1 = (sup_1, sig_1)$ löst (w_0, z) für alle $z \in \{h_{1,0}, h_{1,5}, t_{i,0,0}\}$: für alle $z \in Z_\iota$, $sup_1(z) = 0$ und $\mathcal{E}^{R_1}_{\mathsf{used}} = \{w_0\}$ und $\mathcal{E}^{R_1}_{\mathsf{swap}} = \{k\}$ und $\mathcal{E}^{R_1}_{\mathsf{nop}} = \underline{E} \setminus \{w_0, k\}$.

Die folgende Region $R_2 = (sup_2, sig_2)$ löst (w_0, z) für alle $z \in \{h_{1,3}, h_{1,4}\}$ und $z \in \{t_{i,0,3}, t_{i,0,4}, t_{i,0,5}\}$: für alle $z \in Z_\iota$, $sup(z) = 1$ und $\mathcal{E}^{R_2}_{\mathsf{used}} = \{w_0\}$ und $\mathcal{E}^{R_2}_{\mathsf{swap}} = \{o_0, a_{3i}\}$ und $\mathcal{E}^{R_2}_{\mathsf{nop}} = \underline{E} \setminus (\mathcal{E}^{R_2}_{\mathsf{used}} \cup \mathcal{E}^{R_2}_{\mathsf{swap}})$.

Die folgende Region $R_3 = (sup_3, sig_3)$ löst (w_0, z) für alle $z \in \{t_{i,0,6}, \ldots, t_{i,0,16}\}$: für alle $z \in Z_\iota$, $sup(z) = 1$ und $\mathcal{E}^{R_3}_{\mathsf{used}} = \{w_0\}$ und $\mathcal{E}^{R_3}_{\mathsf{swap}} = \{a_{3i}\}$ und $\mathcal{E}^{R_3}_{\mathsf{set}} = \{X_{i_0}\}$ und $\mathcal{E}^{R_3}_{\mathsf{nop}} = \underline{E} \setminus \{w_0, X_{i_0}, a_{3i}\}$.

Aus der Beliebigkeit von i folgt die Lösbarkeit von w_0. □

Fakt 4.9. *Wenn $e \in \{a_0, \ldots, a_{3m-1}\}$, dann ist e τ-lösbar.*

Beweis. Sei $i \in \{0, \ldots, m-1\}$ beliebig aber fest. Wir argumentieren für a_{3i}. Die Argumente für a_{3i+1} beziehungsweise a_{3i+2} sind analog zu führen. Aus der Beliebigkeit von i folgt dann die τ-Lösbarkeit aller $a_0, \ldots, a_{3m-1}$.

Die folgende Region $R_1 = (sup_1, sig_1)$ löst (a_{3i}, z) für alle $z \in \{t_{i,0,0}, t_{i,0,1}, t_{i,0,4}\}$: $sup_1(z) = 0$ für alle $z \in Z_\iota$ und $\mathcal{E}^{R_1}_{\mathsf{used}} = \{a_{3i}\}$ und $\mathcal{E}^{R_1}_{\mathsf{swap}} = \{w_0, X_{i_0}\}$ und $\mathcal{E}^{R_1}_{\mathsf{nop}} = \underline{E} \setminus \{w_0, X_{i_0}, a_{3i}\}$.

Die folgende Region $R_2 = (sup_2, sig_2)$ löst (a_{3i}, z) für alle $z \in \{t_{i,0,7}, t_{i,0,8}, t_{i,0,9}\}$: $sup_2(z) = 1$ für alle $z \in Z_\iota$ und $\mathcal{E}^{R_2}_{\mathsf{used}} = \{a_{3i}\}$ und $\mathcal{E}^{R_2}_{\mathsf{swap}} = \{a_{3i+1}\}$ und $\mathcal{E}^{R_2}_{\mathsf{nop}} = \underline{E} \setminus \{a_{3i}, a_{3i+1}\}$.

Die folgende Region $R_3 = (sup_3, sig_3)$ löst (a_{3i}, z) für alle $z \in \{t_{i,0,10}, \ldots, t_{i,0,16}\}$: $sup_3(z) = 1$ für alle $z \in Z_\iota$ und $\mathcal{E}^{R_3}_{\mathsf{used}} = \{a_{3i}\}$ und $\mathcal{E}^{R_3}_{\mathsf{swap}} = \{a_{3i+1}\}$ und $\mathcal{E}^{R_3}_{\mathsf{set}} = \{X_{i_1}\}$ und $\mathcal{E}^{R_3}_{\mathsf{nop}} = \underline{E} \setminus (\{a_{3i}, a_{3i+1}, X_{i_1}\})$. Dies zeigt die Lösbarkeit von a_{3i}. □

Fakt 4.10. *Wenn $e \in \mathfrak{U}$, dann ist e τ-lösbar.*

Beweis. Sei $i \in \{0, \ldots, m-1\}$ beliebig aber fest. Die folgende Region $R = (sup, sig)$ löst (X_{i_0}, z) für alle $z \in \{t_{i,0,0}, t_{i,0,1}, t_{i,0,2}\}$ und alle $z \in \{t_{i,0,6}, \ldots, t_{i,0,16}\}$: für alle $z \in Z_\iota$, wenn $z = t_{i,0,0}$, dann $sup(z) = 0$, sonst $sup(z) = 1$; außerdem $\mathcal{E}^{R}_{\mathsf{used}} = \{X_{i_0}\}$ und $\mathcal{E}^{R}_{\mathsf{swap}} = \{a_{3i}\}$ und $\mathcal{E}^{R}_{\mathsf{nop}} = \underline{E} \setminus (\{X_{i_0}, a_{3i}\})$.

Die Lösbarkeit von X_{i_1} und X_{i_2} in $T_{i,0}$ zeigt man analog. Es ist leicht zu sehen, dass X_{i_0} und –wieder ähnlich– auch X_{i_1} und X_{i_2} in $T_{i,1}, T_{i,2}$ und $T_{i,3}$ lösbar sind. Wir ersparen uns die ermüdenden Fallunterscheidungen. Weil i beliebig war, folgt die Behauptung. □

Insgesamt haben wir bisher den Nachweis für die τ-EZTE von A_τ erbracht. Wie man leicht sieht, weisen die Regionen, welche die τ-EZTE bezeugen, auch die τ-ZTE von A_τ nach. Wir erhalten das folgende Lemma, das den Beweis von Satz 4.8 vervollständigt:

Lemma 4.13. *Wenn $(\mathfrak{U}, M)$ ein Eins-in-drei-Modell hat, dann hat A_τ die τ-EZTE und die τ-ZTE.*

4.1.6 Die Netztypen A10

Satz 4.9. *Es seien $\tau_0 = \{\mathsf{nop}, \mathsf{res}\} \cup \omega$ mit $\omega \subseteq \{\mathsf{inp}, \mathsf{used}, \mathsf{free}\}$ und $\tau_1 = \{\mathsf{nop}, \mathsf{set}\} \cup \omega$ mit $\omega \subseteq \{\mathsf{out}, \mathsf{used}, \mathsf{free}\}$. Wenn $\tau \in \{\tau_0, \tau_1\}$, dann sind τ-*SPRACHSIMULATION *und τ-*REALISIERBARKEIT *in Polynomialzeit entscheidbar.*

Im weiteren Verlauf dieses Abschnitts beweisen wir Satz 4.9. Wenn $\tau = \{\mathsf{nop}, \mathsf{res}\}$ und $\tau' = \{\mathsf{nop}, \mathsf{set}\}$, dann gibt es für alle $\omega' \subseteq \{\mathsf{out}, \mathsf{used}, \mathsf{free}\}$ ein $\omega \subseteq \{\mathsf{inp}, \mathsf{used}, \mathsf{free}\}$, sodass $\tau \cup \omega$ und $\tau' \cup \omega'$ gemäß Definition 2.24 kongruent sind. Wir werden deshalb im Folgenden, solange nicht explizit anders vereinbart, $\tau = \{\mathsf{nop}, \mathsf{res}\} \cup \omega$ mit $\omega \subseteq \{\mathsf{inp}, \mathsf{used}, \mathsf{free}\}$ beliebig aber fest annehmen und zeigen, dass τ-SPRACHSIMULATION und τ-REALISIERUNG in Polynomialzeit entscheidbar sind. Zusammen mit Satz 2.2 folgt dann Satz 4.9.

Algorithmus 1 bildet den Kern eines polynomiellen Entscheidungsverfahrens für τ-SPRACHSIMULATION und τ-REALISIERUNG. Die wesentliche Idee dieser Prozedur lässt sich wie folgt zusammenfassen: starte für ein Trennungsproblem α eines Transitionssystems $A = (Z, E, \delta, \iota)$ mit einer –in zu definierendem Sinne– minimalen Menge $Q \subseteq Z$ von Zuständen, deren Support unter einer τ-Region, die α löst, positiv ist; füge solange *notwendigerweise* Zustände zu Q hinzu, wie Q nicht als Support (überhaupt) einer τ-Region $R = (Q, sig)$ aufgefasst werden kann (wobei diese α nicht notwendigerweise lösen muss);

Algorithmus 1 : Der Algorithmus erweitert $Q \subseteq Z$ zu einem Support einer τ-Region für $\tau = \{\mathsf{nop}, \mathsf{res}\} \cup \omega$ mit $\omega \subseteq \{\mathsf{inp}, \mathsf{used}, \mathsf{free}\}$.

Input : ein Transitionssystem $A = (Z, E, \delta, \iota)$ und eine Menge $Q \subseteq Z$
Output : ein Support $sup \supseteq Q$ einer Region von A

1 **while** *Es gibt $z \in Q, z' \notin Q$ und $e \in E$, sodass (1) $z' \xrightarrow{e} z \in A$ oder (2) $z \xrightarrow{e} z' \in A$* und *es gibt $q \xrightarrow{e} q' \in A$ mit $q, q' \in Q$* **do**
2 $\quad Q = Q \cup \{z'\}$;
3 **return** $sup = Q$;

gib Q schließlich zurück, wenn Q als ein solcher Support aufgefasst werden kann. Letzteres ist stets der Fall, denn wenn $Q = Z$, dann ist mit $sig(e) = \mathsf{nop}$ für alle $e \in E$ das Paar $R = (Q, sig)$ eine τ-Region. Wenn es eine lösende Region gibt, dann gibt es eine passende Menge Q, sodass wir ihren Support auf diese Weise erhalten, andernfalls können wir die Existenz einer solchen Region ausschließen.

Lemma 4.14 (Algorithmus 1)**.** *Sei $\tau = \{\mathsf{nop}, \mathsf{res}\} \cup \omega$ mit $\omega \subseteq \{\mathsf{inp}, \mathsf{used}, \mathsf{free}\}$ und $A = (Z, E, \delta, \iota)$ ein Transitionssystem und $Q \subseteq Z$. Wenn Algorithmus 1 mit der Eingabe A, Q gestartet wird, dann ist die Ausgabe sup der Support der τ-Region $R = (sup, sig)$, deren Signatur sig für alle $e \in E$ wie folgt definiert ist:*

$$sig(e) = \begin{cases} \mathsf{used}, & \text{falls } \mathsf{used} \in \tau \text{ und } \{z, z' \in Z \mid z \xrightarrow{e} z' \in A\} \subseteq sup, \\ \mathsf{free}, & \text{falls } \mathsf{free} \in \tau \text{ und } \{z, z' \in Z \mid z \xrightarrow{e} z' \in A\} \cap sup = \emptyset, \\ \mathsf{inp}, & \text{falls } \mathsf{inp} \in \tau \text{ und für alle } z \xrightarrow{e} z' \text{ in } A \text{ gilt } z \in sup \text{ und } z' \notin sup, \\ \mathsf{res}, & \text{falls } \mathsf{inp} \notin \tau \text{ und für alle } z \xrightarrow{e} z' \text{ in } A \text{ gilt } z \in sup \text{ und } z' \notin sup, \\ \mathsf{res}, & \text{falls es } z \xrightarrow{e} z' \text{ und } q \xrightarrow{e} q' \text{ in } A \text{ gibt mit } z \in sup \text{ und } z', q, q' \notin sup, \\ \mathsf{nop}, & \text{sonst.} \end{cases}$$

Wenn $R' = (sup', sig')$ eine τ-Region von A mit $Q \subseteq sup'$ ist, dann gilt außerdem $sup \subseteq sup'$. Die Laufzeit des Algorithmus beträgt $\mathcal{O}(|Z|^5 \cdot |E|^2)$.

Beweis. Algorithmus 1 terminiert, da die Menge Q bei jeder Iteration um einen Zustand erweitert wird und dies höchstens $|Z|$ mal möglich ist. Darüber hinaus gilt offensichtlich $Q \subseteq sup$, nachdem Algorithmus 1 terminiert ist.

Es ist zu zeigen, dass $R = (sup, sig)$ eine wohldefinierte τ-Region ist. Weil $\{\mathsf{nop}, \mathsf{res}\} \subseteq \tau$ gilt, dass jedes $e \in E$ auf einen Wert $sig(e)$ aus τ abgebildet wird. Darüber hinaus ist sig eindeutig, da sich die sechs aufgeführten Fälle der Definition paarweise gegenseitig ausschließen. Sei $e \in E$ beliebig aber fest. Es ist zu zeigen, dass $sup(z) \xrightarrow{sig(e)} sup(z') \in \tau$ für alle $z \xrightarrow{e} z' \in A$ gilt. Wenn $sig(e) \in \{\mathsf{inp}, \mathsf{used}, \mathsf{free}\}$, dann ist dies offensichtlich erfüllt.

Untersuchen wir den Fall $sig(e) = \mathsf{res}$. Angenommen, es gibt $t \xrightarrow{e} t' \in A$, sodass $sup(t) \xrightarrow{\mathsf{res}} sup(t') \notin \tau$. Wegen $0 \xrightarrow{\mathsf{res}} 0 \in \tau$ und $1 \xrightarrow{\mathsf{res}} 0 \in \tau$ impliziert dies $sup(t') = 1$. Falls $sup(t) = 0$, dann erfüllt $t \xrightarrow{e} t'$ die Bedingung (1) der While-Schleife. Dies widerspricht der Voraussetzung, dass sup von Algorithmus 1 ausgegeben wurde. Andernfalls, wenn $sup(t) = 1$, dann muss es gemäß der Fallunterscheidungen für die Definition von sig eine Kante $z \xrightarrow{e} z'$ geben (fünfte Zeile), sodass $sup(z) = 1$ und $sup(z') = 0$. Dann erfüllen $z \xrightarrow{e} z'$ und $t \xrightarrow{e} t'$ aber die Bedingung (2) der While-Schleife. Dies widerspricht erneut der Annahme, dass sup nach Terminierung von Algorithmus 1 ausgegeben wurde. Mithin war unsere Annahme falsch und es gilt $sup(t) \xrightarrow{\mathsf{res}} sup(t') \in \tau$ für alle $t \xrightarrow{e} t' \in A$.

Es bleibt der Fall $sig(e) = \mathsf{nop}$: Nehmen wir an, es gibt $t \xrightarrow{e} t' \in A$ mit $sup(t) \neq sup(t')$. Wenn $sup(t) = 0$, dann gilt $sup(t') = 0$ und insbesondere $sup(t) \xrightarrow{\mathsf{nop}} sup(t') \in \tau$,

denn andernfalls würde $t \xrightarrow{e} t'$ Bedingung (1) der While-Schleife erfüllen und wir hätten einen Widerspruch zur Terminierung von Algorithmus 1. Daraus folgt $sup(t) = 1$ und $sup(t') = 0$. Man sieht sofort: Wenn es eine Kante $z \xrightarrow{e} z' \in A$ mit $sup(z) = 0$ und $sup(z') = 1$ oder $sup(z) = sup(z') = 1$ gibt, dann erhalten wir wegen Bedingung 1 beziehungsweise Bedingung 2 der While-Schleife einen Widerspruch zur Terminierung von Algorithmus 1. Für alle e-beschrifteten Kanten $t \xrightarrow{e} t' \neq z \xrightarrow{e} z' \in A$ gilt somit $sup(z) = 1$ und $sup(z') = 0$ oder $sup(z) = sup(z') = 0$. Dies widerspricht der Definition von sig, denn in diesem Fall soll entweder $sig(e) = \mathsf{inp}$ oder $sig(e) = \mathsf{res}$ gelten. Somit impliziert $sig(e) = \mathsf{nop}$ stets $sup(z) = sup(z')$ und $sup(z) \xrightarrow{\mathsf{nop}} sup(z')$ für alle $z \xrightarrow{e} z' \in A$. Insbesondere ist R eine wohldefinierte τ-Region.

Sei $R' = (sup', sig')$ eine beliebige aber feste τ-Region von A mit $Q \subseteq sup'$. Es ist zu zeigen, dass $sup \subseteq sup'$. Dazu zeigen wir induktiv (über die Anzahl i der Schleifenaufrufe), dass die Menge Q_i, welche aus $Q_0 = Q$ nach i Aufrufen der While-Schleife von Algorithmus 1 resultiert, eine Teilmenge von sup' ist. Weil $sup = Q_n$ nach $n \leq |Z|$ Iterationen, folgt dann $sup \subseteq sup'$. Für den Induktionsanfang gilt (nach Voraussetzung) $Q_0 = Q \subseteq sup'$. Gelte $Q_i \subseteq sup'$. Nehmen wir an, dass $Q_{i+1} \not\subseteq sup'$. Weil Q_{i+1} aus Q_i (und A) und der $i+1$-ten Iteration resultiert, gibt es einen eindeutigen Zustand z', sodass $Q_{i+1} \setminus Q_i = \{z'\}$. Insbesondere gilt $z' \notin sup'$. Weil $Q_{i+1} = Q_i \cup \{z'\}$, gibt es $z' \xrightarrow{e} z \in A$ mit $z' \notin Q_i \subseteq \sup'$ und $z \in Q_i \subseteq \sup'$ oder $z \xrightarrow{e} z' \in A$ und $q \xrightarrow{e} q' \in A$ mit $z' \notin Q_i \subseteq \sup'$ und $z, q, q' \in Q_i \subseteq \sup'$. Daraus folgt aber $sig'(e) \notin \{\mathsf{nop}, \mathsf{inp}, \mathsf{res}, \mathsf{used}, \mathsf{free}\}$ und dies ist ein Widerspruch zu Annahme, dass R' eine τ-Region ist. Somit gilt $Q_{i+1} \subseteq sup'$.

Zur Laufzeit des Algorithmus: Zunächst ist klar, dass es höchstens $|Z|$ Aufrufe der While-Schleife geben kann. Weil δ eine (partielle) Funktion ist, hat A höchstens $|Z|^2 \cdot |E|$ verschiedene Kanten. Um die While-Schleifen-Bedingung zu testen, müssen also höchstens $|Z|^2 \cdot |E|$ Kanten paarweise miteinander verglichen werden, sodass wir mit $|Z|^4 \cdot |E|^2$ Vergleichen auskommen. Deshalb beträgt die Laufzeit von Algorithmus 1 $\mathcal{O}(|Z|^5 \cdot |E|^2)$. □

Gemäß Lemma 4.14 impliziert die Menge sup, die Algorithmus 1 auf Eingabe Q und A ausgibt, die eindeutige τ-Region $R = (sup, sig)$. Wir werden deshalb im Folgenden auch von *der* τ-Region $R = (sup, sig)$ sprechen, die Algorithmus 1 auf Eingabe Q (und A) ausgibt.

Lemma 4.15. *Sei $\tau = \{\mathsf{nop}, \mathsf{res}\} \cup \omega$ mit $\omega \subseteq \{\mathsf{inp}, \mathsf{used}, \mathsf{free}\}$, $A = (Z, E, \delta, \iota)$ ein Transitionssystem und $\alpha = (q, q')$ ein ZTP von A und $\beta = (e, q)$ ein EZTP von A. Das ZTP β ist genau dann τ-lösbar, wenn $sup(q') = 0$ beziehungsweise $sup(q) = 0$ für die Region $R = (sup, sig)$ gilt, die von Algorithmus 1 auf Eingabe $Q = \{q\}$ beziehungsweise $Q = \{q'\}$ ausgegeben wird. Es ist β genau dann τ-lösbar, wenn eine der folgenden Bedingungen wahr ist:*

1. *Die τ-Region $R = (sup, sig)$, die Algorithmus 1 auf Eingabe $Q = \{z \in Z \mid z \xrightarrow{e}\}$ ausgibt, erfüllt $sig(e) = \mathsf{inp}$ und $sup(z) = 0$.*

2. *Die τ-Region $R = (sup, sig)$, die Algorithmus 1 auf Eingabe $Q = \{z, z' \in Z \mid z \xrightarrow{e} z'\}$ ausgibt, erfüllt $sig(e) = \mathsf{used}$ und $sup(z) = 0$.*

3. *Die τ-Region $R = (sup, sig)$, die Algorithmus 1 auf Eingabe $Q = \{z\}$ ausgibt, erfüllt $sig(e) = \mathsf{free}$.*

Beweis. Zur Lösbarkeit von α: Es ist klar, dass eine τ-Region, die eine der Bedingungen erfüllt, das ZTP α löst. Sei $R' = (sup', sig')$ eine τ-Region, die α löst, dass heißt, wir haben entweder $q \in sup'$ und $q' \notin sup'$ oder $q \notin sup'$ und $q' \in sup$. Gemäß Lemma 4.14 gilt im ersten Fall $q \in sup \subseteq sup'$ und $q' \notin sup$, wenn $R = (sup, sig)$ aus Eingabe $\{q\}$ resultiert. Analog gilt im zweiten Fall, dass $q \notin sup$ und $q' \in sup$, wenn $R = (sup, sig)$ aus Eingabe $\{q'\}$ resultiert. Insbesondere löst R in einem der Fälle α.

Zur Lösbarkeit von β: Es ist klar, dass eine τ-Region, die eine der drei genannten Bedingungen erfüllt, das EZTP β löst.

Sei $R = (sup', sig')$ eine τ-Region, welche α löst. Weil $sig'(e) \in \tau \cap \mathfrak{inhibit}$, gilt $sig(e) \in \{\mathsf{inp}, \mathsf{used}, \mathsf{free}\}$.

Beginnen wir mit dem Fall $sig'(e) = \mathsf{inp}$ und $sup'(q) = 0$: Sei $Q = \{z \mid z \xrightarrow{e} \in A\}$ die Menge aller Ursprünge und $X = \{z \mid \xrightarrow{e} z \in A\}$ die Menge aller Ziele von e in A. Weil $sig'(e) = \mathsf{inp}$ und $sup'(z) \xrightarrow{sig'(e')} sup(z') \in \tau$ für alle Kanten $z \xrightarrow{e'} z' \in A$, erhalten wir $Q \subseteq sup'$ und $X \cap sup' = \emptyset$. Sei $R = (sup, sig)$ die Ausgabe von Algorithmus 1 auf Eingabe Q. Nach Lemma 4.14 gilt, $Q \subseteq sup \subseteq sup'$. Dies impliziert $(\{q\} \cup X) \cap sup = \emptyset$ und weil $\mathsf{inp} \in \tau$ gilt außerdem $sig(e) = \mathsf{inp}$. Somit löst R das EZTP β.

Analog zeigt man: Wenn $sig'(e) = \mathsf{used}$ und Algorithmus 1 wird auf $Q = \{z, z' \in Z \mid z \xrightarrow{e} z'\}$ gestartet, dann löst $R = (sup, sig)$ das EZTP β mit $sig(e) = \mathsf{used}$ und $sup(q) = 0$; Wenn $sig'(e) = \mathsf{free}$ und Algorithmus 1 wird auf $Q = \{q\}$ gestartet, dann löst $R = (sup, sig)$ das EZTP β mit $sig(e) = \mathsf{free}$ und $sup(q) = 1$. □

Das folgende Lemma zeigt, dass sowohl die Existenz eines Zeugen für die τ-EZTE als auch für die τ-ZTE eines Transitionssystems A mit polynomiellen Zeitaufwand bestimmt werden kann, wenn τ mit einem Netztypen A10 aus Abbildung 4.3 korrespondiert. Wegen der eingangs diskutierten Kongruenzen komplettiert es somit den Beweis von Satz 4.9.

Lemma 4.16. *Es seien $\tau_0 = \{\mathsf{nop}, \mathsf{res}\} \cup \omega$ mit $\omega \subseteq \{\mathsf{inp}, \mathsf{used}, \mathsf{free}\}$ und $A = (Z, E, \delta, \iota)$ ein Transitionssystem. Es gibt einen Algorithmus der mit Zeitaufwand $\mathcal{O}(|Z|^6 \cdot |E|^2 \cdot max\{|E|, |Z|\})$ entscheidet, ob es eine Zeugen für die τ-EZTE beziehungsweise die τ-ZTE von A gibt.*

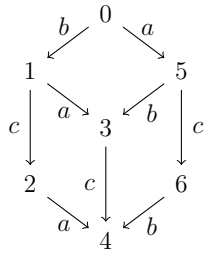
Beweis. (τ-EZTE): Innerhalb der angegebenen Laufzeit wollen wir im Falle einer positiven Entscheidung einen Zeugen $\mathcal{R}$ für die τ-EZTE finden beziehungsweise im Falle einer negativen Entscheidung die Existenz eines Zeugen begründet ablehnen. Falls $\tau = \{\mathsf{nop}, \mathsf{res}\}$, dann hat A genau dann die τ-EZTE, wenn es keine EZTP hat. (Die Interaktionen von τ sind beide jeweils an 0 und 1 definiert, sodass es keine τ-Region gibt, die ein EZTP löst.) Dies lässt sich mit Zeitaufwand $\mathcal{O}(|E| \cdot |Z|^2)$ entscheiden.

Andernfalls starten wir mit einer leeren Menge $\mathcal{R} = \emptyset$, die wir mit Hilfe von Algorithmus 1 zu einem Zeugen für die τ-EZTE von A erweitern, falls dies möglich ist: Sei $\alpha = (a, q)$ ein beliebiges aber festes EZTP von A. Seien $Q_{\mathsf{inp}} = \{z \in Z \mid z \xrightarrow{a}\}$ und $Q_{\mathsf{used}} = \{z, z' \in Z \mid z \xrightarrow{a} z'\}$ und $Q_{\mathsf{free}} = \{q\}$. Für jede der Interaktionen $i \in \{\mathsf{inp}, \mathsf{used}, \mathsf{free}\} \cap \tau$ starten wir Algorithmus 1 genau einmal auf Eingabe Q_i und A. Wenn einer der Aufrufe eine Region R liefert, die α löst, dann fügen wir R zu $\mathcal{R}$ hinzu. Wenn keiner der Aufrufe eine solche Region liefert, dann können wir gemäß Lemma 4.15 die τ-EZTE für A begründet ablehnen. Diese Prozedur wiederholen wir sukzessive für jedes nicht bereits durch Regionen aus $\mathcal{R}$ gelöste EZTP von A. Das Transitionssystem A hat höchstens $|E| \cdot |Z|$ verschiedene EZTP. Gemäß Lemma 4.14 beträgt die jeweilige Laufzeit (und somit die Gesamtlaufzeit für alle drei Aufrufe) $\mathcal{O}(|Z|^5 \cdot |E|^2)$. Daraus folgt, dass wir nach einem Zeitaufwand von $\mathcal{O}(|Z|^6 \cdot |E|^3)$ entweder einen Zeugen $\mathcal{R}$ für die τ-EZTE von A erhalten oder dessen Existenz begründet negieren können.

(τ-ZTE): Wir gehen analog zur Argumentation für die τ-EZTE vor. Für jedes ZTP $\beta = (z, z')$ von A starten wir Algorithmus 1 jeweils einmal mit Eingabe $\{z\}$ und einmal mit $\{z'\}$. Liefert uns einer der Aufrufe eine Region R, die β löst, dann fügen wir sie zu (der initial leeren) Menge $\mathcal{R}$ hinzu, andernfalls wird die τ-ZTE von A gemäß Lemma 4.15 begründet abgelehnt. Weil A höchstens $|Z|^2$ ZTP hat, erhalten wir so nach einer Laufzeit von $\mathcal{O}(|Z|^7 \cdot |E|^2)$ entweder einen Zeugen $\mathcal{R}$ für die τ-ZTE von A oder dürfen ihre Existenz gerechtfertigt negieren. □

Das nächste Beispiel demonstriert die in Lemma 4.16 beschriebene Methode.

Beispiel 4.1. Sei $\tau = \{\mathsf{nop}, \mathsf{inp}, \mathsf{res}, \mathsf{free}\}$. Wir untersuchen, ob das folgende Transitionssystem aus dem Buch [BBD15] die τ-EZTE hat:



Sei $\mathcal{R} = \emptyset$. Wir starten mit dem EZTP $(a, 3)$. Weil die Interaktion inp in τ ist, rufen wir Algorithmus 1 zunächst mit der Menge $Q^a_{\mathsf{inp}} = \{0, 1, 2\}$ auf. Da Q^a_{inp} keine der Bedingungen der While-Schleife erfüllt, terminiert der Algorithmus unverzüglich und gibt $sup_1 = Q^a_{\mathsf{inp}}$ zurück. Gemäß Lemma 4.14 erhalten wir die Region $R_1 = (sup_1, sig_1)$ mit der Signatur sig_1, die definiert ist durch $sig_1(a) = \mathsf{inp}$, $sig_1(b) = sig_1(c) = \mathsf{nop}$. Wir erhalten $\mathcal{R} = \{R_1\}$. Es löst R_1 außerdem (a, z) für alle $z \in \{4, 5, 6\}$ und zeigt damit die τ-Lösbarkeit von a.

Wir fahren mit dem EZTP $(b, 1)$ fort. Analog erhalten wir, dass Algorithmus 1 für die Eingabe $Q^b_{\mathsf{inp}} = \{0, 5, 6\}$ unverzüglich terminiert und $sup_2 = Q^b_{\mathsf{inp}}$ zurückgibt. Die resultierende Region $R_2 = (sup_2, sig_2)$ mit Signatur $sig_2(b) = \mathsf{inp}$ sowie $sig_2(c) = sig_2(c) = \mathsf{nop}$

löst (b, z) für alle $z \in \{1, 2, 3, 4\}$. Wir erweitern $\mathcal{R} = \{R_1, R_2\}$ und haben bereits einen Zeugen für die τ-Lösbarkeit von a und b.

Für das EZTP $(c, 2)$ starten wir Algorithmus 1 auf die Eingabe $Q^c_{\mathsf{inp}} = \{1, 3, 5\}$. Mit $1 \in Q^c_{\mathsf{inp}}$ und $0 \notin Q^c_{\mathsf{inp}}$ erfüllt die Kante $0 \xrightarrow{b} 1$ Bedingung (1) der While-Schleife. Deshalb wird $\{1, 3, 5\}$ zu $\{0, 1, 3, 5\}$ erweitert und Algorithmus 1 terminiert mit der Ausgabe $sup_3 = \{0, 1, 3, 5\}$. Lemma 4.14 liefert die zugehörige Region $R_3 = (sup_3, sig_3)$, wobei $sig_3(c) = \mathsf{inp}$ und $sig_3(a) = sig_3(b) = \mathsf{nop}$. Offenbar löst R_3 das EZTP (c, z) für alle $z \in \{2, 4, 6\}$. Wir erhalten $\mathcal{R} = \{R_1, R_2, R_3\}$ und es bleibt, die τ-Lösbarkeit des EZTP $(c, 0)$ zu untersuchen.

Wenn Algorithmus 1 auf $Q^c_{\mathsf{used}} = \{1, \ldots, 6\}$ gestartet wird, dann erfüllt die Kante $0 \xrightarrow{b} 1$ erneut Bedingung (1) der While-Schleife und die resultierende Rückgabe $sup = \{0, \ldots, 6\}$ erlaubt keine lösende τ-Region. Gestartet auf $Q^c_{\mathsf{free}} = \{0\}$ terminiert der Algorithmus jedoch unverzüglich und gibt $sup_4 = \{0\}$ zurück. Wir erhalten die τ-Region $R_4 = (sup_4, sig_4)$ mit $sig_4(c) = \mathsf{free}$ und $sig_4(a) = sig_4(b) = \mathsf{res}$, die $(c, 0)$ löst.

Schließlich erhalten wir den Zeugen $\mathcal{R} = \{R_1, R_2, R_3, R_4\}$ für die τ-EZTE von A.

4.1.7 Die Netztypen A11

Satz 4.10. *Wenn $\tau = \{\mathsf{nop}, \mathsf{set}, \mathsf{swap}\}$ oder $\tau = \{\mathsf{nop}, \mathsf{res}, \mathsf{swap}\}$ oder $\tau = \{\mathsf{nop}, \mathsf{set}, \mathsf{res}\}$ oder $\tau = \{\mathsf{nop}, \mathsf{set}, \mathsf{res}, \mathsf{swap}\}$, dann ist τ-*Sprachsimulation *in Polynomialzeit entscheidbar.*

Beweis. Weil $\tau \cap \mathfrak{inhibit} = \emptyset$, hat ein Transitionssystem genau dann die τ-EZTE, wenn jedes Ereignis von A an jedem Zustand von A stattfindet. Diese Eigenschaft lässt sich sicherlich mit polynomiellem Zeitaufwand in der Größe von A untersuchen. □

4.2 Realisierung für Boolesche Netztypen ohne nop und swap

In diesem Abschnitt klassifizieren wir die Komplexität von τ-Realisierung für alle booleschen Netztypen, die weder Unabhängigkeit (nop) noch bedingungsloses Invertieren (swap) erlauben. Genauer gesagt ist τ-Realisierung für diese Netztypen gemäß des folgenden Satzes in P:

Satz 4.11. *Wenn τ ein boolescher Netztyp ohne nop und ohne swap ist, dann ist τ-*Realisierung *in Polynomialzeit entscheidbar.*

Der Rest dieses Abschnitts ist dem Beweis von Satz 4.11 gewidmet. Die folgende Definition 4.4 und das folgende Lemma 4.2 zielen auf eine strukturelle Eigenschaft ab, die τ-realisierbaren Transitionssystemen eigen ist, wenn τ mit Satz 4.11 korrespondiert:

Definition 4.4 (stark zielorientiert, eindeutiges Ziel)**.** Sei $A = (Z, E, \delta, \iota)$ ein Transitionssystem und $e \in E$ ein Ereignis von A. Wir sagen ein Ereignis $e \in E$ ist *stark zielorientiert*, wenn $|\{z \in Z \mid \xrightarrow{e} z\}| = 1$. Wenn e stark zielorientiert ist, dann nennen wir $z \in Z$ mit $\xrightarrow{e} z$ das *eindeutige Ziel* von e. Wenn jedes Ereignis von A stark zielorientiert ist, dann nennen wir A *stark zielorientiert*.

Lemma 4.17. *Sei $\tau \subseteq \{\mathsf{inp}, \mathsf{out}, \mathsf{res}, \mathsf{set}, \mathsf{used}, \mathsf{free}\}$ und $A = (Z, E, \delta, \iota)$ ein Transitionssystem. Wenn A die τ-ZTE hat, dann ist A stark zielorientiert.*

Beweis. Es habe A die τ-ZTE. Nehmen wir an, es gibt ein Ereignis $e \in E$ und zwei verschiedene Zustände $z, z' \in Z$ mit $\xrightarrow{e} z$ und $\xrightarrow{e} z'$. Weil A die τ-ZTE hat, gibt es eine τ-Region $R = (sup, sig)$ mit $sup(z) \neq sup(z')$. Dies widerspricht $sig(e) \notin \{\mathsf{nop}, \mathsf{swap}\}$ und beweist das Lemma. □

Tatsächlich hat ein τ-realisierbares Transitionssystem A nach Definition die τ-ZTE. Weil A höchstens $|Z|^2 \cdot |E|$ Kanten hat, kann darüber hinaus mit Zeitaufwand $\mathcal{O}(|Z|^4 \cdot |E|^2)$ entschieden werden, ob A stark zielorientiert ist. Lemma 4.2 rechtfertigt deshalb, im Folgenden stets anzunehmen, dass eine Eingabe A stark zielorientiert ist, denn andernfalls darf A begründet als nicht realisierbar abgelehnt werden.

Satz 4.11 umfasst 64 boolesche Netztypen und die weitere Gliederung des Abschnitts ist wie folgt: Der folgende Abschnitt 4.2.1 beweist die Aussage von Satz 4.11 für die 16 Netztypen mit res und set, das heißt $\tau = \{\mathsf{res}, \mathsf{set}\} \cup \omega$ mit $\omega \subseteq \{\mathsf{inp}, \mathsf{out}, \mathsf{used}, \mathsf{free}\}$. Daraufhin wird in Abschnitt 4.2.2 und Abschnitt 4.2.3 ein polynomielles Entscheidungsverfahren für τ-REALISIERUNG zunächst explizit für die 16 Netztypen mit set aber ohne res entwickelt, das heißt $\tau = \{\mathsf{set}\} \cup \omega$ mit $\omega \subseteq \{\mathsf{inp}, \mathsf{out}, \mathsf{used}, \mathsf{free}\}$. Weil es für alle $\tau' = \{\mathsf{res}\} \cup \omega'$ mit $\omega' \subseteq \{\mathsf{inp}, \mathsf{out}, \mathsf{res}, \mathsf{set}\}$ ein $\omega \subseteq \{\mathsf{inp}, \mathsf{out}, \mathsf{res}, \mathsf{set}\}$ gibt, sodass $\{\mathsf{set}\} \cup \omega \equiv \tau'$, zeigt dies ebenso für die 16 Netztypen mit res aber ohne set, dass τ'-REALISIERUNG in Polynomialzeit entscheidbar ist. Schließlich wird in Abschnitt 4.2.4 der Beweis von Satz 4.11 komplettiert, indem die Aussage für die 16 Netztypen mit $\tau \subseteq \{\mathsf{inp}, \mathsf{out}, \mathsf{used}, \mathsf{free}\}$ bewiesen wird.

4.2.1 Netztypen mit res und set

Solange nicht ausdrücklich anders vereinbart sei im Folgenden $\tau = \{\mathsf{res}, \mathsf{set}\} \cup \omega$ mit $\omega \subseteq \{\mathsf{inp}, \mathsf{out}, \mathsf{used}, \mathsf{free}\}$ beliebig aber fest. Darüber hinaus definieren wir $\tau_0 = \{\mathsf{inp}, \mathsf{res}, \mathsf{set}\}$ und $\tau_1 = \{\mathsf{out}, \mathsf{res}, \mathsf{set}\}$ und $\tau_2 = \{\mathsf{res}, \mathsf{set}, \mathsf{used}\}$ sowie $\tau_3 = \{\mathsf{res}, \mathsf{set}, \mathsf{free}\}$ und verweisen auf die graphische Repräsentation dieser Netztypen in Abbildung 4.5.

Die wesentlichen Einsichten in die Struktur von τ und seiner Eigenschaften bezüglich τ-REALISIERUNG können wie folgt zusammengefasst werden: Ein stark zielorientiertes Transitionssystem A hat stets die τ-ZTE. Außerdem kann die Entscheidung für die τ-Lösbarkeit eines EZTP α von A im Wesentlichen auf die τ_0- beziehungsweise τ_2-Lösbarkeit von α reduziert werden. Darüber hinaus muss für höchstes zwei Regionen festgelegter

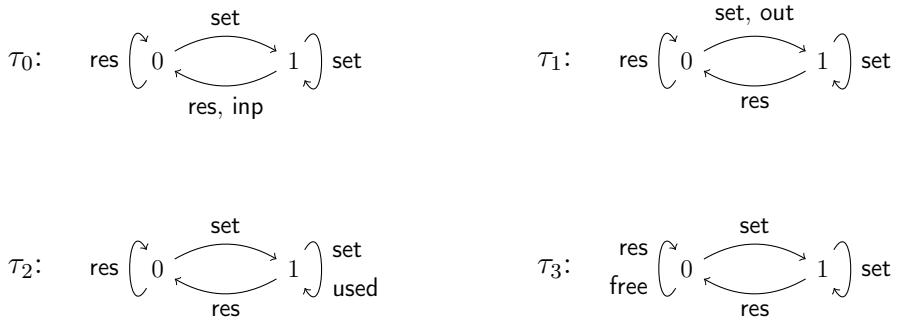


Abbildung 4.5: Die kongruenten Netztypen τ_0 und τ_1 sowie τ_2 und τ_3.

Bauart getestet werden, ob sie α lösen. Insbesondere ist α nicht τ-lösbar, wenn es keine dieser Regionen löst.

Lemma 4.18. *Ein stark zielorientiertes Transitionssystem $A = (Z, E, \delta, \iota)$ hat die τ-ZTE.*

Beweis. Ein beliebiges aber festes ZTP (p, q) von A wird durch die folgende τ-Region $R = (sup, sig)$ gelöst: Für alle $z \in Z$, wenn $z = q$, dann $sup(z) = 0$, andernfalls $sup(z) = 1$ und für alle $e \in E$, falls $\xrightarrow{e}q$, dann $sig(e) = \mathsf{res}$, andernfalls $sig(e) = \mathsf{set}$. Weil A stark zielorientiert ist, gilt für alle $e \in E$, dass genau dann $sig(e) = \mathsf{res}$, wenn aus $z \xrightarrow{e} z'$ stets $z' = q$ folgt. Deshalb ist R, wie in Abbildung 4.6 (1) skizziert, wohldefiniert. □

Lemma 4.19. *Sei $A = (Z, E, \delta, \iota)$ ein stark zielorientiertes Transitionssystem und $\alpha = (a, q)$ ein EZTP von A. Es ist α genau dann τ-lösbar, wenn es $\tau' \in \{\tau_0, \tau_2\}$ und $\tau'' \subseteq \tau$ gibt, sodass α τ'-lösbar ist und $\tau' \equiv \tau''$.*

Beweis. „⇒“: Sei α τ-lösbar und $R = (sup, sig)$ eine τ-Region, die α löst. Zunächst gilt für alle $b, b' \in \{0, 1\}$ und alle $i \in \{\mathsf{inp}, \mathsf{res}, \mathsf{free}\}$ beziehungsweise $j \in \{\mathsf{out}, \mathsf{set}, \mathsf{used}\}$, dass wenn $b \xrightarrow{i} b'$, dann $b \xrightarrow{\mathsf{res}} b'$ beziehungsweise wenn $b \xrightarrow{j} b'$, dann $b \xrightarrow{\mathsf{set}} b'$. Deshalb ist die folgende $\{\mathsf{res}, \mathsf{set}, sig(a)\}$-Region $R' = (sup, sig')$ (mit dem gleichen Support wie R)

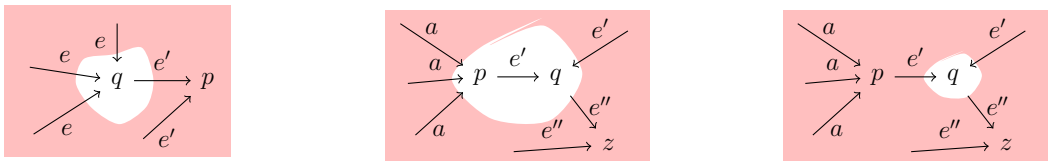


Abbildung 4.6: (1) In Anlehnung an Lemma 4.18, die Skizze einer Region R, die das eindeutige Ziel q von e von allen anderen Zuständen trennt. (2) Skizze einer τ_0-Region die das EZTP (a, q) gemäß Lemma 4.20.1 löst und (3) Skizze einer τ_2-Region die das EZTP (a, q) gemäß Lemma 4.20.2 löst.

wohldefiniert, wobei $sig'(e)$ für alle $e \in E$ wie folgt festgelegt wird: wenn $e = a$, dann $sig'(e) = sig(e)$; andernfalls, wenn $sig(e) \in \{\mathsf{inp}, \mathsf{res}, \mathsf{free}\}$, dann $sig'(e) = \mathsf{res}$ und wenn $sig(e) \in \{\mathsf{out}, \mathsf{set}, \mathsf{used}\}$, dann $sig'(e) = \mathsf{set}$. Weil R und R' denselben Support haben und $sig'(a) = sig(a)$ gilt, löst R' zudem α. Wenn $sig(a) \in \{\mathsf{inp}, \mathsf{used}\}$, dann ist R' bereits eine τ_0- oder eine τ_2-Region und wir sind fertig. Andernfalls ist $sig(a) \in \{\mathsf{out}, \mathsf{free}\}$, denn R' löst α und es gilt $\{\mathsf{res}, \mathsf{set}, sig(a)\} \equiv \tau_0$ oder $\{\mathsf{res}, \mathsf{set}, sig(a)\} \equiv \tau_2$. In diesem Fall folgt die Behauptung mit Satz 2.2.

„⇐“: Die Aussage folgt mit Satz 2.2. □

Das folgende Lemma zeigt, dass für jedes EZTP α eines stark zielorientierten Transitionssystems höchstens zwei genau festgelegte τ_0- beziehungsweise τ_2-Regionen für die Lösung von α in Betracht kommen können:

Lemma 4.20. *Sei $A = (Z, E, \delta, \iota)$ ein stark zielorientiertes Transitionssystem und $\alpha = (a, q)$ ein EZTP von A und p das eindeutige Ziel von a.*

1. *α ist genau τ_0-lösbar, wenn eine wie folgt definierte τ-Region $R = (sup, sig)$ von A existiert: $sup = Z \setminus \{p, q\}$ und $sig(a) = \mathsf{inp}$ und für alle $e \in E \setminus \{a\}$ ist $sig(e) = \mathsf{res}$, falls $\xrightarrow{e} p$ oder $\xrightarrow{e} q$ und sonst $sig(e) = \mathsf{set}$.*

2. *α ist genau τ_2-lösbar, wenn eine wie folgt definierte τ-Region $R = (sup, sig)$ von A existiert: $sup = Z \setminus \{q\}$ und $sig(a) = \mathsf{used}$ und für alle $e \in E \setminus \{a\}$ ist $sig(e) = \mathsf{res}$, falls $\xrightarrow{e} q$ und sonst $sig(e) = \mathsf{set}$.*

Beweis. (1): Zunächst sei bemerkt, dass Abbildung 4.6 (2) die Region R skizziert.

„⇐“: wegen $sup(q) = 0$ und $sig(a) = \mathsf{inp}$ löst R das EZTP α.

„⇒“: Sei $R' = (sup', sig')$ eine τ_0-Region, die α löst. Dies impliziert $sig'(a) = \mathsf{inp}$ und $sup'(z) = 1$ für alle Ursprünge $z \in Z$ von a. Insbesondere kann p kein Ursprung von a sein. Es reicht zu zeigen, dass R eine wohldefinierte τ_0-Region ist. Angenommen, es gibt eine Kante $z \xrightarrow{e} z'$ mit $sup(z) \xrightarrow{sig(e)} sup(z') \notin \tau_0$. Dies impliziert $sup(z') = 0$ und somit $z' \in \{p, q\}$, denn aus $sup(z') = 1$ folgt $sig(e) = \mathsf{set}$ und damit $sup(z) \xrightarrow{sig(e)} sup(z') \in \tau_0$. Wenn $e = a$, dann $z \notin \{p, q\}$, denn z ist ein Ursprung von a, und wir erhalten mit $sig(a) = \mathsf{inp}$ auch $sup(z) \xrightarrow{sig(e)} sup(z') \in \tau_0$. Wenn $e \neq a$, dann erhalten wir mit $sup(z') = 0$, dass $sig(e) = \mathsf{res}$ und es gilt $sup(z) \xrightarrow{sig(e)} sup(z') \in \tau_0$. Mithin war die Annahme $sup(z) \xrightarrow{sig(e)} sup(z') \notin \tau_0$ falsch und R ist wohldefiniert.

(2): Die Region R wird in Abbildung 4.6 (3) skizziert.

„⇐“: wegen $sup(q) = 0$ und $sig(a) = \mathsf{used}$ löst R das EZTP α.

„⇒“: Eine τ_2-Region $R' = (sup', sig')$, die α löst, erfüllt $sup'(q) = 0$ und $sig'(a) = \mathsf{used}$ und $sup'(z) = sup'(z') = 1$ für alle Kanten $z \xrightarrow{a} z'$ in A. Somit ist q kein Ziel von a. Weil A zudem stark zielorientiert ist, folgt leicht, dass R eine wohldefinierte τ_2-Region ist. □

Lemma 4.21. *Sei $\tau = \{\mathsf{res}, \mathsf{set}\} \cup \omega$ mit $\omega \subseteq \{\mathsf{inp}, \mathsf{out}, \mathsf{used}, \mathsf{free}\}$ und $A = (Z, E, \delta, \iota)$ ein stark zielorientiertes Transitionssystem. Es gibt einen Algorithmus, der mit Zeitaufwand $\mathcal{O}(|Z|^3 \cdot |E|^2)$ entscheidet, ob es einen Zeugen für die τ-ZTE beziehungsweise die τ-EZTE von A gibt.*

Beweis. Gemäß Lemma 4.18 hat A die τ-ZTE, das heißt, es gibt einen Zeugen für die τ-ZTE. Sei α ein EZTP von A. Gemäß Lemma 4.19 und Lemma 4.20 können wir die τ-Lösbarkeit von α wie folgt entscheiden: Erstens entscheiden wir, ob es ein $\tau' \subseteq \tau$ mit $\tau' \equiv \tau_0$ gibt und –falls ja– ob α τ_0-lösbar ist. A hat höchstens $|Z|^2 \cdot |E|$ Kanten. Folglich kann dies gemäß Lemma 4.20.1 mit Zeitaufwand $\mathcal{O}(|Z|^2 \cdot |E|)$ entschieden werden. Im Falle einer positiven Entscheidung ist α τ-lösbar.

Andernfalls prüfen wir zweitens, ob es $\tau' \subseteq \tau$ mit $\tau' \equiv \tau_2$ gibt und –falls ja– ob α τ_2-lösbar ist. Gemäß Lemma 4.20.2 ist dies mit Zeitaufwand $\mathcal{O}(|Z|^2 \cdot |E|)$ möglich. Im Falle einer positiven Entscheidung ist α τ-lösbar, andernfalls ist α nicht τ-lösbar. Da A höchstens $|Z| \cdot |E|$ EZTP hat, folgt die Behauptung. □

4.2.2 Netztypen mit set aber ohne res und free

Solange nicht ausdrücklich anders vereinbart, sei in diesem Abschnitt $\tau = \{\mathsf{set}\} \cup \omega$ mit $\omega \subseteq \{\mathsf{inp}, \mathsf{out}, \mathsf{used}\}$. Die polynomielle Entscheidbarkeit von τ-Realisierung basiert im Wesentlichen auf folgender Eigenschaft: Wenn A ein stark zielorientiertes Transitionssystem ist und α ein EZTP beziehungsweise ein ZTP von A, dann genügt es, die Existenz von höchstens drei sehr spezifisch definierten τ-Regionen zu prüfen, um die τ-Lösbarkeit von α zu entscheiden. Wenn keine dieser τ-Regionen existiert, dann ist α nicht τ-lösbar, andernfalls löst eine dieser Regionen α. Der Rest dieses Abschnitts liefert die entsprechenden Details.

Das folgende Lemma diskutiert die τ-Lösbarkeit von EZTP.

Lemma 4.22. *Sei $A = (Z, E, \delta, \iota)$ ein stark zielorientiertes Transitionssystem und $\alpha = (a, q)$ ein EZTP von A und p das eindeutige Ziel von a. Es ist α genau dann τ-lösbar, wenn eine der folgenden Definitionen von $R = (sup, sig)$ eine τ-Region von A ergibt:*

1. *$sup = Z \setminus \{p, q\}$ und für alle $e \in E$: wenn $\xrightarrow{e} p$ oder $\xrightarrow{e} q$, dann $sig(e) = \mathsf{inp}$, andernfalls $sig(e) = \mathsf{set}$.*

2. *$sup = Z \setminus \{q\}$ und für alle $e \in E$: wenn $e = a$, dann $sig(e) = \mathsf{used}$ und wenn $\xrightarrow{e} q$, dann $sig(e) = \mathsf{inp}$, andernfalls $sig(e) = \mathsf{set}$.*

3. *$sup = Z \setminus \{z \in Z \mid z \xrightarrow{a}\}$ und für alle $e \in E$: wenn $e = a$, dann $sig(e) = \mathsf{out}$ und wenn $\xrightarrow{e} z$ für ein $z \in \{z \in Z \mid z \xrightarrow{a}\}$, dann $sig(e) = \mathsf{inp}$, andernfalls $sig(e) = \mathsf{set}$.*

Beweis. „⇐“: Eine τ-Region die gemäß (1), (2) oder (3) definiert ist, löst α, denn sie erfüllt $\neg sup(q) \xrightarrow{sig(a)}$.

„⇒“: Sei $R' = (sup', sig')$ eine τ-Region, die α löst. Dies impliziert $sig'(a) \in \{\mathsf{inp}, \mathsf{used}\}$ und $sup'(q) = 0$ oder $sig'(a) = \mathsf{out}$ und $sup'(q) = 1$. Wir zeigen nun, dass jeder Fall eine der unter (1), (2) beziehungsweise (3) definierten Regionen impliziert.

Wenn $sig'(a) = \mathsf{inp}$, das heißt $sup'(p) = sup'(q) = 0$, dann ist $R = (sup, sig)$, wie in (1) definiert, eine τ-Region von A: Sei $z \xrightarrow{e} z' \in A$ beliebig aber fest. Wir zeigen $sup(z) \xrightarrow{sig(e)} sup(z') \in \tau$. Wenn $sig(e) = \mathsf{set}$, dann gilt $z' \notin \{p, q\}$ für das eindeutige Ziel z' von e, also $sup(z') = 1$ und $sup(z) \xrightarrow{sig(e)} sup(z') \in \tau$. Wenn $sig(e) = \mathsf{inp}$, dann $z' \in \{p, q\}$ und $sup(z') = 0$. Die Annahme $sup(z) = 0$, das heißt $z \in \{p, q\}$, widerspricht der Voraussetzung, dass R' eine τ-Region ist, die $sup'(p) = sup'(q) = 0$ erfüllt, denn es gibt kein $i \in \tau$ mit $0 \xrightarrow{i} 0$. Somit gilt $sup(z) = 1$ und $sup(z) \xrightarrow{sig(e)} sup(z') \in \tau$. Insbesondere ist R wohldefiniert.

Analog argumentiert man, dass der Fall $sig'(a) = \mathsf{used}$ und $sup'(q) = 0$ die Region unter (2) impliziert und der Fall $sig'(a) = \mathsf{out}$ und $sup'(q) = 1$ impliziert die Region, die in (3) definiert ist. □

Das folgende Lemma behandelt die τ-Lösbarkeit von ZTP.

Lemma 4.23. *Sei $A = (Z, E, \delta, \iota)$ ein stark zielorientiertes Transitionssystem und $\alpha = (p, q)$ ein ZTP von A. Es ist α genau dann τ-lösbar, wenn eine der folgenden Definitionen von $R = (sup, sig)$ eine τ-Region von A ergibt:*

1. *$sup = Z \setminus \{p\}$ und für alle $e \in E$: wenn $\xrightarrow{e} p$, dann $sig(e) = \mathsf{inp}$, sonst $sig(e) = \mathsf{set}$.*

2. *$sup = Z \setminus \{q\}$ und für alle $e \in E$: wenn $\xrightarrow{e} q$, dann $sig(e) = \mathsf{inp}$, sonst $sig(e) = \mathsf{set}$.*

Beweis. Die Richtung „⇐“ ist klar.

„⇒“: Sei $R' = (sup', sig')$ eine τ-Region, die α löst, das heißt $sup'(p) = 0$ und $sup'(q) = 1$ oder $sup'(p) = 1$ und $sup'(q) = 0$.

Wenn $sup'(p) = 0$ und $sup'(q) = 1$, dann muss $sig'(e) = \mathsf{inp}$ für alle $e \in E$ mit $\xrightarrow{e} p$ gelten. Insbesondere gilt $\mathsf{inp} \in \tau$ und $p \xrightarrow{e} p \notin A$ für alle $e \in E$. Es folgt, dass die Definition aus (1) eine τ-Region R von A ergibt.

Wenn $sup'(p) = 1$ und $sup'(q) = 0$, dann folgt analog, dass die Definition aus (2) eine τ-Region R von A ergibt. □

Ein Transitionssystem $A = (Z, E, \delta, \iota)$ hat höchstens $|Z|^2 \cdot |E|$ Kanten und höchstens $|Z| \cdot |E|$ EZTP beziehungsweise $|Z|^2$ ZTP. Das folgende Lemma ist deshalb eine einfache Folgerung aus Lemma 4.22 und Lemma 4.23:

Lemma 4.24 (Ohne Beweis)**.** *Sei $\tau = \{\mathsf{set}\} \cup \omega$ mit $\omega \subseteq \{\mathsf{inp}, \mathsf{out}, \mathsf{used}\}$ und $A = (Z, E, \delta, \iota)$ ein zielorientiertes Transitionssystem. Es gibt einen Algorithmus, der mit Zeitaufwand $\mathcal{O}(|Z|^4 \cdot |E|^2)$ entscheidet, ob A die τ-ZTE beziehungsweise die τ-EZTE hat.*

4.2.3 Netztypen mit set und free aber ohne res

Solange nicht ausdrücklich anders vereinbart, sei in diesem Abschnitt $\tau = \{\mathsf{set}, \mathsf{free}\} \cup \omega$ mit $\omega \subseteq \{\mathsf{inp}, \mathsf{out}, \mathsf{used}\}$. Der folgende Algorithmus 2 bildet den Kern für ein polynomielles Entscheidungsverfahren für τ-REALISIERBARKEIT.

Algorithmus 2 : Für ein Transitionssystem $A = (Z, E, \delta, \iota)$ und eine Menge $Q \subseteq Z$ reduziert der Algorithmus Q zu einem Support einer τ-Region von A.

Input : ein Netztyp $\tau = \{\mathsf{set}, \mathsf{free}\} \cup \omega$ mit $\omega \subseteq \{\mathsf{inp}, \mathsf{out}, \mathsf{used}, \mathsf{free}\}$, ein stark zielorientiertes Transitionssystem $A = (Z, E, \delta, \iota)$ und eine Menge $Q \subseteq Z$

Output : ein Support $sup \subseteq Q$ einer τ-Region von A

1 **while** *(1) Es gibt* $p \xrightarrow{e} p' \in A$ *mit* $Q(p) = 1$ *und* $Q(p') = 0$ *und* $\mathsf{inp} \notin \tau$ *oder (2) es gibt* $p \xrightarrow{e} p' \in A$ *und* $z \xrightarrow{e} z' \in A$ *mit* $Q(p) = 1$ *und* $Q(p') = Q(z) = Q(z') = 0$ **do**

2 $\quad Q = Q \setminus \{p\}$;

3 **return** $sup = Q$;

Die Ausgabe von Algorithmus 2 ist der Support einer Q-maximalen Regionen:

Definition 4.5 (Q-maximal)**.** Sei τ ein boolescher Netztyp und $A = (Z, E, \delta, \iota)$ ein Transitionssystem und $Q \subseteq Z$. Eine τ-Region $R = (sup, sig)$ heißt *Q-maximal*, wenn $sup \subseteq Q$ und für alle τ-Regionen $R' = (sup', sig')$ mit $sup' \subseteq Q$ gilt $|sup'| \leq |sup|$.

Wir erinnern daran, dass für eine Teilmenge $Q \subseteq Z(A)$ von Zuständen eines Transitionssystem A das Signaturpotential eines Ereignisses $e \in E(A)$ gemäß Definition 4.3 bestimmt ist durch $S_e^Q = \{(Q(z), Q(z')) \mid z \xrightarrow{e} z' \in A\}$.

Lemma 4.25 (Algorithmus 2)**.** *Sei* $\tau = \{\mathsf{set}, \mathsf{free}\} \cup \omega$ *mit* $\omega \subseteq \{\mathsf{inp}, \mathsf{out}, \mathsf{used}, \mathsf{free}\}$ *und* $A = (Z, E, \delta, \iota)$ *ein stark zielorientiertes Transitionssystem und* $Q \subseteq Z$*. Wenn Algorithmus 2 für die Eingabe* τ, A *und* Q *gestartet wird, dann gelten die folgenden Aussagen:*

1. *Algorithmus 2 terminiert und seine Laufzeit beträgt* $\mathcal{O}(|Z|^5 \cdot |E|^2)$*.*
2. *Wenn Algorithmus 2 sup ausgibt, dann ist sup der Support der* τ*-Region* $R = (sup, sig)$*, deren Signatur für alle* $e \in E$ *wie folgt definiert ist:*

$$sig(e) = \begin{cases} \mathsf{used}, & \text{falls } \mathsf{used} \in \tau \text{ und } S_e^{sup} = \{(1,1)\} \\ \mathsf{set}, & \text{falls } \mathsf{used} \notin \tau \text{ und } S_e^{sup} = \{(1,1)\} \\ \mathsf{out}, & \text{falls } \mathsf{out} \in \tau \text{ und } S_e^{sup} = \{(0,1)\} \\ \mathsf{set}, & \text{falls } \mathsf{out} \notin \tau \text{ und } S_e^{sup} = \{(0,1)\} \\ \mathsf{set}, & \text{falls } S_e^{sup} = \{(0,1),(1,1)\} \\ \mathsf{inp}, & \text{falls } S_e^{sup} = \{(1,0)\} \\ \mathsf{free}, & \text{falls } S_e^{sup} = \{(0,0)\} \end{cases}$$

.

3. *Wenn $R = (sup, sig)$ eine Q-maximale τ-Region von A ist, dann ist sup die Ausgabe von Algorithmus 2.*

Beweis. (1): Das Transitionssystem hat höchstens $|Z|^2 \cdot |E|$ Kanten. Um zu prüfen, ob eine der Bedingungen der While-Schleife erfüllt ist, müssen höchstens alle Kanten paarweise miteinander verglichen werden. Da Q in jedem Durchlauf der While-Schleife um einen Knoten verkleinert wird und $|Q| \leq |Z|$, kann es höchstens $|Z|$ Iterationen geben. Daraus folgt die Behauptung.

(2): Es gilt zu zeigen, dass R wohldefiniert ist. Dazu argumentieren wir erstens, dass jedem Ereignis $e \in E$ eindeutig ein Bild $sig(e) \in \tau$ zugewiesen wird. Zweitens ist zu zeigen, dass $sup(z) \xrightarrow{sig(e)} sup(z') \in \tau$ für alle $z \xrightarrow{e} z' \in A$. Weil A stark zielorientiert ist, gilt zunächst $sup(p') = sup(z')$ paarweise für alle Kanten $p \xrightarrow{e} p', z \xrightarrow{e} z' \in A$. Daraus folgt sofort, dass wenn $sup(z') = 1$ für eine Kante $z \xrightarrow{e} z'$, dann $S_e^{sup} \in \{\{(1,1)\}, \{(0,1)\}, \{(0,1),(1,1)\}\}$; alle möglichen Fälle für S_e^{sup} werden offensichtlich in der Definition sig berücksichtigt. Ebenso folgt, dass $S_e^{sup} \in \{\{(1,0)\}, \{(0,0)\}, \{(0,0),(1,0)\}\}$, wenn $sup(z') = 0$ für eine Kante $z \xrightarrow{e} z'$. Zwei Fälle benötigen hier besondere Aufmerksamkeit, nämlich der Fall $S_e^{sup} = \{(0,0),(1,0)\}$, denn er wird in der Definition nicht berücksichtigt, und der Fall $S_e^{sup} = \{(1,0)\}$, denn er setzt $\mathsf{inp} \in \tau$ voraus. Wir argumentieren für den ersten Fall, dass $S_e^{sup} = \{(0,0),(1,0)\}$ nicht vorkommen kann und für den zweiten Fall, dass die Annahme $\mathsf{inp} \in \tau$ korrekt ist: Nehmen wir $S_e^{sup} = \{(0,0),(1,0)\}$ an. Dann gibt es Kanten $p \xrightarrow{e} p', z \xrightarrow{e} z' \in A$ mit $sup(p) = 1$ und $sup(p') = sup(z) = sup(z') = 1$, das heißt, sup erfüllt Bedingung (2) der While-Schleife von Algorithmus 2. Dies ist ein Widerspruch zu Voraussetzung, dass sup (nach Terminierung) von Algorithmus 2 ausgegeben wurde. Somit gilt $S_e^{sup} \neq \{(0,0),(1,0)\}$ für alle $e \in E$. Des Weiteren impliziert $S_e^{sup} = \{(1,0)\}$, dass inp in τ, denn andernfalls gibt es $z \xrightarrow{e} z' \in A$ mit $sup(z) = 1$ und $sup(z') = 0$, obwohl $\mathsf{inp} \notin \tau$. Dies widerspricht der Voraussetzung, dass sup die Bedingung (1) der While-Schleife nicht erfüllt. Schließlich gilt noch zu bemerken, dass sich die in der Definition von sig unterschiedenen Fälle paarweise gegenseitig ausschließen, sodass jedem Ereignis $e \in E$ genau ein Bild $sig(e) \in \tau \supseteq \{\mathsf{set}, \mathsf{free}\}$ zugeordnet wird. Anhand von S_e^{sup} ist darüber hinaus leicht zu sehen, dass $sup(z) \xrightarrow{sig(e)} sup(z') \in \tau$ für alle $z \xrightarrow{e} z' \in A$. Es folgt, dass $R = (sup, sig)$ eine wohldefinierte τ-Region ist.

(3): Sei $R = (sup, sig)$ eine entsprechende τ-Region von A. Bezeichne Q_i das Resultat der initialen Menge Q nach (der Verarbeitung durch) i While-Schleifen-Iterationen. Wir zeigen zunächst $sup \subseteq Q_i$ mittels vollständiger Induktion über i. Nach Voraussetzung gilt $sup \subseteq Q_0 = Q$. Gelte $sup \subseteq Q_i$ und sei $Q_{i+1} = Q_i \setminus \{p\}$. Nehmen wir an $sup \not\subseteq Q_{i+1}$, das heißt $p \in sup$. Wenn p wegen Bedingung (1) der While-Schleife aus Q_i entfernt wurde, dann gibt es eine Kante $p \xrightarrow{e} p'$ in A mit $sup(p) = 1$ und $sup(p') = 0$, denn $sup \subseteq Q_i$ nach Induktionsvoraussetzung und $p' \notin Q_i$. Wegen $\mathsf{inp} \notin \tau$ widerspricht dies der Voraussetzung, dass R eine τ-Region ist. Wenn p wegen Bedingung (2) der While-Schleife entfernt wurde,

dann gibt es Kanten $p \xrightarrow{e} p'$ und $z \xrightarrow{e} z'$ in A, sodass (mit gleicher Begründung) $sup(p) = 1$ und $sup(p') = sup(z) = sup(z') = 0$. Dies ist ebenso ein Widerspruch zur Voraussetzung, dass R eine τ-Region ist. Mithin war unsere Annahme falsch und es gilt $sup \subseteq Q_{i+1}$.

Nach (1) terminiert der Algorithmus nach höchstens $n \leq |Z|$ Iterationen und nach (2) ist die Ausgabe Q_n der Support einer τ-Region, die offensichtlich $Q_n \subseteq Q$ erfüllt. Nun gilt einerseits $|Q_n| \leq |sup|$, denn R ist Q-maximal. Andererseits gilt, wie eben bewiesen, auch $sup \subseteq Q_n$. Es folgt $sup = Q_n$. □

Gemäß Lemma 4.25.2 definierte jede Ausgabe sup von Algorithmus 2 in eindeutiger Weise eine τ-Region $R = (sup, sig)$. Wir werden deshalb im Folgenden der Bequemlichkeit halber auch von der durch Algorithmus 2 (gemäß Lemma 4.25.2) ausgegebenen τ-Region $R = (sup, sig)$ sprechen.

Lemma 4.26. *Sei $\tau = \{\mathsf{set}, \mathsf{free}\} \cup \omega$ mit $\omega \subseteq \{\mathsf{inp}, \mathsf{out}, \mathsf{used}, \mathsf{free}\}$ und $A = (Z, E, \delta, \iota)$ ein stark zielorientiertes Transitionssystem und $\alpha = (a, q)$ ein EZTP von A und p das eindeutige Ziel von a. Es ist α genau dann τ-lösbar, wenn eine der folgenden Aussagen erfüllt ist:*

1. *Die τ-Region $R = (sup, sig)$, die Algorithmus 2 auf Eingabe $Q_{\mathsf{inp}} = Z \setminus \{p, q\}$ ausgibt, erfüllt $sig(a) = \mathsf{inp}$.*
2. *Die τ-Region $R = (sup, sig)$, die Algorithmus 2 auf Eingabe $Q_{\mathsf{out}} = Z \setminus \{z \in Z \mid z \xrightarrow{a}\}$ ausgibt, erfüllt $sig(a) = \mathsf{out}$ und $sup(q) = 1$.*
3. *Die τ-Region $R = (sup, sig)$, die Algorithmus 2 auf Eingabe $Q_{\mathsf{used}} = Z \setminus \{q\}$ ausgibt, erfüllt $sig(a) = \mathsf{used}$.*
4. *Die τ-Region $R = (sup, sig)$, die Algorithmus 2 auf Eingabe $Q_{\mathsf{free}} = Z \setminus \{z, z' \in Z \mid z \xrightarrow{a} z'\}$ ausgibt, erfüllt $sig(q) = 1$.*

Beweis. Die Richtung „⇐“ ist klar, denn für alle Fälle gilt $\neg sup(q) \xrightarrow{sig(a)}$, wobei man beachte, dass (4) und die Definition von sig gemäß Lemma 4.25.2 $sig(a) = \mathsf{free}$ impliziert.

„⇒“: Man überlegt sich, dass für alle $i \in \{\mathsf{inp}, \mathsf{out}, \mathsf{used}, \mathsf{free}\}$ und alle τ-Regionen $R' = (sup', sig')$ mit $sig'(a) = i$ und $\neg sup'(q) \xrightarrow{sig'(a)}$ die Eigenschaft $sub' \subseteq Q_i$ gilt. Wenn überhaupt eine entsprechende Region R' existiert, dann gibt es deshalb auch eine, die Q_i-maximal ist. Die Behauptung folgt dann mit Lemma 4.25, sowie der Beobachtung, dass die Definition von sig gemäß Lemma 4.25.2 einem Ereignis e einen Wert aus $\{\mathsf{inp}, \mathsf{out}, \mathsf{used}, \mathsf{free}\}$ zuordnet, wann immer es sup erlaubt. □

Lemma 4.27. *Sei $\tau = \{\mathsf{set}, \mathsf{free}\} \cup \omega$ mit $\omega \subseteq \{\mathsf{inp}, \mathsf{out}, \mathsf{used}, \mathsf{free}\}$ und $A = (Z, E, \delta, \iota)$ ein stark zielorientiertes Transitionssystem und $\alpha = (p, q)$ ein ZTP von A. Es ist α genau dann τ-lösbar, wenn die τ-Region, die Algorithmus 2 auf Eingabe $Q_0 = Z \setminus \{p\}$ oder auf Eingabe $Q_1 = Z \setminus \{q\}$ ausgibt, das ZTP α löst.*

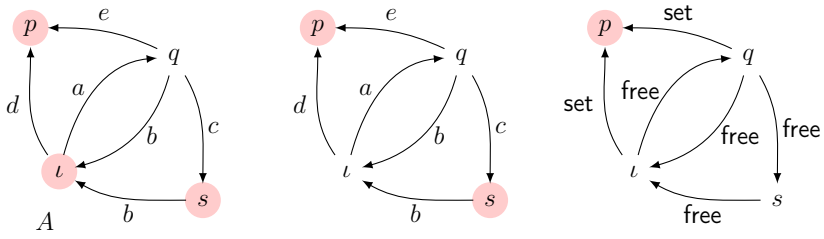


Abbildung 4.7: Nach Beispiel 4.2: Die $\{\mathsf{set}, \mathsf{free}\}$-Region (rechts) von A (links), die auf Eingabe $\{\iota, p, s\}$ durch Algorithmus 2 ausgegeben wird.

Beweis. Die Richtung „⇐“ ist trivial.

„⇒“: Jede τ-Region $R' = (sup', sig')$ mit $sup'(p) = 0$ und $sup'(q) = 1$ erfüllt $sup' \subseteq Q_0$ und jede mit $sup'(p) = 1$ und $sup'(q) = 0$ erfüllt $sup' \subseteq Q_1$. Wenn überhaupt eine entsprechende Region R' existiert, dann gibt es auch eine, die entsprechend der Eingabemengen maximal ist. Die Behauptung folgt mit Lemma 4.25. □

Lemma 4.28. *Sei $\tau = \{\mathsf{set}, \mathsf{free}\} \cup \omega$ mit $\omega \subseteq \{\mathsf{inp}, \mathsf{out}, \mathsf{used}, \mathsf{free}\}$ und $A = (Z, E, \delta, \iota)$ ein stark zielorientiertes Transitionssystem. Es gibt einen Algorithmus, der mit Zeitaufwand $\mathcal{O}(|Z|^6 \cdot |E|^3)$ beziehungsweise mit Zeitaufwand $\mathcal{O}(|Z|^8 \cdot |E|^2)$ entscheidet, ob A die τ-EZTE beziehungsweise die τ-ZTE hat.*

Beweis. Für ein beliebiges aber festes EZTP α von A benötigen wir gemäß Lemma 4.26 für jedes $i \in \{\mathsf{inp}, \mathsf{out}, \mathsf{used}, \mathsf{free}\} \cap \tau$ höchstens einen Lauf von Algorithmus 2 gestartet auf Q_i. Genau dann ist α τ-lösbar, wenn Algorithmus 2 für ein solches Q_i eine lösende τ-Region ausgibt. Da die Laufzeit den Aufwand überwiegt, R zu konstruieren und $\neg sup(q) \xrightarrow{sig(a)}$ zu testen, können wir mit Zeitaufwand $\mathcal{O}(|Z|^5 \cdot |E|^2)$ die τ-Lösbarkeit von α entscheiden. Da A höchstens $|Z| \cdot |E|$ EZTP hat, können wir in Zeit $\mathcal{O}(|Z|^6 \cdot |E|^3)$ entscheiden, ob A die τ-EZTE hat. Analog überlegt man sich, dass mit Hilfe von Algorithmus 2 in Zeit $\mathcal{O}(|Z|^8 \cdot |E|^2)$ entschieden werden kann, ob A die τ-ZTE hat, da A höchstens $|Z|^2$ ZTP hat. Die Behauptung folgt. □

Beispiel 4.2 (Anwendung von Algorithmus 2). Betrachten wir das Transitionssystem A mit Zustandsmenge $Z = \{\iota, p, q, s\}$ und Ereignismenge $E = \{a, b, c, d, e\}$, das in Abbildung 4.7 links dargestellt ist. Wir starten Algorithmus 2 für die Eingabe $\tau = \{\mathsf{set}, \mathsf{free}\}$ and $Q = \{\iota, p, s\}$, wobei die Zustände von Q in Abbildung 4.7 rot markiert sind. Die erste Prüfung der While-Bedingung(en) findet die Kante $\iota \xrightarrow{a} q$ mit $Q(\iota) = 1$ and $Q(q) = 0$. Weil $\mathsf{inp} \notin \tau$, wird ι aus Q entfernt. Die neue Menge Q ist in der Mitte von Abbildung 4.7 (farblich rot) skizziert. Für $Q = \{p, s\}$, werden erneut die While-Bedingung(en) geprüft und es finden sich die Kanten $s \xrightarrow{b} \iota$, $q \xrightarrow{b} \iota$ mit $Q(s) = 1$ und $S(q) = S(\iota) = 0$. Entsprechend wird s aus Q entfernt und wir erhalten $Q = \{p\}$. Ein erneutes Überprüfen von Q zeigt, dass keine der While-Bedingungen erfüllt ist. Somit ist $sup = Q = \{p\}$

ein τ-Support. Die gemäß Lemma 4.25 definierte τ-Region wird rechts in Abbildung 4.7 skizziert.

4.2.4 Netztypen ohne res und set

Solange nicht ausdrücklich anders vereinbart, sei im weiteren Verlauf dieses Abschnitts $\tau \subseteq \{\mathsf{inp}, \mathsf{out}, \mathsf{used}, \mathsf{free}\}$ beliebig aber fest.

Die folgende Definition 4.6 formalisiert den Begriff der Eindeutigkeit für Ereignisse und Transitionssysteme. Das sich anschließende Lemma 4.29 zeigt, dass τ-realisierbare Transitionssysteme stets eindeutig sind.

Definition 4.6 (Eindeutigkeit)**.** Sei $A = (Z, E, \delta, \iota)$ ein Transitionssystem. Ein Ereignis $e \in E$ ist *eindeutig* (in A), wenn es genau eine e-beschriftete Kante in A gibt. A heißt *eindeutig*, wenn jedes Ereignis von A eindeutig ist.

Lemma 4.29. *Sei $A = (Z, E, \delta, \iota)$ ein Transitionssystem. Wenn A die τ-ZTE hat, dann ist A eindeutig.*

Beweis. Habe A die τ-ZTE. Sei $e \in E$ und seien $p \xrightarrow{e} p' \in A$ und $z \xrightarrow{e} z' \in A$. Wegen Lemma 4.2 wissen wir bereits, dass A stark zielorientiert ist, das heißt $p' = z'$. Nehmen wir $p \neq z$ an. Weil A die τ-ZTE hat, gibt es eine τ-Region $R = (sup, sig)$, die das ZTP (p, z) löst. Somit gilt $sup(p) \neq sup(z)$. Dies ist eine Widerspruch zu $sup(p) \xrightarrow{sig(e)}$ und $sup(z) \xrightarrow{sig(e)}$ und $sig(e) \in \tau$, denn für alle $i \in \{\mathsf{inp}, \mathsf{out}, \mathsf{used}, \mathsf{free}\}$ und alle $b, b' \in \{0, 1\}$ gilt: wenn $b \xrightarrow{i}$ und $b' \xrightarrow{i}$, dann $b = b'$. Wir erhalten $p = z$ und damit die Behauptung. □

Ein τ-realisierbares Transitionssystem hat stets die τ-ZTE. Es kann mit Zeitaufwand $\mathcal{O}(|Z|^2 \cdot |E|)$ entschieden werden, ob ein Transitionssystem eindeutig ist. Deshalb werden wir unsere Untersuchungen im Rest dieses Abschnitts auf eindeutige Transitionssysteme beschränken, denn andernfalls darf die Eingabe gemäß Lemma 4.29 gerechtfertigt als nicht τ-realisierbar abgelehnt werden.

Lemma 4.30. *Wenn $\tau \subseteq \{\mathsf{inp}, \mathsf{out}\}$ dann hat ein τ-realisierbares Transitionssystem $A = (Z, E, \delta, \iota)$ höchstens zwei Zustände und deshalb höchstens vier τ-Regionen.*

Beweis. Sei $\tau \subseteq \{\mathsf{inp}, \mathsf{out}\}$ und A τ-realisierbar, dass heißt, A hat die τ-ZTE. Wenn $\iota \xrightarrow{a} z \in A$ und $z \xrightarrow{b} z' \in A$ oder $\iota \xrightarrow{b} z' \in A$, sodass ι, z und z' paarweise verschiedenen sind, dann ist, im Widerspruch zur τ-ZTE von A, das ZTP (ι, z') nicht lösbar. Somit hat A höchstens zwei Zustände. Wenn $\tau = \emptyset$, dann hat A keine τ-Region oder E ist leer. Wenn $\tau \neq \emptyset$, dann, falls $R = (sup, sig)$ eine τ-Region ist, gilt $sup(z) \in \{0, 1\}$ für alle $z \in Z$. Wegen $sig(e) \in \{\mathsf{inp}, \mathsf{out}\}$, ist $sig(e)$ für alle $e \in E$ durch sup festgelegt. Insgesamt kann es also höchstens 4 τ-Regionen geben. Dies beweist das Lemma. □

Algorithmus 3 : Für $\tau = \{\mathsf{free}\} \cup \omega$ mit $\omega \subseteq \{\mathsf{inp}, \mathsf{out}, \mathsf{used}\}$ und ein eindeutiges Transitionssystem $A = (Z, E, \delta, \iota)$ und eine Menge $Q \subseteq Z$ reduziert der Algorithmus Q zum Support einer τ-Region von A.

Input : ein Netztyp $\tau = \{\mathsf{free}\} \cup \omega$ mit $\omega \subseteq \{\mathsf{inp}, \mathsf{out}, \mathsf{used}\}$, ein eindeutiges Transitionssystem $A = (Z, E, \delta, \iota)$ und eine Menge $Q \subseteq Z$

Output : ein Support $sup \subseteq Q$ einer Region von A

1 **while** *(1) Es gibt* $p \xrightarrow{e} p' \in A$ *mit* $Q(p) = 1$ *und* $Q(z') = 0$ *und* $\mathsf{inp} \notin \tau$ *oder (2) es gibt* $p' \xrightarrow{e} p \in A$ *mit* $Q(p') = 0$ *und* $Q(p) = 1$ *und* $\mathsf{out} \notin \tau$ *oder (3) es gibt* $p \xrightarrow{e} p' \in A$ *mit* $Q(p) = Q(p') = 1$ *und* $\mathsf{used} \notin \tau$ **do**

2 $\quad Q = Q \setminus \{p\}$;

3 **return** $sup = Q$;

Algorithmus 3 ist die Grundlage eines polynomiellen Entscheidungsverfahrens für τ-REALISIERBARKEIT, falls $\tau = \{\mathsf{free}\} \cup \omega$ mit $\omega \subseteq \{\mathsf{inp}, \mathsf{out}, \mathsf{used}\}$.

Lemma 4.31 (Algorithmus 3)**.** *Sei* $\tau = \{\mathsf{free}\} \cup \omega$ *mit* $\omega \subseteq \{\mathsf{inp}, \mathsf{out}, \mathsf{used}\}$ *und* $A = (Z, E, \delta, \iota)$ *ein eindeutiges Transitionssystem und* $Q \subseteq Z$*. Wenn Algorithmus 3 für die Eingabe* τ, A *und* Q *gestartet wird, dann gelten die folgenden Aussagen:*

1. *Algorithmus 3 terminiert und seine Laufzeit beträgt* $\mathcal{O}(|Z|^3 \cdot |E|)$*.*
2. *Wenn Algorithmus 3 sup ausgibt, dann ist sup der Support der* τ*-Region* $R = (sup, sig)$*, deren Signatur für alle* $e \in E$ *wie folgt definiert ist:*

$$sig(e) = \begin{cases} \mathsf{inp}, & \text{falls } \mathsf{inp} \in \tau \text{ und } S_e^{sup} = \{(1,0)\} \\ \mathsf{out}, & \text{falls } \mathsf{out} \in \tau \text{ und } S_e^{sup} = \{(0,1)\} \\ \mathsf{used}, & \text{falls } \mathsf{used} \in \tau \text{ und } S_e^{sup} = \{(1,1)\} \\ \mathsf{free}, & \text{sonst.} \end{cases}$$

.

3. *Wenn* $R = (sup, sig)$ *eine* Q*-maximale* τ*-Region von* A *ist, dann ist sup die Ausgabe von Algorithmus 3.*

Beweis. (1): A hat höchstens $|Z|^2 \cdot |E|$ Kanten und es gibt höchstens $|Q| \leq |Z|$ Iterationen der While-Schleife, denn jede Iteration entfernt einen Zustand aus Q.

(2): Die aufgeführten Fälle schließen sich gegenseitig aus, sodass jedes $e \in E$ einen Wert $sig(e) \in \tau$ zugewiesen bekommt. Für jedes $e \in E$ gibt es nur eine Kante $z \xrightarrow{e} z' \in A$. Nehmen wir an, es gibt $z \xrightarrow{e} z' \in A$ mit $sup(z) \xrightarrow{sig(e)} sup(z') \notin \tau$. Dies impliziert $sig(e) = \mathsf{free}$ und $S_e = \{(1,0)\}$ und $\mathsf{inp} \notin \tau$ oder $S_e = \{(0,1)\}$ und $\mathsf{out} \notin \tau$ oder $S_e = \{(1,1)\}$ und $\mathsf{used} \notin \tau$. In allen Fällen erhalten wir einen Widerspruch dazu, dass

keine der Bedingungen (1), (2) oder (3) der While-Schleife durch *sup* erfüllt ist. Es folgt, dass $sup(z) \xrightarrow{sig(e)} sup(z') \in \tau$ und R wohldefiniert ist.

(3): Der Beweis wird analog zu Lemma 4.25 (3) geführt. □

Sei $A = (Z, E, \delta, \iota)$ ein eindeutiges Transitionssystem und $\alpha = (a, q)$ ein EZTP beziehungsweise $\beta = (p, q)$ ein ZTP von A und seien die Mengen $Q_{\mathsf{inp}}, Q_{\mathsf{out}}, Q_{\mathsf{used}}$ und Q_{free} sowie Q_0 und Q_1 wie in Lemma 4.26 beziehungsweise Lemma 4.27 definiert. Ähnlich zu (den Beweisen von) diesen Lemmata überlegt man sich, dass α genau dann τ-lösbar ist, wenn die durch Algorithmus 3 auf Eingabe $Q_{\mathsf{inp}}, Q_{\mathsf{out}}, Q_{\mathsf{used}}$ oder Q_{free} definierte τ-Region α löst und β genau dann τ-lösbar ist, wenn die durch Algorithmus 3 auf Eingabe Q_0 oder Q_1 definierte β löst. Entsprechend zu Lemma 4.28 erhalten wir, gemäß der geringeren Laufzeit von Algorithmus 3 im Vergleich zu Algorithmus 2, dass mit einem Zeitaufwand von $\mathcal{O}(|Z|^4 \cdot |E|^2)$ beziehungsweise $\mathcal{O}(|Z|^5 \cdot |E|)$ entschieden werden kann, ob A die τ-EZTE beziehungsweise die τ-ZTE hat. Darüber hinaus gibt es für alle $\tau' = \{\mathsf{used}\} \cup \omega'$ mit $\omega' \subseteq \{\mathsf{inp}, \mathsf{out}, \mathsf{free}\}$ ein $\omega \subseteq \{\mathsf{inp}, \mathsf{out}, \mathsf{used}\}$, sodass $\tau = \{\mathsf{free}\} \cup \omega \equiv \tau'$. Mit diesen Überlegungen und Satz 2.2 sowie Lemma 4.30 können wir das folgende Lemma als bewiesen betrachten:

Lemma 4.32. *Wenn $\tau \subseteq \{\mathsf{inp}, \mathsf{out}, \mathsf{used}, \mathsf{free}\}$, dann gibt es einen Algorithmus, der für ein eindeutiges Transitionssystem A mit Zeitaufwand $\mathcal{O}(|Z|^4 \cdot |E|^2)$ beziehungsweise $\mathcal{O}(|Z|^5 \cdot |E|)$ entscheidet, ob es die τ-EZTE beziehungsweise die τ-ZTE hat.*

Insgesamt folgt nun aus Lemma 4.21, Lemma 4.24, Lemma 4.28 und Lemma 4.32 der Beweis von Satz 4.11.

4.3 Einbettung für Boolesche Netztypen

In diesem Abschnitt charakterisieren wir die Komplexität des Problems τ-Einbettung vollständig für alle 256 booleschen Netztypen. Die Tabelle aus Abbildung 4.8 zeigt eine Übersicht über die jeweiligen Resultate. Netztypen sind wieder nach Kongruenz beziehungsweise gleicher Beweisstrategie zusammengefasst.

Für die Netztypen τ mit nop, das heißt die Typen B1 - B7 aus Abbildung 4.8, erhalten wir die folgenden zusammenfassenden Charakterisierungen:

1. Wenn $\tau \cap \{\mathsf{res}, \mathsf{set}\} = \emptyset$, dann ist τ-Einbettung NP-vollständig, falls $\tau \cap \{\mathsf{inp}, \mathsf{out}\} \neq \emptyset$ und $\mathsf{swap} \notin \tau$, sonst ist das Problem in Polynomialzeit entscheidbar.

2. Wenn $\mathsf{res} \in \tau$, dann ist τ-Einbettung NP-vollständig, falls $\tau \cap \mathfrak{produce} \neq \emptyset$, sonst ist das Problem in Polynomialzeit entscheidbar.

3. Wenn $\mathsf{set} \in \tau$, dann ist τ-Einbettung NP-vollständig, falls $\tau \cap \mathfrak{consume} \neq \emptyset$, sonst ist das Problem in Polynomialzeit entscheidbar.

B	Netztypen τ	Aufwand	Anzahl	Satz
1	$\{\mathsf{nop}, \mathsf{inp}\} \cup \omega$, $\{\mathsf{nop}, \mathsf{out}\} \cup \omega$ mit $\omega \subseteq \{\mathsf{used}, \mathsf{free}\}$	NPV	8	4.12
2	$\{\mathsf{nop}, \mathsf{inp}, \mathsf{out}\} \cup \omega$ mit $\omega \subseteq \{\mathsf{used}, \mathsf{free}\}$	NPV	4	4.13
3	$\{\mathsf{nop}, \mathsf{swap}\} \cup \omega$ mit $\omega \subseteq \{\mathsf{inp}, \mathsf{out}, \mathsf{used}, \mathsf{free}\}$, $\{\mathsf{nop}\} \cup \omega$ mit $\omega \subseteq \{\mathsf{used}, \mathsf{free}\}$	P	20	4.14
4	$\{\mathsf{nop}, \mathsf{res}, \mathsf{set}\} \cup \omega$ mit $\omega \subseteq \{\mathsf{inp}, \mathsf{out}, \mathsf{swap}, \mathsf{used}, \mathsf{free}\}$	NPV	32	4.15
5	$\{\mathsf{nop}, \mathsf{res}, \mathsf{swap}\} \cup \omega$ mit $\omega \subseteq \{\mathsf{inp}, \mathsf{out}, \mathsf{used}, \mathsf{free}\}$, $\{\mathsf{nop}, \mathsf{set}, \mathsf{swap}\} \cup \omega$ mit $\omega \subseteq \{\mathsf{inp}, \mathsf{out}, \mathsf{used}, \mathsf{free}\}$	NPV	32	4.15
6	$\{\mathsf{nop}, \mathsf{out}, \mathsf{res}\} \cup \omega$ mit $\omega \subseteq \{\mathsf{inp}, \mathsf{used}, \mathsf{free}\}$, $\{\mathsf{nop}, \mathsf{inp}, \mathsf{set}\} \cup \omega$ mit $\omega \subseteq \{\mathsf{out}, \mathsf{used}, \mathsf{free}\}$	NPV	16	4.15
7	$\{\mathsf{nop}, \mathsf{res}\} \cup \omega$ mit $\omega \subseteq \{\mathsf{inp}, \mathsf{used}, \mathsf{free}\}$, $\{\mathsf{nop}, \mathsf{set}\} \cup \omega$ mit $\omega \subseteq \{\mathsf{out}, \mathsf{used}, \mathsf{free}\}$	P	16	4.16
8	$\{\mathsf{swap}\} \cup \omega$ mit $\omega \subseteq \{\mathsf{inp}, \mathsf{out}, \mathsf{res}, \mathsf{set}, \mathsf{used}, \mathsf{free}\}$ und $\omega \cap \{\mathsf{res}, \mathsf{set}, \mathsf{used}, \mathsf{free}\} \neq \emptyset$	NPV	60	4.17
9	$\{\mathsf{swap}\} \cup \omega$ with $\omega \subseteq \{\mathsf{inp}, \mathsf{out}\}$	P	4	4.18
10	$\omega \subseteq \{\mathsf{inp}, \mathsf{out}, \mathsf{res}, \mathsf{set}, \mathsf{used}, \mathsf{free}\}$	P	64	4.19

Abbildung 4.8: Eine Übersicht über die Komplexität von τ-EINBETTUNG für boolesche Netztypen sowie jeweils eine Referenz auf den Satz mit dem Ergebnis.

Die Netztypen τ ohne nop, das heißt die Typen B8 - B10 aus Abbildung 4.8, lassen sich wie folgt charakterisieren:

1. Wenn $\mathsf{swap} \notin \tau$ oder wenn $\mathsf{swap} \in \tau$ und $\tau \cap \mathfrak{save} = \emptyset$, dann ist τ-EINBETTUNG in Polynomialzeit entscheidbar.

2. Wenn $\mathsf{swap} \in \tau$ und $\tau \cap \mathfrak{save} \neq \emptyset$, dann ist τ-EINBETTUNG NP-vollständig.

Wenn τ ein boolescher Netztyp mit nop ist, dann genügt es für die Lösbarkeit von ZTP nach *normalisierten* Regionen zu suchen, deren Signaturen –salopp gesagt– so viele Ereignisse des Eingabe-Transitionsssystems auf nop abbilden, wie möglich:

Definition 4.7 (Normalisierte Region)**.** Sei $A = (Z, E, \delta, \iota)$ ein Transitionssystem und τ ein boolescher Netztyp mit nop. Eine τ-Region $R = (sup, sig)$ von A heißt *normalisiert*, wenn für alle $e \in E$ die folgende Eigenschaft erfüllt ist: wenn $sig(e) \neq \mathsf{nop}$, dann gibt es $z \xrightarrow{e} z' \in A$ mit $sup(z) \neq sup(z')$.

Gemäß des folgenden Lemmas sind normalisierte Regionen für die Untersuchung der ZTE eines Transitionssystems tatsächlich ausreichend:

Lemma 4.33. *Sei $A = (Z, E, \delta, \iota)$ ein Transitionssystem und τ ein boolescher Netztyp mit nop. Es gibt genau dann einen Zeugen für die τ-ZTE von A, wenn es einen Zeugen normalisierter Regionen für die τ-ZTE von A gibt.*

Beweis. Die eine Richtung ist trivial. Sei $\mathcal{R}$ ein Zeuge für die τ-ZTE von A. Die folgende Prozedur liefert offensichtlich einen Zeugen $\mathcal{R}'$ normalisierter τ-Regionen von A: $\mathcal{R}' = \emptyset$. Für jede τ-Region $R = (sup, sig) \in \mathcal{R}$ definiere folgendermaßen eine normalisierte τ-Region $R' = (sup, sig')$ mit dem gleichen Support wie R und füge sie zu $\mathcal{R}'$ hinzu: Für alle $e \in E$, falls $sup(z) = sup(z')$ für alle $z \xrightarrow{e} z' \in A$, dann $sig'(e) = \mathsf{nop}$, sonst $sig'(e) = sig(e)$. □

Der Rest dieses Abschnitts ist den Beweisen der bereits angedeuteten Komplexitätsresultate zur Entscheidbarkeit von τ-EINBETTUNG gewidmet.

4.3.1 Die Netztypen B1

Satz 4.12. *Wenn $\tau = \{\mathsf{nop}, \mathsf{inp}\} \cup \omega$ oder $\tau = \{\mathsf{nop}, \mathsf{out}\} \cup \omega$ mit $\omega \subseteq \{\mathsf{used}, \mathsf{free}\}$, dann ist τ-EINBETTUNG NP-vollständig.*

Im Folgenden zeigen wir die Behauptung von Satz 4.12 für den Fall $\tau = \{\mathsf{nop}, \mathsf{inp}\}$. Der Beweis wird gemäß der in Kapitel 3 vorgestellten Reduktionsstrategie geführt. Des Weiteren lässt sich aus der NP-Vollständigkeit von $\{\mathsf{nop}, \mathsf{inp}\}$-EINBETTUNG dann die Aussage des Satzes auch für die übrigen Netztypen wie folgt ableiten: Wegen Lemma 4.33, folgt auch für $\tau = \{\mathsf{nop}, \mathsf{inp}\} \cup \omega$ für alle $\omega \subseteq \{\mathsf{used}, \mathsf{free}\}$. Darüber hinaus gibt es für alle $\tau' = \{\mathsf{nop}, \mathsf{out}\} \cup \omega'$ mit $\omega' \subseteq \{\mathsf{used}, \mathsf{free}\}$ ein $\omega \subseteq \{\mathsf{used}, \mathsf{free}\}$, sodass $\tau' \equiv \{\mathsf{nop}, \mathsf{inp}\} \cup \omega$. Somit folgt nach Satz 2.2 die Behauptung auch für τ'. Dies beweist den Satz.

Sei also im Folgenden, solange nicht ausdrücklich anders vereinbart, $\tau = \{\mathsf{nop}, \mathsf{inp}\}$. Das Transitionssystem A_τ hat den folgenden Pfad P mit dem ZTP $\alpha = (t_{m,0}, t_{m+1,0})$:

$$P = t_{0,0} \xrightarrow{w_0} \cdots \xrightarrow{w_{i-1}} t_{i,0} \xrightarrow{w_i} \cdots \xrightarrow{w_{m-2}} t_{m-1,0} \xrightarrow{w_{m-1}} t_{m,0} \xrightarrow{k} t_{m+1,0} \xrightarrow{v} \top$$

Des Weiteren hat A_τ für jedes $i \in \{0, \ldots, m-1\}$ den folgenden Pfad T_i auf dem Variablen der Klausel $M_i = \{X_{i_0}, X_{i_1}, X_{i_2}\}$ als Ereignisse stattfinden und der den (Teil-) Pfad $P_i = t_{i,0} \xrightarrow{X_{i_0}} \ldots \xrightarrow{X_{i_2}} t_{i,3}$ hat:

$$T_i = t_{i,0} \xrightarrow{X_{i_0}} t_{i,1} \xrightarrow{X_{i_1}} t_{i,2} \xrightarrow{X_{i_2}} t_{i,3} \xrightarrow{u_i} \top$$

Schließlich hat das Transitionssystem A_φ für jedes $i \in \{0, \ldots, m-1\}$ den folgenden Pfad G_i:

$$G_i = t_{0,0} \xrightarrow{y_i} g_{i,0} \xrightarrow{u_i} g_{i,1} \xrightarrow{k} g_{i,2}$$

Das resultierende Transitionssystem mit dem Initialzustand $t_{0,0}$ ist A_τ. Insbesondere sind die eingeführten Komponenten von A_τ bereits über den Initialzustand $t_{0,0}$ miteinander verbunden. Darüber hinaus haben die Pfade P und $T_0, \ldots, T_{m-1}$ den gemeinsamen Endzustand $\top$. Abbildung 4.9 zeigt die Konstruktion von A_τ für die Instanz $(\mathfrak{U}, M)$ aus Beispiel 3.1.

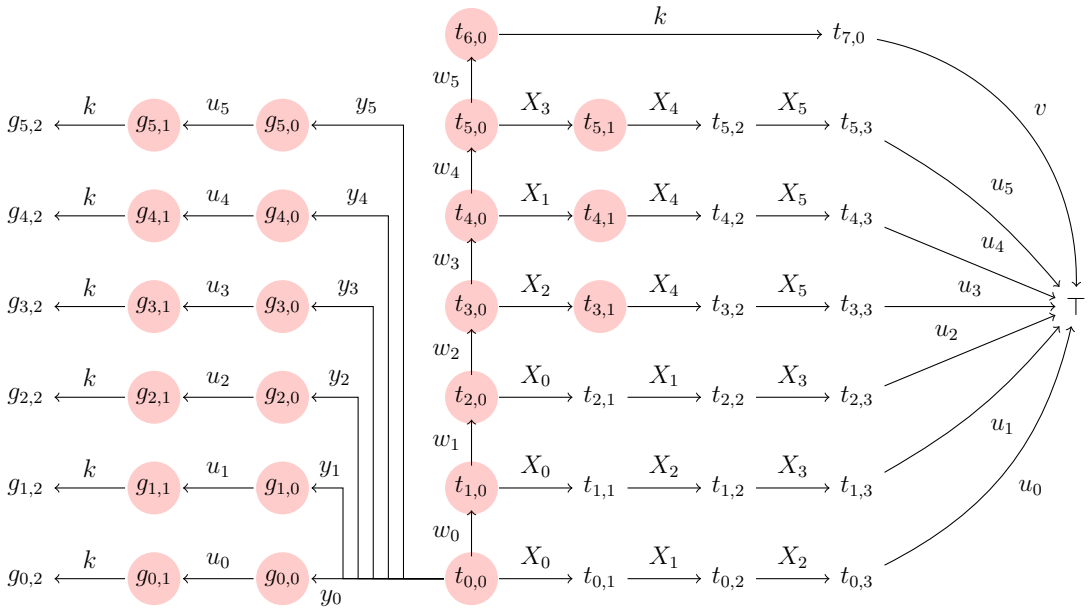


Abbildung 4.9: Das Transitionssystem A_τ für den Beweis von Satz 4.12, welches aus der Eingabe von Beispiel 3.1 resultiert. Der farbige Bereich skizziert den Support sup_3 der τ-Region R_3 aus Lemma 4.35, die $\alpha = (t_{6,0}, t_{7,0})$ löst.

Lemma 4.34. *Wenn A_τ die τ-ZTE hat, dann gibt es ein Eins-in-drei-Modell für $(\mathfrak{U}, M)$.*

Beweis. Weil A_τ die τ-ZTE hat, gibt es eine τ-Region, die das ZTP $\alpha = (t_{m,0}, t_{m+1,0})$ löst. Sei $R = (sup, sig)$ eine solche Region und sei $S = \{X \in \mathfrak{U} \mid sig(X) = \mathsf{inp}\}$. Wir zeigen, dass die Menge S ein Eins-in-drei-Modell von $(\mathfrak{U}, M)$ definiert. Weil α von R gelöst wird, gilt $sup(t_{m,0}) \neq sup(t_{m+1}, 0)$. Wegen der Kante $t_{m,0} \xrightarrow{k} t_{m+1,0}$ und $\tau = \{\mathsf{nop}, \mathsf{inp}\}$ folgt sofort, dass $sig(k) = \mathsf{inp}$ und $sup(t_{m,0}) = 1$ und $sup(t_{m+1}, 0) = 0$. Analog erhalten wir mit $g_{i,1} \xrightarrow{k} g_{i,2}$, dass $sup(g_{i,1}) = 1$ und $sig(g_{i,2}) = 0$ für alle $i \in \{0, \ldots, m-1\}$. Weil in τ kein Zustandsübergang von 0 nach 1 möglich ist, impliziert dies $sup(t_{0,0}) = \cdots = sup(t_{m,0}) = 1$ und $sup(\top) = 0$ sowie $sup(g_{0,0}) = \cdots = sup(g_{m-1,0}) = 1$. Wir erhalten mit $g_{i,0} \xrightarrow{u_i} g_{i,1}$ und $sup(g_{i,0}) = sup(g_{i,1}) = 1$ außerdem $sig(u_i) = \mathsf{nop}$ für alle $i \in \{0, \ldots, m-1\}$. Für alle $i \in \{0, \ldots, m-1\}$ gilt nun: Weil $sup(t_{i,0}) = 1$ und $sup(\top) = 0$ und $sig(u_i) = \mathsf{nop}$, ist das Bild P_i^R des Pfades P_i (siehe Konstruktion), ein Pfad von 1 nach 0 in τ. Insbesondere gibt es genau ein $X \in \{X_{i_0}, X_{i_1}, X_{i_2}\}$ mit $sig(X) = \mathsf{inp}$. Somit ist S ein Eins-in-drei-Modell von $(\mathfrak{U}, M)$. □

Soweit ist bewiesen, dass $(\mathfrak{U}, M)$ ein Eins-in-drei-Modell hat, wenn A_τ die τ-ZTE hat. Für die andere Richtung beweisen wir folgendes Lemma:

Lemma 4.35. *Wenn $(\mathfrak{U}, M)$ ein Eins-in-drei-Modell hat, dann hat A_τ die τ-ZTE.*

Beweis. Sei S ein Eins-in-drei-Modell von $(\mathfrak{U}, M)$. Wir werden eine Liste von τ-Regionen präsentieren, welche insgesamt die Lösbarkeit aller ZTP von A_τ implizieren. Zugunsten der besseren Lesbarkeit vereinfachen wir die Darstellung von Regionen wie folgt: Zunächst wird eine Region $R = (sup, sig)$ implizit mittels $sup(t_{0,0})$ und sig definiert. Dabei gilt stets $sup(t_{0,0}) = 1$, sodass wir diesen Fakt nicht jedesmal erneut erwähnen. Weiterhin, weil $\tau = \{\mathsf{nop}, \mathsf{inp}\}$, werden alle Ereignisse von A_τ, die durch sig nicht auf inp abgebildet werden, auf nop abgebildet. Das heißt, wir präsentieren nur die Menge $\mathcal{E}^R_{\mathsf{inp}} = \{e \in E(A_\tau) \mid sig(e) = \mathsf{inp}\}$, sodass $\mathcal{E}^R_{\mathsf{nop}} = E(A_\tau) \setminus \mathcal{E}^R_{\mathsf{inp}}$ folgt.

Die folgende Region $R_1 = (sup_1, sig_1)$ löst (z, z') für alle $z \in \bigcup_{i=0}^{m-1} Z(T_i) \cup \{t_{m,0}, t_{m+1,0}\}$ und alle $z' \in \bigcup_{i=0}^{m-1} Z(G_i)$ mit $z' \neq t_{0,0}$: $\mathcal{E}^{R_1}_{\mathsf{inp}} = \{y_0, \ldots, y_{m-1}\}$.

Es bleibt, die ZTP (z, z') mit $z, z' \in \bigcup_{i=0}^{m-1} Z(G_i)$ beziehungsweise $z, z' \in \bigcup_{i=0}^{m-1} Z(T_i) \cup \{t_{m,0}, t_{m+1,0}\}$ zu lösen.

Die folgende Region $R_2 = (sup_2, sig_2)$ löst $(g_{j,0}, g_{j,1})$ und $(g_{j,0}, g_{j,2})$ für alle $j \in \{0, \ldots, m-1\}$ und $(\top, z)$ für alle $z \in \bigcup_{i=0}^{m-1} Z(P_i) \cup \{t_{m,0}, t_{m+1,0}\}$: $\mathcal{E}^{R_2}_{\mathsf{inp}} = \{v\} \cup \{u_0, \ldots, u_{m-1}\}$.

Die folgende Region $R_3 = (sup_3, sig_3)$ nutzt das Eins-in-drei-Modell S von $(\mathfrak{U}, M)$ und löst α sowie $(g_{j,1}, g_{j,2})$ für alle $j \in \{0, \ldots, m-1\}$: $\mathcal{E}^{R_3}_{\mathsf{inp}} = \{k\} \cup S$. Der rote Bereich in Abbildung 4.9 skizziert R_3 für das Transitionssystem A_τ, das aus Beispiel 3.1 resultiert.

Mit Blick auf die Zustände aus $\bigcup_{j=0}^{m-1} Z(G_j)$ bleibt nur noch zu zeigen, dass für alle $i \neq j \in \{0, \ldots, m-1\}$ und alle $z \in Z(G_i) \setminus \{t_{0,0}\}$ und alle $z' \in Z(G_j) \setminus \{t_{0,0}\}$ das ZTP (z, z') lösbar ist.

Sei $i \in \{0, \ldots, m-2\}$ beliebig aber fest. Die folgende Region $R_4 = (sup_4, sig_4)$ löst (z, z') für alle $z \in \{g_{i,0}, g_{i,1}, g_{i,2}\}$ und alle $z' \in \bigcup_{j=i+1}^{m-1} Z(G_j)$ mit $z' \neq t_{0,0}$: $\mathcal{E}^{R_4}_{\mathsf{inp}} = \{y_{i+1}, \ldots, y_{m-1}\}$.

Aus der Beliebigkeit von i und den Regionen R_1, R_2, R_3 folgt nun, dass jeder Zustand $z \in \bigcup_{j=0}^{m-1} Z(G_j)$ τ-lösbar ist. Des Weiteren folgt aus R_1 und R_2, dass $\top$ τ-lösbar ist. Es verbleibt deshalb, die Zustände aus $\bigcup_{i=0}^{m-1} Z(P_i) \cup \{t_{m,0}, t_{m+1,0}\}$ zu untersuchen, wobei das ZTP $(t_{m,0}, t_{m+1,0})$ schon gelöst ist.

Sei $i \in \{0, \ldots, m-1\}$ beliebig aber fest. Die folgende Region $R_5 = (sup_5, sig_5)$ löst (z, z') für alle $z \in \{t_{i,0}, \ldots, t_{i,3}\}$ und alle $z' \in (\bigcup_{j=i+1}^{m-1} Z(T_j)) \cup \{t_{m,0}, t_{m+1,0}\}$: $\mathcal{E}^{R_5}_{\mathsf{inp}} = \{w_i\} \cup \{u_0, \ldots, u_i\} \cup \{y_{i+1}, \ldots, y_{m-1}\}$.

Aus der Beliebigkeit von i folgt, dass $t_{m,0}$ und $t_{m+1,0}$ τ-lösbar sind. Des Weiteren folgt aus der Beliebigkeit von i, dass nur noch die Lösbarkeit von (z, z') für alle $z \neq z' \in \{t_{i,0}, \ldots, t_{i,3}\}$ und alle $i \in \{0, \ldots, m-1\}$ zu zeigen ist. Sei $i \in \{0, \ldots, m-1\}$ beliebig aber fest. Die folgende Region R_6 löst $(t_{i,0}, z)$ für alle $z \in \{t_{i,1}, t_{i,2}, t_{i,3}\}$. Die verbleibenden ZTP von P_i sind analog lösbar, sodass wir diese nicht explizit diskutieren. Seien die Indizes $j \neq \ell \in \{0, \ldots, m-1\} \setminus \{i\}$ derart, dass sie genau die beiden anderen Klauseln von M auswählen, in denen X_{i_0} vorkommt: $X_{i_0} \in M_j \cap M_\ell$. Wir erhalten $R_6 = (sup_6, sig_6)$ wie folgt: $\mathcal{E}^{R_5}_{\mathsf{inp}} = \{X_{i_0}\} \cup \{v\} \cup \{u_0, \ldots, u_{m-1}\} \setminus \{u_i, u_j, u_\ell\}$.

Auf ähnliche Weise erhält man Regionen, welche X_{i_1} beziehungsweise X_{i_2} eine inp-Signatur zuweisen. Diese lösen die verbleibenden ZTP aus $Z(T_i) \setminus \{\top\}$. Weil i beliebig war, vervollständigt dies den Beweis des Lemmas. □

Offenbar ist die präsentierte Reduktion polynomiell. Aus Lemma 4.1, Lemma 4.34 und Lemma 4.35 sowie der NP-Vollständigkeit von CM 1 IN 3 3SAT folgt somit Satz 4.12.

4.3.2 Die Netztypen B2 - B3

Der folgende Satz adressiert die Netztypen B2 aus Abbildung 4.8:

Satz 4.13. *Wenn $\tau = \{\mathsf{nop}, \mathsf{inp}, \mathsf{out}\} \cup \omega$ mit $\omega \subseteq \{\mathsf{used}, \mathsf{free}\}$, dann ist τ-*EINBETTUNG *auch dann dann NP-vollständig, wenn das Eingabe-Transitionssystem linear ist.*

Für $b = 1$ ist der Netztyp $\{\mathsf{nop}, \mathsf{inp}, \mathsf{out}\}$ der elementaren Netzsysteme kongruent zu dem Netztyp der puren b-beschränkten P/T-Netze, der in Kapitel 5 eingeführt wird. Dort zeigen wir in Satz 5.1, dass τ-EINBETTUNG für diesen Netztyp auch dann NP-vollständig ist, wenn die Eingabe ein lineares Transitionssystem ist, dass heißt (gemäß Definition 2.25) ein kreisfreier initialisierter Pfad. Wegen Satz 2.2 folgt daraus die NP-Vollständigkeit von $\{\mathsf{nop}, \mathsf{inp}, \mathsf{out}\}$-EINBETTUNG für lineare Eingaben. Wir werden in Kapitel 5 an entsprechender Stelle nochmal auf diesen Zusammenhang hinweisen. Weil wir für den Nachweis der τ-ZTE eines Transitionssystems mit normalisierten Regionen auskommen, wenn τ ein boolescher Netztyp mit nop ist, folgt die Aussage von Satz 4.13 für alle $\tau = \{\mathsf{nop}, \mathsf{inp}, \mathsf{out}\} \cup \omega$ mit $\omega \subseteq \{\mathsf{used}, \mathsf{free}\}$ schließlich aus der NP-Vollständigkeit von τ'-EINBETTUNG mit $\tau' = \{\mathsf{nop}, \mathsf{inp}, \mathsf{out}\}$.

Der nächste Satz wendet sich an die Netztypen B3 aus Abbildung 4.8:

Satz 4.14. *Für einen booleschen Netztyp τ ist τ-*EINBETTUNG *in Polynomialzeit entscheidbar, wenn eine der beiden folgenden Bedingungen erfüllt ist:*

1. $\tau = \{\mathsf{nop}, \mathsf{swap}\} \cup \omega$ *mit* $\omega \subseteq \{\mathsf{inp}, \mathsf{out}, \mathsf{used}, \mathsf{free}\}$
2. $\tau = \{\mathsf{nop}\} \cup \omega$ *mit* $\omega \subseteq \{\mathsf{used}, \mathsf{free}\}$.

Beweis. (1): Für diese Netztypen folgt die Aussage gemäß Lemma 4.7 aus Abschnitt 4.1.2, nach dem in Zeit $\mathcal{O}(|Z|^6 \cdot |E|^4)$ entschieden werden kann, ob A die τ-ZTE hat.

(2): Für diese Netztypen ist die Aussage trivial: Weil eine τ-Region $R = (sup, sig)$ von A stets $sup(z) = sup(z')$ für jede Kante $z \xrightarrow{e} z'$ impliziert, hat A genau dann die τ-ZTE, wenn A höchstens einen Zustand hat. □

4.3.3 Die Netztypen B4 - B6

Satz 4.15. *Sei τ ein boolescher Netztyp. Es ist τ-*EINBETTUNG *NP-vollständig, wenn eine der folgenden Bedingungen erfüllt ist:*

1. $\tau \supseteq \{\mathsf{nop}, \mathsf{res}, \mathsf{set}\}$ *oder* $\tau = \{\mathsf{nop}, \mathsf{out}, \mathsf{res}\} \cup \omega$ *mit* $\omega \subseteq \{\mathsf{inp}, \mathsf{used}, \mathsf{free}\}$ *oder* $\tau = \{\mathsf{nop}, \mathsf{res}, \mathsf{swap}\} \cup \omega$ *mit* $\omega \subseteq \{\mathsf{inp}, \mathsf{out}, \mathsf{used}, \mathsf{free}\}$

2. $\tau = \{\mathsf{nop}, \mathsf{inp}, \mathsf{set}\} \cup \omega$ *mit* $\omega \subseteq \{\mathsf{out}, \mathsf{used}, \mathsf{free}\}$ *oder* $\tau = \{\mathsf{nop}, \mathsf{set}, \mathsf{swap}\} \cup \omega$ *mit* $\omega \subseteq \{\mathsf{inp}, \mathsf{out}, \mathsf{used}, \mathsf{free}\}$

Der Rest dieses Abschnitts behandelt den Beweis von Satz 4.15. Zunächst stellen wir fest, dass es offenbar für jeden Netztyp τ' aus Satz 4.15.2 einen Netztyp τ aus Satz 4.15.1 gibt, sodass $\tau \equiv \tau'$. Wir werden wegen dieser Kongruenzen deshalb nur Satz 4.15.1 explizit beweisen. Die Aussage von Satz 4.15.2 folgt dann aus Satz 2.2.

Für den Beweis von Satz 4.15.1 gehen wir gemäß der folgenden Strategie vor: Wenn $\tau = \{\mathsf{nop}, \mathsf{out}, \mathsf{res}\} \cup \omega$ mit $\omega \subseteq \{\mathsf{inp}, \mathsf{used}, \mathsf{free}\}$ oder wenn $\{\mathsf{nop}, \mathsf{res}, \mathsf{set}\} \subseteq \tau$, dann reduzieren wir $\{\mathsf{nop}, \mathsf{inp}, \mathsf{out}\}$-Einbettung auf τ-Einbettung. Mit Satz 4.13 folgt daraus die NP-Vollständigkeit von τ-Einbettung. Andernfalls, wenn $\tau = \{\mathsf{nop}, \mathsf{res}, \mathsf{swap}\} \cup \omega$ mit $\omega \subseteq \{\mathsf{inp}, \mathsf{used}, \mathsf{free}\}$, dann reduzieren wir $\{\mathsf{nop}, \mathsf{inp}\}$-Einbettung auf τ-Einbettung. Die NP-Vollständigkeit von τ-Einbettung folgt dann aus Satz 4.12.

Für die angekündigten Reduktionen werden die folgenden Begrifflichkeiten bemüht, um passende *Erweiterungen* eines Transitionssystems A zu definieren. Abbildung 4.10 veranschaulicht die entsprechenden Definitionen.

Definition 4.8 (Assoziierte Ereignisse)**.** Sei $A = (Z, E, \delta, \iota)$ ein Transitionssystem. Die Menge $\overline{E} = \{\overline{e} \mid e \in E\}$ der mit E assoziierten Ereignisse enthalte für jedes Ereignis $e \in E$ ein unverwechselbares neues Ereignis $\overline{e}$, welches das *mit e assoziierte Ereignis* genannt wird.

Die *Rückwärts-Erweiterung* B eines Transitionssystems A erweitert A um eine $\overline{e}$-beschriftete „Rückwärtskante“ für jede e-beschriftete Kante von A:

Definition 4.9 (Rückwärts-Erweiterung)**.** Sei $A = (Z, E, \delta, \iota)$ ein Transitionssystem und $\overline{E}$ die Menge der mit E assoziierten Ereignisse. Die *Rückwärts-Erweiterung* ist das Transitionssystem $B = (Z, E \cup \overline{E}, \delta', \iota)$, wobei δ' wie folgt definiert ist: Für alle $z, z' \in Z$ und alle $e \in E$, wenn $\delta(z, e) = z'$, dann $\delta'(z, e) = z'$ und $\delta'(z', \overline{e}) = z$, wobei $\overline{e}$ das mit e assoziierte Ereignis ist.

Die *einseitige Schleifen-Erweiterung* eines Transitionssystems ergänzt dieses ebenso um die Rückwärtskanten. Darüber hinaus enthält es für jede ursprüngliche e-beschriftete Kante eine e-beschriftete Schleife um das Ziel der Kante:

Definition 4.10 (Einseitige Schleifen-Erweiterung)**.** Sei $A = (Z, E, \delta, \iota)$ ein Transitionssystem und $\overline{E}$ die Menge der mit E assoziierten Ereignisse. Die *einseitige Schleifen-Erweiterung* ist das Transitionssystem $C = (Z, E \cup \overline{E}, \delta', \iota)$, wobei δ' wie folgt definiert ist: Für alle $z, z' \in Z$ und alle $e \in E$, wenn $\delta(z, e) = z'$, dann $\delta'(z, e) = z'$ und $\delta'(z', e) = z'$ sowie $\delta'(z', \overline{e}) = z$, wobei $\overline{e}$ das mit e assoziierte Ereignis ist.

Die (vollständige) *Schleifen-Erweiterung* eines Transitionssystems fügt, neben all den bisher ergänzten Kanten, schließlich auch für die mit assoziierten Ereignissen beschrifteten Kanten eine Schleife um deren Ziele hinzu:

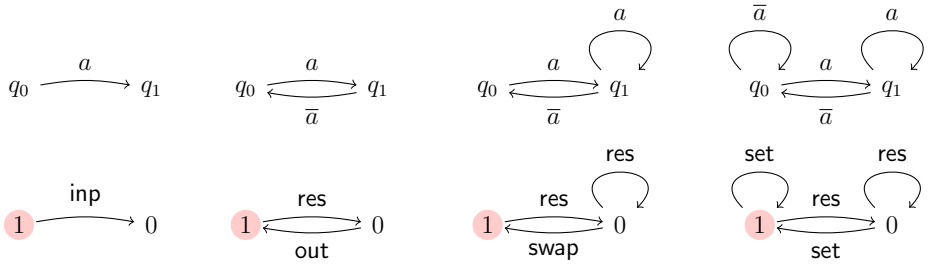


Abbildung 4.10: Oben ist von links nach rechts abgebildet: ein Transitionssystem A, das nur aus einer e-beschrifteten Kante besteht, die Rückwärts-Erweiterung B von A, die einseitige Schleifen-Erweiterung C von A und schließlich die Schleifen-Erweiterung D von A. Unten sind von links nach rechts die Bilder von A und den Erweiterungen B, C und D unter Regionen abgebildet, die den Argumenten beziehungsweise Netztypen von Lemma 4.36 entsprechen und jeweils das ZTP (q_0, q_1) lösen: eine $\{\mathsf{nop}, \mathsf{inp}\}$- ($\{\mathsf{nop}, \mathsf{inp}, \mathsf{out}\}$-) Region von A, eine $\{\mathsf{nop}, \mathsf{res}, \mathsf{out}\}$-Region von B, eine $\{\mathsf{nop}, \mathsf{res}, \mathsf{swap}\}$-Region von C, eine $\{\mathsf{nop}, \mathsf{res}, \mathsf{set}\}$-Region von D.

Definition 4.11 (Schleifen-Erweiterung)**.** Sei $A = (Z, E, \delta, \iota)$ ein Transitionssystem und $\overline{E}$ die Menge der mit E assoziierten Ereignisse. Die *Schleifen-Erweiterung* ist das Transitionssystem $D = (Z, E \cup \overline{E}, \delta', \iota)$, wobei δ' wie folgt definiert ist: Für alle $z, z' \in Z$ und alle $e \in E$, wenn $\delta(z, e) = z'$, dann $\delta'(z, e) = z'$ und $\delta'(z', e) = z'$ sowie $\delta'(z', \overline{e}) = z$ als auch $\delta'(z, \overline{e}) = z$, wobei $\overline{e}$ das mit e assoziierte Ereignis ist.

Lemma 4.36. *Seien $\tau_0 = \{\mathsf{nop}, \mathsf{inp}, \mathsf{out}\}$ und $\tau_1 = \{\mathsf{nop}, \mathsf{inp}\}$. Sei $A = (Z, E, \delta, \iota)$ ein schleifenfreies Transitionssystem und seien B, C und D seine Rückwärts-Erweiterung, seine einseitige Schleifen-Erweiterung beziehungsweise seine Schleifen-Erweiterung und sei $sup : Z \to \{0, 1\}$.*

1. *Es sei $\tau = \{\mathsf{nop}, \mathsf{out}, \mathsf{res}\} \cup \omega$ mit $\omega \subseteq \{\mathsf{inp}, \mathsf{used}, \mathsf{free}\}$. Die Abbildung sup erlaubt genau dann eine τ_0-Region $R = (sup, sig)$ von A, wenn sie eine normalisierte τ-Region $R' = (sup, sig')$ von B erlaubt.*

2. *Es gelte $\tau \supseteq \{\mathsf{nop}, \mathsf{res}, \mathsf{set}\}$. Die Abbildung sup erlaubt genau dann eine τ_0-Region $R = (sup, sig)$ von A, wenn sie eine normalisierte τ-Region $R' = (sup, sig')$ von D erlaubt.*

3. *Es sei $\tau = \{\mathsf{nop}, \mathsf{res}, \mathsf{swap}\} \cup \omega$ mit $\omega \subseteq \{\mathsf{inp}, \mathsf{used}, \mathsf{free}\}$. Die Abbildung sup erlaubt genau dann eine τ_1-Region $R = (sup, sig)$ von A, wenn sie eine normalisierte τ-Region $R' = (sup, sig')$ von C erlaubt.*

Beweis. Abbildung 4.10 veranschaulicht die im Folgenden definierten τ-Regionen.

(1) „⇒“: Sei $R = (sup, sig)$ eine τ_0-Region von A, das heißt, für alle Kanten $z \xrightarrow{e} z'$ von A gilt: Wenn $sig(e) = \mathsf{inp}$, dann $sup(z) = 1$ und $sup(z') = 0$, wenn $sig(e) = \mathsf{out}$, dann $sup(z) = 0$ und $sup(z') = 1$ und wenn $sig(e) = \mathsf{nop}$, dann $sup(z) = sup(z')$. Daraus folgt leicht, dass die folgende Definition von sig' eine normalisierte τ-Region $R' = (sup, sig')$ von B liefert. Für jedes Ereignis $e \in E$ und sein assoziiertes Ereignis $\overline{e}$: wenn $sig(e) = \mathsf{nop}$, dann $sig'(e) = sig'(\overline{e}) = \mathsf{nop}$, wenn $sig(e) = \mathsf{inp}$, dann $sig'(e) = \mathsf{res}$ und $sig'(\overline{e}) = \mathsf{out}$ und wenn $sig(e) = \mathsf{out}$, dann $sig'(e) = \mathsf{out}$ und $sig'(\overline{e}) = \mathsf{res}$.

„⇐“: Sei $R' = (sup, sig')$ eine normalisierte τ-Region von B und seien $z \xrightarrow{e} z'$ und $t \xrightarrow{e} t'$ mit $e \in E$ zwei beliebige aber feste Kanten von B. Man bemerke, dass diese Kanten ebenso in A sind. Zunächst wird gezeigt, dass sich die Support-Werte der Ursprünge und Ziele von e jeweils einheitlich verhalten, das heißt, wenn $sup(z) \neq sup(z')$, dann $sup(z) = sup(t)$ und $sup(z') = sup(t')$: Weil B die Rückwärts-Erweiterung von A ist, enthält B die Kanten $z' \xrightarrow{\overline{e}} z$ und $t' \xrightarrow{\overline{e}} t$. Wenn $sup(z) = 1$ und $sup(z') = 0$, dann $sig'(\overline{e}) = \mathsf{out}$ und dies impliziert $sup(t) = 1$ und $sup(t') = 0$. Wenn $sup(z) = 0$ und $sup(z') = 1$, dann $sig(e) = \mathsf{out}$ und daraus folgt $sup(t) = 0$ und $sup(t') = 1$. In jedem Fall gilt also $sup(z) = sup(t)$ und $sup(z') = sup(t')$. Außerdem impliziert dies, dass aus $sup(z) = sup(z')$ stets $sup(t) = sup(t')$ folgt. Für alle $e \in E$ definieren wir $sig(e)$ daher wir folgt und erhalten eine τ_0-Region $R = (sup, sig)$ von A: Wenn es eine Kante $z \xrightarrow{e} z'$ in B mit $sup(z) \neq sup(z')$ in B gibt, dann sei $sig(e) = \mathsf{inp}$, falls $sup(z) = 1$ und $sup(z) = 0$, sonst $sig(e) = \mathsf{out}$ und falls es solche Kante nicht gibt, sei $sig(e) = \mathsf{nop}$.

(2) „⇒“: Es sind die Kanten $0 \xrightarrow{\mathsf{res}} 0$ und $1 \xrightarrow{\mathsf{set}} 1$ in τ. Daher erhalten wir R' analog zu der Definition für (1, „⇒“), wobei der Signaturwert out stets durch set ersetzt wird.

„⇐“: Sei $R' = (sup, sig')$ eine normalisierte τ-Region. Analog zu (1, „⇐“) zeigt man für alle $e \in E$, dass sich Ursprünge und Ziele von e-beschrifteten Kanten homogen verhalten, falls $sig'(e) \neq \mathsf{nop}$. Ebenso analog erhält man dann die behauptete τ_0-Region R.

(3) „⇒“: Sei $R = (sup, sig)$ eine τ_1-Region von A, dass heißt, $sig(e) \in \{\mathsf{nop}, \mathsf{inp}\}$ für alle $e \in E$. Da $0 \xrightarrow{\mathsf{res}} 0$ und $0 \xrightarrow{\mathsf{swap}} 1$ in τ sind, ergibt die folgende Definition von sig' offensichtlich eine normalisierte τ-Region $R' = (sup, sig')$ von A. Für alle $e \in E$ und das mit e assoziierte Ereignis $\overline{e}$: wenn $sig(e) = \mathsf{inp}$, dann $sig'(e) = \mathsf{res}$ und $sig'(\overline{e}) = \mathsf{swap}$, andernfalls $sig'(e) = sig'(\overline{e}) = \mathsf{nop}$.

„⇐“: Sei $R' = (sup, sig')$ eine normalisierte τ-Region. Wir überlegen uns zunächst das Folgende: Für alle $e \in E$ gilt, wenn $z \xrightarrow{e} z'$ in C, dann folgt aus $sup(z) \neq sup(z')$, dass $sup(z) = sup(t) = 1$ und $sup(z') = sup(t') = 0$ für alle e-beschrifteten Kanten $t \xrightarrow{e} t'$ in C. Angenommen, es gibt für $e \in E$ eine Kante $z \xrightarrow{e} z'$ mit $sup(z) = 0$ und $sup(z') = 1$. Weil $sig'(e) \in \tau$, impliziert dies $sig'(e) = \mathsf{swap}$. Da C die einseitige Schleifen-Erweiterung von A ist, enthält C auch die Kante $z' \xrightarrow{e} z'$. Mit $sup(z') = 1$ erhalten wir den Widerspruch $1 \xrightarrow{\mathsf{swap}} 1$. Es folgt die Behauptung und wir erhalten die τ_1-Region $R = (sup, sig)$ von A, indem wir sig wie folgt definieren: Für alle $e \in E$, falls $z \xrightarrow{e} z'$ in C mit $sup(z) \neq sup(z')$, dann $sig(e) = \mathsf{inp}$, andernfalls $sig(e) = \mathsf{nop}$. Dies komplettiert

den Beweis des Lemmas. □

Das folgende Korollar ist eine einfache Folgerung aus Lemma 4.36:

Korollar 4.1 (Ohne Beweis)**.** *Sei* $\tau_0 = \{\mathsf{nop}, \mathsf{inp}, \mathsf{out}\}$ *und* $\tau_1 = \{\mathsf{nop}, \mathsf{inp}\}$ *und sei* A *ein schleifenfreies Transitionssystem und seien* B*,* C *und* D *die Rückwärts-Erweiterung, die einseitige Schleifen-Erweiterung und die Schleifen-Erweiterung von* A*.*

1. *Wenn* $\tau = \{\mathsf{nop}, \mathsf{out}, \mathsf{res}\} \cup \omega$ *mit* $\omega \subseteq \{\mathsf{inp}, \mathsf{used}, \mathsf{free}\}$*, dann hat* A *genau dann die* τ_0*-ZTE, wenn* B *die* τ*-ZTE hat.*
2. *Wenn* $\tau \supseteq \{\mathsf{nop}, \mathsf{res}, \mathsf{set}\}$*, dann hat* A *genau dann die* τ_0*-ZTE, wenn* D *die* τ*-ZTE hat.*
3. *Wenn* $\tau = \{\mathsf{nop}, \mathsf{res}, \mathsf{swap}\} \cup \omega$ *mit* $\omega \subseteq \{\mathsf{inp}, \mathsf{used}, \mathsf{free}\}$*, dann hat* A *genau dann die* τ_1*-ZTE, wenn* C *die* τ*-ZTE hat.*

Das Transitionssystem A_τ der Reduktion für den Beweis von Satz 4.12 ist schleifenfrei und die Aussage von Satz 4.13 adressiert sogar lineare Transitionssysteme, die insbesondere schleifenfrei sind. Es ist klar, dass die Rückwärts-Erweiterung beziehungsweise (einseitige) Schleifen-Erweiterung von A_τ jeweils in Polynomialzeit konstruierbar ist. Das bedeutet: wenn $\tau = \{\mathsf{nop}, \mathsf{out}, \mathsf{res}\} \cup \omega$ mit $\omega \subseteq \{\mathsf{inp}, \mathsf{used}, \mathsf{free}\}$ oder wenn $\{\mathsf{nop}, \mathsf{res}, \mathsf{set}\} \subseteq \tau$, dann können wir $\{\mathsf{nop}, \mathsf{inp}, \mathsf{out}\}$-EINBETTUNG polynomiell auf τ-EINBETTUNG reduzieren und wenn $\tau = \{\mathsf{nop}, \mathsf{res}, \mathsf{swap}\} \cup \omega$ mit $\omega \subseteq \{\mathsf{inp}, \mathsf{used}, \mathsf{free}\}$, dann können wir $\{\mathsf{nop}, \mathsf{inp}\}$-EINBETTUNG polynomiell auf τ-EINBETTUNG reduzieren. Mit Korollar 4.1 und Satz 4.12 sowie Satz 4.13 komplettiert dies den Beweis von Satz 4.15.

4.3.4 Die Netztypen B7

Der folgende Satz umfasst die booleschen Netztypen B7 aus Abbildung 4.8:

Satz 4.16. *Wenn* $\tau = \{\mathsf{nop}, \mathsf{res}\} \cup \omega$ *mit* $\omega \subseteq \{\mathsf{inp}, \mathsf{used}, \mathsf{free}\}$ *oder* $\tau = \{\mathsf{nop}, \mathsf{set}\} \cup \omega$ *mit* $\omega \subseteq \{\mathsf{out}, \mathsf{used}, \mathsf{free}\}$*, dann ist* τ*-*EINBETTUNG *in Polynomialzeit entscheidbar.*

Beweis. Für jeden Netztyp $\tau' = \{\mathsf{nop}, \mathsf{set}\} \cup \omega'$ mit $\omega' \subseteq \{\mathsf{out}, \mathsf{used}, \mathsf{free}\}$ gibt es offenbar einen Netztyp $\tau = \{\mathsf{nop}, \mathsf{res}\} \cup \omega$ mit $\omega \subseteq \{\mathsf{inp}, \mathsf{used}, \mathsf{free}\}$, sodass $\tau \equiv \tau'$. Darüber hinaus haben wir bereits für Lemma 4.16 aus Abschnitt 4.1.6 argumentiert, dass sich in Zeit $\mathcal{O}(|Z|^7 \cdot |E|^2)$ entscheiden lässt, ob ein Transitionssystem $A = (Z, E, \delta, \iota)$ die τ-ZTE hat, wenn $\tau = \{\mathsf{nop}, \mathsf{res}\} \cup \omega$ mit $\omega \subseteq \{\mathsf{inp}, \mathsf{used}, \mathsf{free}\}$. Insgesamt erhalten wir die Behauptung von Satz 4.16 somit als einfache Folgerung aus Lemma 4.16 und Satz 2.2. □

4.3.5 Die Netztypen B8

Der folgende Satz zeigt, dass die Suche nach einer τ-Einbettung im Allgemeinen auch dann nicht effizient lösbar ist, wenn dass gesuchte Netz keine Unabhängigkeit zwischen Plätzen und Transitionen ermöglicht:

Satz 4.17. *Wenn $\tau = \{\mathsf{swap}\} \cup \omega$ ein Netztyp mit $\omega \subseteq \{\mathsf{inp}, \mathsf{out}, \mathsf{res}, \mathsf{set}, \mathsf{used}, \mathsf{free}\}$ und $\omega \cap \{\mathsf{res}, \mathsf{set}, \mathsf{used}, \mathsf{free}\} \neq \emptyset$ ist, dann ist τ-*EINBETTUNG *NP-vollständig.*

Der Rest dieses Abschnitts ist dem Beweis von Satz 4.17 gewidmet. Solange nicht ausdrücklich anders vereinbart, sei im Rest dieses Abschnitts $\tau = \{\mathsf{swap}\} \cup \omega$ mit $\omega \subseteq \{\mathsf{inp}, \mathsf{out}, \mathsf{res}, \mathsf{set}, \mathsf{used}, \mathsf{free}\}$ und $\omega \cap \{\mathsf{res}, \mathsf{set}, \mathsf{used}, \mathsf{free}\} \neq \emptyset$ beliebig aber fest. Wir folgen der in Kapitel 3 vorgestellten Beweisstrategie. In Kapitel 2 haben wir für Notation 2.1 besprochen, dass wir $z \xleftarrow{e} z'$ eine bidirektionale Kante nennen, die kurz für $z \xrightarrow{e} z'$ und $z' \xrightarrow{e} z$ steht, wobei wir in diesem Zusammenhang sinnvollerweise $z \neq z'$ annehmen. In diesem Abschnitt erweitern wir diese Begrifflichkeit in natürlicher Weise und nennen einen Pfad $P = z_0 \xleftrightarrow{e_1} z_1 \xleftrightarrow{e_2} \dots \xleftrightarrow{e_n} z_n$ der nur aus bidirektionalen Kanten besteht *bidirektionalen Pfad*. Ebenso nennen wir ein Transitionssystem *bidirektional*, wenn es nur bidirektionale Kanten hat.

Die Reduktion für den Beweis von Satz 4.17 resultiert in einem bidirektionalen Transitionssystem A_τ. Insbesondere hat dieses Transitionssystem A_τ für jede Klausel $M_i = \{X_{i_0}, X_{i_1}, X_{i_2}\}$ einen bidirektionalen Pfad $P_i = \dots \xleftrightarrow{X_{i_0}} \dots \xleftrightarrow{X_{i_1}} \dots \xleftrightarrow{X_{i_2}} \dots$ auf dem die Variablen von M_i als Ereignisse stattfinden. Insgesamt sollen die Pfade $P_0, \dots, P_{m-1}$ dann $(\mathfrak{U}, M)$ repräsentieren. Darüber hinaus wird es ein ZTP $\alpha = (z, z')$ geben, wobei z und z' später genauer spezifiziert werden, das genau dann τ-lösbar ist, wenn es ein Eins-in-drei-Modell für $(\mathfrak{U}, M)$ gibt. Die Kombination aus Abwesenheit von nop und Präsenz von swap erschwert es jedoch erheblich, diese Äquivalenz zwischen τ-Lösbarkeit und Erfüllbarkeit zu konstruieren. Es stellt sich heraus, dass wir $(\mathfrak{U}, M)$ ein zweites Mal in A_τ repräsentieren müssen. Dazu nutzen wir eine Kopie $(\mathfrak{U}', M')$ von $(\mathfrak{U}, M)$, die wir durch einfaches Umbenennen der Variablen erhalten. Insbesondere erhalten wir für jede Variable $X \in \mathfrak{U}$ eine neue Variable X', das heißt $\mathfrak{U}' = \{X' \mid X \in \mathfrak{U}\}$; für jedes $i \in \{0, \dots, m-1\}$ und die Klausel $M_i = \{X_{i_0}, X_{i_1}, X_{i_2}\}$ erhalten wir in natürlicher Weise die neue Klausel $M'_i = \{X'_{i_0}, X'_{i_1}, X'_{i_2}\}$; schließlich erhalten wir $M' = \{M'_0, \dots, M'_{m-1}\}$. Offensichtlich erlaubt $(\mathfrak{U}, M)$ genau dann eine positive Entscheidung, wenn $(\mathfrak{U}', M')$ eine positive Entscheidung erlaubt.

Beispiel 4.3. Wir erhalten $(\mathfrak{U}', M')$ aus der Instanz $(\mathfrak{U}, M)$ aus Beispiel 3.1 durch $\mathfrak{U} = \{X'_0, \dots, X'_5\}$ und $M' = \{M'_0, \dots, M'_5\}$, wobei $M'_0 = \{X'_0, X'_1, X'_2\}$, $M'_1 = \{X'_0, X'_2, X'_3\}$, $M'_2 = \{X'_0, X'_1, X'_3\}$, $M'_3 = \{X'_2, X'_4, X'_5\}$, $M'_4 = \{X'_1, X'_4, X'_5\}$ und $M'_5 = \{X'_3, X'_4, X'_5\}$.

Das angekündigte Transitionssystem A_τ wird auch für jede (kopierte) Klausel $M'_i = \{X'_{i_0}, X'_{i_1}, X'_{i_2}\}$ einen Pfad $P'_i = \dots \xleftrightarrow{X'_{i_0}} \dots \xleftrightarrow{X'_{i_1}} \dots \xleftrightarrow{X'_{i_2}} \dots$ enthalten, sodass $P'_0, \cdots P'_{m-1}$ zusammen $(\mathfrak{U}', M')$ repräsentieren. Wenn nun eine τ-Region $R = (sup, sig)$ das ZTP $\alpha = (z, z')$ löst, dann werden entweder die Signaturwerte der Variablen-Ereignisse aus $\mathfrak{U}$ ein Eins-in-drei-Modell von $(\mathfrak{U}, M)$ oder die Signaturwerte der Variablen-Ereignisse aus $\mathfrak{U}'$ ein Eins-in-drei-Modell von $(\mathfrak{U}', M')$ definieren, je nachdem ob $sup(z) = 1$ und $sup(z') = 0$ oder $sup(z) = 0$ und $sup(z') = 1$. In beiden Fällen impliziert dies eine positive Entscheidung für $(\mathfrak{U}, M)$. Darüber hinaus wird die Existenz eines Eins-in-drei-Modells von $(\mathfrak{U}, M)$ die τ-ZTE von A_τ implizieren.

Das Transitionssystem A_τ ist die Komposition verschiedener Gadgets. Das folgende Lemma 4.37 offenbart einige grundlegende Eigenschaften von τ-Regionen bidirektionaler Transitionssysteme, die wir nutzen werden, um die Funktionalität der Gadgets von A_τ zu beweisen.

Lemma 4.37. *Sei τ ein boolescher Netztyp ohne* nop *und $A = (Z, E, \delta, \iota)$ ein bidirektionales Transitionssystem und seien $P_0 = z_0 \xleftrightarrow{e_1} \dots \xleftrightarrow{e_m} z_m$ sowie $P_1 = q_0 \xleftrightarrow{e_1} \dots \xleftrightarrow{e_m} q_m$ zwei bidirektionale Pfade von A, die beide die gleiche Sequenz $e_1 \dots e_m$ von Ereignissen verwenden. Es sei zudem $R = (sup, sig)$ eine τ-Region von A.*

1. *Für alle Ereignisse $e \in E$ gilt: Wenn $sig(e) \in \mathfrak{save}$, dann gilt $sup(z) = sup(z') = sup(t) = sup(t')$ für alle Kanten mit $z \xrightarrow{e} z'$ und $t \xrightarrow{e} t'$ in A.*
2. *Wenn $sup(z_m) \neq sup(q_m)$, dann gilt $sig(e_i) =$ swap *für alle $i \in \{1, \dots, m\}$.*
3. *Wenn $sup(z_0) = sup(z_m)$, dann ist die Anzahl der auf P_0 stattfindenden Ereignisse, deren Signaturwert* swap *ist, gerade.*

Beweis. (1): Weil A bidirektional ist, folgt aus $s \xrightarrow{e} s' \in A$ stets $s' \xrightarrow{e} s \in A$. Außerdem gilt für alle $i \in \tau \cap \mathfrak{save}$, dass $\xrightarrow{i} p$ und $\xrightarrow{i} p'$ stets $p = p'$ impliziert, denn $i \neq$ nop. Wenn nun $R = (sup, sig)$ eine τ-Region von A ist, sodass $sig(e) \in \tau \cap \mathfrak{save}$, dann folgt mit $\xrightarrow{sig(e)} sup(s)$ und $\xrightarrow{sig(e)} sup(s')$ sowie $\xrightarrow{sig(e)} sup(t)$ und $\xrightarrow{sig(e)} sup(t')$ die Behauptung.

(2): Es gilt $\xrightarrow{sig(e_m)} sup(s_m)$ und $\xrightarrow{sig(e_m)} sup(q_m)$. Mit nop $\notin \tau$ und $sup(s_m) \neq sup(q_m)$ impliziert dies $sig(e_m) =$ swap. Aus $sup(s_m) \neq sup(q_m)$ und $sig(e_m) =$ swap erhalten wir $sup(s_{m-1}) \neq sup(q_{m-1})$. Die Behauptung folgt damit leicht per Induktion über m.

(3): Das Bild P_0^R ist ein Pfad in τ, dessen Start- und Endzustand übereinstimmen. Deshalb ist die Anzahl der Zustandswechsel zwischen 0 und 1 auf P_0^R gerade. Weil P_0 ein bidirektionaler Pfad ist, gilt für alle $i \in \{0, \dots, m-1\}$ genau dann $sup(s_i) \neq sup(s_{i+1})$ (das heißt, es findet ein Wechsel zwischen 0 und 1 statt), wenn $sig(e_{i+1}) =$ swap. Die Behauptung folgt. □

Sei $(\mathfrak{U}, M)$ unsere beliebige aber feste Instanz für CM 1 IN 3 3SAT mit Variablenmenge $\mathfrak{U} = \{X_0, \dots, X_{m-1}\}$ und $(\mathfrak{U}', M')$ ihre aus Umbenennung resultierende Kopie mit Variablenmenge $\mathfrak{U}' = \{X'_0, \dots, X'_{m-1}\}$. Wir führen A_τ ein.

Das Transitionssystem A_τ hat für alle $i \in \{0, \dots, 7m-1\}$ die folgenden Gadgets G_i, F_i, G'_i und F'_i mit den Anfangszuständen g_{i_0}, $f_{i,0}$, $g'_{i,0}$ und $f'_{i,0}$, wobei G_0 das angekündigte ZTP $\alpha = (g_{0,2}, g_{0,4})$ liefert:

$$G_i = g_{i,0} \xleftrightarrow{v_i} g_{i,1} \xleftrightarrow{w_i} g_{i,2} \xleftrightarrow{k_0} g_{i,3} \xleftrightarrow{k_1} g_{i,4}$$

$$F_i = f_{i,0} \xleftrightarrow{v_i} f_{i,1} \xleftrightarrow{w_i} f_{i,2} \xleftrightarrow{k_1} f_{i,3} \xleftrightarrow{k_0} f_{i,4}$$

$$G'_i = g'_{i,0} \xleftrightarrow{v'_i} g'_{i,1} \xleftrightarrow{w'_i} g'_{i,2} \xleftrightarrow{k_1} g'_{i,3} \xleftrightarrow{k_0} g'_{i,4}$$

$$F'_i = f'_{i,0} \xleftrightarrow{v'_i} f'_{i,1} \xleftrightarrow{w'_i} f'_{i,2} \xleftrightarrow{k_0} f'_{i,3} \xleftrightarrow{k_1} f'_{i,4}$$

Des Weiteren hat A_τ für jedes $i \in \{0, \ldots, m-1\}$ die folgenden Gadgets $T_{i,0}$ und $T_{i,1}$ mit Startzuständen $t_{i,0,0}$ und $t_{i,1,0}$, welche die Variablen der Klausel $M_i = \{X_{i_0}, X_{i_1}, X_{i_2}\}$ als Ereignisse anwenden:

$$T_{i,0} = t_{i,0,0} \xleftrightarrow{k_0} t_{i,0,1} \xleftrightarrow{v_{7i}} t_{i,0,2} \xleftrightarrow{v_{7i+1}} t_{i,0,3} \xleftrightarrow{X_{i_0}} t_{i,0,4} \xleftrightarrow{v_{7i+2}} t_{i,0,5} \xleftrightarrow{X_{i_1}} t_{i,0,6} \xleftrightarrow{v_{7i+3}} t_{i,0,7} \updownarrow X_{i_2} \; t_{i,0,11} \xleftrightarrow[k_0]{} t_{i,0,10} \xleftrightarrow[v_{7i+5}]{} t_{i,0,9} \xleftrightarrow[v_{7i+4}]{} t_{i,0,8}$$

$$T_{i,1} = t_{i,1,0} \xleftrightarrow{X_{i_0}} t_{i,1,1} \xleftrightarrow{v_{7i+6}} t_{i,1,2} \xleftrightarrow{X_{i_2}} t_{i,1,3}$$

Ebenso hat das Transitionssystem A_τ für alle $i \in \{0, \ldots, m-1\}$ die Gadgets $T'_{i,0}$ und $T'_{i,1}$ mit Startzuständen $t'_{i,0,0}$ und $t'_{i,1,0}$, auf denen die Variablen der (kopierten) Klausel $M'_i = \{X'_{i_0}, X'_{i_1}, X'_{i_2}\}$ als Ereignisse stattfinden (man beachte außerdem das Ereignis k_1 in $T'_{i,0}$ im Unterschied zu k_0 in $T_{i,0}$):

$$T'_{i,0} = t'_{i,0,0} \xleftrightarrow{k_1} t'_{i,0,1} \xleftrightarrow{v'_{7i}} t'_{i,0,2} \xleftrightarrow{v'_{7i+1}} t'_{i,0,3} \xleftrightarrow{X'_{i_0}} t'_{i,0,4} \xleftrightarrow{v'_{7i+2}} t'_{i,0,5} \xleftrightarrow{X'_{i_1}} t'_{i,0,6} \xleftrightarrow{v'_{7i+3}} t'_{i,0,7} \updownarrow X'_{i_2} \; t'_{i,0,11} \xleftrightarrow[k_1]{} t'_{i,0,10} \xleftrightarrow[v'_{7i+5}]{} t'_{i,0,9} \xleftrightarrow[v'_{7i+4}]{} t'_{i,0,8}$$

$$T'_{i,1} = t'_{i,1,0} \xleftrightarrow{X'_{i_0}} t'_{i,1,1} \xleftrightarrow{v'_{7i+6}} t'_{i,1,2} \xleftrightarrow{X'_{i_2}} t'_{i,1,3}$$

Die eingeführten Gadgets werden dann mittels ihrer Startzustände sowie zusätzlicher Ereignisse und Zustände, wie in Abbildung 4.11 skizziert, zu A_τ mit Initialzustand ι verbunden.

Lemma 4.38. *Wenn A_τ die τ-ZTE hat, dann gibt es ein Eins-in-drei-Modell für $(\mathfrak{U}, M)$.*

Beweis. Weil A_τ die τ-ZTE hat, gibt es eine τ-Region, die das ZTP $\alpha = (g_{0,2}, g_{0,6})$ löst. Sei $R = (sup, sig)$ eine solche Region. A_τ ist ein bidirektionales Transitionssystem, somit gilt $sig(e) \notin \{\mathsf{inp}, \mathsf{out}\}$ und $sig(e) \in \{\mathsf{swap}\} \cup \mathfrak{save}$ für alle Ereignisse e von A_τ.

Wir zeigen zunächst, dass entweder $sig(k_0) \in \mathfrak{save}$ und $sig(k_1) = \mathsf{swap}$ oder $sig(k_0) = \mathsf{swap}$ und $sig(k_1) \in \mathfrak{save}$ gilt: Weil R das ZTP α löst, gilt $sup(g_{0,2}) \neq sup(g_{0,4})$. Wenn $sig(k_0) = sig(k_1) = \mathsf{swap}$, dann $sup(g_{0,2}) \neq sup(g_{0,3}) \neq sup(g_{0,4})$ und wir erhalten den Widerspruch $sup(g_{0,2}) = sup(g_{0,4})$. Ebenso folgt aus $sig(k_0) \in \mathfrak{save}$ und $sig(k_1) \in \mathfrak{save}$ der Widerspruch $sup(g_{0,2}) = sup(g_{0,3}) = sup(g_{0,4})$. Dies zeigt die Behauptung.

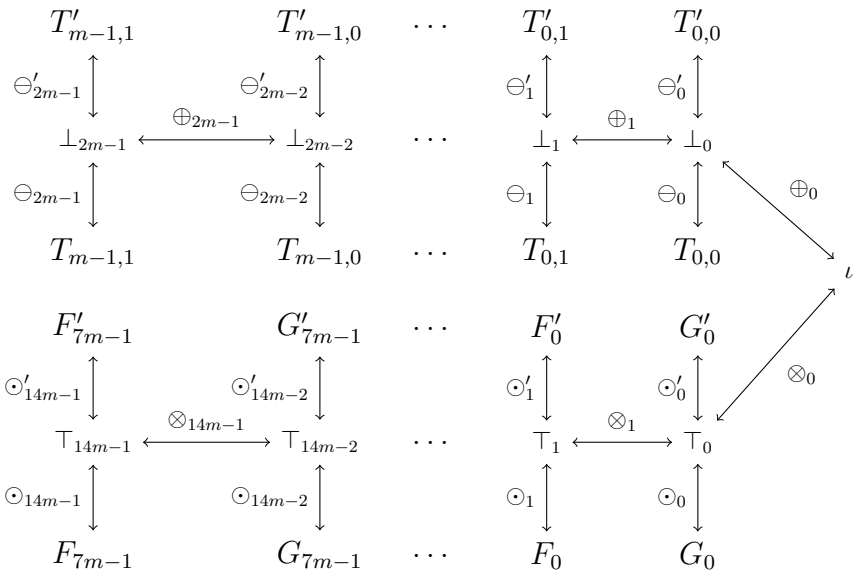


Abbildung 4.11: Das Transitionssystem A_τ für den Beweis von Satz 4.17.

Als nächsten Schritt zeigen wir, dass wenn $sig(k_0) \in \mathfrak{save}$ und $sig(k_1) = \mathsf{swap}$ oder $sig(k_0) = \mathsf{swap}$ und $sig(k_1) \in \mathfrak{save}$, dann $sig(v_i) = sig(w_i) = \mathsf{swap}$ und $sig(v'_i) = sig(w'_i) = \mathsf{swap}$ für alle $i \in \{0, \ldots, 7m-1\}$: Sei $i \in \{0, \ldots, 7m-1\}$ beliebig aber fest. Wenn $sig(k_0) \in \mathfrak{save}$ und $sig(k_1) = \mathsf{swap}$, dann erhalten wir zunächst $sup(g_{i,2}) = sup(f_{i,3})$ und $sup(g'_{i,3}) = sup(f'_{i,2})$ mit Lemma 4.37.1. Mit $sig(k_1) = \mathsf{swap}$ impliziert dies $sup(g_{i,2}) \neq sup(f_{i,2})$ sowie $sup(g'_{i,2}) \neq sup(f'_{i,2})$. Gemäß Lemma 4.37.2 folgt $sig(v_i) = sig(w_i) = \mathsf{swap}$ und $sig(v'_i) = sig(w'_i) = \mathsf{swap}$. Analog argumentiert man, dass der Fall $sig(k_0) = \mathsf{swap}$ und $sig(k_1) \in \mathfrak{save}$ zu diesen Signaturwerten für v_i, w_i und v'_i, w'_i führt. Weil i beliebig war, folgt die Behauptung.

Als weiteren Schritt zeigen wir nun: Wenn $sig(k_0) \in \mathfrak{save}$ und $sig(v_0) = \cdots = sig(v_{7m-1}) = \mathsf{swap}$, dann definiert die Menge $S = \{X \in \mathfrak{U} \mid sig(X) \in \mathfrak{save}\}$ ein Eins-in-drei-Modell von $(\mathfrak{U}, M)$. Sei $i \in \{0, \ldots, m-1\}$ beliebig aber fest. Weil $sig(k_0) \in \mathfrak{save}$, erhalten wir $sup(t_{i,0,1}) = sup(t_{i,0,10})$ mit Lemma 4.37.1. Dies impliziert mit Lemma 4.37.3, dass die Anzahl $|\{e \in \{v_{7i}, \ldots, v_{7i+5}, X_{i_0}, X_{i_1}, X_{i_2}\} \mid sig(e) = \mathsf{swap}\}|$ gerade ist. Weil $sig(v_{7i}) = \cdots = sig(v_{7i+5}) = \mathsf{swap}$, bedeutet dies, dass entweder genau zwei Ereignisse aus $\{X_{i_0}, X_{i_1}, X_{i_2}\}$ eine swap-Signatur haben oder keines dieser Ereignisse eine swap-Signatur hat, dass heißt $sig(X) \in \mathfrak{save}$ für alle $X \in \{X_{i_0}, X_{i_1}, X_{i_2}\}$. Wir argumentieren nun, dass der zweite Fall nicht eintreten kann: Wenn $sig(X) \in \mathfrak{save}$ für alle $X \in \{X_{i_0}, X_{i_1}, X_{i_2}\}$, dann erhalten wir mit $sig(v_{7i+2}) = sig(v_{7i+3}) = \mathsf{swap}$ zunächst $sup(t_{i,0,4}) \neq sup(t_{i,0,5}) = sup(t_{i,0,6}) \neq sup(t_{i,0,7})$, also insbesondere $sup(t_{i,0,4}) = sup(t_{i,0,7})$. Dies impliziert $sup(t_{i,1,1}) = sup(t_{i,1,2})$ mit Lemma 4.37.1 und widerspricht $sig(v_{7i+6}) = \mathsf{swap}$. Somit gibt es genau zwei Ereignisse aus $\{X_{i_0}, X_{i_1}, X_{i_2}\}$, die eine swap-Signatur haben beziehungsweise genau ein $X \in \{X_{i_0}, X_{i_1}, X_{i_2}\}$ mit $sig(X) \in \mathfrak{save}$. Weil i beliebig war,

gilt dies für alle $i \in \{0, \ldots, m-1\}$ gleichzeitig. Somit ist $S = \{X \in \mathfrak{U} \mid sig(X) \in \mathfrak{save}\}$ ein Eins-in-drei-Modell von $(\mathfrak{U}, M)$.

Analog zeigt man, dass $S' = \{X' \in \mathfrak{U}' \mid sig(X') \in \mathfrak{save}\}$ ein Eins-in-drei-Modell von $(\mathfrak{U}', M')$ ist, wenn $sig(k_1) \in \mathfrak{save}$ und $sig(v'_0) = \cdots = sig(v'_{7m-1}) = \mathsf{swap}$. Ein Eins-in-drei-Modell von $(\mathfrak{U}, M)$ erhalten wir dann durch $S = \{X \in \mathfrak{U} \mid X' \in S'\}$. Weil $sig(k_0) \in \mathfrak{save}$ und $sig(v_0) = \cdots = sig(v_{7m-1}) = \mathsf{swap}$ oder $sig(k_1) \in \mathfrak{save}$ und $sig(v'_0) = \cdots = sig(v'_{7m-1}) = \mathsf{swap}$ erfüllt ist und $(\mathfrak{U}, M)$ genau dann eine positive Entscheidung erlaubt, wenn $(\mathfrak{U}', M')$ eine positive Entscheidung erlaubt, folgt das Lemma. □

Für die andere Richtung bleibt zu zeigen, dass A_τ die τ-ZTE hat, wenn $(\mathfrak{U}, M)$ ein Eins-in-drei-Modell hat. Gemäß des folgenden Lemmas reicht es sogar zu zeigen, dass A_τ die $\{\mathsf{swap}, \mathsf{free}\}$-ZTE hat:

Lemma 4.39. *Wenn A_τ die $\{\mathsf{swap}, \mathsf{free}\}$-ZTE hat, dann hat A_τ auch die τ-ZTE.*

Beweis. Man überlegt sich das Folgende: Wenn $R = (sup, sig)$ eine $\{\mathsf{swap}, \mathsf{free}\}$-Region von A_τ ist, dann gilt für ein Ereignis e von A_τ genau dann $sig(e) = \mathsf{free}$, wenn $sup(z) = sup(z') = 0$ für alle e-beschrifteten Kanten $z \xrightarrow{e} z'$ in A_τ. Dies impliziert, dass die folgende Region $R' = (sup, sig')$ mit demselben Support wie R wohldefiniert ist: für alle $e \in E(A_\tau)$, wenn $sig(e) = \mathsf{free}$, dann $sig'(e) = \mathsf{res}$, sonst $sig'(e) = \mathsf{swap}$. Dabei beachten wir, dass R alle Ereignisse, die nicht auf free abgebildet werden, auf swap abbildet. Es folgt, dass R genau dann ein ZTP α von A_τ löst, wenn R' das ZTP α löst. Wenn A_τ die $\{\mathsf{swap}, \mathsf{free}\}$-ZTE hat, dann hat A_τ also auch die $\{\mathsf{swap}, \mathsf{res}\}$-ZTE und somit sogar die τ-ZTE für alle τ mit $\tau \cap \mathfrak{save}_0 \neq \emptyset$. Nun gelten aber die Kongruenzen $\{\mathsf{swap}, \mathsf{free}\} \equiv \{\mathsf{swap}, \mathsf{used}\}$ und $\{\mathsf{swap}, \mathsf{res}\} \equiv \{\mathsf{swap}, \mathsf{set}\}$, sodass mit Satz 2.2 auch für alle nop-freien booleschen Netztypen τ mit $\tau \cap \mathfrak{save}_1 \neq \emptyset$ folgt, dass A_τ die τ-ZTE hat. □

Lemma 4.40. *Wenn $(\mathfrak{U}, M)$ ein Eins-in-drei-Modell hat, dann hat A_τ die $\{\mathsf{swap}, \mathsf{free}\}$-ZTE.*

Beweis. Sei S ein Eins-in-drei-Modell von $(\mathfrak{U}, M)$ und $\tau = \{\mathsf{swap}, \mathsf{free}\}$. Wir werden im Folgenden argumentieren, dass A_τ die τ-ZTE hat (wobei wir beachten, dass A_τ für alle Netztypen aus Satz 4.17 dieselbe Gestalt hat).

Der Einfachheit halber fassen wir gewisse Ereignisse und Zustände von A_τ wie folgt zusammen: $\oplus = \{\oplus_0, \ldots, \oplus_{2m-1}\}$, $\otimes = \{\otimes_0, \ldots, \otimes_{14m-1}\}$, $\ominus = \{\ominus_0, \ldots, \ominus_{2m-1}\}$, $\odot = \{\odot_0, \ldots, \odot_{14m-1}\}$, $V = \{v_0, \ldots, v_{7m-1}\}$, $W = \{w_0, \ldots, w_{7m-1}\}$, $\ominus' = \{\ominus'_0, \ldots, \ominus'_{2m-1}\}$, $\odot' = \{\odot'_0, \ldots, \odot'_{14m-1}\}$, $V' = \{v'_0, \ldots, v'_{7m-1}\}$, und $W' = \{w'_0, \ldots, w'_{7m-1}\}$ und $\bot = \{\bot_0, \ldots, \bot_{2m-1}\}$ sowie $\top = \{\top_0, \ldots, \top_{14m-1}\}$.

Der Übersichtlichkeit halber definieren wir Regionen $R = (sup, sig)$ wieder implizit mittels $sup(\iota)$ und sig. Weil $\tau = \{\mathsf{swap}, \mathsf{free}\}$, genügt es stets eine der Mengen $\mathcal{E}^R_{\mathsf{swap}} = \{e \in E(A_\tau) \mid sig(e) = \mathsf{swap}\}$ beziehungsweise $\mathcal{E}^R_{\mathsf{free}} = \{e \in E(A_\tau) \mid sig(e) = \mathsf{free}\}$ anzugeben, denn es folgt sofort $\mathcal{E}^R_{\mathsf{free}} = E(A_\tau) \setminus \mathcal{E}^R_{\mathsf{swap}}$ beziehungsweise $\mathcal{E}^R_{\mathsf{swap}} = E(A_\tau) \setminus \mathcal{E}^R_{\mathsf{free}}$. Um die Einsicht in die Funktionalität der Regionen zu erleichtern, fügen wir darüber hinaus aussagekräftige Skizzen hinzu.

Es ist leicht zu sehen, dass z lösbar ist, wenn $z \in \bot \cup \top$. Wir diskutieren diese Zustände daher nicht explizit.

Wenn A ein beliebiges aber festes Gadget von A_τ ist, dann ist das ZTP (z, z') für alle $z \in Z(A)$ und alle $z' \in Z(A_\tau) \setminus Z(A)$ τ-lösbar, wie die folgenden Überlegungen zeigen: Sei $a \in \ominus \cup \odot \cup \ominus' \cup \odot'$ das eindeutige Ereignis, welches A über seinen Startzustand mit einem Zustand aus $\bot \cup \top$ verbindet. Einerseits gibt es eine Region $R_1 = (sup_1, sig_1)$, die wie folgt definiert ist: $sup_1(\iota) = 0$ und $\mathcal{E}^{R_1}_{\mathsf{free}} = \oplus \cup \otimes \cup \ominus \cup \odot \cup \ominus' \cup \odot'$. Siehe Abbildung 4.12 für eine Skizze von R_1 eingeschränkt auf G_{7i+j} und F_{7i+j}, wobei $i \in \{0, \ldots, m-1\}$ und $j \in \{0, \ldots, 6\}$.

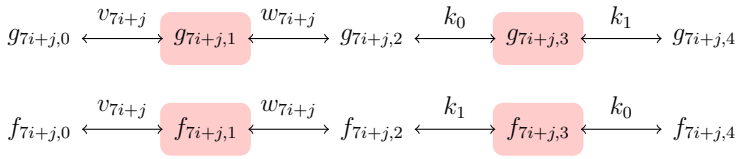


Abbildung 4.12: Skizze der Region R_1 eingeschränkt auf G_{7i+j} und F_{7i+j}.

Andererseits ist auch die folgende Region $R_2 = (sup_2, sig_2)$ wohldefiniert: $sup_2(\iota) = 0$ und $\mathcal{E}^{R_2}_{\mathsf{free}} = (\oplus \cup \otimes \cup \ominus \cup \odot \cup \ominus' \cup \odot') \setminus \{a\}$. Siehe auch Abbildung 4.13. Nun gilt für alle Zustände $z \in Z(A_\tau)$, dass $sup_1(z) = sup_2(z)$, wenn $z \notin Z(A)$ und andernfalls $sup_2(z) = 1 - sup_1(z)$. Wenn $z \in Z(A)$ und $z' \in Z(A_\tau) \setminus Z(A)$, dann ist das ZTP (z, z') deshalb durch R_1 oder durch R_2 gelöst.

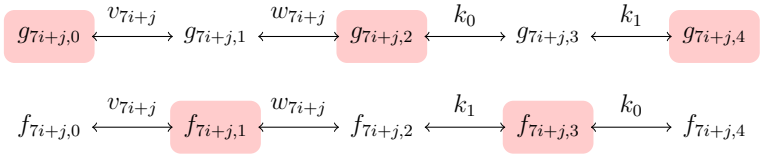


Abbildung 4.13: Skizze der Region R_2 mit $a = \odot_{14i+2j}$ eingeschränkt auf G_{7i+j} und F_{7i+j}.

Wegen dieser Beobachtungen bleibt nur noch zu zeigen, dass ein (jedes) ZTP (z, z') auch dann lösbar ist, wenn z und z' Zustände des gleichen Gadgets A von A_τ sind.

Solange nicht explizit anders vereinbart, seien im Folgenden $i \in \{0, \ldots, m-1\}$ und $j \in \{0, \ldots, 6\}$ beliebig aber fest. Wir zeigen sukzessive, dass z für alle $z \in Z(G_{7i+j})$ τ-lösbar ist. Aus der Beliebigkeit von i und j sowie der Symmetrie von G_{7i+j} und G'_{7i+j} folgt dann die Lösbarkeit aller Zustände $z \in \bigcup_{i=0}^{7m-1}(Z(G_i) \cup Z(G'_i))$.

Die wie folgt definierte τ-Region $R_3 = (sup_3, sig_3)$ löst $(g_{7i+j,3}, z)$ für alle Zustände $z \in \{g_{7i+j,0}, g_{7i+j,1}, g_{7i+j,2}, g_{7i+j,4}\}$: $sup_3(\iota) = 0$ und $\mathcal{E}^{R_3}_{\mathsf{swap}} = \{k_0, k_1\} \cup \{\ominus_{2i}, \ominus'_{2i} \mid i \in \{0, \ldots, m-1\}\}$. Siehe auch Abbildung 4.14. Somit ist $g_{7i+j,3}$ τ-lösbar.

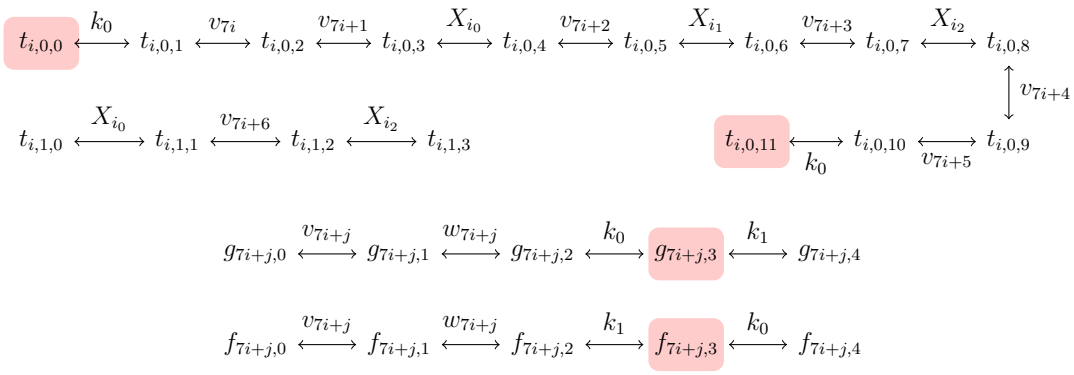


Abbildung 4.14: Skizze der Region R_3 eingeschränkt auf $T_{i,0}, T_{i,1}, G_{7i+j}$ und F_{7i+j}.

Die wie folgt definierte τ-Region $R_4 = (sup_4, sig_4)$ löst $(g_{7i+j,0}, z)$ für alle Zustände $z \in \{g_{7i+j,1}, g_{7i+j,2}, g_{7i+j,4}\}$: $sup(\iota) = 0$ und $\mathcal{E}^{R_4}_{\mathsf{swap}} = \{k_0, k_1\} \cup \odot \cup \odot' \cup V \cup V' \cup \mathfrak{U} \cup \mathfrak{U}'$. Siehe auch Abbildung 4.15. Wegen R_3 und R_4 ist $g_{7i+j,0}$ somit τ-lösbar.

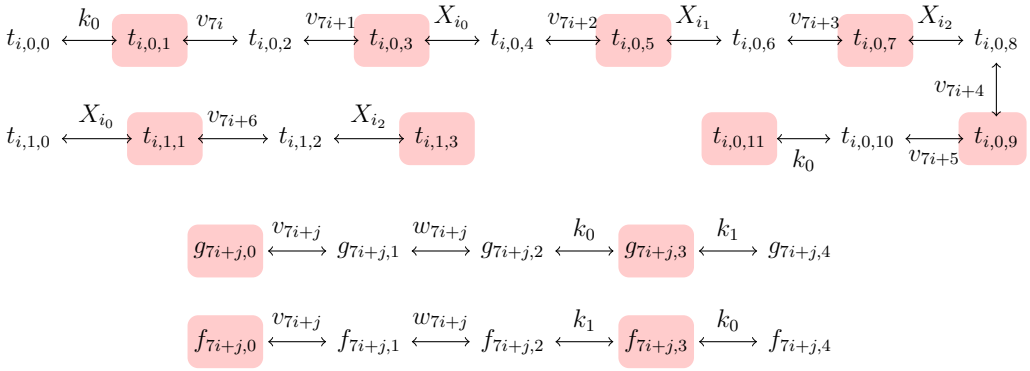


Abbildung 4.15: Die Region R_4 eingeschränkt auf $T_{i,0}, T_{i,1}, G_{7i+j}$ und F_{7i+j}.

Die wie folgt definierte τ-Region $R_5 = (sup_5, sig_5)$ löst $(g_{7i+j,1}, z)$ für alle Zustände $z \in \{g_{7i+j,2}, g_{7i+j,4}\}$: $sup_5(\iota) = 0$ und $\mathcal{E}^{R_5}_{\mathsf{swap}} = \{k_0, k_1\} \cup V \cup V' \cup W \cup W' \cup \mathfrak{U} \cup \mathfrak{U}'$. Siehe auch Abbildung 4.16. Mit R_3 und R_4 und R_5 erhalten wir die τ-Lösbarkeit von $g_{7i+j,1}$.

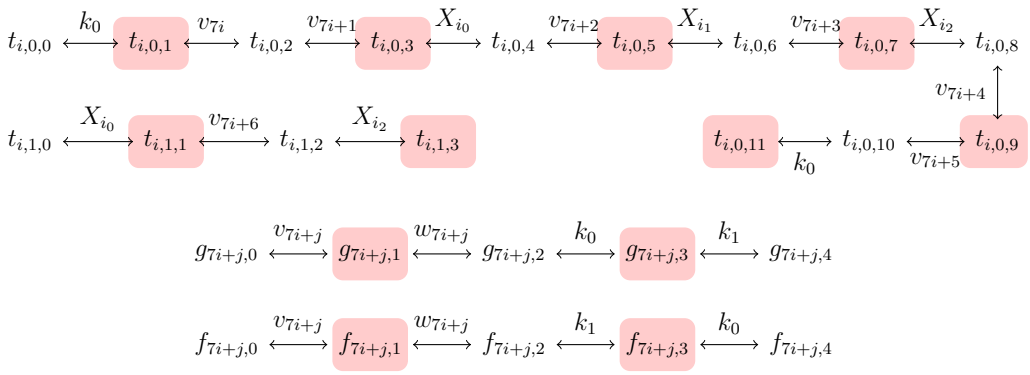


Abbildung 4.16: Die Region R_5 eingeschränkt auf $T_{i,0}, T_{i,1}, G_{7i+j}$ und F_{7i+j}.

Schließlich bleibt uns bezüglich G_{7i+j} nur noch das ZTP $(g_{7i+j,2}, g_{7i+j,4})$ zu lösen. Die folgende τ-Region $R_6 = (sup_6, sig_6)$ löst $(g_{7i+j,2}, g_{7i+j,4})$ und benötigt die Existenz des Eins-in-drei-Modells S: $sup_6(\iota) = 0$ und $\mathcal{E}^{R_6}_{\mathsf{swap}} = \{k_1\} \cup V \cup V' \cup W \cup W' \cup (\mathfrak{U} \setminus S) \cup \mathfrak{U}' \cup \{\odot_{2i+1}, \odot'_{2i} \mid i \in \{0, \ldots, 7m-1\}\}$. Siehe auch Abbildung 4.17 und Abbildung 4.18. Ingesamt komplettieren die Regionen $R_3, \ldots, R_6$ den Beweis der τ-Lösbarkeit von $g_{7i+j,4}$.

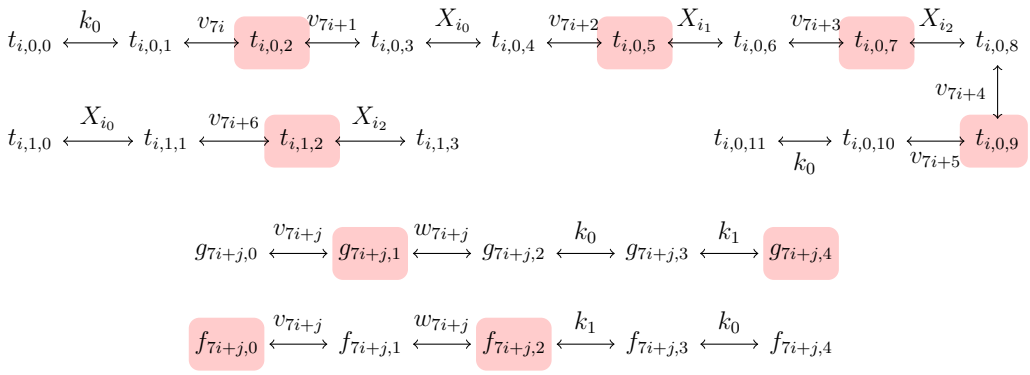


Abbildung 4.17: Skizze der Region R_6 eingeschränkt auf $T_{i,0}, T_{i,1}, G_{7i+j}, F_{7i+j}$ für den Fall $X_{i_0} \in S$.

Man beachte, dass die Regionen $R_3, \ldots, R_6$ ebenso den Beweis der τ-Lösbarkeit aller Zustände $z \in Z(F_{7i+j})$ komplettieren. Insgesamt dürfen wir aus der Beliebigkeit von i und j sowie der Ähnlichkeit (beziehungsweise Symmetrie) zwischen G_{7i+j} und G'_{7i+j} sowie zwischen F_{7i+j} und F'_{7i+j} folgern, dass der Zustand z für alle $z \in Z(G_\ell) \cup Z(F_\ell) \cup Z(G'_\ell) \cup Z(F'_\ell)$ und alle $\ell \in \{0, \ldots, 7m-1\}$ τ-lösbar ist.

Wir erinnern daran, dass $i \in \{0, \ldots, m-1\}$ beliebig aber fest gewählt war. Im Folgenden argumentieren wir, dass z für alle $z \in Z(T_{i,0})$ τ-lösbar ist.

Die folgende τ-Region $R_7 = (sup_7, sig_7)$ löst $(t_{i,0,2}, z)$ für alle $z \in Z(T_{i,0}) \setminus \{t_{i,0,2}\}$: $sup_7(\iota) = 0$ und $\mathcal{E}^{R_7}_{\mathsf{swap}} = \{v_{7i}, w_{7i}, v_{7i+1}, w_{7i+1}\}$. Siehe auch Abbildung 4.19. Analog zeigt

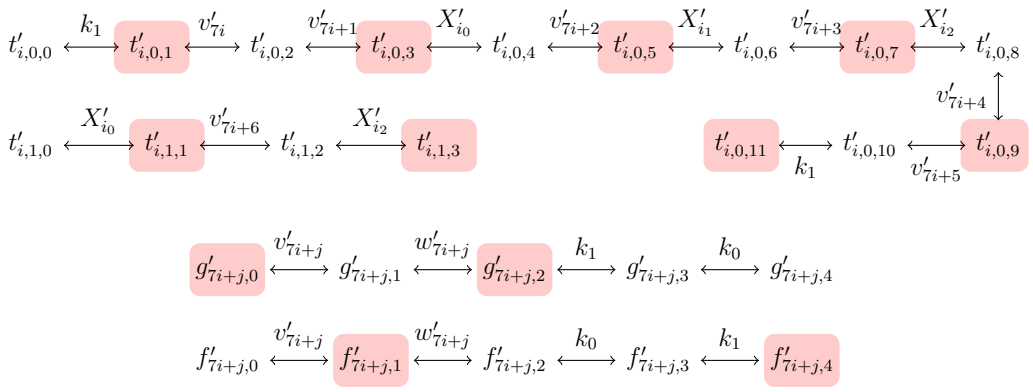


Abbildung 4.18: Skizze der Region R_6 eingeschränkt auf $T'_{i,0}, T'_{i,1}, G'_{7i+j}, F'_{7i+j}$.

man, dass $(t_{i,0,9}, z)$ für alle relevanten $z \in Z(T_{i,0}) \setminus \{t_{i,0,9}\}$ τ-lösbar ist. Die τ-Lösbarkeit von $t_{i,0,2}$ und $t_{i,0,9}$ folgt.

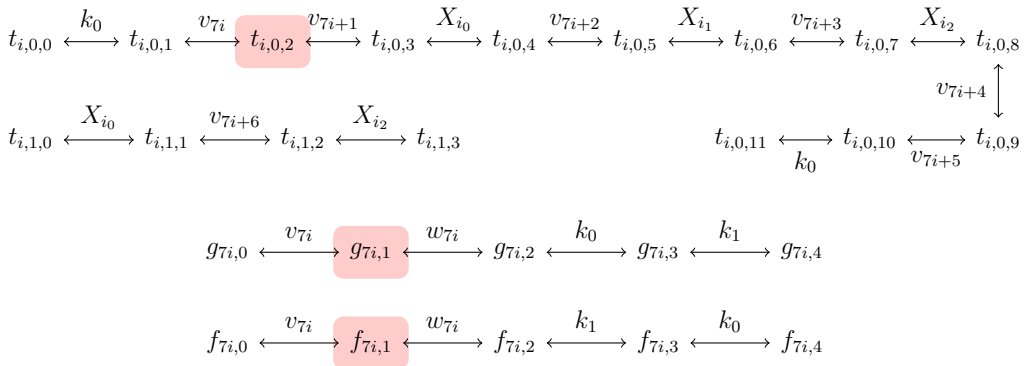


Abbildung 4.19: Skizze der Region R_7 eingeschränkt auf $T_{i,0}, T_{i,1}, G_{7i}$ und F_{7i}.

Untersuchen wir nun die τ-Lösbarkeit von $(t_{i,0,3}, z)$ für alle Zustände $z \in Z(T_{i,0}) \setminus \{t_{i,0,3}\}$. Dazu überlegen wir uns, dass es eine τ-Region $R_8 = (sup_8, sig_8)$ von A_τ gibt, sodass $sup_8(t_{i,0,3}) = 1$ und $sup_8(z) = 0$ für alle $z \in Z(T_{i,0}) \setminus \{t_{i,0,3}\}$ sowie $sig_8(X_{i_0}) = sig_8(v_{7i+1}) = \mathsf{swap}$ und $sig_8(e) = \mathsf{free}$ für alle Ereignisse $e \in E(T_{i,0}) \setminus \{X_{i_0}, v_{7i+1}\}$. Zunächst müssen wir mit den (höchstens) fünf weiteren Vorkommen von X_{i_0} in A_τ umgehen: dem Vorkommen in $T_{i,1}$ und den (höchstens) vier weiteren Vorkommen in $T_{j,0}, T_{j,1}$ sowie $T_{\ell,0}$ und $T_{\ell,1}$ für die paarweise verschiedenen Indizes $j, \ell \in \{0, \ldots, m-1\} \setminus \{i\}$, sodass $X_{i_0} \in M_j \cap M_\ell$. Leider wissen wir nicht „genau“, welches der Ereignisse $X_{j_0}, X_{j_1}, X_{j_2}$ beziehungsweise $X_{\ell_0}, X_{\ell_1}, X_{\ell_2}$ dem Ereignis X_{i_0} entspricht. Daher würde eine formale (vollständige) Definition von R_8 eine kraftraubende und wenig erkenntnisreiche Kaskade von Fallunterscheidungen erfordern. Da dies wenig sinnvoll erscheint, argumentieren wir stattdessen informal, dass R_8 existiert. Die folgende Beobachtung ist grundlegend: Für jede

Kante $s \xleftarrow{X_{i_0}} s'$ in A_τ, die von $t_{i,0,3} \xleftarrow{X_{i_0}} t_{i,0,4}$ verschieden ist, gibt es ein $v \in V$ sodass $\xleftarrow{v} s$ oder $\xleftarrow{v} s'$. Um R_8 zu definieren wählen wir nun für jede solcher Kanten genau eines solcher Ereignisse $v \in V$ und das eindeutige Ereignis $w \in W$, dass den gleichen Index wie v hat. Zum Beispiel wählen wir für die Kante $t_{i,1,0} \xrightarrow{X_{i_0}} t_{i,1,1}$ das Ereignis v_{7i+6} sowie w_{7i+6}. Wir bilden nun ι auf 0 und die genannten Ereignisse ebenso wie X_{i_0} und v_{7i+3} auf swap ab. Alle anderen Ereignisse erhalten den Signaturwert free. Diese Wahl der Signatur sorgt dafür, dass jedes Gadget von A_τ höchstens eine Zustand mit positivem Support enthält, denn „swap-Ereignisse“ treten stets paarweise hintereinander auf. Das resultierende Paar von Abbildungen $R_8 = (sup_8, sig_8)$ ist eine τ-Region und verhält sich, wie angekündigt. Abbildung 4.20 skizziert R_8 eingeschränkt auf $T_{i,0}, T_{i,1}, G_{7i+1}$ und F_{7i+1}.

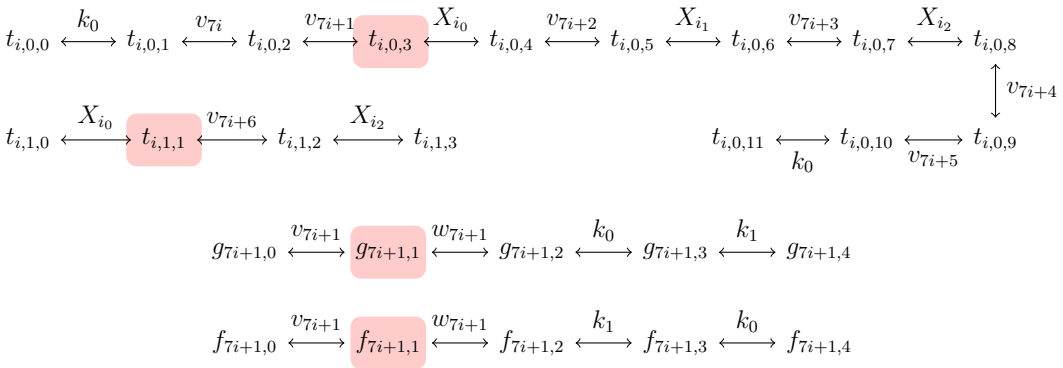


Abbildung 4.20: Die Region R_8 eingeschränkt auf $T_{i,0}, T_{i,1}, G_{7i+1}$ und F_{7i+1}.

Analog zur Lösbarkeit von $t_{i,0,3}$ argumentiert man für alle $z \in \{t_{i,0,4}, \ldots, t_{i,0,8}\}$, dass z τ-lösbar ist.

Soweit haben wir gezeigt, dass die Zustände $t_{i,0,2}, \ldots, t_{i,0,9}$ τ-lösbar sind. Mit Blick auf $T_{i,0}$ bleibt deshalb nur noch zu zeigen, dass $t_{i,0,0}, t_{i,0,1}, t_{i,0,10}$ und $t_{i,0,11}$ paarweise voneinander getrennt werden können. Es stellt sich heraus, dass die bisher besprochenen Regionen bereits ausreichen, um die zugehörigen ZTP zu lösen. Genauer gesagt löst die Region R_4 die ZTP $(t_{i,0,0}, t_{i,0,1})$ und $(t_{i,0,0}, t_{i,0,11})$ sowie $(t_{i,0,1}, t_{i,0,10})$ als auch $(t_{i,0,10}, t_{i,0,11})$. Des Weiteren löst die Region R_3 die ZTP $(t_{i,0,0}, t_{i,0,10})$ und $(t_{i,0,1}, t_{i,0,11})$. (Man beachte die jeweiligen Skizzen zu den genannten Regionen.) Folglich haben wir für alle $z \in T(T_{i,0})$ gezeigt, dass z τ-lösbar ist.

Schließlich ist es leicht zu sehen, dass alle von $T_{i,1}$ herrührenden ZTP (auf ähnliche Weise) τ-lösbar sind. Analog zur τ-Lösbarkeit der ZTP von $T_{i,0}$ und $T_{i,1}$ lassen sich jene von $T'_{i,0}$ und $T'_{i,1}$ lösen. Da i beliebig war, erhalten wir die τ-Lösbarkeit von z für alle $z \in \bigcup_{i=0}^{m-1}(Z(T_{i,0}) \cup Z(T_{i,1}) \cup Z(T'_{i,0}) \cup Z(T'_{i,1}))$. Dies beweist das Lemma. □

Die präsentierte Reduktion ist offenbar polynomiell. Aus Lemma 4.38, Lemma 4.39 und Lemma 4.40, der Zugehörigkeit zu NP sowie der NP-Vollständig von CM 1 IN 3

3SAT folgt schließlich Satz 4.17.

4.3.6 Die Netztypen B9 - B10

Der folgende Satz adressiert die Netztypen B9 aus Abbildung 4.8 und konstatiert, dass Transitionssysteme, für die es eine Einbettung dieser Typen gibt, in gewissem Sinne trivial sind:

Satz 4.18. *Sei $\tau = \{\mathsf{swap}\} \cup \omega$ mit $\omega \subseteq \{\mathsf{inp}, \mathsf{out}\}$ und $A = (Z, E, \delta, \iota)$ ein Transitionssystem. Wenn A die τ-ZTE hat, dann hat A höchstens zwei Zustände, das heißt $|Z| \leq 2$. Es lässt sich mit Zeitaufwand $\mathcal{O}(|E|)$ entscheiden, ob A die τ-ZTE hat.*

Beweis. Habe A die τ-ZTE. Wenn A eine Schleife $z \xrightarrow{e} z$ hat, dann hat A keine τ-Region. Deshalb kann ein Transitionssystem mit Schleife, das die τ-ZTE hat, höchstens einen Zustand haben.

Betrachten wir nun den Fall, dass A schleifenfrei ist, das heißt, wenn $z \xrightarrow{e} z' \in A$, dann $z \neq z'$. Zunächst zeigen wir, dass es höchstens eine Kante $\iota \xrightarrow{e} s$ geben kann: Nehmen wir an, dass A Kanten $\iota \xrightarrow{e} z$ und $\iota \xrightarrow{e'} z'$ mit $z \neq z'$ enthält. Weil A die τ-ZTE hat, ist das ZTP (z, z') lösbar. Sei $R = (sup, sig)$ eine τ-Region, die (s, s') löst, das heißt $sup(z) \neq sup(z')$. Wenn $sup(\iota) = sup(z)$, dann erhalten wir $sig(e) \in \mathfrak{save}$ und wenn $sup(\iota) \neq sup(z)$, das heißt $sup(\iota) = sup(z')$, dann erhalten wir $sig(e') \in \mathfrak{save}$. Dies widerspricht $\tau \cap \mathfrak{save} = \emptyset$. Somit kann es höchstens eine Kante mit Ursprung ι geben.

Zweitens zeigen wir, dass wenn $\iota \xrightarrow{e} z$ und $z \xrightarrow{e'} z'$ Kanten in A sind, dann gilt $z' = \iota$: Nehmen wir an, $z' \neq \iota$. Weil A die τ-ZTE hat, ist das ZTP (ι, z') lösbar. Sei $R = (sup, sig)$ eine τ-Region, die (ι, s') löst, das heißt $sup(\iota) \neq sup(z')$. Wenn $sup(\iota) = sup(z)$, dann $sig(e) \in \mathfrak{save}$ und wenn $sup(\iota) \neq sup(z)$, das heißt $sup(z) = sup(z')$, dann $sig(e') \in \mathfrak{save}$. Beide Fälle widersprechen der Voraussetzung $\tau \cap \mathfrak{save} = \emptyset$ und wir erhalten $z' = \iota$.

Für den Zeitaufwand: A hat höchstens zwei Zustände z, z'. Diese sind genau dann trennbar, wenn $R = (sup, sig)$ wie folgt eine wohldefinierte Region ist: $sup(z) = 0$, $sup(z') = 1$ und $sig(e) = \mathsf{swap}$ für alle $e \in E$. Die Behauptung folgt. □

Der nächste Satz adressiert die Netztypen B10 aus Abbildung 4.8 und folgt direkt aus Abschnitt 4.2, in dem gezeigt wurde, dass für boolesche Netztypen ohne nop und swap in Polynomialzeit entschieden werden kann, ob ein Transitionssystem die τ-ZTE hat.

Satz 4.19 (Ohne Beweis)**.** *Wenn $\tau \subseteq \{\mathsf{inp}, \mathsf{out}, \mathsf{res}, \mathsf{set}, \mathsf{used}, \mathsf{free}\}$, dann ist τ-EINBETTUNG in Polynomialzeit entscheidbar.*

4.4 Zusammenfassung und Ausblick

In diesem Kapitel wurden die Komplexität der klassischen Entscheidungsprobleme τ-EINBETTUNG, τ-SPRACHSIMULATION und τ -REALISIERUNG der Netz-Synthese für boo-

lesche Netztypen untersucht. Zunächst zeigt sich, dass diese Probleme für alle Netztypen in NP sind.

Eine beträchtliche Teil-Charakterisierung gelingt für τ-Sprachsimulation und τ -Realisierung: Für 84 von 128 booleschen Netztypen mit nop sind beide Probleme NP-vollständig, für 40 beziehungsweise 36 von diesen Netztypen sind τ-Sprachsimulation beziehungsweise τ-Realisierung in Polynomialzeit lösbar. τ-Realisierung ist außerdem für alle 64 booleschen Netztypen ohne nop und swap ebenso in polynomieller Zeit entscheidbar.

Darüber hinaus gelingt eine vollständige Charakterisierung der Komplexität von τ-Einbettung: Für 152 der 256 möglichen booleschen Netztypen ist das Problem NP-vollständig und für die verbleibenden 104 Netztypen ist es in Polynomialzeit entscheidbar.

Der Inhalt des aktuellen Kapitels geht jedoch über eine quantitative Analyse hinaus: Die Darstellung der Ergebnisse, das heißt die Präsentation der Netztypen in der Form $\tau = \tau' \cup \omega$, wobei τ' als „fester Kern“ und ω als „ beliebiges Beiwerk“ aufgefasst werden kann, verdeutlicht welche Strukturen beziehungsweise booleschen Interaktionen wesentlich für die NP-Schwere der Probleme sind. So sind zum Beispiel τ-Sprachsimulation und τ -Realisierung für alle booleschen Netztypen NP-schwer, die wenigstens $\{\mathsf{nop}, \mathsf{inp}, \mathsf{set}\}$ oder $\{\mathsf{nop}, \mathsf{out}, \mathsf{res}\}$ enthalten. Für die nop-freien Netztypen sehen wir zum Beispiel, dass τ-Einbettung genau dann NP-vollständig ist, wenn $\mathsf{swap} \in \tau$ und $\tau \cap \mathfrak{save} \neq \emptyset$.

Ein wesentlicher Beitrag dieses Kapitels ist darüber hinaus eine sehr allgemeine Beweisstrategie, die für den Nachweis der jeweiligen NP-Schwere für alle betreffenden Netztypen und Entscheidungsprobleme funktioniert.

Für zukünftige Forschung bleibt zunächst die Aufgabe, die Charakterisierung der Komplexität von τ-Sprachsimulation und τ-Realisierung für alle booleschen Netztypen zu vervollständigen. Für Netztypen mit nop betrifft dies bezüglich τ-Sprachsimulation nur die Typen A5. Eine Schwierigkeit dieser Typen ist, dass sie das Lösen von EZTP nur „in eine Richtung “ erlauben: Nehmen wir an $P = z_0 \xrightarrow{e_1} \dots \xrightarrow{e_m} z_m$ ist ein Pfad in A. Wenn es nun $i < j \in \{0, \dots, m\}$ gibt, sodass $\neg z_i \xrightarrow{e_j}$, dann ist (z_i, e_j) nicht lösbar für diese Netztypen. Daher sind in diesem Kontext nur (azyklische) Transitionssysteme interessant, die für jedes Ereignis e_i des Pfades P erfüllen, dass e_i an allen Zuständen des Pfades mit kleinerem Index als i stattfindet. Ähnliche Überlegungen gelten für τ-Realisierung und die Netztypen A5.

Für τ-Realisierung und die Netztypen A11 überlegt man sich, dass aus der Argumentation für Satz 4.10 bereits folgt, dass nur Transitionssysteme interessant sind, für die jedes Ereignis an jedem Zustand stattfindet. Obwohl die beschriebenen Transitionssysteme zunächst sehr restriktiv erscheinen, sodass der Autor der vorliegenden Arbeit vermutet, dass die zugehörigen Entscheidungsprobleme polynomiell sind, ist ein Beweis für diese Vermutung nicht leicht zu finden.

Die Komplexität von τ-Sprachsimulation ist weiterhin für alle Netztypen ohne nop offen. Wenn τ ohne nop und swap ist, dann gibt es polynomielle Methoden für τ-Realisierung. Leider lassen sich die Verfahren nicht einfach für τ-Sprachsimulation

verwenden, denn sie basieren wesentlich darauf, dass nur stark zielorientierte Transitionssysteme als Eingaben interessant sind. Wie wir gesehen haben, ist diese Eigenschaft notwendig für die τ-ZTE. Für τ-Sprachsimulation ist diese Eigenschaft jedoch irrelevant und man findet hier implementierbare Transitionssysteme, die nicht stark zielorientiert sind. Es wäre daher sehr interessant zu erfahren, ob τ-Sprachsimulation für diese Netztypen schwerer zu entscheiden ist, als τ-Realisierung und τ-Einbettung.

Wir haben in diesem Kapitel gesehen, dass die untersuchten Probleme insbesondere für die in der Literatur diskutierten Netztypen NP-vollständig sind. Zukünftige Forschung könnte nun die Frage adressieren, ob die Synthese dieser Netze unter gewissen Restriktionen von anderer Komplexität ist. Dabei bieten sich offenbar wenigstens zwei Möglichkeiten an: Einerseits kann die Struktur des gesuchten Netzes eingeschränkt werden. Solche Einschränkungen werden wir in Kapitel 6 untersuchen. Andererseits kann das Eingabe-Verhalten, das heißt das Transitionssystem, eingeschränkt werden: Ein Transitionssystem A ist ein beschrifteter gerichteter Graph. Daher erscheint es natürlich für Einschränkungen den (Zustands-) Grad, das heißt die maximale Anzahl an eingehenden beziehungsweise ausgehenden Kanten an einem Zustand, sowie die Häufigkeit von Ereignissen, das heißt die maximale Anzahl an Kanten, die dasselbe Ereignis tragen, in den Blick zu nehmen. In [TRW18] beziehungsweise in [TR18] haben Rosenke und Wolf beziehungsweise Rosenke und jeweils der Autor der vorliegenden Arbeit (für alle Implementierungen) gezeigt, dass die Synthese elementarer Netz-Systeme ($\{\mathsf{nop}, \mathsf{inp}, \mathsf{out}\}$), selbst dann NP-vollständig bleibt, wenn A den Grad Eins hat, das heißt, jeder Zustand hat höchstens eine eingehende/ausgehende Kante, beziehungsweise wenn jedes Ereignis höchstens zwei Mal stattfindet. Kombiniert man beide Parameter, dann sind alle Probleme schwer, wenn der Grad Eins ist und jedes Ereignis höchstens drei Mal stattfindet beziehungsweise wenn der Grad Zwei beträgt und jedes Ereignis höchstens zwei Mal stattfindet. Dagegen ist τ-Einbettung polynomiell, wenn A Grad Eins hat und jedes Ereignis höchstens zwei Mal stattfindet. Die Komplexität von τ-Sprachsimulation und τ-Realisierung für Eingaben mit Grad Eins, in denen jedes Ereignis höchstens zwei Mal stattfindet, ist dagegen noch ungeklärt und –im Sinne der theoretischen Vollständigkeit– sicherlich eine interessante Forschungsfrage.

Die Untersuchung der Komplexität von τ-Sprachsimulation und τ-Realisierung für Transitionssysteme mit im Voraus fest eingeschränktem Grad hat der Autor der vorliegenden Arbeit in [Tre21a] für alle booleschen Netztypen mit nop vollständig geklärt. Generell zeigt sich, dass die Probleme auch für sehr kleinen Grad schwer bleiben, jedoch gibt es zum Beispiel bei $\{\mathsf{nop}, \mathsf{inp}, \mathsf{set}\}$ Unterschiede: bis einschließlich Grad Zwei bleibt Synthese schwer, aber für Grad Eins ist sie –im Unterschied zum Netztypen $\{\mathsf{nop}, \mathsf{inp}, \mathsf{out}\}$– in Polynomialzeit lösbar.

Die Komplexität von τ-Einbettung für Transitionssysteme eingeschränkten Grades für boolesche Netztypen (außer $\{\mathsf{nop}, \mathsf{inp}, \mathsf{out}\}$) ist jedoch offen. Ebenso wurde die (Einschränkung der) Häufigkeit von Ereignissen bisher nur für $\{\mathsf{nop}, \mathsf{inp}, \mathsf{out}\}$ untersucht. Die Komplexität der Synthese anderer boolescher Netze für derart eingeschränkte Eingaben bleibt Gegenstand für zukünftige Forschung.

Kapitel 5

A priori beschränkte Platz/Transitions-Netze

Es ist ein definierendes Merkmal der booleschen Netze, dass jeder Platz in jeder erreichbaren Markierung höchstens eine Marke enthalten darf. Ein Platz dieser per Definition *1-beschränkten* Netze kann deshalb als boolesche Bedingung interpretiert werden, die entweder wahr oder falsch sein kann. In natürlicher Weise lässt sich die *im Voraus fest gewählte Schranke* für die mögliche Anzahl an Marken auf einem Platz von 1 auf b für eine beliebige natürliche Zahl b erhöhen. Plätze repräsentieren nun Ressourcen, die von Transitionen produziert oder konsumiert werden. Die Anzahl der Marken, die ein Platz in einer Markierung enthält, entspricht dann der Anzahl der vorhandenen Einheiten der jeweiligen Ressource. In diesem Sinne werden Plätze nun auch als Zähler verstanden.

Dieses Kapitel basiert auf den Veröffentlichungen [Tre19a, Tre19b] und wir untersuchen in Abschnitt 5.1 zunächst die Komplexität der klassischen Syntheseprobleme, die auf Einbettung, Sprachsimulation und Realisierung abzielen, für zwei Varianten *a priori* beschränkter *Platz/Transitions-Netze*, kurz *P/T-Netze*, nämlich den sogenannten *b-beschränkten P/T-Netzen* und den *puren b-beschränkten P/T-Netzen*. Wir zeigen, dass die zugehörigen Entscheidungsprobleme für diese Netze für jedes $b \geq 1$ NP-vollständig sind[1]. Dies beantwortet eine Frage aus [SW17].

Die Synthese purer 1-beschränkter P/T-Netze ist für alle Implementierungen NP-vollständig. Schmitt [Sch96] hat mit den sogenannten Flip-Flop-Netzen (engl. *flip-flop nets*) die Erweiterung dieser Netze um die Gruppe $\mathbb{Z}_2$ der Addition ganzer Zahlen modulo 2 eingeführt. (In der Welt der booleschen Netze entsprechen die Flip-Flop-Netze dem Netztyp $\{\mathsf{nop}, \mathsf{inp}, \mathsf{out}, \mathsf{swap}\}$.) Durch Reduktion der Lösbarkeit von Trennungsproblemen auf das Lösen von Gleichungssystemen über dem endlichen Körper $\mathbb{F}_2$ hat Schmitt gezeigt: diese Erweiterung führt dazu, dass die Synthese der resultierenden Netze für alle Implementierungen in Polynomialzeit lösbar ist. Auf natürliche Weise wirft dies die Frage auf, ob die Erweiterung der (puren) b-beschränkten P/T-Netze um die Gruppe $\mathbb{Z}_{b+1}$ auch

[1] Für die puren 1-beschränkten P/T-Netze, den *elementary net systems*, und dem Ziel der Realisierung wurde dies ursprünglich von Badouel, Bernardinello und Darondeau gezeigt [BBD97].

Schranke	Problem	τ^b_{PT}	τ^b_{PPT}	$\tau^b_{\mathbb{Z}PT}$	$\tau^b_{\mathbb{Z}PPT}$
$b = 1$	τ-Einbettung	NPV	NPV	P	P
	τ-Sprachsimulation	NPV	NPV	P	P
	τ-Realisierung	NPV	NPV	P	P
$b \geq 2$	τ-Einbettung	NPV	NPV	P	P
	τ-Sprachsimulation	NPV	NPV	NPV	NPV
	τ-Realisierung	NPV	NPV	NPV	NPV

Abbildung 5.1: Die Komplexität der klassischen Syntheseprobleme für alle (puren) b-beschränkten P/T-Netze sowie deren Erweiterungen um die Gruppe $\mathbb{Z}_{b+1}$, wobei $b \in \mathbb{N}^+$ beliebig aber fest.

für $b \geq 2$ zu Netztypen führt, für welche die klassischen Syntheseprobleme in Polynomialzeit lösbar sind. In Abschnitt 5.2 beantworten wir diese Frage wie folgt: Synthese $\mathbb{Z}_{b+1}$-erweiterter (purer) b-beschränkter P/T-Netze, die auf Sprachsimulation oder Realisierung abzielt, ist für alle $b \geq 2$ NP-vollständig. Im Kontrast dazu ist Synthese für diese Netze für alle b in Polynomialzeit entscheidbar, wenn eine Einbettung angestrebt wird. Abbildung 5.1 zeigt eine tabellarische Übersicht der entsprechenden Charakterisierungen, die in diesem Kapitel bewiesen werden.

Die (puren) b-beschränkten P/T-Netze sowie ihre jeweiligen Erweiterungen um die Gruppe $\mathbb{Z}_{b+1}$ der Addition ganzer Zahlen modulo $b + 1$ lassen sich mittels Netztypen charakterisieren, die im Folgenden definiert werden. Dabei ist klar, dass Synthese trivial ist, wenn Plätze eines angestrebten Netzes keine Marken enthalten dürfen, das heißt, wenn $b = 0$. Solange nicht ausdrücklich anders vereinbart, sei deshalb im weiteren Verlauf dieses Kapitels $b \in \mathbb{N}^+$ beliebig aber fest.

Definition 5.1 (τ^b_{PT})**.** Der Netztyp $\tau^b_{PT} = (Z_{\tau^b_{PT}}, E_{\tau^b_{PT}}, \delta_{\tau^b_{PT}})$ der *b-beschränkten P/T-Netze* ist gegeben durch die Zustandsmenge $Z_{\tau^b_{PT}} = \{0, \ldots, b\}$, die Ereignismenge $E_{\tau^b_{PT}} = \{(m, n) \mid 0 \leq m, n \leq b\}$ sowie der Übergangsfunktion $\delta_{\tau^b_{PT}}$, die für alle $z, z' \in Z_{\tau^b_{PT}}$ und alle $(m, n) \in E_{\tau^b_{PT}}$ wie folgt definiert ist:

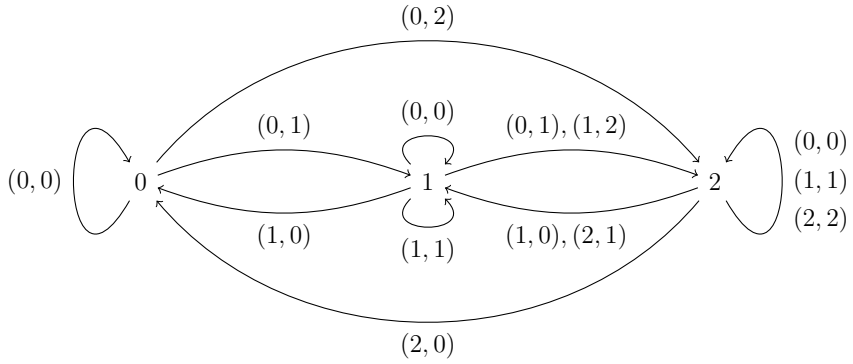
$$\delta_{\tau^b_{PT}}(z, (m, n)) = z' \iff (z \leq m \text{ und } z' = z - m + n)$$

Bemerkung 5.1. Für die b-beschränkten P/T-Netze entspricht der Flusswert $f(p, t) = (0, 0)$ dem Flusswert $f(p, t) = \mathsf{nop}$ der booleschen Netze und beschreibt damit Unabhängigkeit zwischen p und t.

Der Netztyp purer b-beschränkter P/T-Netze geht aus τ^b_{PT} hervor, indem Ereignisse (m, n) ausgeschlossen werden, für die sowohl m als auch n positiv sind. Auf Netzebene bedeutet dies, dass ein Platz p nicht gleichzeitig ein Eingabe- und Ausgabeplatz einer Transition t sein kann, das heißt, das Schalten von t kann nicht Marken auf p produzieren und von p konsumieren.

Definition 5.2 (τ^b_{PPT})**.** Der Netztyp $\tau^b_{PPT} = (Z_{\tau^b_{PPT}}, E_{\tau^b_{PPT}}, \delta_{\tau^b_{PPT}})$ der *puren b-beschränkten P/T-Netze* ist gegeben durch $Z_{\tau^b_{PPT}} = \{0, \ldots, b\}$ und $E_{\tau^b_{PPT}} = E_{\tau^b_{PT}} \setminus \{(m, n) \mid 1 \leq m, n \leq b\}$ sowie der Übergangsfunktion $\delta_{\tau^b_{PPT}}$, die für alle $z \in Z_{\tau^b_{PPT}}$ und alle $e \in E_{\tau^b_{PPT}}$ durch $\delta_{\tau^b_{PPT}}(z, e) = \delta_{\tau^b_{PT}}(z, e)$ definiert ist.

Beispiel 5.1. Die folgende Abbildung skizziert τ^2_{PT}. Durch Kommata getrennte Ereignisse beschriften verschiedene Kanten. Wenn die Ereignisse $(1, 1)$, $(1, 2)$, $(2, 1)$ und $(2, 2)$ sowie die zugehörigen Kanten entfernt werden, dann erhält man τ^2_{PPT}.



Der folgende Netztyp $\tau^b_{\mathbb{Z}PT}$ der $\mathbb{Z}_{b+1}$-erweiterten b-beschränkten P/T-Netze erweitert den Netztyp τ^b_{PT} um die Addition modulo $(b+1)$[2]. Man beachte, dass das neutrale Element 0 der Gruppe $\mathbb{Z}_{b+1}$ der Addition ganzer Zahlen modulo $(b+1)$ die Funktion des Ereignisses $(0, 0)$ von τ^b_{PT} übernimmt, welches Unabhängigkeit zwischen Plätzen und Transitionen beschreibt.

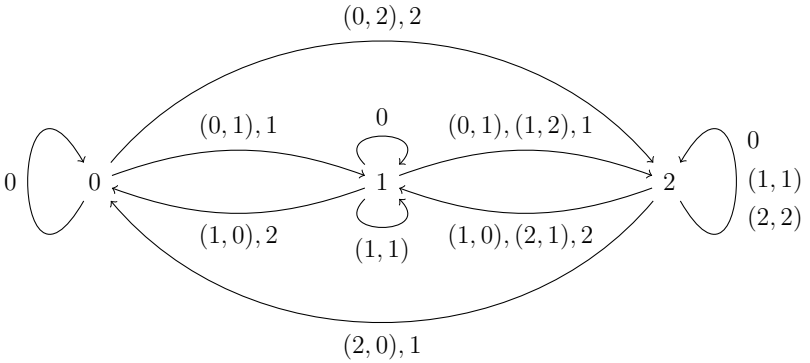
Definition 5.3 ($\tau^b_{\mathbb{Z}PT}$)**.** Der Netztyp $\tau^b_{\mathbb{Z}PT} = (Z_{\tau^b_{\mathbb{Z}PT}}, E_{\tau^b_{\mathbb{Z}PT}}, \delta_{\tau^b_{\mathbb{Z}PT}})$ der *$\mathbb{Z}_{b+1}$-erweiterten b-beschränkten P/T-Netze* ist gegeben durch $Z_{\tau^b_{\mathbb{Z}PT}} = \{0, \ldots, b\}$ und $E_{\tau^b_{\mathbb{Z}PT}} = (E_{\tau^b_{PT}} \setminus \{(0, 0)\}) \cup \{0, \ldots, b\}$ sowie der Übergangsfunktion $\delta_{\tau^b_{\mathbb{Z}PT}}$, die für alle $z \in \{0, \ldots, b\}$ und alle $e \in E_{\tau^b_{\mathbb{Z}PT}}$ wie folgt definiert ist: Wenn $e \in E_{\tau^b_{PT}}$, dann $\delta_{\tau^b_{\mathbb{Z}PT}}(z, e) = \delta_{\tau^b_{PT}}(z, e)$, andernfalls $\delta_{\tau^b_{\mathbb{Z}PT}}(z, e) = z + e \bmod (b + 1)$.

Der folgende Netztyp $\tau^b_{\mathbb{Z}PPT}$ der *puren* $\mathbb{Z}_{b+1}$-erweiterten b-beschränkten P/T-Netze geht schließlich aus $\tau^b_{\mathbb{Z}PT}$ durch Streichen der Ereignisse (m, n) mit $m > 0$ und $n > 0$ hervor:

Definition 5.4 ($\tau^b_{\mathbb{Z}PPT}$)**.** Der Netztyp $\tau^b_{\mathbb{Z}PPT} = (Z_{\tau^b_{\mathbb{Z}PPT}}, E_{\tau^b_{\mathbb{Z}PPT}}, \delta_{\tau^b_{\mathbb{Z}PPT}})$ der *puren $\mathbb{Z}_{b+1}$-erweiterten b-beschränkten P/T-Netze* ist gegeben durch $Z_{\tau^b_{\mathbb{Z}PPT}} = \{0, \ldots, b\}$ und $E_{\tau^b_{\mathbb{Z}PPT}} = E_{\tau^b_{\mathbb{Z}PT}} \setminus \{(m, n) \mid 1 \leq m, n \leq b\}$ sowie der Übergangsfunktion $\delta_{\tau^b_{\mathbb{Z}PPT}}$, die für alle $z \in \{0, \ldots, b\}$ und alle $e \in E_{\tau^b_{\mathbb{Z}PPT}}$ durch $\delta_{\tau^b_{\mathbb{Z}PPT}}(z, e) = \delta_{\tau^b_{\mathbb{Z}PT}}(z, e)$ definiert ist.

[2]Jetzt und im Folgenden verwenden wir das Zeichen = auch im Kontext von Zahlenkongruenzen: für eine natürliche Zahl c und ganze Zahlen a, b schreiben wir $a = b \bmod c$ (sprich a ist kongruent zu b modulo c), wenn c ein Teiler von $a - b$ ist, das heißt, wenn es eine ganze Zahl m gibt, sodass $c \cdot m = a - b$.

Beispiel 5.2. Die folgende Abbildung skizziert $\tau^2_{\mathbb{Z}PT}$. Das Entfernen von $(1,1),(1,2),(2,1)$ und $(2,2)$ sowie den zugehörigen Kanten liefert $\tau^2_{\mathbb{Z}PPT}$.



Gemäß des folgenden Lemmas sind die klassischen Syntheseprobleme für alle eingeführten b-beschränkten Netztypen in NP:

Lemma 5.1. *Wenn* $\tau \in \{\tau^b_{PT}, \tau^b_{PPT}, \tau^b_{\mathbb{Z}PT}, \tau^b_{\mathbb{Z}PPT}\}$*, dann sind* τ-Einbettung *und* τ-Sprachsimulation *sowie* τ-Realisierung *in* NP.

Beweis. Gemäß Definition 2.21 beziehungsweise Definition 2.22 hat ein gegebenes Transitionssystem $A = (Z, E, \delta, \iota)$ höchstens $|Z|^2$ ZTP beziehungsweise $|Z| \cdot |E|$ EZTP. Wenn A die jeweilige Eigenschaft hat, dann gibt es dafür also einen Zeugen $\mathcal{R}$ mit höchstens $|Z|^2 + |Z| \cdot |E|$ τ-Regionen. Weil b im Voraus fest gewählt ist, sodass die Größe einer Region polynomiell in der Größe von A ist, kann $\mathcal{R}$ von einer nicht-deterministischen Turing-Maschine in Polynomialzeit geraten und anschließend in deterministischer Polynomialzeit verifiziert werden. □

Boolesche und 1-beschränkte Netztypen: Zusammenhänge

Bei den booleschen Netzen fassen wir Plätze vordergründig als boolesche Bedingungen auf, die wahr oder falsch sein können, je nachdem ob der entsprechende Platz markiert ist oder nicht. Das Schalten von Transitionen kann dabei als das Verändern (inp, out, set, res, swap) oder aber das Prüfen (used, free) eines Wahrheitswertes (des Platzes) aufgefasst werden. Im Gegensatz dazu steht bei den b-beschränkten Netzen das Zählen (Platzinhalte) und Rechnen (Schalten von Transitionen) im Vordergrund. Nichtsdestotrotz reduziert sich die Zustandsmenge der eingeführten Netztypen für $b = 1$ zu $\{0, 1\}$, sodass die entsprechenden Netztypen gemäß Definition 4.1 boolesche Netztypen sind. Da wir in Kapitel 4 (bis auf Umbenennung) alle booleschen Netztypen eingeführt haben, müssen die Netztypen τ^1_{PPT} und τ^1_{PT} sowie ihre Gruppenerweiterungen boolesche (kongruente) Entsprechungen haben. Folglich gelten die Komplexitäts-Resultate für die (gruppenerweiterten) 1-beschränkten Netztypen des aktuellen Kapitels gemäß Satz 2.2 unmittelbar für ihre booleschen Entsprechungen. Tatsächlich werden wir diese Eigenschaft ausnutzen, um die in Kapitel 4 schuldig

gebliebenen Beweise für Satz 4.2, Lemma 4.4, Lemma 4.6 sowie Satz 4.13 wie angekündigt nachzuliefern. Der folgende Fakt liefert dazu die formelle Grundlage und es wird im Verlauf dieses Kapitel an den bedeutsamen Stellen jeweils explizit darauf hingewiesen:

Fakt 5.1. *Es gelten die folgenden Kongruenzen:*

1. $\tau^1_{PPT} \equiv \{\mathsf{nop}, \mathsf{inp}, \mathsf{out}\}$ *und* $\tau^1_{PT} \equiv \{\mathsf{nop}, \mathsf{inp}, \mathsf{out}, \mathsf{used}\}$,
2. $\tau^1_{\mathbb{Z}PPT} \equiv \{\mathsf{nop}, \mathsf{inp}, \mathsf{out}, \mathsf{swap}\}$ *und* $\tau^1_{\mathbb{Z}PT} \equiv \{\mathsf{nop}, \mathsf{inp}, \mathsf{out}, \mathsf{swap}, \mathsf{used}\}$.

Beweis. Die Kongruenzen sind gerechtfertigt durch die Abbildungen φ und ψ, für die gilt: $\varphi(0) = 0$ und $\varphi(1) = 1$ sowie $\psi((0,0)) = \mathsf{nop}$ beziehungsweise $\psi(0) = \mathsf{nop}$, $\psi((1,0)) = \mathsf{inp}$, $\psi((0,1)) = \mathsf{out}$, $\psi((1,1)) = \mathsf{used}$ und $\psi(1) = \mathsf{swap}$. □

Einige technische Hilfsmittel: Definitionen und Ergebnisse

Definition 5.5 (sig^- und sig^+ und $|sig|$)**.** Sei $\tau \in \{\tau^b_{PT}, \tau^b_{PPT}, \tau^b_{\mathbb{Z}PT}, \tau^b_{\mathbb{Z}PPT}\}$ und $A = (Z, E, \delta, \iota)$. Für eine τ-Region $R = (sup, sig)$ von A und jedes $e \in E$ definieren wir $sig^-(e) = m$ und $sig^+(e) = n$ und $|sig(e)| = 0$, falls $e = (m,n) \in \{0, \ldots, b\}^2$, sowie $sig^-(e) = sig^+(e) = 0$ und $|sig(e)| = sig(e)$, falls $sig(e) \in \{0, \ldots, b\}$.

Lemma 5.2. *Sei* $\tau \in \{\tau^b_{PT}, \tau^b_{PPT}, \tau^b_{\mathbb{Z}PT}, \tau^b_{\mathbb{Z}PPT}\}$ *und* $A = (Z, E, \delta, \iota)$ *ein Transitionssystem.*

1. *Zwei Abbildungen* $sup : Z \to Z_\tau$ *und* $sig : E \to E_\tau$ *definieren genau dann eine* τ*-Region* $R = (sup, sig)$*, wenn für jeden Pfad* $z_0 \xrightarrow{e_1} \ldots \xrightarrow{e_n} z_n$ *von* A *für alle* $i \in \{1, \ldots, n\}$ *folgende Bedingungen gelten:*
   (a) *wenn* $sig(e) \in \{0, \ldots, b\}^2$*, dann* $sig^-(e_i) \geq sup(z_{i-1})$ *und* $sup(z_i) = sup(z_{i-1}) - sig^-(e_i) + sig^+(e_i) + |sig(e_i)|$
   (b) *wenn* $sig(e) \in \{0, \ldots, b\}$*, dann* $sup(z_i) = sup(z_{i-1}) - sig^-(e_i) + sig^+(e_i) + |sig(e_i)| \mod (b+1)$
2. *Wenn* $z_0 \xrightarrow{e} \ldots \xrightarrow{e} z_b$ *ein Pfad in* A *ist, sodass* $z_i \neq z_{i+1}$ *und* $z_i \xrightarrow{e} z_{i+1}$ *für alle* $i \in \{0, \ldots, b-1\}$*, dann gilt für eine* τ*-Region* $R = (sup, sig)$ *genau dann* $sig(e) = (m,n)$ *mit* $m \neq n$*, wenn* $(m,n) \in \{(1,0),(0,1)\}$*. Wenn* $sig(e) = (1,0)$*, dann gilt* $sup(z_0) = b$ *und* $sup(z_b) = 0$ *und falls* $sig(e) = (0,1)$*, dann gilt* $sup(z_0) = 0$ *und* $sup(z_b) = 1$.
3. *Wenn* $\tau \in \{\tau^b_{PT}, \tau^b_{PPT}\}$ *und* $R = (sup, sig)$ *eine* τ*-Region von* A *ist sowie* $e \in E$ *ein Ereignis mit einem Ursprung* $z \in Z$ *und einem Ziel* $z' \in Z$*, das heißt* $z \xrightarrow{e}$ *und* $\xrightarrow{e} z'$*, sodass* $sup(z) = sup(z') = b$ *oder* $sup(z) = sup(z') = 0$*, dann gilt* $sig^-(e) = sig^+(e)$*. Insbesondere ist dann* $sup(q) = sup(q')$ *für jede Kante* $q \xrightarrow{e} q'$ *von* A.

Beweis. (1): Die Aussage folgt direkt aus den Definitionen von τ und τ-Regionen sowie sig^+, sig^- und $|sig|$.

(2): „$\Rightarrow$“: Offenbar gilt $m \neq n$, falls $(m,n) \in \{(1,0),(0,1)\}$.

„$\Leftarrow$“: Gelte $sig(e) = (m,n)$ mit $m \neq n$. Dies impliziert offenbar $|n-m| \neq 0$. Aus (1.a) folgt induktiv, dass $sup(z_b) = sup(z_0) + b \cdot (n-m)$. Angenommen, es gilt $|n-m| > 1$, das heißt, entweder $n-m < -1$ oder $n-m > 1$. Der erste Fall impliziert $b \cdot (n-m) < -b$ und damit $sup(z_b) < 0$. Der zweite Fall impliziert $b \cdot (n-m) > b$ und somit $b < sup(z_b)$. Beide Fälle widersprechen $sup(z_b) \in \{0,\ldots,b\}$. Somit erhalten wir $|n-m| = 1$. Wir zeigen nun: wenn $n < m$, dann $m = 1$ und $n = 0$. Mit $n \leq m-1$ und $sup(z_0) \leq b$ sowie (1.a) erhalten wir die folgende Abschätzung:

$$sup(z_{b-1}) = sup(z_0) + (b-1)(n-m) \;\leq\; b + (b-1)(m-1-m) = 1$$

Weil R eine Region ist, gilt $sup(z_{b-1}) \xrightarrow{(m,n)} sup(z_b)$. Dies impliziert $n < m \leq sup(z_{b-1}) = 1$, das heißt $m = 1$ und $n = 0$. Analog folgt $(m,n) = (0,1)$ aus der Annahme $m < n$ und insgesamt haben wir $(m,n) \in \{(1,0),(0,1)\}$, wenn $m \neq n$.

Beweisen wir noch die Aussage über die Support-Werte: Wenn $(m,n) = (1,0)$, dann erhalten wir $sup(z_b) = sup(z_0) - b$ mit (1) und aus $0 \leq sup(z_b)$ sowie $sup(z_0) \leq b$ folgt $sup(z_0) = b$ und $sup(z_b) = 0$. Analog folgen $sup(z_0) = 0$ und $sup(z_b) = b$ aus $(m,n) = (0,1)$.

(3): Wir untersuchen zunächst den Fall $sup(z) = sup(z') = b$. Für $p,q \in \{0,\ldots,b\}$ und $(m,n) \in E_\tau$ gilt nach Definition genau dann $\delta(p,(m,n)) = q$, wenn $q = p - m + n$ und $m \leq p$. Aus $b \xrightarrow{sig(e)}$ folgt deshalb $sig^-(e) \geq sig(e)^+$, da sonst $\delta_\tau(b, sig(e)) > b$. Weil $\xrightarrow{sig(e)} sup(z')$ mit $sup(z') = b$, gibt es ein $y \in \{0,\ldots,b\}$, sodass $y \xrightarrow{sig(e)} b$. Dies bedeutet $b = y - sig^-(e) + sig^+(e)$ beziehungsweise $sig^-(e) - sig^+(e) = y - b$. Aus $sig^-(e) \geq sig(e)^+$ folgt $sig^-(e) - sig^+(e) \geq 0$ und damit $y - b \geq 0$. Mit $y \in \{0,\ldots,b\}$ erhalten wir $y = b$ und somit $sig^-(e) - sig^+(e) = 0$. Dies impliziert $sig^-(e) = sig^+(e)$.

Wenn nun $q \xrightarrow{e} q' \in A$, dann gilt $sup(q') = sup(q) - sig^-(e) + sig^+(e)$ und mit $sig^-(e) = sig^+(e)$ folgt offensichtlich $sup(q') = sup(q)$. Der Fall $sup(z) = sup(z') = 0$ wird analog bewiesen. □

Die Ergebnisse zur NP-Vollständigkeit dieses Kapitels werden gemäß der in Kapitel 3 vorgestellten Reduktions-Strategie geführt: ausgehend von einer Eingabe $(\mathfrak{U}, M)$ werden eine Reihe von Gadgets konstruiert –die eigentlichen Funktionsträger der Reduktion– und anschließend zu einem Transitionssystem A_τ verbunden. Die Vielzahl an verschiedenen Netztypen und deren unterschiedliche Interaktionen in Kapitel 4 haben eine einheitliche Art der Verbindung der Gadgets für die verschiedenen Konstruktionen und Netztypen unmöglich gemacht. Für die Netztypen in diesem Kapitel und den zugehörigen Konstruktionen ist die Situation erheblich günstiger. Genauer gesagt erlauben die Netztypen aus τ^b_{PPT} und τ^b_{PT} sowie $\tau^b_{\mathbb{Z}PPT}$ und $\tau^b_{\mathbb{Z}PT}$ jeweils die gleichen Verbindungsarten. Wir werden diesen Umstand ausnutzen, um den Teil der Beweisführung zu vereinfachen, der, ausgehend von einem Eins-in-drei-Modell von $(\mathfrak{U}, M)$, die erfragten Trennungseigenschaften

des Transitionssystems A_τ nachweist. Im Wesentlichen basiert diese Vereinfachung auf der Möglichkeit, nur solche EZTP und ZTP zu betrachten, die von den jeweiligen Gadgets herrühren und nicht von den Verbindungs-Kanten. Die folgenden Definitionen dienen als Grundlage für diese Herangehensweise und werden dann in Abschnitt 5.1 beziehungsweise in Abschnitt 5.2 für die jeweiligen Netztypen im zugehörigen Kontext verwendet.

Die folgende Definition führt den Begriff der Basis ein. So eine Basis ist –salopp gesagt– eine lose Ansammlung von Transitionssystemen. Die Gadgets der Reduktionen werden später diese Transitionssysteme sein. Das Ziel ist, so eine Basis wie ein großes Transitionssystem behandeln zu können. Dieses Transitionssystem entspricht später der (je nach Netztyp) einheitlichen Verbindung der Gadgets durch Verbindungskanten.

Definition 5.6 (Basis)**.** Wenn $A_0 = (Z_0, E_0, \delta_0, \iota_0), \ldots, A_n = (Z_n, E_n, \delta_n, \iota_n)$ jeweils Transitionssysteme mit paarweise disjunkten Zustandsmengen sind, dann nennen wir das Tupel $(A_0, \ldots, A_n)$ eine *Basis*. Wir bezeichnen $\underline{Z} = \bigcup_{i=0}^{n} Z_i$ als Zustandsbasis und $\underline{E} = \bigcup_{i=0}^{n} E_i$ als Ereignisbasis von $(A_0, \ldots, A_n)$.

Der folgende Begriff des Rumpfes ist das Regionen-Äquivalent für Basen. Wir streben an, den Rumpf R einer Basis als Region R des durch die Basis implizierten Transitionssystems auffassen zu können, das durch die Verbindung der (einzelnen) Transitionssysteme der Basis entsteht. Der einzige Unterschied eines Rumpfes gegenüber einer Region ist dabei, dass der Rumpf genau für die Ereignisse und (zusätzlichen Zustände) der Verbindungskanten nicht definiert ist.

Definition 5.7 (Rumpf)**.** Sei $\tau = (Z_\tau, E_\tau, \delta_\tau)$ ein Netztyp und $(A_0, \ldots, A_n)$ eine Basis. Ein Paar $R = (sup, sig)$ von Abbildungen $sup : \underline{Z} \to Z_\tau$ und $sig : \underline{E} \to E_\tau$ heißt τ-(Regionen-) *Rumpf* für A, wenn für alle $i \in \{0, \ldots, n\}$ die folgende Implikation erfüllt ist:

wenn $z \xrightarrow{e} z' \in A_i$, dann $sup(z) \xrightarrow{sig(e)} sup(z') \in \tau$.

Schließlich definieren die ZTP und die EZTP, die von den Bestandteilen einer Basis herrühren, die „interessanten“ Trennungsprobleme, des durch die Basis implizierten Transitionssystems:

Definition 5.8 (ZTP und EZTP von Basen)**.** Es sei $B = (A_0, \ldots, A_n)$ eine Basis. Ein Paar (e, z) aus Ereignis $e \in \underline{E}$ und Zustand $z \in \underline{Z}$ mit $\neg z \xrightarrow{e}$ definiert ein *Ereignis-Zustands-Trennungsproblem* (kurz *EZTP*) (e, z) von B. Für $i \in \{0, \ldots, n\}$ definieren zwei verschiedene Zustände z und z' desselben Transitionssystems A_i ein *Zustands-Trennungs-Problem* (kurz *ZTP*) (z, z') von B. Es heißen (e, z) beziehungsweise (z, z') *τ-lösbar*, wenn es einen τ-Rumpf $R = (sup, sig)$ von B mit $\neg sup(z) \xrightarrow{sig(e)}$ beziehungsweise $sup(z) \neq sup(z')$ gibt.

Bemerkung 5.2. In naheliegender Weise verwenden wir im weiteren Verlauf der vorliegenden Arbeit auch die restlichen Begriffe und Definitionen „rund um Regionen“ wie zum Beispiel *Lösbarkeit von Ereignissen* oder die Eigenschaften *EZTE* und *ZTP* etc. für Basen und τ-Rümpfe, ohne sie nochmal explizit zu definieren.

Die weitere Struktur dieses Kapitels ist wie folgt: Im folgenden Abschnitt 5.1 untersuchen wir die klassischen Syntheseprobleme für die (puren) b-beschränkten P/T-Netze und zeigen, dass diese für alle $b \geq 1$ NP-vollständig sind. Anschließend widmen wir uns den gruppenerweiterten (puren) b-beschränkten P/T-Netzen: In Abschnitt 5.2 wird zunächst gezeigt, dass Synthese die auf Sprachsimulation beziehungsweise Realisierung abzielt, für diese Netze NP-vollständig ist, wenn $b \geq 2$. Alle Ergebnisse zur NP-Vollständigkeit basieren auf der in Kapitel 3 vorgestellten Strategie.

Anschließend wird argumentiert, dass für die gruppenerweiterten Netze in Polynomialzeit entschieden werden kann, ob eine Einbettung existiert und schließlich wird bewiesen, dass über die Existenz einer Sprachsimulation beziehungsweise Realisierung ebenso in Polynomialzeit entscheiden werden kann, wenn $b = 1$ ist. Diese Resultate basieren auf einer Erweiterung des Ansatzes von Schmitt [Sch96], der die Lösbarkeit von Trennungsproblemen auf die Lösbarkeit von Gleichungssystem über $\mathbb{Z}_{b+1}$ reduziert.

5.1 b-beschränkte Netztypen

Der folgende Satz stellt das Hauptresultat dieses Abschnitts dar:

Satz 5.1. *Wenn $b \in \mathbb{N}^+$ und $\tau \in \{\tau^b_{PT}, \tau^b_{PPT}\}$, dann sind die Entscheidungsprobleme τ-*Einbettung *und τ-*Sprachsimulation *sowie τ-*Realisierung *NP-vollständig.*

Bevor wir uns dem Beweis des Satzes widmen, sei an dieser Stelle nochmals bemerkt, dass die Netztypen τ^1_{PPT} und $\{\mathsf{nop}, \mathsf{inp}, \mathsf{out}\}$ sowie τ^1_{PT} und $\{\mathsf{nop}, \mathsf{inp}, \mathsf{out}, \mathsf{used}\}$ gemäß Fakt 5.1 kongruent sind. Die schuldig gebliebenen Beweise der Aussagen aus Satz 4.2 und Satz 4.13 bezüglich dieser Netztypen sind deshalb unmittelbare Folgerungen aus Satz 2.2 und Satz 5.1.

Der Rest dieses Abschnitts ist dem Beweis von Satz 5.1 gewidmet und wird gemäß der in Kapitel 3 vorgestellten Beweisstrategie geführt. Die konkreten Ausprägungen des zu konstruierenden Transitionssystems A_τ hängen dabei selbstverständlich von τ und dem konkret untersuchten Entscheidungsproblem ab. Allen Konstruktionen ist jedoch gemeinsam, dass ihre Gadgets in Übereinstimmung mit Definition 2.25 als lineare Transitionssysteme betrachtet werden können. A_τ ist dann ebenfalls ein lineares Transitionssystem, nämlich die lineare Verbindung dieser Gadgets:

Definition 5.9 (Lineare Verbindung)**.** Es sei $(A_0, \ldots, A_n)$ eine Basis mit Zustandsbasis $\underline{Z}$ und Ereignisbasis $\underline{E}$, wobei $A_i = (Z_i, E_i, \delta_i, \iota_i)$ für alle $i \in \{0, \ldots, n\}$ ein lineares Transitionssystem mit Terminalzustand t_i sei. Es seien weiterhin $\bot = \{\bot_1, \ldots, \bot_n\}$ beziehungsweise $\ominus = \{\ominus_1, \ldots, \ominus_n\}$ sowie $\oplus = \{\oplus_1, \ldots, \oplus_n\}$ Mengen paarweise verschiedener Zustände beziehungsweise Ereignisse, die jeweils zu $\underline{Z}$ und $\underline{E}$ disjunkt sind. Die *lineare Verbindung von* $(A_0, \ldots, A_n)$ ist das Transitionssystem $A = (Z, E, \delta, \iota)$ mit $Z = \underline{Z} \cup \bot$ und $E = \underline{E} \cup \ominus \cup \oplus$ sowie $\iota = \iota_0$, wobei für alle $e \in E$ und alle $z \in Z$ die Übergangsfunktion

δ wie folgt definiert ist:

$$\delta(z,e) = \begin{cases} \delta_i(z,e), & \text{falls } z \in Z_i \text{ und } e \in E_i \text{ für } i \in \{0,\ldots,n\} \\ \perp_{i+1}, & \text{falls } z = t_i \text{ und } e = \ominus_{i+1} \text{ für } i \in \{0,\ldots,n-1\} \\ \iota_i, & \text{falls } z = \perp_i \text{ und } e = \oplus_i \text{ für } i \in \{1,\ldots,n\} \\ \text{undefiniert}, & \text{sonst} \end{cases}$$

Bemerkung 5.3. Die lineare Verbindung lässt sich wie folgt skizzieren:

$$A = \quad A_0 \xrightarrow{\ominus_1} \perp_1 \xrightarrow{\oplus_1} A_1 \xrightarrow{\ominus_2} \perp_2 \xrightarrow{\oplus_2} \quad \cdots \quad \xrightarrow{\ominus_n} \perp_n \xrightarrow{\oplus_n} A_n$$

Lemma 5.3. *Es sei $\tau \in \{\tau^b_{PT}, \tau^b_{PPT}\}$ und $B = (A_0,\ldots,A_n)$ eine Basis linearer Transitionssysteme, wobei t_i für alle $i \in \{0,\ldots,n\}$ der Terminalzustand von $A_i = (Z_i, E_i, \delta_i, \iota_i)$ sei. Außerdem gebe es für jedes Ereignis $e \in \underline{E}$ einen Zustand $z \in \underline{Z}$ mit $\neg z \xrightarrow{e}$. Darüber hinaus sei $A = (Z, E, \delta, \iota_0)$ die lineare Verbindung von B.*

1. *Wenn jedes EZTP von B τ-lösbar ist, dann hat A die τ-EZTE.*
2. *Wenn jedes ZTP von B τ-lösbar ist, dann hat A die τ-ZTE.*

Beweis. Sei $R' = (sup', sig')$ ein τ-Rumpf von B, der ein (beliebiges aber festes) EZTP (a,q) beziehungsweise ZTP (p,q) von B löst, das heißt, $\neg sup(q) \xrightarrow{sig(a)}$ beziehungsweise $sup(p) \neq sup(q)$. Es lässt sich R' zu einer τ-Region $R = (sup, sig)$ von A fortsetzen, indem für alle $z \in Z$, für alle $e \in E$, für alle $i \in \{0,\ldots,n-1\}$ und alle $j \in \{1,\ldots,n\}$ die Werte $sup(z)$ und $sig(e)$ wie folgt definiert werden:

$$sup(z) = \begin{cases} sup'(z), & \text{falls } z \in \underline{Z} \\ sup'(q), & \text{falls } z \in \{\perp_1,\ldots,\perp_n\} \end{cases}$$

$$sig(e) = \begin{cases} sig'(e), & \text{falls } e \in \underline{E} \\ (sup'(t_i) - sup'(q), 0), & \text{falls } e = \ominus_{i+1} \text{ und } sup(t_i) > sup(q) \\ (0, sup'(q) - sup'(t_i)), & \text{falls } e = \ominus_{i+1} \text{ und } sup(t_i) \leq sup(q) \\ (0, sup'(\iota_j) - sup'(q)), & \text{falls } e = \oplus_j \text{ und } sup(\iota_j) > sup(q) \\ (sup'(q) - sup'(\iota_j), 0), & \text{falls } e = \oplus_j \text{ und } sup(\iota_j) \leq sup(q) \end{cases}$$

(1): R erfüllt offenbar $sup(z) \xrightarrow{sig(e)} sup(z') \in \tau$, wenn $z \xrightarrow{e} z' \in A$. Darüber hinaus gilt $\neg sup(z) \xrightarrow{sig(a)}$ für alle $z \in \{\perp_1,\ldots,\perp_n\}$. Aus der Beliebigkeit von (a,q) und dem Fakt, dass es für alle $e \in \underline{E}$ ein zu lösendes EZTP in B gibt, folgt bereits die τ-Lösbarkeit für alle EZTP (e,z) von A mit $e \in \underline{E}$. Es ist leicht zu sehen, dass sich alle EZTP (e,z) mit $e \in \{\oplus_1,\ldots,\oplus_n\}$ lösen lassen. Ebenso verifiziert man leicht für alle $i \in \{0,\ldots,n-1\}$, dass das EZTP $(\ominus_{i+1}, z)$ für alle $z \in E \setminus Z_i$ lösbar ist. Es gilt die EZTP $(\ominus_{i+1}, z)$ mit

$z \in Z_i$ zu diskutieren: Seien $i \in \{1, \ldots, n\}$ und $p \in Z_i \setminus \{t_i\}$ beliebig aber fest. (Für t_i ist wegen $t_i \xrightarrow{\ominus_{i+1}}$ nichts zu zeigen.) Es bleibt zu zeigen, dass $(\ominus_{i+1}, p)$ τ-lösbar ist.

Die folgenden Überlegungen führen zum Gewünschten: Weil es für jedes EZTP (e, z) von A_i einen Rumpf (sup', sig') mit $\neg sup'(z) \xrightarrow{sig'(e)}$ gibt, hat A_i die τ-EZTE. Dies gilt, weil die Beschränkung eines Rumpfes auf A_i offenbar eine τ-Region von A_i definiert. Insbesondere gibt es einen Zeugen $\mathcal{R}$ für die τ-EZTE von A_i, der eben aus der Beschränkung solcher Rümpfe hervorgeht. Weil A_i linear ist, gilt mit Lemma 2.5, dass $\mathcal{R}$ auch ein Zeuge für die τ-ZTE von A_i ist. Daraus folgt, dass es einen Rumpf (sup', sig') von A mit $sup'(p) \neq sup'(t_i)$ gibt. Dieser Rumpf lässt sich wie zu einer τ-Region $R = (sup, sig)$ von A fortsetzen, die $(\ominus_{i+1}, p)$ löst: Bis auf $sup(\bot_{i+1}), sig(\ominus_{i+1})$ und $sig(\oplus_{i+1})$ sind sup und sig wie oben definiert; wenn $sup'(p) > sup'(t_i)$, dann definieren wir $sup(\bot_{i+1}) = b$ und $sig(\ominus_{i+1}) = (0, b - sup(t_i))$ sowie $sig(\oplus_{i+1}) = (b - sup(t_{i+1}), 0)$; andernfalls, das heißt $sup'(p) < sup'(t_i)$, definieren wir $sup(\bot_{i+1}) = 0$ und $sig(\ominus_{i+1}) = (sup(t_i), 0)$ sowie $sig(\oplus_{i+1}) = (0, sup(t_{i+1}))$. Weil i beliebig war, folgt insgesamt, dass A die τ-EZTE hat.

(2): Sei $i \in \{0, \ldots, n\}$ beliebig aber fest. Zunächst folgt aus der Beliebigkeit von p und q und der oben definierten Region R, dass (p, q) für alle $p \neq q \in Z_i$ τ-lösbar ist. Da jedes Ereignis $e \in \ominus \cup \oplus$ genau einmal in A stattfindet, ist klar, dass es eine τ-Region $R = (sup, sig)$ von A gibt, deren Support sup genau die Zustände von A_i auf 1 abbildet und alle anderen auf 0. Es folgt insgesamt, dass sich jedes ZTP (p, q) mit $p \neq q \in \underline{Z}$ lösen lässt. Analog sieht man, dass es für jeden Zustand $p \in \bot$ eine τ-Region $R = (sup, sig)$ von A gibt, deren Support sup genau p auf 1 abbildet und alle andere Zustände auf 0 Dies beweist die τ-Lösbarkeit der verbleibenden ZTP von A. □

Lemma 5.3 rechtfertigt, dass es für den Nachweis der τ-EZTE beziehungsweise der τ-ZTE eines linearen (konstruierten) Transitionssystems A_τ ausreicht, τ-Rümpfe anzugeben, welche nur die Lösbarkeit von EZTP beziehungsweise ZTP adressieren, die von den verwendeten Gadgets herstammen. Weitere Vereinfachungen der Beweise der „Rückrichtungen“ erhalten wir durch Eigenschaften der Gadgets, welche die folgenden Definitionen und Lemmata adressieren.

Definition 5.10 (k-konsistent, konsistent)**.** Es sei $A = (Z, E, \delta, \iota)$ die lineare Verbindung von $A_0, \ldots, A_n$ und $e \in E$ sowie $k \in \mathbb{N}^+$. Wir sagen, e ist *k-konsistent* (in A), wenn e in A stets genau k-mal hintereinander stattfindet: wenn $z \xrightarrow{e} z' \in A$, dann gibt es Zustände $z, z' \in \{z_0, \ldots, z_k\} \subseteq Z$ mit $z_0 \xrightarrow{e} \ldots \xrightarrow{e} z_k$ und $\neg \xrightarrow{e} z_0$ sowie $\neg z_k \xrightarrow{e}$. A heißt *konsistent*, wenn es für jedes Ereignis $e \in E$ ein $k \in \mathbb{N}^+$ gibt, sodass e k-konsistent ist.

Definition 5.11 (dünn verteilt)**.** Sei $A = (Z, E, \delta, \iota)$ die lineare Verbindung von $A_0 = (Z_0, E_0, \delta_0, \iota_0), \ldots, A_n = (Z_n, E_n, \delta_n, \iota_n)$ und $e \in E$; es gebe ein $k \in \{1, b\}$, sodass e k-konsistent ist. Wir sagen e ist *dünn verteilt* (in A), wenn für alle $i \in \{0, \ldots, n\}$ die folgende Eigenschaft erfüllt ist: wenn $e \in E_i$, dann gibt es genau einen Pfad mit paarweise verschiedenen Zuständen $z_0 \xrightarrow{e} \ldots \xrightarrow{e} z_k$ in A_i.

Das folgende technische Lemma listet zwei Bedingungen auf, von denen fast alle Ereignisse der Gadgets des zu konstruierenden Transitionssystems A_τ eine erfüllen. Daraus resultiert ihre τ-Lösbarkeit in A_τ, sodass diese nicht durch explizites Auflisten von τ-Regionen bewiesen werden muss.

Lemma 5.4. *Es sei $\tau \in \{\tau^b_{PPT}, \tau^b_{PT}\}$. Sei $A = (Z, E, \delta, \iota)$, die konsistente lineare Verbindung der Basis $B = (A_0, \ldots, A_n)$, wobei $A_0 = (Z_0, E_0, \delta_0, \iota_0), \ldots, A_n = (Z_n, E_n, \delta_n, \iota_n)$, sodass jedes Ereignis 1-konsistent oder b-konsistent ist. Sei $a \in \underline{E}$ ein dünn verteiltes Ereignis und $p \in Z_i$ ein Zustand mit $\neg p \xrightarrow{a}$, wobei $i \in \{0, \ldots, n\}$ beliebig aber fest sei. Wenn eine der folgenden Bedingungen erfüllt ist, dann gibt es einen τ-Rumpf $R = (sup, sig)$ von B, der das EZTP (a, p) löst:*

1. *entweder $a \notin E_i$ oder $a \in E_i$* und *a findet vor p statt.*
2. *$a \in E_i$ und a findet nach p statt und es gibt ein Ereignis $x \in E_i \setminus \{a\}$ sodass $\xrightarrow{x} z \xrightarrow{a}$ in A_i und*
   (a) *x ist dünn verteilt und*
   (b) *für alle $j \in \{0, \ldots, n\}$ gilt, wenn $a, x \in E_j$, dann findet x nicht nach a in A_j statt und*
   (c) *wenn a b-konsistent ist, dann ist x 1-konsistent.*

Insbesondere ist das EZTP (a, p) von A τ-lösbar.

Beweis. (1): Der folgende (implizit) definierte τ-Rumpf $R = (sup, sig)$ löst (a, p): für alle $j \in \{0, \ldots, n\}$, wenn $a \in E_j$, dann $sup(\iota_j) = 0$, sonst $sup(\iota_j) = b$; für alle $e \in \underline{E}$, wenn $e = a$, dann $sig(e) = (0, 1)$, falls a b-konsistent ist, und $sig(e) = (0, b)$, falls a 1-konsistent ist; andernfalls, das heißt $e \neq a$, dann $sig(e) = (0, 0)$.

(2): Der folgende (implizit) definierte τ-Rumpf $R = (sup, sig)$ und löst (a, p): für alle $j \in \{0, \ldots, n\}$, wenn $a \in E_j$ und $x \notin E_j$, dann $sup(\iota_j) = 0$, sonst $sup(\iota_j) = b$; für alle $e \in \underline{E}$, wenn $e = a$, dann $sig(e) = (0, 1)$, falls a b-konsistent ist, sonst $sig(e) = (0, b)$; wenn $e = x$, dann $sig(e) = (1, 0)$, falls x b-konsistent ist, sonst $sig(e) = (b, 0)$; andernfalls $sig(e) = (0, 0)$.

Analog zu Lemma 5.3 erweitern wir R zu einer τ-Region von A, die (a, p) löst. □

5.1.1 τ^b_{PPT}-Sprachsimulation und τ^b_{PPT}-Realisierung

Solange nicht ausdrücklich anders vereinbart, gelte $\tau = \tau^b_{PPT}$ im Rest dieses Abschnitts. Wir starten mit der Konstruktion des Transitionssystems A_τ, das heißt der Einführung der Gadgets, und beweisen anschließend seine Funktionalität.

Das Transitionssystem A_τ hat das folgende Gadget H_0, welches insbesondere das EZTP $\alpha = (k, h_{0,2b+4})$ liefert:

$$H_0 = h_{0,0} \xrightarrow{k} \cdots \xrightarrow{k} h_{0,b} \xrightarrow{y_0} h_{0,b+1} \xrightarrow{o_0} h_{0,b+2} \xrightarrow{k} \cdots \xrightarrow{k} h_{0,2b+2} \downarrow y_1$$
$$h_{0,3b+5} \xleftarrow[k]{} \cdots \xleftarrow[k]{} h_{0,2b+5} \xleftarrow[o_1]{} h_{0,2b+4} \xleftarrow[y_0]{} h_{0,2b+3}$$

Man bemerke, dass das Ereignis k jeweils an den Zuständen $h_{0,0}$ und $h_{0,b+2}$ sowie $h_{0,2b+5}$ genau b-mal hintereinander stattfindet.

Für jedes $j \in \{0,1,2,3\}$ hat das Transitionsystem A_τ das folgende Gadget D_j, welches insbesondere das Ereignis k_j verwendet:

$$D_j = \quad d_{j,0} \xrightarrow{o_0} d_{j,1} \xrightarrow{k_j} d_{j,2} \xrightarrow{o_1} d_{j,3}$$

Für alle $j \in \{0,\ldots,2m-1\}$ hat A_τ außerdem die folgenden Gadgets F_j und G_j, die das Ereignis w_j verwenden:

$$F_j = \quad f_{j,0} \xrightarrow{k_0} f_{j,1} \xrightarrow{w_j} f_{j,2} \qquad\qquad G_j = \quad g_{j,0} \xrightarrow{w_j} g_{j,1} \xrightarrow{o_0} g_{j,2}$$

Des Weiteren hat A_τ für jedes $i \in \{0,\ldots,m-1\}$ das folgende Gadget Q_i auf dem die Variable X_i aus $\mathfrak{U}$ b-mal hintereinander als Ereignis stattfindet:

$$Q_i = \quad q_{i,0} \xrightarrow{k_1} q_{i,1} \xrightarrow{X_i} \cdots \xrightarrow{X_i} q_{i,b+1}$$

Schließlich hat A_τ für jedes $i \in \{0,\ldots,m-1\}$ das folgende Gadget T_i, auf dem die Variablen der Klausel $M_i = \{X_{i_0}, X_{i_1}, X_{i_2}\}$ als Ereignisse stattfinden:

$$T_i = \quad t_{i,0} \xrightarrow{k_2} t_{i,1} \xrightarrow{X_{i_0}} \cdots \xrightarrow{X_{i_0}} t_{i,b+1} \xrightarrow{w_{2i}} t_{i,b+2} \xrightarrow{X_{i_1}} \cdots \xrightarrow{X_{i_1}} t_{i,2b+2} \downarrow w_{2i+1}$$
$$t_{i,3b+4} \xleftarrow[k_3]{} t_{i,3b+3} \xleftarrow[X_{i_2}]{} \cdots \xleftarrow[X_{i_2}]{} t_{i,2b+3}$$

Man beachte, dass X_{i_0} an $t_{i,1}$ und X_{i_1} an $t_{i,b+2}$ sowie X_{i_2} an $t_{i,2b+3}$ jeweils b-mal hintereinander stattfinden. A_τ enthält keine weiteren Gadgets und wir erhalten das Transitionssystem als lineare Verbindung der Basis $B_\tau =$, die gemäß Definition 5.9 wie folgt definiert ist:

$$B_\tau = (H_0, D_0, \ldots, D_3, F_0, \ldots, F_{2m-1}, G_0, \ldots, G_{2m-1}, Q_0, \ldots, Q_{m-1}, T_0, \ldots, T_{m-1})$$

Man beachte, dass A_τ konsistent ist, wobei jedes Ereignis von A entweder 1-konsistent oder b-konsistent ist. Wir beweisen nun die Funktionalität von A_τ.

Lemma 5.5. *Wenn A_τ die τ-EZTE hat, dann hat $(\mathfrak{U}, M)$ ein Eins-in-drei-Modell.*

Beweis. Weil A_τ die τ-EZTE hat, gibt es eine τ-Region, die α löst. Sei $R = (sup, sig)$ so eine Region, das heißt $\neg sup(h_{0,2b+4}) \xrightarrow{sig(k)}$. Weil $(0,0)$ an allen Zuständen von τ stattfindet und k b-konsistent ist, impliziert dies $sig(k) \in \{(1,0),(0,1)\}$ mit (dem ersten Item von) Lemma 5.2. Im Folgenden untersuchen wir zunächst den Fall $sig(k) = (0,1)$ und argumentieren, dass die Menge $S = \{X \in \mathfrak{U} \mid sig(X) = (1,0)\}$ unter dieser Voraussetzung ein Eins-in-drei-Modell definiert.

Wir beginnen damit zu zeigen, dass $sig(o_0) = sig(o_1) = (b,0)$: Weil $sig(k) = (0,1)$ und $\neg sup(h_{0,2b+4}) \xrightarrow{sig(k)}$ sowie $z \xrightarrow{0,1} \in \tau$ für alle $z \in \{0,\dots,b-1\}$, erhalten wir $sup(h_{0,2b+4}) = b$. Mit Lemma 5.2 erhalten wir außerdem $sup(h_{0,b}) = sup(h_{0,2b+2}) = b$ und $sup(h_{0,b+2}) = sup(h_{0,2b+5}) = 0$. Aus $sup(h_{0,2b+4}) = b$ und $sup(h_{0,2b+5}) = 0$ schließen wir sofort $sig(o_1) = (b,0)$. Außerdem impliziert $sup(h_{0,b}) = sup(h_{0,2b+4}) = b$ zum einen $b \xrightarrow{sig(y_0)}$ und zum anderen $\xrightarrow{sig(y_0)} b$ und somit $sig^-(y_0) = sig^+(y_0)$ gemäß Lemma 5.2. Daraus folgt $sup(h_{0,b+1}) = b$ und mit $sup(h_{0,b+2}) = 0$ folgt $sig(o_0) = (b,0)$.

Nun zeigen wir $sig(k_i) = (0,b)$ für alle $i \in \{0,\dots,3\}$ mit Hilfe der Gadgets $D_0,\dots,D_3$: Genauer gesagt folgt aus $sig(o_0) = sig(o_0) = (b,0)$ sofort $sup(d_{j,1}) = 0$ und $sup(d_{j,2}) = b$ und somit $sig(k_j) = (0,b)$ für alle $j \in \{0,\dots,3\}$.

Für alle $j \in \{0,\dots,2m-1\}$ sichern die Gadgets F_j und G_j zu, dass $sig^-(w_j) = sig^+(w_j)$: Mit $sig(k_0) = (0,b)$ und $sig(o_0) = (b,0)$ folgt $sup(f_{j,1}) = sup(g_{j,1}) = b$. Dies impliziert $b \xrightarrow{sig(w_j)}$ und $\xrightarrow{sig(w_j)} b$ und folglich $sig^-(w_j) = sig^+(w_j)$ mit Lemma 5.2.

Für alle $i \in \{0,\dots,m-1\}$ sorgt das Gadget Q_i dafür, dass $sig(X_i) \in \{(0,0),(1,0)\}$: Wegen $sig(k_1) = (0,b)$ erhalten wir $sup(q_{i,1}) = b$. Weil X_i b-fach hintereinander stattfindet, erhalten wir mit Lemma 5.2, dass $sig(X_i) = (0,1)$ jedoch $sup(q_{i,1}) = 0$ impliziert, ein Widerspruch. Mit Lemma 5.2 folgt $sig(X_i) \in \{(0,0),(1,0)\}$.

Sei $i \in \{0,\dots,m-1\}$ beliebig aber fest. Das Gadget T_i sorgt dafür, dass es genau ein Ereignis $X \in \{X_{i_0}, X_{i_1}, X_{i_2}\}$ gibt, welches $sig(X) = (1,0)$ erfüllt: Mit $sig(k_2) = sig(k_3) = (0,b)$ erhalten wir $sup(t_{i,1}) = b$ und $sup(t_{i,3b+3}) = 0$. Folglich ist das Bild P_i^R des Pfades $P_i = t_{i,1} \xrightarrow{X_{i_0}} \dots \xrightarrow{X_{i_2}} t_{i,3b+3}$ ein Pfad von b nach 0 in τ. Deshalb muss es ein Ereignis e mit $sig^-(e) > sig^+(e)$ auf diesem Pfad geben. Weil $sig^-(w_{2i}) = sig^+(w_{2i})$ und $sig^-(w_{2i+1}) = sig^+(w_{2i+1})$, muss e aus $\{X_{i_0}, X_{i_1}, X_{i_2}\}$ sein. Für alle $X \in \{X_{i_0}, X_{i_1}, X_{i_2}\}$ gilt $sig(X) \in \{(0,0),(1,0)\}$. Deshalb folgt aus $sig^-(X) > sig^+(X)$ sofort $sig(X) = (1,0)$. Wir zeigen nun, dass X eindeutig ist:

Wenn $sig(X_{i_0}) = (1,0)$, dann gilt $sup(t_{i,b+1}) = sup(t_{i,b+2}) = 0$ mit Lemma 5.2 und $sig^-(w_{2i}) = sig^+(w_{2i})$. Mit $\neg 0 \xrightarrow{1,0} \in \tau$ impliziert dies $sig(X_{i_1}) = (0,0)$ und außerdem $sup(t_{i,b+2}) = \dots = sup(t_{i,2b+3}) = 0$ mit $sig^-(w_{2i+1}) = sig^+(w_{2i+1})$. Analog folgt daraus $sig(X_{i_2}) = (0,0)$.

Wenn $sig(X_{i_1}) = (1,0)$, dann gelten $sup(t_{i,b+2}) = b$ und $sup(t_{i,2b+2}) = 0$ gemäß Lemma 5.2. Mit $sig^-(w_{2i}) = sig^+(w_{2i})$ und $sig^-(w_{2i+1}) = sig^+(w_{2i+1})$ erhalten wir $sup(t_{i,b+1}) = b$ und $sup(t_{i,2b+3}) = 0$. Daraus folgt $sig(X_{i_0}) = (0,0)$, denn $sig(X_{i_0}) = (1,0)$ impliziert den Widerspruch $sup(t_{i,b+1}) = 0$. Ebenso folgt $sig(X_{i_2}) = (0,0)$, denn

$sig(X_{i_2}) = (1,0)$ impliziert den Widerspruch $sup(t_{i,b+3}) = b$.

Schließlich, wenn $sig(X_{i_2}) = (1,0)$, dann muss $sig(X_{i_0}) = sig(X_{i_1}) = (0,0)$ gelten, denn, wie gerade argumentiert, impliziert $sig(X_{i_0}) = (1,0)$ oder $sig(X_{i_1}) = (1,0)$, dass $sig(X_{i_2}) = (0,0)$.

Weil i beliebig war, folgt $|M_i \cap S| = 1$ für alle $i \in \{0,\dots,m-1\}$. Insbesondere ist S ein Eins-in-drei-Modell von $(\mathfrak{U}, M)$.

Analog führt die Annahme $sig(k) = (1,0)$ zu $sig(k_0) = \cdots = sig(k_3) = (b,0)$ sowie $sig^-(w_j) = sig^+(w_j)$ für alle $j \in \{0,\dots,2m-1\}$ und $sig(X_i) \in \{(0,0),(0,1)\}$ für alle $i \in \{0,\dots,m-1\}$. Man argumentiert dann, ähnlich wie eben, dass die Menge $S' = \{\, X \in \mathfrak{U} \mid sig(X) = (0,1)\}$ ein Eins-in-drei-Modell von $(\mathfrak{U}, M)$ ist. □

Soweit haben wir bewiesen, dass die τ-EZTE von A_τ ein Eins-in-drei-Modell für $(\mathfrak{U}, M)$ impliziert. Für die andere Richtung sei S ein beliebiges aber festes Eins-in-drei-Modell von $(\mathfrak{U}, M)$. Weil A_τ die lineare Verbindung der Basis B_τ ist, genügt es zu zeigen, dass alle EZTP von B_τ τ-lösbar sind, wie die folgenden Überlegungen beweisen: Wenn alle EZTP von B_τ τ-lösbar sind, dann hat A_τ gemäß Lemma 5.3 die τ-EZTP. Weil A_τ ein lineares Transitionssystem ist, impliziert dies mit Lemma 2.5, dass A_τ auch die τ-ZTE hat. Wir werden im Folgenden die EZTP von B_τ und deren Lösbarkeit diskutieren. Es bezeichne $\underline{E}$ die Ereignisbasis von B_τ und Z_ι die Menge der Initialzustände der Gadgets von B_τ:

$$Z_\iota = \{h_{0,0}, d_{0,0}, \dots, d_{3,0}, f_{0,0}, \dots, f_{2m-1,0}, g_{0,0}, \dots, g_{2m-1,0}, q_{0,0}, \dots, q_{m-1,0}, t_{0,0}, \dots, t_{m-1,0}\}$$

Der besseren Lesbarkeit wegen, werden wir einen Rumpf $R = (sup, sig)$ von B_τ implizit mittels $sup(z)$ für alle $z \in Z_\iota$ sowie sig definieren. Der Support sup ergibt sich dann wieder in naheliegender Weise für jedes Gadget $z_0 \xrightarrow{e_1} \dots \xrightarrow{e_n} z_n$ von A über $sup(z_{i+1}) = \delta_\tau(sup(z_i), sig(e_{i+1}))$ für alle $i \in \{0,\dots,n-1\}$. Darüber hinaus definieren wir sig, indem wir Ereignisse mit gleichem Signaturwert $(m,n) \in E_\tau$ wie folgt in einer Menge zusammenfassen: $\mathcal{E}^R_{m,n} = \{e \in \underline{E} \mid sig(e) = (m,n)\}$.

Fakt 5.2. *Das Ereignis k ist τ lösbar.*

Beweis. Der folgende τ-Rumpf $R_1 = (sup_1, sig_1)$ nutzt das Eins-in-drei-Modell S und löst (k,z) für alle relevanten Zustände $z \in Z(H_0)$ und löst damit insbesondere $\alpha = (k, h_{0,2b+4})$: wenn $z \in \{h_{0,0}, f_{0,0}, \dots, f_{2m-1,0}\} \cup \{q_{0,0}, \dots, q_{m-1,0}\} \cup \{t_{0,0}, \dots, t_{m-1,0}\}$, dann $sup_1(z) = 0$; wenn $z \in \{d_{0,0}, \dots, d_{3,0}, g_{0,0}, \dots, g_{2m-1,0}\}$, dann $sup_1(z) = b$; außerdem $\mathcal{E}^{R_1}_{0,1} = \{k\}$ und $\mathcal{E}^{R_1}_{b,0} = \{o_0, o_1\}$ und $\mathcal{E}^{R_1}_{0,b} = \{k_0, \dots, k_3\}$ und $\mathcal{E}^{R_1}_{1,0} = S$ und $\mathcal{E}^{R_1}_{0,0} = \underline{E} \setminus (\mathcal{E}^{R_1}_{0,1} \cup \mathcal{E}^{R_1}_{b,0} \cup \mathcal{E}^{R_1}_{0,b} \cup \mathcal{E}^{R_1}_{1,0})$.

Der folgende τ-Rumpf $R_2 = (sup_2, sig_2)$ löst alle verbleibenden EZTP von k: für alle $z \in Z_\iota$, wenn $z = h_{0,0}$, dann $sup_2(z) = 0$, andernfalls $sup_2(z) = b$; außerdem $\mathcal{E}^{R_2}_{0,1} = \{k\}$ und $\mathcal{E}^{R_2}_{b,0} = \{y_0\}$ und $\mathcal{E}^{R_2}_{0,0} = \underline{E} \setminus \{k, y_0\}$. □

Fakt 5.3. *Die Ereignisse y_0 und y_1 sind τ-lösbar.*

Beweis. Der folgende τ-Rumpf $R_1 = (sup_1, sig_1)$ löst (y_0, z) für alle $z \in \{h_{0,0}, \ldots, h_{0,b-1}\}$ und für alle $z \in \underline{Z} \setminus Z(H_0)$: $sup_1(z) = 0$ für alle $z \in Z_\iota$ und $\mathcal{E}^{R_1}_{b,0} = \{y_0\}$ und $\mathcal{E}^{R_1}_{0,1} = \{k\}$ und $\mathcal{E}^{R_1}_{0,0} = \underline{E} \setminus \{y_0, k\}$.

Der folgende Rumpf $R_2 = (sup_2, sig_2)$ löst (y_0, z) für alle $z \in Z(H_0) \setminus \{h_{0,0}, \ldots, h_{0,b-1}\}$: $sup_2(z) = b$ für alle $z \in Z_\iota$ und $\mathcal{E}^{R_2}_{b,0} = \{y_0\}$ und $\mathcal{E}^{R_2}_{0,b} = \{y_1\}$ und $\mathcal{E}^{R_2}_{0,0} = \underline{E} \setminus \{y_0, y_1\}$.

Für alle $z \in \{h_{0,2b+3}, \ldots h_{0,3b+5}\}$ und alle $\underline{Z} \setminus Z(H_0)$ folgt die τ-Lösbarkeit von (y_1, z) durch (die erste Eigenschaft von) Lemma 5.4.

Der folgende τ-Rumpf $R_3 = (sup_3, sig_3)$ löst (y_1, z) für alle $z \in \{h_{0,b+2}, \ldots, h_{0,2b+1}\}$: für alle $z \in Z_\iota$, wenn $z = h_{0,0}$, dann $sup_3(h_{0,0}) = 0$, andernfalls $sup_3(z) = b$ und $\mathcal{E}^{R_3}_{b,0} = \{o_0, y_1\}$ und $\mathcal{E}^{R_3}_{0,1} = \{k\}$ und $\mathcal{E}^{R_3}_{0,0} = \underline{E} \setminus \{k, o_0, y_1\}$.

Der folgende τ-Rumpf $R_4 = (sup_4, sig_4)$ löst (y_1, z) für alle $z \in \{h_{0,0}, \ldots, h_{0,b+1}\}$: $sup_4(z) = 0$ für alle $z \in Z_\iota$ und $\mathcal{E}^{R_4}_{b,0} = \{y_1\}$ und $\mathcal{E}^{R_4}_{0,b} = \{o_0\}$ und $\mathcal{E}^{R_4}_{0,0} = \underline{E} \setminus \{o_0, y_1\}$. □

Fakt 5.4. *Wenn $e \in \underline{E} \setminus \{k, y_0, k_1\}$, dann ist e τ-lösbar*

Beweis. Sei $e \in \underline{E} \setminus \{k, y_0, y_1\}$ beliebig aber fest. Zunächst ist e dünn verteilt. Darüber hinaus gilt $i_0 < i_1 < i_2$ für Klausel $M_i = \{X_{i_0}, X_{i_1}, X_{i_2}\}$ für alle $i \in \{0, \ldots, m-1\}$. Damit erhalten wir für alle $z \in \underline{Z}$ mit $\neg z \xrightarrow{e}$ das Folgende: Wenn $e \notin \{o_0, o_1\}$ oder wenn $e \in \{o_0, o_1\}$ und $z \notin Z(H_0)$, dann ist für das EZTP (e, z) eine der Bedingungen von Lemma 5.4 erfüllt und es folgt, mit ebendiesem Lemma, dass (e, z) τ-lösbar ist.

Um den Beweis des Fakts zu komplettieren, müssen wir nur noch die τ-Lösbarkeit von o_0 und o_1 in H_0 zeigen: Für alle $z \in \{h_{0,b+2}, \ldots h_{0,3b+5}\}$ folgt die τ-Lösbarkeit von (o_0, z) durch (die erste Eigenschaft von) Lemma 5.4.

Darüber hinaus löst der folgende τ-Rumpf $R = (sup, sig)$ das EZTP (o_0, z) für alle $z \in \{h_{0,0}, \ldots, h_{0,b}\}$: für alle $z \in Z_\iota$, wenn $z = h_{0,0}$, dann $sup(z) = b$, andernfalls $sup(z) = b$; außerdem $\mathcal{E}^{R}_{0,b} = \{o_0\}$ und $\mathcal{E}^{R}_{b,0} = \{y_0\}$ und $\mathcal{E}^{R}_{0,0} = \underline{E} \setminus \{o_0, y_0\}$. Die komplettiert die τ-Lösbarkeit von o_0. Mit ähnlichen Argumenten beweist man die τ-Lösbarkeit von o_1 in H_0. Die Behauptung folgt. □

Weil die τ-EZTE von B_τ die τ-EZTE und somit auch die τ-ZTE von A_τ impliziert, erhalten wir das folgende Lemma als einfache Folgerung aus Fakt 5.2 und Fakt 5.3 sowie Fakt 5.4.

Lemma 5.6. *Wenn $(\mathfrak{U}, M)$ ein Eins-in-drei-Modell hat, dann hat A_τ die τ-EZTE und die τ-ZTE.*

Soweit haben wir mit Lemma 5.5 und Lemma 5.6 gezeigt, dass τ-SPRACHSIMULATION und τ-REALISIERUNG für $\tau = \tau^b_{PPT}$ NP-vollständig sind, denn die präsentierte Reduktion ist offenbar polynomiell und die adressierten Entscheidungsprobleme sind gemäß Lemma 5.1 in NP.

5.1.2 τ^b_{PT}-Sprachsimulation und τ^b_{PT}-Realisierung

Solange nicht ausdrücklich anders vereinbart, gelte im Rest dieses Abschnitts $\tau = \tau^b_{PT}$. In diesem Abschnitt werden wir die Aussage von Satz 5.1 für τ-Sprachsimulation und τ-Realisierung beweisen. Dazu folgen wir erneut der in Kapitel 3 vorgestellten Strategie. Der Grund dafür, dass wir die Konstruktion für den Netztyp τ^b_{PPT} des vorherigen Abschnitts nicht verwenden können, lässt sich knapp wie folgt skizzieren: Für den Netztyp τ^b_{PT} kann ein EZTP (a, p) potentiell durch eine Region $R = (sup, sig)$ gelöst werden, sodass $sig(a) = (m, m)$ (und $\neg sup(p) \xrightarrow{(m,m)}$), das heißt $m \geq 1$. Dies führt dazu, dass es für die Konstruktion $A_{\tau^b_{PPT}}$ aus dem vorherigen Abschnitt unabhängig von der Existenz eines Eins-in-drei-Modells eine solche τ^b_{PT}-Region gibt, die α löst. Insbesondere müssen wir das Gadget H_0 entsprechend modifizieren. Diese Modifikation führt nun aber dazu, dass es EZTP gibt, deren Lösbarkeit die Existenz von Ereignissen (m, m) mit $m \geq 1$ *erfordern*. Folglich kann diese neue Konstruktion wiederum nicht für τ^b_{PPT} verwendet werden. Generell liefern diese Ereignisse (m, m) mit $m \geq 1$ eine neue Mächtigkeit, EZTP zu lösen, die dazu führt, dass eine einheitliche Konstruktion für beide Netztypen nicht (leicht) möglich ist. Nichtsdestotrotz können wir die meisten der im vorherigen Abschnitt definierten Gadgets wiederverwenden.

Wir beginnen mit der Konstruktion von A_τ und zeigen anschließend sein Funktionalität gemäß der in Kapitel 3 vorgestellten Strategie.

Das Transitionssystem A_τ hat das folgende Gadget H_1, welches insbesondere das EZTP $\alpha = (k, h_{1,4b+1})$ liefert:

$$H_1 = h_{1,0} \xrightarrow{k} \cdots \xrightarrow{k} h_{1,b} \xrightarrow{w} \cdots \xrightarrow{w} h_{1,2b} \xrightarrow{o_0} h_{1,2b+1} \xrightarrow{k} \cdots \xrightarrow{k} h_{1,3b+1} \downarrow w$$
$$h_{1,6b+1} \xleftarrow[k]{} \cdots \xleftarrow[k]{} h_{1,5b+1} \xleftarrow[o_1]{} \cdots \xleftarrow[o_1]{} h_{1,4b+1} \xleftarrow[w]{} \cdots$$

Man beachte, dass k b-mal hintereinander an $h_{1,0}, h_{1,2b+1}$ und $h_{1,5b+1}$ stattfindet. Außerdem findet w b-mal hintereinander an $h_{1,b}$ und $h_{1,3b+1}$ statt. Des Weiteren findet o_1 b-mal hintereinander an $h_{1,4b+1}$ statt.

Das Transitionssystem A_τ hat für jedes $j \in \{0, 1, 2, 3\}$ das folgende Gadget C_j, welches k_j liefert:

$$C_j = \quad c_{j,0} \xrightarrow{o_0} c_{j,1} \xrightarrow{k_j} c_{j,2} \xrightarrow{o_1} \cdots \xrightarrow{o_1} c_{j,b+2}$$

Darüber hinaus hat A_τ für alle $j \in \{0, \ldots, 2m-1\}$ und für alle $i \in \{0, \ldots, m-1\}$ die Gadgets F_j, G_j, Q_i and T_i, die in Abschnitt 5.1.1 für die Konstruktion von $A_{\tau^b_{PPT}}$ eingeführt wurden. Schließlich erhalten wir A_τ gemäß Definition 5.9 als lineare Verbindung der Basis

$$B_\tau = (H_1, C_0, \ldots, C_3, F_0, \ldots, F_{2m-1}, G_0, \ldots, G_{2m-1}, Q_0, \ldots, Q_{m-1}, T_0, \ldots, T_{m-1})$$

.

Lemma 5.7. *Wenn A_τ die τ-EZTE hat, dann hat $(\mathfrak{U}, M)$ ein Eins-in-drei-Modell.*

Beweis. Wenn A_τ die τ-EZTE hat, dann gibt es eine τ-Region, die α löst. Sei $R = (sup, sig)$ so eine Region. Wir werden im Folgenden argumentieren, dass $sig^-(w_j) = sig^+(w_j)$ für alle $j \in \{0, \ldots, 2m-1\}$ und entweder $sig(k_0) = \cdots = sig(k_3) = (0, b)$ und $sig(X_i) \in \{(0,0), (1,0)\}$ für alle $i \in \{0, \ldots, m-1\}$ oder $sig(k_0) = \cdots = sig(k_3) = (b, 0)$ und $sig(X_i) \in \{(0,0), (0,1)\}$ für alle $i \in \{0, \ldots, m-1\}$. Daraus folgt dann analog zu der Argumentation für Lemma 5.5 in Abschnitt 5.1.1, dass $S = \{X \in \mathfrak{U} \mid sig(X) = (1,0)\}$ oder $S' = \{X \in \mathfrak{U} \mid sig(X) = (0,1)\}$ ein gesuchtes Eins-in-drei-Modell von $(\mathfrak{U}, M)$ impliziert.

Weil R das EZTP α löst, gilt $\neg sup(h_{1,4b+1}) \xrightarrow{sig(k)}$. Es findet k in H_1 an $h_{1,0}$ b-mal hintereinander statt. Mit Lemma 5.2 erhalten wir deshalb $sig(k) \in \{(1,0), (0,1)\}$ oder $sig(k) = (m, m)$ mit $m \in \{1, \ldots, b\}$.

Untersuchen wir zunächst den zweiten Fall, das heißt $sig(k) = (m, m)$ mit $m \in \{1, \ldots, b\}$. Dies impliziert $m \leq sup(z) = sup(z')$ für alle $z \xrightarrow{k} z' \in A_\tau$. Insbesondere erhalten wir (sukzessive) $sup(h_{1,2b+1}) = sup(h_{1,3b+1}) \geq m$. Es gilt $sup(h_{1,4b+1}) < m$, weil $sig(k)$ nicht an $sup(h_{1,4b+1})$ definiert ist. Aus $sup(h_{1,4b+1}) < sup(h_{1,3b+1})$ folgt $sig^-(w) > sig^+(w)$. Weil zudem w dem b-mal an $h_{1,3b+1}$ stattfindet, folgt $sig(w) = (1,0)$ mit Lemma 5.2. Dies impliziert $sup(h_{1,3b+1}) = b$ und, wegen $sig(k) = (m, m)$, auch $sup(h_{1,2b+1}) = b$ sowie $sup(h_{1,2b}) = sup(h_{1,4b+1}) = 0$. Mit $sup(h_{1,2b}) = 0$ und $sup(h_{1,2b+1}) = b$ erhalten wir sofort $sig(o_0) = (0, b)$. Weiterhin folgt $sup(h_{1,5b+1}) > 0$ aus $(m, m) \neq (0, 0)$. Mit $sup(h_{1,4b+1}) = 0$ und dem b-fachen Stattfinden von o_1 an $h_{1,4b+1}$ impliziert dies $sig^+(o_1) > sig^-(o_1)$ und damit $sig(o_1) = (0, 1)$.

Insgesamt haben wir bisher gezeigt, dass der Fall $sig(k) = (m, m)$ mit $m \in \{1, \ldots, b\}$ die Signaturwerte $sig(o_0) = (0, b)$ und $sig(o_1) = (0, 1)$ impliziert. Man überlegt sich, dass daraus $sup(c_{j,1}) = b$ und $sup(c_{j,2}) = 0$ folgen und deshalb $sig(k_j) = (b, 0)$ für alle $j \in \{0, \ldots, 3\}$ gilt. Wie bereits in Abschnitt 5.1.1 diskutiert, impliziert dies $sig^-(w_j) = sig^+(w_j)$ für alle $j \in \{0, \ldots, 2m-1\}$ und $sig(X_i) \in \{(0,0), (0,1)\}$ für alle $i \in \{0, \ldots, m-1\}$, sodass die Menge $S' = \{X \in \mathfrak{U} \mid sig(X) = (0,1)\}$ ein eins-in-drei-Modell von $(\mathfrak{U}, M)$ definiert.

Wir untersuchen nun den Fall $sig(k) = (1, 0)$, der $sup(h_{1,4b+1}) = 0$ impliziert. Mit $sig(k) = (1, 0)$ erhalten wir $sup(h_{1,b}) = sup(h_{1,3b+1}) = 0$ sowie $sup(h_{1,2b+1}) = sup(h_{1,5b+1}) = b$. Aus $h_{1,b} \xrightarrow{w}$ und $\xrightarrow{w} h_{1,4b+1}$ sowie $sup(h_{1,b}) = sup(h_{1,4b+1}) = 0$ erhalten wir gemäß Lemma 5.2 $sig^-(w) = sig^+(w)$ und somit $sup(h_{1,2b}) = 0$. Mit $sup(h_{1,2b+1}) = b$ impliziert dies $sig(o_0) = (0, b)$. Außerdem erhalten wir aus $sup(h_{1,4b+1}) = 0$ und $sup(h_{1,5b+1}) = b$ erneut $sig(o_1) = (0, 1)$. Wie gerade für den Fall $sig(k) = (m, m)$ mit $m \in \{1, \ldots, b\}$ diskutiert, impliziert dies, dass $S' = \{X \in \mathfrak{U} \mid sig(X) = (0,1)\}$ ein eins-in-drei-Modell von $(\mathfrak{U}, M)$ definiert.

Es bleibt noch den Fall $sig(k) = (0, 1)$ zu untersuchen, der $sup(h_{1,4b+1}) = b$ impliziert. Analog zu dem eben diskutierten Fall $sig(k) = (1, 0)$ überlegt man sich Folgendes: Es gilt $sig^-(w_j) = sig^+(w_j)$ für alle $j \in \{0, \ldots, 2m-1\}$ und $sig(k_0) = \cdots = sig(k_3) = (0, b)$ und

$sig(X_i) \in \{(0,0),(1,0)\}$ für alle $i \in \{0,\ldots,m-1\}$, sodass $S = \{X \in \mathfrak{U} \mid sig(X) = (1,0)\}$ ein gesuchtes Eins-in-drei-Modell von $(\mathfrak{U}, M)$ definiert. □

Soweit haben wir gezeigt, dass die τ-EZTE von A_τ ein Eins-in-drei-Modell für $(\mathfrak{U}, M)$ impliziert. Für die andere Richtung beweisen wir das folgende Lemma:

Lemma 5.8. *Wenn $(\mathfrak{U}, M)$ ein Eins-in-drei-Modell hat, dann hat A_τ die τ-EZTE und die τ-ZTE.*

Beweis. Sei S ein Eins-in-drei-Modell von $(\mathfrak{U}, M)$. Es genügt zu zeigen, das jedes EZTP der Basis B_τ τ-lösbar ist, denn dann hat A_τ gemäß Lemma 5.3 die τ-EZTE. Weil A_τ ein lineares Transitionssystem ist, impliziert dies gemäß Lemma 2.5 auch die τ-ZTE für A_τ.

Sei $\underline{E}$ die Menge aller Ereignisse von B_τ und Z_ι die Menge der Initialzustände der Gadgets von B_τ, das heißt,

$$Z_\iota = \{h_{1,0}, c_{0,0}, \ldots, c_{3,0}, f_{0,0}, \ldots, f_{2m-1,0}, g_{0,0}, \ldots, g_{2m-1,0}, q_{0,0}, \ldots, q_{m-1,0}, t_{0,0}, \ldots, t_{m-1,0}\}.$$

Analog zu Abschnitt 5.1.1 präsentieren wir einen τ-Rumpf $R = (sup, sig)$ wieder implizit mittels $sup(z)$ für alle $z \in Z_\iota$ und sig, wobei wir Ereignisse $e \in \underline{E}$ gleicher Signatur (m,n) gemäß der Menge $\mathcal{E}^R_{m,n} = \{e \in \underline{E} \mid sig(e) = (m,n)\}$ zusammenfassen.

Offensichtlich ist jedes Ereignis $e \in \{k, w, o_1, X_0, \ldots, X_{m-1}\}$ b-konsistent und jedes Ereignis $e \in \{o_0, k_0, \ldots, k_3, w_0, \ldots, w_{2m-1}\}$ ist 1-konsistent. Wenn $e \in \underline{E} \setminus \{k, w\}$, dann ist e darüber hinaus dünn verteilt. Seien $e \in \underline{E} \setminus \{k, w\}$ und $z \in \underline{Z}$ mit $\neg z \xrightarrow{e}$ beliebig aber fest. Wenn $e \notin \{o_0, o_1\}$ oder $e \in \{o_0, o_1\}$ und $z \notin Z(H_1)$, dann ist für das EZTP (e, z) eine der Bedingungen von Lemma 5.4 erfüllt. Somit ist (e, z) τ-lösbar.

Die Ereignisse k und w finden nur in H_1 statt. Daher löst der folgende τ-Rumpf $R_1 = (sup_1, sig_1)$ das EZTP (a, q) für alle $a \in \{k, w\}$ und alle $q \in \underline{Z} \setminus Z(H_1)$: für alle $z \in Z_\iota$, wenn $z = h_{1,0}$, dann $sup_1(z) = 1$, sonst $sup_1(z) = 0$; außerdem $\mathcal{E}^{R_1}_{1,1} = \{k, w\}$ und $\mathcal{E}^{R_1}_{0,0} = \underline{E} \setminus \{k, w\}$.

Um den Beweis des Lemmas abzuschließen, bleibt zu zeigen, dass ein EZTP (a, q) lösbar ist, wenn $a \in \{k, w, o_0, o_1\}$ und $q \in Z(H_1)$.

Wir beginnen mit den Ereignissen o_0 und o_1: Wenn $e = o_0$ und $z \in \{h_{1,2b+1}, \ldots, h_{1,6b+1}\}$ oder wenn $e = o_1$ und $z \in \{h_{1,5b+1}, \ldots, h_{1,6b+1}\}$, dann ist für das EZTP (e, z) die erste Bedingung von Lemma 5.4 erfüllt. Somit ist (e, z) τ-lösbar.

Der folgende τ-Rumpf $R_2 = (sup_2, sig_2)$ nutzt das Eins-in-drei-Modell S und löst (o_0, z) für alle $z \in \{h_{1,0}, \ldots, h_{2b-1}\}$ sowie (o_1, z) für alle $z \in \{h_{1,0}, \ldots, h_{4b}\} \setminus \{h_{1,2b}\}$; für alle $z \in Z_\iota$, wenn $z \in \{c_{0,0}, \ldots, c_{3,0}\} \cup \{f_{0,0}, \ldots, g_{0,0}, \ldots, g_{2m-1,0}\}$, dann $sup_2(z) = b$, sonst $sup_2(z) = 0$; außerdem $\mathcal{E}^{R_2}_{0,1} = \{w\}$ und $\mathcal{E}^{R_2}_{b,0} = \{o_0\}$ und $\mathcal{E}^{R_2}_{b,b} = \{o_1\}$ und $\mathcal{E}^{R_2}_{0,b} = \{k_0, \ldots, k_3\}$ und $\mathcal{E}^{R_2}_{1,0} = S$ und $\mathcal{E}^{R_2}_{0,0} = \underline{E} \setminus (\mathcal{E}^{R_2}_{0,1} \cup \mathcal{E}^{R_2}_{b,0} \cup \mathcal{E}^{R_2}_{b,b} \cup \mathcal{E}^{R_2}_{0,b} \cup \mathcal{E}^{R_2}_{1,0})$. Dies komplettiert den Beweis der τ-Lösbarkeit von o_0.

Der folgende τ-Rumpf $R_3 = (sup_3, sig_3)$ löst $(o_1, h_{1,2b})$: $sup(z) = 0$ für alle $z \in Z_\iota$ und $\mathcal{E}^{R_3}_{0,b} = \{o_0\}$ und $\mathcal{E}^{R_3}_{b,b} = \{o_1\}$ und $\mathcal{E}^{R_3}_{0,0} = \underline{E} \setminus \{o_0, o_1\}$. Es folgt, dass o_1 τ-lösbar ist.

Nun zur τ-Lösbarkeit von k: Der folgende τ-Rumpf $R_4 = (sup_4, sig_4)$ nutzt das Eins-in-drei-Modell S und löst (k, z) für alle Zustände $z \in Z(H_1) \setminus \{h_{1,4b+2}, \ldots, h_{1,5b}\}$, an denen k nicht stattfindet: für alle $z \in Z_\iota$, wenn $z \in \{c_{0,0}, \ldots, c_{3,0}, g_{0,0}, \ldots, g_{2m-1,0}\}$, dann $sup_4(z) = 0$, sonst $sup_4(z) = b$; außerdem $\mathcal{E}^{R_4}_{1,0} = \{k\}$ und $\mathcal{E}^{R_4}_{0,b} = \{o_0\}$ und $\mathcal{E}^{R_4}_{0,1} = S \cup \{o_1\}$ und $\mathcal{E}^{R_4}_{b,0} = \{k_0, \ldots, k_3\}$ und $\mathcal{E}^{R_4}_{0,0} = \underline{E} \setminus (\mathcal{E}^{R_4}_{1,0} \cup \mathcal{E}^{R_4}_{0,b} = \cup\mathcal{E}^{R_4}_{0,1} \cup \mathcal{E}^{R_4}_{b,0})$.

Modifiziert man R_4 zu einem τ-Rumpf R_5, der k auf (b, b) und w auf $(1, 0)$ abbildet und sonst implizit genauso definiert ist wie R_4, dann löst R_5 das EZTP (k, z) für alle $z \in \{h_{1,4b+2}, \ldots, h_{1,5b}\}$. Dies komplettiert den Beweis für die τ-Lösbarkeit von k.

Es bleibt noch die τ-Lösbarkeit von w zu vervollständigen: Der folgende τ-Rumpf $R_6 = (sup_6, sig_6)$ nutzt ebenso das Eins-in-drei-Modell S und löst (w, z) für alle Zustände $z \in Z(H_1) \setminus \{h_{1,6b+1}\}$, an denen w nicht stattfindet: für alle $z \in Z_\iota$, wenn $z \in \{c_{0,0}, \ldots, c_{3,0}, g_{0,0}, \ldots, g_{2m-1,0}\}$, dann $sup_6(z) = b$, sonst $sup_6(z) = 0$; außerdem $\mathcal{E}^{R_6}_{0,1} = \{k\}$ und $\mathcal{E}^{R_6}_{b,b} = \{w\}$ und $\mathcal{E}^{R_6}_{b,0} = \{o_0\}$ und $\mathcal{E}^{R_6}_{1,0} = \{o_1\} \cup S$ und $\mathcal{E}^{R_6}_{0,b} = \{k_0, \ldots, k_3\}$ und $\mathcal{E}^{R_6}_{0,0} = \underline{E} \setminus (\mathcal{E}^{R_6}_{0,1} \cup \mathcal{E}^{R_6}_{b,b} \cup \mathcal{E}^{R_6}_{b,0} \cup \mathcal{E}^{R_6}_{1,0} \cup \mathcal{E}^{R_6}_{0,b})$.

Der folgende τ-Rumpf $R_7 = (sup_7, sig_7)$ löst $(w, h_{1,6b+1})$: $sup_7(z) = b$ für alle $z \in Z_\iota$ und $\mathcal{E}^{R_7}_{b,b} = \{w\}$ und $\mathcal{E}^{R_7}_{1,0} = \{o_1\}$ und $\mathcal{E}^{R_7}_{0,0} = \underline{E} \setminus \{w, o_1\}$. Es folgt die τ-Lösbarkeit von w und somit auch das Lemma. □

Weil die präsentierte Reduktion offenbar polynomiell ist, haben wir mit Lemma 5.1, Lemma 5.7 und Lemma 5.8 sowie der NP-Vollständigkeit von CM 1 IN 3 3SAT bewiesen, dass τ-SPRACHSIMULATION und τ-REALISIERUNG NP-vollständig sind.

Im folgenden Abschnitt zeigen wir für alle $\tau \in \{\tau^b_{PPT}, \tau^b_{PT}\}$, dass τ-EINBETTUNG NP-vollständig ist.

5.1.3 τ^b_{PPT}-EINBETTUNG und τ^b_{PT}-EINBETTUNG

Solange nicht ausdrücklich anders festgelegt, sei im Rest dieses Abschnitts $\tau \in \{\tau^b_{PPT}, \tau^b_{PT}\}$ beliebig aber fest. Wir zeigen gemäß der in Kapitel 3 vorgestellten Strategie, dass τ-EINBETTUNG NP-vollständig ist. Insbesondere werden wir für beide Netztypen mit derselben konkreten Reduktion auskommen, weil die im vorherigen Abschnitt diskutierten Probleme bezüglich der Lösbarkeit von EZTP für die Lösbarkeit von ZTP nicht auftreten.

Das Transitionssystem A_τ hat das folgende Gadget H_2, das insbesondere das ZTP $\alpha = (h_{2,0}, h_{2,b})$ zur Verfügung stellt:

$$H_2 = \; h_{2,0} \xrightarrow{k} \cdots \xrightarrow{k} h_{2,b} \xrightarrow{o_0} h_{2,b+1} \xrightarrow{k} \cdots \xrightarrow{k} h_{2,2b+1} \downarrow o_2 \; h_{2,3b+2} \xleftarrow[k]{} \cdots \xleftarrow[k]{} h_{2,2b+2}$$

Bis auf das Gadget H_0 hat A_τ darüber hinaus alle Gadgets die in Abschnitt 5.1.1 für die Diskussion von τ^b_{PPT}-REALISIERUNG eingeführt wurden. Insbesondere ist A_τ die

lineare Verbindung der folgenden Basis

$$B_\tau = (H_2, D_0, \ldots, D_3, F_0, \ldots, F_{2m-1}, G_0, \ldots, G_{2m-1}, Q_0, \ldots, Q_{m-1}, T_0, \ldots, T_{m-1})$$

Wie bisher adressieren wir mit Z_ι beziehungsweise $\underline{E}$ die Initialzustände beziehungsweise die Ereignisse der Gadgets von B_τ.

Einerseits impliziert die τ-ZTE von A_τ ein gesuchtes Modell für $(\mathfrak{U}, M)$:

Lemma 5.9. *Wenn A_τ die τ-ZTE hat, dann hat $(\mathfrak{U}, M)$ ein Eins-in-drei-Modell.*

Beweis. Wenn A_τ die τ-ZTE hat, dann gibt es eine τ-Region, die α löst. Sei $R = (sup, sig)$ eine solche Region. Wenn $sig^-(k) = sig^+(k)$, dann gilt $sup(h_{2,0}) = sup(h_{2,b})$, ein Widerspruch. Mit Lemma 5.2 folgt deshalb $sig(k) \in \{(1,0),(0,1)\}$. Wir starten mit dem Fall $sig(k) = (1,0)$ und erhalten sofort $sup(h_{2,b}) = sup(h_{2,2b+1}) = 0$ und $sup(h_{2,b+1}) = sup(h_{2,2b+2}) = b$. Dies impliziert $sig(o_0) = sig(o_1) = (0,b)$. Wie bereits für den Beweis von Lemma 5.5 in Abschnitt 5.1.1 diskutiert, folgt daraus, dass $S = \{X \in \mathfrak{U} \mid sig(X) = (0,1)\}$ ein Eins-in-drei-Modell von $(\mathfrak{U}, M)$ ist.

Analog führt der Fall $sig(k) = (0,1)$ dazu, dass die Menge $S = \{X \in \mathfrak{U} \mid sig(X) = (1,0)\}$ ein Eins-in-drei-Modell von $(\mathfrak{U}, M)$ definiert. □

Anderseits impliziert so ein Modell einen Zeugen für die τ-ZTE von A_τ:

Lemma 5.10. *Wenn $(\mathfrak{U}, M)$ ein Eins-in-drei-Modell hat, dann hat A_τ die τ-ZTE.*

Beweis. Es genügt zu zeigen, das alle ZTP der Basis B_τ τ-lösbar sind, denn wegen Lemma 5.3 impliziert dies, dass A_τ die τ-ZTE hat.

Seien $e \in \underline{E} \setminus \{k\}$ und $z \in \underline{Z} \setminus Z(H_2)$ beliebig aber fest. Offenbar ist e dünn verteilt und für das EZTP (e,z) gilt eine der Bedingungen von Lemma 5.4. Daher ist (e,z) gemäß dieses Lemmas τ-lösbar. Weil k nur in H_2 stattfindet, impliziert dies mit der Beliebigkeit von e und z, dass es eine Menge $\mathcal{R}$ von τ-Rümpfen gibt, welche die τ-Lösbarkeit aller EZTP (e,z) von B_τ bezeugen, für die $e \neq k$ und $z \notin Z(H_2)$. Weil jedes dieser Gadgets linear ist, bezeugt $\mathcal{R}$ gemäß Lemma 2.5 auch, dass alle ZTP (z,z') von B_τ τ-lösbar sind, für die $z, z' \notin Z(H)$ gilt.

Wegen Lemma 5.3 genügt es deshalb zu zeigen, dass es für alle ZTP (z,z') mit $z \neq z' \in Z(H_2)$ einen τ-Rumpf $R = (sup, sig)$ von B_τ mit $sup(z) \neq sup(z')$ gibt. Die entsprechenden Rümpfe lassen sich ähnlich zu denen für Abschnitt 5.1.1 definieren. Insbesondere lässt sich das ZTP $(h_{2,0}, h_{2,b})$ durch einen Rumpf lösen, der implizit analog zu dem Rumpf $R_1 = (sup_1, sig_1)$ aus Fakt 5.2 definiert ist, der das EZTP $(k, h_{1,2b+4})$ löst: man braucht nur $sup_1(h_{1,0}) = 0$ durch $sup_1(h_{2,0}) = 0$ ersetzen und die Ereignisse y_0 und y_1 ignorieren. Der resultierende Rumpf löst (z,z') für alle $z \neq z' \in \{h_{2,0}, \ldots, h_{2,b}\}$ und $z \neq z' \in \{h_{2,b+1}, \ldots, h_{2,2b+1}\}$ und $z \neq z' \in \{h_{2,2b+2}, \ldots, h_{2,3b+2}\}$.

Darüber hinaus ist es leicht zu sehen, dass alle Zustände aus $\{h_{2,0}, \ldots, h_{2,b}\}$ sich von allen Zuständen aus $\{h_{2,b+1}, \ldots, h_{2,2b+1}\} \cup \{h_{2,2b+2}, \ldots, h_{2,3b+2}\}$ sowie dass alle Zustände aus $\{h_{2,b+1}, \ldots, h_{2,2b+1}\}$ sich von allen Zuständen aus $\{h_{2,2b+2}, \ldots, h_{2,3b+2}\}$ trennen lassen. Wir verzichten deshalb auf eine vollständige Auflistung der entsprechenden Rümpfe. □

Weil die Reduktion polynomiell ist, haben wir mit Lemma 5.1 und Lemma 5.9 sowie Lemma 5.10 schließlich bewiesen, dass τ-EINBETTUNG NP-vollständig ist. Insgesamt vervollständigt dies mit den Ergebnissen aus Abschnitt 5.1.1 und Abschnitt 5.1.2 den Beweis von Satz 5.1.

Der folgende Abschnitt adressiert die Komplexität klassischen Syntheseprobleme für die gruppenerweiterten b-beschränkten P/T-Netze.

5.2 $\mathbb{Z}_{b+1}$-erweiterte b-beschränkte Netztypen

In diesem Abschnitt charakterisieren wir die Komplexität der klassischen Syntheseprobleme für die (puren) $\mathbb{Z}_{b+1}$-erweiterten b-beschränkten Netztypen. Der folgende Satz fasst die entsprechenden Ergebnisse zusammen:

Satz 5.2. *Wenn $b \in \mathbb{N}$ und $\tau \in \{\tau^b_{\mathbb{Z}PPT}, \tau^b_{\mathbb{Z}PT}\}$, dann gelten die folgenden Aussagen:*

1. *τ-SPRACHSIMULATION und τ-REALISIERUNG sind NP-vollständig, wenn $b \geq 2$.*
2. *τ-EINBETTUNG ist in Polynomialzeit entscheidbar.*
3. *Wenn $b \leq 1$, dann sind auch τ-SPRACHSIMULATION und τ-REALISIERUNG in Polynomialzeit entscheidbar.*

Die weitere Struktur dieses Abschnitts ist wie folgt: Im nächsten Abschnitt 5.2.1 beweisen wir Satz 5.2.1 mit Hilfe der in Kapitel 3 vorgestellten Strategie. Danach wird in Abschnitt 5.2.2 der Beweis von Satz 5.2.2 diskutiert. Wegen der in Fakt 5.1 bewiesenen Kongruenz von $\tau^1_{\mathbb{Z}PPT}$ und dem booleschen Netztyp $\{\mathsf{nop}, \mathsf{inp}, \mathsf{out}, \mathsf{swap}\}$, folgt aus dieser Diskussion auch der schuldig gebliebene Beweis für Lemma 4.4 aus Abschnitt 4.1.2. Weiterhin werden in diesem Abschnitt auch die notwendigen Begrifflichkeiten entwickelt, um Satz 5.2.3 beweisen zu können. In Abschnitt 5.2.3 wird schließlich Satz 5.2.3 bewiesen und die entsprechende Argumentation impliziert darüber hinaus den Beweis von Lemma 4.6 aus Abschnitt 4.1.2.

5.2.1 SPRACHSIMULATION und REALISIERUNG für gruppenerweiterte b-beschränkte Netze mit $b \geq 2$

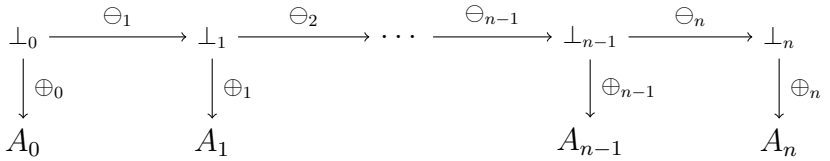
Im Folgenden werden wir Satz 5.2.1 beweisen. Solange nicht explizit anders vereinbart, seien im Rest diese Abschnitts $b \geq 2$ und $\tau \in \{\tau^b_{\mathbb{Z}PPT}, \tau^b_{\mathbb{Z}PT}\}$ beliebig aber fest.

Der Beweis wird mittels der in Kapitel 3 eingeführten Strategie geführt und resultiert in einem Transitionssystem A_τ. Um A_τ zu erhalten werden Gadgets –die eigentlichen Funktionsträger der Reduktion– zu einem Transitionssystem verbunden. Gemäß Definition 5.6 definieren die Gadgets eine Basis B_τ. Das Transitionssystem A_τ resultiert dann gemäß der folgenden Definition aus der *Perlekettenverbindung* von B_τ:

Definition 5.12 (Kettenverbindung)**.** Es sei $B = (A_0, \ldots, A_n)$ eine Basis, wobei $A_i = (Z_i, E_i, \delta_i, \iota_i)$ für alle $i \in \{0, \ldots, n\}$. Außerdem seien $\bot = \{\bot_0, \ldots, \bot_n\}$ beziehungsweise $\oplus = \{\oplus_0, \ldots, \oplus_n\}$ und $\ominus = \{\ominus_1, \ldots, \ominus_n\}$ Mengen paarweise verschiedener Zustände beziehungsweise Ereignisse, die jeweils zu $\underline{Z} \cup \underline{E}$ disjunkt sind. Die *Ketten-Verbindung* von B ist das Transitionssystem $A = (Z, E, \delta, \bot_0)$, mit der Zustandsmenge $Z = \underline{Z} \cup \bot$ sowie der Ereignismenge $E = \underline{E} \cup \oplus \cup \ominus$, dessen Übergangsfunktion δ für alle $e \in E$ und alle $z \in Z$ wie folgt definiert ist:

$$\delta(z,e) = \begin{cases} \delta_i(z,e), & \text{falls } z \in Z_i \text{ und } e \in E_i \text{ mit } i \in \{0,\ldots,n\} \\ \iota_i & \text{falls } z = \bot_i \text{ und } e = \oplus_i \text{ mit } i \in \{0,\ldots,n\} \\ \bot_{i+1}, & \text{falls } z = \bot_i \text{ und } e = \ominus_{i+1} \text{ mit } i \in \{0,\ldots,n-1\} \\ \text{undefiniert}, & \text{sonst} \end{cases}$$

Bemerkung 5.4. Die Kettenverbindung lässt sich wie folgt skizzieren:



Lemma 5.11. *Sei $B = (A_0, \ldots, A_n)$ eine Basis, wobei $A_i = (Z_i, E_i, \delta_i, \iota_i)$ für alle $i \in \{0, \ldots, n\}$, und $A = (Z, E, \delta, \bot_0)$ die Kettenverbindung von B; für jedes Ereignis $e \in \underline{E}$ gebe es einen Zustand $z \in \underline{Z}$ mit $\neg z \xrightarrow{e}$. Wenn jedes EZTP beziehungsweise ZTP von B τ-lösbar ist, dann hat A die τ-EZTP beziehungsweise die τ-ZTE.*

Beweis. Es sei (a, q) ein EZTP beziehungsweise (p, q) ein ZTP von B. Ein τ-Rumpf $R = (sup, sig)$ von B, der (a, q) beziehungsweise (p, q) löst, kann für alle $z \in Z$ und alle $e \in E$ auf folgende Art zu einer τ-Region $R' = (sup', sig')$ von A erweitert werden, die (a, q) beziehungsweise (p, q) löst:

$$sup'(z) = \begin{cases} sup(z), & \text{wenn } z \in \underline{Z}, \\ sup(q), & \text{wenn } q \in \bot \end{cases}$$

$$sig'(e) = \begin{cases} sig(e), & \text{wenn } e \in \underline{E}, \\ 0, & \text{wenn } e \in \ominus, \\ (sup(q) - sup(\iota_i), 0), & \text{wenn } e = \oplus_i \text{ und } sup(\iota_i) < sup(q) \text{ und } 0 \le i \le n, \\ (0, sup(\iota_i) - sup(q)), & \text{wenn } e = \oplus_i \text{ und } sup(\iota_i) \ge sup(q) \text{ und } 0 \le i \le n. \end{cases}$$

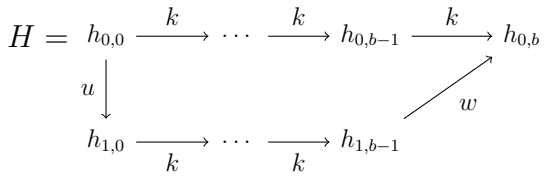
Weil R' das EZTP (a, q) löst und $sup'(\bot_i) = sup(q)$ gilt, wird auch $(a, \bot_i)$ für alle $i \in \{0, \ldots, n\}$ gelöst. Da es für jedes $e \in \underline{E}$ einen Zustand $z \in \underline{Z}$ mit $\neg z \xrightarrow{e}$ gibt und das zugehörige EZTP gelöst werden kann, folgt bereits, dass alle EZTP (e, z) von A mit

$e \in \underline{E}$ und $z \in Z$ gelöst werden können. Es ist leicht zu sehen, dass ein EZTP (e, z) für alle $e \in \ominus \cup \oplus$ gelöst werden kann. Somit hat A die τ-EZTE.

Aus der Beliebigkeit von (p, q) und dessen Lösbarkeit folgt, dass alle ZTP (z, z') von A mit $z, z' \in Z_i$ für alle $i \in \{0, \ldots, n\}$ τ-lösbar sind. Ebenso ist ein beliebiges aber festes ZTP (p, q) von A auch dann lösbar, wenn $p \in Z_i$ und $q \in Z_j$ mit $i \neq j \in \{0, \ldots, n\}$, wie die folgende τ-Region (sup, sig) zeigt: für alle $z \in Z$, wenn $z \in Z_i$, dann $sup(z) = 1$, sonst $sup(z) = 0$; für alle $e \in E$, wenn $e = \oplus_i$, dann $sig(e) = (0, 1)$, sonst $sig(e) = 0$. Schließlich sieht man leicht, dass ein ZTP (p, q) von A τ-lösbar, wenn $p \in \bot$. □

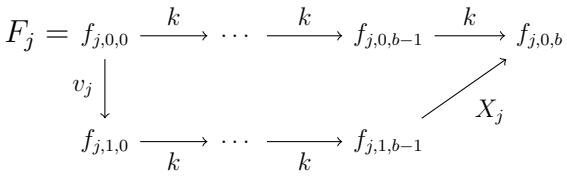
Im Rest dieses Abschnitts werden wir zunächst die Gadgets von A_τ vorstellen und im Anschluss ihre Funktionalität beweisen.

Das Transitionssystem A_τ hat das folgende Gadget H, das insbesondere das EZTP $\alpha = (k, h_{1,b-1})$ liefert:



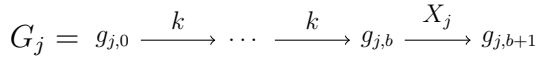
Man beachte, dass das Ereignis k an dem Zustand $h_{0,0}$ b-mal hintereinander stattfindet, während es an dem Zustand $h_{1,0}$ nur $b-1$-mal stattfindet.

Des Weiteren hat A_τ für jedes $j \in \{0, \ldots, m-1\}$ das folgende Gadget F_j auf dem die Variable X_j aus $\mathfrak{U}$ als Ereignis stattfindet:



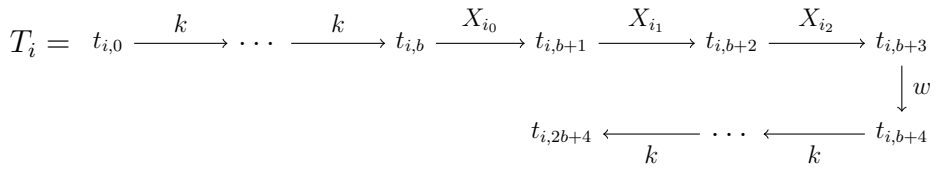
Auch hier beachte man, dass k an dem Zustand $f_{j,0,0}$ b-mal hintereinander und an $f_{j,1,0}$ nur $b-1$-mal hintereinander stattfindet.

Darüber hinaus hat A_τ für alle $j \in \{0, \ldots, m-1\}$ das folgende Gadget G_j auf dem die Variable X_j aus $\mathfrak{U}$ ebenso als Ereignis stattfindet:



Das Ereignis k findet an dem Zustand $g_{j,0}$ b-mal hintereinander statt.

Schließlich hat das Transitionssystem A_τ für jedes $i \in \{0, \ldots, m-1\}$ das folgende Gadget T_i, das die Variablen der Klausel $M_i = \{X_{i_0}, X_{i_1}, X_{i_2}\}$ als Ereignisse verwendet:



A_τ ergibt sich schließlich gemäß Definition 5.12 als die Ketten-Verbindung der Basis

$$B_\tau = (H, F_0, \ldots, F_{m-1}, G_0, \ldots, G_{m-1}, T_0, \ldots, T_{m-1})$$

Wir werden nun zeigen, dass A_τ genau dann eine positive Entscheidung erlaubt, wenn $(\mathfrak{U}, M)$ ein Eins-in-drei-Modell hat. Das folgende Lemma adressiert die Implikation von links nach rechts:

Lemma 5.12. *Wenn A_τ die τ-EZTE hat, dann hat $(\mathfrak{U}, M)$ ein Eins-in-drei-Modell.*

Beweis. Es sei $E_0 = \{(m, m) \mid m \in \{1, \ldots, b\}\} \cup \{0\}$ die Menge der Ereignisse, deren Stattfinden in τ keinen Zustandswechsel verursacht, das heißt, $z \xrightarrow{e} z' \in \tau$ impliziert $z = z'$ für alle $e \in E_0$.

Wenn A_τ die τ-EZTE hat, dann gibt es eine τ-Region, die $\alpha = (k, h_{1,b-1})$ löst. Sei $R = (sup, sig)$ eine solche Region, das heißt $\neg sup(h_{1,b-1}) \xrightarrow{sig(k)}$ und folglich $sig(k) \notin \{0, \ldots, b\}$. Wir argumentieren zunächst, dass $sig(k) \in \{(1,0),(0,1)\}$: Man überlegt sich, dass die Annahme $sig^-(k) = sig^+(k)$ zu dem Widerspruch $sup(h_{1,b-1}) \xrightarrow{sig(k)}$ führt, denn dies impliziert $sup(h_{1,0}) = sup(h_{1,b-1})$. Es gilt also $sig^-(k) \neq sig^+(k)$ und mit Lemma 5.2 erhalten wir $sig(k) \in \{(1,0),(0,1)\}$.

Wir zeigen nun: wenn $sig(k) = (1,0)$, dann ist $S = \{X \in \mathfrak{U} \mid sig(X) = b\}$ ein Eins-in-drei-Modell von $(\mathfrak{U}, M)$. Weil $\neg sup(h_{1,b-1}) \xrightarrow{sig(k)}$, erhalten wir mit $sig(k) = (1,0)$, dass $sup(h_{1,b-1}) = 0$ und, mit Lemma 5.2, auch $sup(h_{0,b}) = 0$. Daraus folgt $0 \xrightarrow{sig(w)} 0$, das heißt $sig(w) = 0$.

Sei $j \in \{0, \ldots, m-1\}$ beliebig aber fest. Die Gadgets F_j und G_j sichern zu, dass entweder $sig(X_j) = 0$ oder $sig(X_j) = b$: Weil k jeweils b-mal an $f_{j,0,0}$ und an $g_{j,0}$ stattfindet, erhalten wir $sup(f_{j,0,b}) = sup(g_{j,b}) = 0$. Man überlegt sich außerdem, dass $sup(f_{j,1,b-1}) \in \{0,1\}$, weil k an $f_{j,1,0}$ $b-1$-mal stattfindet. Wegen $sup(g_{j,b}) = 0$ und $g_{j,b} \xrightarrow{X_j}$ erhalten wir $sig^-(X_j) = 0$. Analog erhalten wir $sig^+(X_j) = 0$, wegen $sup(f_{j,0,b}) = 0$ und $\xrightarrow{X_j} f_{j,0,b}$ und folglich $sig(X_j) \in \{0, \ldots, b\}$. Wenn $sup(f_{j,1,b-1}) = sup(f_{j,0,b}) = 0$, dann ist offenbar $sig(X_j) = 0$. Andernfalls, wenn $sup(f_{j,1,b-1}) = 1$, dann erhalten wir $sig(X_j) = b$, denn es gilt $0 = 1 + sig(X_j) \bmod (b+1)$. Insgesamt folgt mit der Beliebigkeit von j, dass entweder $sig(X_j) = 0$ oder $sig(X_j) = b$ für alle $j \in \{0, \ldots, m-1\}$.

Sei $i \in \{0, \ldots, m-1\}$ beliebig aber fest. Weil $sig(k) = (1,0)$ und $sig(w) = 0$ folgt, dass das Bild P_i^R des Pfades $P_i = t_{i,b} \xrightarrow{X_{i_0}} t_{i,b+1} \xrightarrow{X_{i_1}} t_{i,b+2} \xrightarrow{X_{i_2}} t_{i,b+2}$ unter R ein Pfad von 0

nach b in τ ist. Mit Lemma 5.2 und $sig(X) \in \{0, b\}$ für alle $X \in \{X_{i_0}, X_{i_1}, X_{i_2}\}$ erhalten wir, dass die folgende Kongruenz-Gleichung gilt:

$$(|sig(X_{i_0})| + |sig(X_{i_1})| + |sig(X_{i_2})|) = b \bmod (b+1) \tag{5.1}$$

Dies impliziert, dass es wenigstens ein $X \in \{X_{i_0}, X_{i_1}, X_{i_2}\}$ mit $sig(X) = b$ geben muss, denn $b \neq 0$. Wir zeigen, dass die Annahme, es gäbe mehr als ein Ereignis aus $\{X_{i_0}, X_{i_1}, X_{i_2}\}$ mit Signaturwert b zum Widerspruch führt: Nehmen wir also zunächst an, es gibt genau zwei verschiedene Ereignisse $X, Y \in \{X_{i_0}, X_{i_1}, X_{i_2}\}$ mit $sig(X) = sig(Y) = b$, das heißt $sig(Z) = 0$ für $Z \in \{X_{i_0}, X_{i_1}, X_{i_2}\} \setminus \{X, Y\}$. Weil die Modulo-Kongruenzrelation symmetrisch und transitiv ist, gelten die folgenden Gleichungen:

$$\begin{aligned}
&b = (|sig(X_{i_0})| + |sig(X_{i_1})| + |sig(X_{i_2})|) \bmod (b+1) && |5.1 && (5.2)\\
&(|sig(X_{i_0})| + |sig(X_{i_1})| + |sig(X_{i_2})|) = 2b \bmod (b+1) && |\text{Annahme} && (5.3)\\
&b = 2b \bmod (b+1) && |5.2, 5.3 && (5.4)\\
&2b = (b-1) \bmod (b+1) && |\text{Def. } = && (5.5)\\
&b = (b-1) \bmod (b+1) && |5.4, 5.5 && (5.6)\\
&\exists m \in \mathbb{Z} : m(b+1) = 1 && |5.6, \text{Def. } = && (5.7)
\end{aligned}$$

Mit Gleichung 5.7 erhalten wir den Widerspruch $b = 0$. Wenn wir nun annehmen, es gilt $sig(X_{i_0}) = sig(X_{i_1}) = sig(X_{i_2}) = b$, dann sind die folgenden Gleichungen korrekt:

$$\begin{aligned}
&(|sig(X_{i_0})| + |sig(X_{i_1})| + |sig(X_{i_2})|) = 3b \bmod (b+1) && |\text{Annahme} && (5.8)\\
&b = 3b \bmod (b+1) && |5.2, 5.8 && (5.9)\\
&3b = (b-2) \bmod (b+1) && |\text{Def. } = && (5.10)\\
&b = (b-2) \bmod (b+1) && |5.9, 5.10 && (5.11)\\
&\exists m \in \mathbb{Z} : m(b+1) = 2 && |5.11, \text{Def. } = && (5.12)
\end{aligned}$$

Mit Gleichung 5.12 erhalten wir $b \in \{0, 1\}$. Dies ist ein Widerspruch zu $b \geq 2$. Das heißt es gibt genau ein $X \in \{X_{i_0}, X_{i_1}, X_{i_2}\}$ mit $sig(X) = b$.

Aus der Beliebigkeit von i folgt nun insgesamt $|S \cap M_i| = 1$ für alle $i \in \{0, \ldots, m-1\}$. Somit ist S ein gesuchtes Eins-in-drei-Modell.

Wir argumentieren gestrafft für den Fall $sig(k) = (0, 1)$, der $sup(h_{0,b}) = sup(h_{1,b-1}) = b$ impliziert. Aus $sup(h_{0,b}) = sup(h_{1,b-1}) = b$ erhalten wir $sig^-(w) = sig^+(w)$ und $|sig(w)| = 0$, das heißt $sig(w) \in E_0$. Sei $j \in \{0, \ldots, m-1\}$ beliebig aber fest. Mit $sig(k) = (0, 1)$ erhalten wir $sup(f_{j,0,b}) = sup(g_{j,b}) = b$ und $sup(f_{j,1,b-1}) \in \{b-1, b\}$. Aus $b \xrightarrow{sig(X_j)}$ folgt $sig^-(X_j) \geq sig^+(X_j)$ und aus $\xrightarrow{sig(X_j)} b$ folgt $sig^+(X_j) \geq sig^-(X_j)$, das heißt $sig^-(X_j) = sig^+(X_j)$. Wir argumentieren, dass wenn $sig(X_j) \in \{0, \ldots, b\}$, dann $sig(X_j) \in \{0, 1\}$: Wenn $sup(f_{j,0,b}) = sup(f_{j,1,b-1}) = b$, dann offenbar $sig(X_j) = 0$. Andernfalls, wenn $sup(f_{j,1,b-1}) = b-1$, dann gilt $sig(X_j) = 1$, wegen $b = b-1+|sig(X_j)| \bmod (b+1)$. Weil j beliebig war, gilt nun $X \in E_0 \cup \{1\}$ für alle $X \in \mathfrak{U}$. Sei $i \in \{0, \ldots, m-1\}$ beliebig aber fest. Aus $sig(k) = (0, 1)$ und $sig(w) \in E_0$ folgt $sup(t_{i,b}) = b$ und $sup(t_{i,b+3}) = 0$.

Damit ist das Bild P_i^R des Pfades P_i (wie oben definiert) ein Pfad von b nach 0 in τ. Es gibt also wenigstens ein Ereignis $X \in \{X_{i_0}, X_{i_1}, X_{i_2}\}$ mit $sig(X) = 1$. Analog zu dem Fall $sig(k) = (1,0)$ zeigt man, dass die Annahme, es gäbe mehr als ein solches Ereignis in $\{X_{i_0}, X_{i_1}, X_{i_2}\}$ zum Widerspruch $b \in \{0,1\}$ führt. Es folgt das die Menge $S' = \{X \in \mathfrak{U} \mid sig(X) = 1\}$ ein Eins-in-drei-Modell von $(\mathfrak{U}, M)$ definiert. □

Im Gegenzug ist auch A_τ eine Ja-Instanz, wenn $(\mathfrak{U}, M)$ eine positive Entscheidung erlaubt. Solange nicht ausdrücklich anders vereinbart, sei S im Folgenden ein beliebiges aber festes Eins-in-drei-Modell von $(\mathfrak{U}, M)$. Gemäß Lemma 5.11 genügt es zu zeigen, das jedes EZTP und ZTP der Basis B_τ τ-lösbar ist. Wir werden im Folgenden zunächst die EZTP von B_τ sukzessive diskutieren. Die dafür präsentierten τ-Rümpfe von B_τ lösen im Wesentlichen auch alle ZTP von B_τ. Wie gehabt bezeichnen wir mit $\underline{E}$ und Z_ι die Ereignisse von B_τ und die Initialzustände der Gadgets von B_τ:

$$Z_\iota = \{h_{0,0}, f_{0,0,0}, \ldots, f_{m-1,0,0}, g_{0,0}, \ldots, g_{m-1,0}, t_{0,0}, \ldots, t_{m-1,0}\}$$

Der Einfachheit halber wird ein τ-Rumpf $R = (sup, sig)$ implizit mittels $sup(z)$ für alle $z \in Z_\iota$ sowie sig definiert, wobei wir Ereignisse mit gleichem Signaturwert $(m,n) \in \{0,\ldots,b\}^2$ beziehungsweise $x \in \{0,\ldots,b\}$ wie folgt zusammenfassen: $\mathcal{E}^R_{m,n} = \{e \in \underline{E} \mid sig(e) = (m,n)\}$ und $\mathcal{E}^R_x = \{e \in \underline{E} \mid sig(e) = x\}$.

Fakt 5.5. *Das Ereignis k ist τ-lösbar.*

Beweis. Der folgende τ-Rumpf $R_1 = (sup_1, sig_1)$ nutzt das Eins-in-drei-Modell S und löst (k,z) für alle $z \in \{h_{0,b}, h_{1,b-1}\}$: $sup_1(z) = 0$ für alle $z \in Z_\iota$ und $\mathcal{E}^{R_1}_{0,1} = \{k\}$ und $\mathcal{E}^{R_1}_0 = \{w\} \cup (\mathfrak{U} \setminus S) \cup \{v_j \mid j \in \{0,\ldots,m-1\} \text{ und } X_j \in S\}$ und $\mathcal{E}^{R_1}_1 = \{v_j \mid j \in \{0,\ldots,m-1\} \text{ und } X_j \notin S\} \cup S \cup \{u\}$.

Für jede Sequenz $q_0 \xrightarrow{k} \ldots \xrightarrow{k} q_b$ in A_τ wird darüber hinaus das ZTP (z,z') für alle $z \neq z' \in \{q_0,\ldots,q_b\}$ gelöst.

Der folgende τ-Rumpf $R_2 = (sup_2, sig_2)$ löst (k,z) für alle verbleibenden Zustände aus $z \in \underline{Z}$, an denen k nicht stattfindet: $sup_2(z) = 0$ für alle $z \in Z_\iota$ und $\mathcal{E}^{R_2}_{0,1} = \{k\}$ und $\mathcal{E}^{R_2}_1 = \{w\} \cup \{v_0,\ldots,v_{m-1}\}$ und $\mathcal{E}^{R_2}_0 = \underline{E} \setminus (\mathcal{E}^{R_2}_{0,1} \cup \mathcal{E}^{R_2}_1)$. □

Fakt 5.6. *Das Ereignis w ist τ-lösbar.*

Beweis. Es sei $i \in \{0,\ldots,m-1\}$ beliebig aber fest und $j \neq \ell \in \{0,\ldots,m-1\} \setminus \{i\}$ die beiden Indizes, sodass $X_{i_2} \in E(T_j)$ und $X_{i_2} \in E(T_\ell)$, das heißt, j und ℓ wählen die beiden weiteren Vorkommen von X_{i_2} in M.

Der folgende τ-Rumpf $R_1 = (sup_1, sig_1)$ löst (w,z) für alle $z \in \{h_{0,0}\} \cup Z(T_i)$: für alle $z \in Z_\iota$, wenn $z \in \{h_{0,0}, t_{i,0}, t_{j,0}, t_{\ell,0}\}$, dann $sup_1(z) = b$; wenn $z \in \{t_{l,0} \mid l \in \{0,\ldots,m-1\} \setminus \{i,j,\ell\}\}$, dann $sup_1(z) = 0$; andernfalls $sup_1(z) = 1$; außerdem $\mathcal{E}^{R_1}_{0,b} = \{w\}$ und $\mathcal{E}^{R_1}_1 = \{u, X_{i_2}\}$ und $\mathcal{E}^{R_1}_b = \{v_{i_2}\}$ und $\mathcal{E}^{R_1}_0 = \underline{E} \setminus (\mathcal{E}^{R_1}_{0,b} \cup \mathcal{E}^{R_1}_1 \cup \mathcal{E}^{R_1}_b)$.

Wenn $z \in \{h_{0,0},\ldots,h_{0,b}\}$ und $z' \in \{h_{1,0},\ldots,h_{1,b-1}\}$ oder wenn $z \in \{f_{i_2,0,0},\ldots,f_{i_2,0,b}\}$ und $z' \in \{f_{i_2,1,0},\ldots,f_{i_2,1,b-1}\}$, dann löst R_1 auch das ZTP (z,z'). Weil i beliebig war,

ist (z, z') somit für alle $z \in \{f_{i,0,0}, \ldots, f_{i,0,b}\}$ und $z' \in \{f_{i,1,0}, \ldots, f_{i,1,b-1}\}$ und alle $i \in \{0, \ldots, m-1\}$ τ-lösbar. Damit zeigen der Rumpf R_1 aus Fakt 5.5 sowie der gerade definierte Rumpf R_1 bereits die τ-Lösbarkeit aller ZTP von H und $F_0, \ldots F_{m-1}$.

Der folgende τ-Rumpf $R_2 = (sup_2, sig_2)$ löst (w, z) für alle $z \in Z(H) \setminus \{h_{0,0}\}$: für alle $z \in Z_\iota$, wenn $z \in \{t_{i,0} \mid i \in \{0, \ldots, m-1\}\}$, dann $sup_2(z) = 1$, sonst $sup_2(z) = 0$; des Weiteren $\mathcal{E}_{0,b}^{R_2} = \{w\}$ und $\mathcal{E}_1^{R_2} = \{k\}$ und $\mathcal{E}_2^{R_2} = \{u\}$ und $\mathcal{E}_0^{R_2} = \underline{E} \setminus (\mathcal{E}_{0,b}^{R_2} \cup \mathcal{E}_1^{R_2} \cup \mathcal{E}_2^{R_2})$.

Der folgende τ-Rumpf $R_3 = (sup_3, sig_3)$ löst (w, z) für alle verbleibenden $z \in \underline{Z}$, an denen w nicht stattfindet: für alle $z \in Z_\iota$, wenn $z \in \{t_{i,0} \mid i \in \{0, \ldots, m-1\}\}$, dann $sup_3(z) = 0$, sonst $sup_3(z) = b$; außerdem $\mathcal{E}_{0,b}^{R_3} = \{w\}$ und $\mathcal{E}_1^{R_3} = \{u\}$ und $\mathcal{E}_0^{R_3} = \underline{E} \setminus \{w, u\}$. Dies beweist den Fakt. □

Fakt 5.7. *Das Ereignis u ist τ-lösbar.*

Beweis. Der folgende τ-Rumpf $R_1 = (sup_1, sig_1)$ löst (u, z) für alle $z \in \{h_{0,1}, \ldots, h_{0,b}\}$, dabei nutzen wir, dass $2b + 1 = b \bmod b + 1$: $sup_1(z) = 0$ für alle $z \in Z_\iota$ und $\mathcal{E}_{0,b}^{R_1} = \{u\}$ und $\mathcal{E}_1^{R_1} = \{k\}$ und $\mathcal{E}_2^{R_1} = \{w\}$ und $\mathcal{E}_0^{R_1} = \underline{E} \setminus \{k, u, w\}$.

Der folgende τ-Rumpf $R_2 = (sup_2, sig_2)$ löst (u, z) für alle $z \in \underline{Z} \setminus \{h_{0,1}, \ldots, h_{0,b}\}$: für alle $z \in Z_\iota$, wenn $z = h_{0,0}$, dann $sup_2(z) = 0$, andernfalls $sup_2(z) = 1$; darüber hinaus $\mathcal{E}_{0,b}^{R_2} = \{u\}$ und $\mathcal{E}_1^{R_2} = \{w\}$ und $\mathcal{E}_0^{R_2} = \underline{E} \setminus \{w, u\}$. □

Fakt 5.8. *Wenn $e \in \{v_0, \ldots, v_{m-1}\}$ oder $e \in \mathfrak{U}$, dann ist e τ-lösbar.*

Beweis. Sei $i \in \{0, \ldots, m-1\}$ beliebig aber fest. Der folgende τ-Rumpf $R_1 = (sup_1, sig_1)$ löst (v_i, z) für alle $z \in \{f_{i,0,1}, \ldots, f_{i,0,b}\}$, wobei wir erneut $2b+1 = b \bmod b+1$ ausnutzen: $sup_1(z) = 0$ für alle $z \in Z_\iota$ und $\mathcal{E}_{0,b}^{R_1} = \{v_0, \ldots, v_{m-1}\}$ und $\mathcal{E}_1^{R_1} = \{w, k\}$ und $\mathcal{E}_2^{R_1} = \mathfrak{U}$ und $\mathcal{E}_0^{R_1} = \underline{E} \setminus (\mathcal{E}_{0,b}^{R_1} \cup \mathcal{E}_1^{R_1} \cup \mathcal{E}_2^{R_1})$.

Der folgende τ-Rumpf $R_2 = (sup_2, sig_2)$ löst (v_i, z) für alle verbleibenden $z \in \underline{Z} \setminus \{f_{i,0,1}, \ldots, f_{i,0,b}\}$, an denen v_i nicht stattfindet, wobei wir wieder $2b + 1 = b \bmod b + 1$ beachten: für alle $z \in Z_\iota$, wenn $z = f_{i,0,0}$, dann $sup_2(z) = 0$, andernfalls $sup_2(z) = 1$; außerdem $\mathcal{E}_{0,b}^{R_2} = \{v_i\}$ und $\mathcal{E}_1^{R_2} = \{X_i\}$ und $\mathcal{E}_0^{R_2} = \underline{E} \setminus \{v_i, X_i\}$. Weil i beliebig war, folgt für alle $e \in \{v_0, \ldots, v_{m-1}\}$, dass e τ-lösbar ist.

Wenn $i \in \{0, \ldots, m-1\}$ beliebig aber fest ist, dann findet man leicht heraus, dass X_i τ-lösbar ist, denn dieses Ereignis findet in seinen jeweiligen Gadgets höchstens einmal statt. Wir verzichten daher auf eine explizite Fallanalyse. Dies beweist das Lemma. □

Es resultiert das folgende Lemma aus den bisher bewiesenen Fakten:

Lemma 5.13. *Wenn $(\mathfrak{U}, M)$ ein Eins-in-drei-Modell hat, dann hat A_τ die τ-ZTE und die τ-EZTE.*

Beweis. Wie bereits diskutiert, genügt es zu zeigen, dass alle EZTP und alle ZTP von B_τ τ-lösbar sind. Aus Fakt 5.5, Fakt 5.6, Fakt 5.7 und Fakt 5.8 folgt, dass alle Ereignisse von B_τ τ-lösbar sind.

Darüber beweisen der τ-Rumpf R_1 aus Fakt 5.5 und R_1 aus Fakt 5.6, dass das ZTP (z, z') für alle $z \neq z' \in Z(H)$ τ-lösbar ist. Wenn $i \in \{0, \ldots, m-1\}$ beliebig aber fest ist, dann zeigen R_1 aus Fakt 5.5 und R_2 aus Fakt 5.8, dass das ZTP (z, z') für alle $z \neq z' \in Z(F_i)$ τ-lösbar ist. Dies zeigt die τ-Lösbarkeit aller Zustände der Gadegts $F_0, \ldots, F_{m-1}$.

Die anderen Gadgets von B_τ sind aber lineare Transitionssysteme. Weil alle Ereignisse dieser Gadgets τ-lösbar sind, erhalten wir die τ-Lösbarkeit ihrer ZTP mit Lemma 2.5. Dies beweist das Lemma. □

Weil die Reduktion offenbar polynomiell ist und die adressierten Entscheidungsprobleme in NP sind erhalten wir die Aussage von Satz 5.2.1 schließlich mit Lemma 5.12 und Lemma 5.13.

5.2.2 EINBETTUNG für gruppenerweiterte b-beschränkte Netze

In diesem Abschnitt diskutieren wir den Beweis von Satz 5.2.2. In [Sch96] hat Schmitt die Komplexität der klassischen Syntheseprobleme τ-EINBETTUNG, τ-SPRACHSIMULATION und τ-REALISIERUNG für (den Netztypen der) Flip-Flop-Netze ($\{\mathsf{nop}, \mathsf{inp}, \mathsf{out}, \mathsf{swap}\}$) untersucht. Schmitt zeigt, dass die entsprechenden Entscheidungsprobleme in Polynomialzeit lösbar sind, indem die Lösbarkeit von Trennungsproblemen auf die Lösbarkeit von Gleichungssystemen über der additiven Gruppe $\mathbb{Z}_2$ der ganzen Zahlen modulo 2 reduziert wird. Für den Beweis von Satz 5.2.2 erweitern wir diesen Ansatz auf natürliche Weise und nutzen, dass sich Gleichungssysteme über $\mathbb{Z}_{b+1}$ für alle $b \in \mathbb{N}$ in Polynomialzeit lösen lassen:

Lemma 5.14 ([GR02]). *Es seien $k, n, b \in \mathbb{N}$ und $K \in \mathbb{Z}_{b+1}^{k \times n}$ sowie $c \in \mathbb{Z}_{b+1}^{k}$. Es gibt einen Algorithmus, der mit Zeitaufwand $\mathcal{O}(nk \cdot max\{n, k\})$ entscheidet, ob es ein Element $\boldsymbol{x} \in \mathbb{Z}_{b+1}^{n}$ mit $K \cdot \boldsymbol{x} = c$ gibt.*

Solange nicht explizit anders vereinbart, seien im weiteren Verlauf dieses Abschnitts $b \in \mathbb{N}$, $\tau \in \{\tau_{\mathbb{Z}PPT}, \tau_{\mathbb{Z}PT}\}$ und $A = (Z, E, \delta, \iota)$ ein Transitionssystem mit wenigstens zwei Zuständen und der Ereignismenge $E = \{e_1, \ldots, e_n\}$ beliebig aber fest gewählt.

Die Herangehensweise für den Beweis von Satz 5.2.2 lässt sich wie folgt skizzieren: Für jedes ZTP α von A werden höchstens b lineare Systeme von Gleichungen über $\mathbb{Z}_{b+1}$ konstruiert. Die Konstruktion ist derart, dass α genau dann lösbar ist, wenn wenigstens eines dieser Gleichungssysteme lösbar ist. Da A höchstens $|Z|^2$ ZTP hat und somit höchstens $b \cdot |Z|^2$ Gleichungssysteme auf ihre Lösbarkeit geprüft werden müssen, liefert dieser Ansatz mit der Aussage von Lemma 5.14 eine polynomielles Entscheidungsverfahren für τ-EINBETTUNG. Darüber hinaus impliziert jede Lösung eines Gleichungssystems, welches aus α resultiert, bereits eine τ-Region $R = (sup, sig)$, die α löst. Genauer gesagt werden die Einträge eines Lösungselements $\mathbf{x} = (x_1, \ldots, x_n)$ eines entsprechenden linearen Gleichungssystems $K \cdot \mathbf{x} = c$ mit Koeffizientenmatrix K als Signaturwerte der Ereignisse von

A interpretiert: $sig(e_i) = x_i$ für alle $i \in \{1, \ldots, n\}$. Offenbar beschränkt dies den Wertebereich der Signatur auf die Elemente aus $\mathbb{Z}_{b+1}$, sodass wir es mit sogenannten *abstrakten* Signaturen beziehungsweise *abstrakten* Regionen zu tun haben:

Definition 5.13 (Abstrakte Region)**.** Eine τ-Region $R = (sup, sig)$ von A heißt *abstrakt*, wenn $sig(e) \in \{0, \ldots, b\}$ für alle $e \in E$. Wenn R eine abstrakte Region ist, dann nennen wir *sig abstrakte Signatur*.

Notation 5.1 (Abstrakte Regionen)**.** Im Folgenden schreiben wir *abs* statt *sig* für abstrakte Signaturen und (sup, abs) für abstrakte Regionen. Darüber hinaus identifizieren wir $abs = (abs(e_1), \ldots, abs(e_n))$.

Bemerkung 5.5. Wenn $R = (sup, abs)$ eine abstrakte Region von A ist, dann gilt $sup(z') = sup(z) + abs(e) \bmod (b+1)$ für jede Kante $z \xrightarrow{e} z'$ von A gemäß Definition 2.18 und der Definition von τ. Daraus folgt für alle $i \in \{1, \ldots, n\}$, dass $sup(z') + i = sup(z) + abs(e) + i \bmod (b+1)$ für alle Kanten $z \xrightarrow{e} z'$ von A gilt, sodass $sup_i(\iota) = sup(\iota) + i$ und *abs* gemäß Bemerkung 2.7 implizit eine weitere Region $R_i = (sup_i, abs)$ von A definieren. Weil der Support jeder Region dem Initialzustand ι einen Wert aus $\{0, \ldots, b\}$ zuordnet, gibt es deshalb genau $b + 1$ abstrakte τ-Regionen von A mit der Signatur *abs*.

Um die Gleichungen eines zu einem Trennungsproblem α gehörendes Gleichungssystems $K \cdot \mathbf{x} = c$ zu erhalten, bestimmen wir zunächst einen beliebigen aber festen *Spannbaum* T von A. Jede sogenannte *Sehne* von T definiert einen *Fundamentalkreis*, der eine Gleichung zu $K \cdot \mathbf{x} = c$ beiträgt. Diese durch Fundamentalkreise definierten Gleichungen stellen sicher, dass eine Lösung $\mathbf{x}$ des Systems eine abstrakte Signatur von A definiert. Sie sind für jedes Trennungsproblem Bestandteil der erwähnten Gleichungssysteme. Alle weiteren Gleichungen sind durch das jeweilige konkret betrachtete Trennungsproblem α definiert. Sie stellen sicher, dass die durch eine Lösung implizierte definierte Region, das Problem α tatsächlich löst.

Ein Spannbaum eines Transitionssystems $A = (Z, E, \delta, \iota)$ ist ein Teil-Transitionssystem von A, dessen unterliegender, ungerichteter und unbeschrifteter Graph ein Baum mit Wurzel ι im üblichen graph-theoretischen Sinne ist:

Definition 5.14 (Spannbaum, Sehne)**.** Ein *Spannbaum* von $A = (Z, E, \delta, \iota)$ ist ein schleifenfreies Transitionssystem $T = (Z, E', \delta', \iota)$ mit den gleichen Zuständen wie A und einer eingeschränkten Übergangsfunktion, sodass für alle $e \in E'$ und alle $z, z' \in Z$ gilt: (1) wenn $z \xrightarrow{e} z'$ eine Kante in T ist, dann ist $z \xrightarrow{e} z'$ eine Kante in A; (2) es gibt genau einen initialisierten Pfad P_z von ι nach z in T. Jede Kante $z \xrightarrow{e} z'$ von A, die nicht in T ist, nennen wir *Sehne* für T.

Mit dem Algorithmus von Tarjan [Tar77] kann ein Spannbaum von A mit Zeitaufwand $\mathcal{O}(|E| \cdot |Z|^3)$ konstruiert werden. Solange nicht ausdrücklich anders vereinbart, sei A' im Folgenden ein beliebiger aber fester Spannbaum von A. Gemäß Definition 5.14 gibt es für jeden Zustand $z \in Z$ genau einen gerichteten Pfad von ι nach z in A, wobei dieser Pfad

der leere Pfad $\iota \xrightarrow{\varepsilon} \iota$ ist, wenn $z = \iota$ (siehe Definition 2.5), denn A' ist nach Definition schleifenfrei. Um für jedes Ereignis e von A die Anzahl seiner Vorkommen modulo b entlang dieses Pfades in A' zu zählen, nutzen wir einen *Parikh-Vektor*:

Definition 5.15 (Parikh-Vektor)**.** Sei $z \in Z$ und sei $P_z = \iota \xrightarrow{e_{i_1}} \dots \xrightarrow{e_{i_m}} z$ der eindeutige Pfad von ι nach z in A'. Der *Parikh-Vektor* σ_z (von z in A') ist die Abbildung $\sigma_z : \{e_1, \dots, e_n\} \to \{0, \dots, b\}$, die für jedes Ereignis $e \in \{e_1, \dots, e_n\}$ definiert ist durch:

$$\sigma_z(e) = |\{\ell \in \{1, \dots, m\} \mid e_{i_\ell} = e\}| \bmod (b+1)$$

Der Bequemlichkeit halber identifizieren wir $\sigma_z = (\sigma_z(e_1), \dots, \sigma_z(e_n))$.

Die Sehnen eines Spannbaums definieren sogenannte *Fundamentalkreise*. Diese liefern Gleichungen, deren Bedingungen dazu führen, dass eine Lösung des Gleichungssystems eine abstrakte Signatur von A ist:

Definition 5.16 (Fundamentalkreis)**.** Der *Fundamentalkreis* σ_t einer Sehne $t = z \xrightarrow{e} z'$ ist die Abbildung $\sigma_t : \{e_1, \dots, e_n\} \to \{0, \dots, b\}$, die für alle $i \in \{1, \dots, n\}$ wie folgt definiert ist:

$$\sigma_t(e_i) = \begin{cases} \sigma_z(e_i) - \sigma_{z'}(e_i) \bmod (b+1), & \text{falls } e \neq e_i \\ \sigma_z(e_i) - \sigma_{z'}(e_i) + 1 \bmod (b+1), & \text{sonst.} \end{cases}$$

Der Bequemlichkeit halber identifizieren wir $\sigma_t = (\sigma_t(e_1), \dots, \sigma_t(e_n))$.

Definition 5.17 (Produktsumme)**.** Für zwei Tupel $\mathbf{x} = (x_1, \dots, x_n), \mathbf{y} = (y_1, \dots, y_n) \in \mathbb{Z}_{b+1}^n$ definieren wir die *Produktsumme* durch $\mathbf{x} \cdot \mathbf{y} = (x_1 \cdot y_1 + \dots + x_n \cdot y_n) \bmod (b+1)$.

Für eine Sehne $t = z \xrightarrow{e} z'$ gibt es die eindeutigen Pfade $P_z = \iota \xrightarrow{e_{i_1}} \dots \xrightarrow{e_{i_n}} z$ beziehungsweise $P_{z'} = \iota \xrightarrow{e'_{i_1}} \dots \xrightarrow{e'_{i_m}} z'$ von ι nach z beziehungsweise z' in A'. Kehren wir die Richtung jeder Kante von $P_{z'}$ in A um, dann erhalten wir einen gerichteten Kreis $P = z' \xrightarrow{e'_{i_m}} \dots \xrightarrow{e'_{i_1}} \iota \xrightarrow{e_{i_1}} \dots \xrightarrow{e_{i_n}} z \xrightarrow{e} z'$ von z' nach z'. Entlang dieses Pfades summiert der Fundamentalkreis σ_t für jedes $i \in \{1, \dots, n\}$ die Vorkommen von e_i entlang $P_{z'}$ mit negativen Vorzeichen und jene entlang P_z und $z \xrightarrow{e} z'$ mit positiven Vorzeichen (modulo $b+1$).

Wenn $R = (sup, abs)$ eine abstrakte Region von A ist, dann gelten (mit Lemma 5.2 und Definition 2.18) die folgenden Gleichungen:

$$sup(z) = sup(\iota) + \sigma_z \cdot abs \bmod (b+1) \tag{5.13}$$
$$sup(z') = sup(\iota) + \sigma_{z'} \cdot abs \bmod (b+1) \tag{5.14}$$
$$sup(z') = sup(z) + abs(e) \bmod (b+1) \tag{5.15}$$
$$0 = sup(z) + abs(e) - sup(z') \bmod (b+1) \tag{5.16}$$

Das Ersetzen von $sup(z)$ beziehungsweise $sup(z')$ in Gleichung 5.16 durch die jeweils rechte Seite von Gleichung 5.13 beziehungsweise Gleichung 5.14 ergibt $0 = \sigma_t \cdot abs \bmod (b+1)$. Weil R und t beliebig waren sehen wir: Wenn abs eine abstrakte Region ist, dann gilt $\sigma_t \cdot abs = 0$ für jede Sehne t.

Das folgende Lemma zeigt, dass auch die Umkehrung wahr ist, sodass Fundamentalkreise genutzt werden können, um abstrakte Signaturen –und damit gemäß Bemerkung 5.5 auch abstrakte Regionen– von A zu generieren. Darüber hinaus sehen wir, dass die konkrete Wahl der Spannbaums beliebig sein darf.

Lemma 5.15. *Es seien $t_1, \ldots, t_k$ die Sehnen für T. Es ist $abs \in \mathbb{Z}_{b+1}^n$ genau dann eine abstrakte Signatur von A, wenn $\sigma_{t_i} \cdot abs = 0$ für alle $i \in \{1, \ldots, k\}$. Darüber hinaus liefern zwei verschiedene Spannbäume von A die gleichen abstrakten Regionen.*

Beweis. Wir beginnen mit dem Beweis der ersten Aussage. Sei $abs \in \mathbb{Z}_{b+1}^n$, sodass $\sigma_{t_i} \cdot abs = 0$ für alle $i \in \{1, \ldots, k\}$. Sei $sup(\iota) \in \{0, \ldots, b\}$ beliebig aber fest und $sup(z) = sup(\iota) + \sigma_z \cdot abs$ für jeden Zustand $z \in Z \setminus \{\iota\}$. Wir zeigen, dass $R = (sup, abs)$ eine (abstrakte) τ-Region von A ist, das heißt, es gilt $sup(z') = sup(z) + abs(e) \bmod (b+1)$ für jede Kante $z \xrightarrow{e} z'$ von A.

Sei $t = z \xrightarrow{e} z'$ ein Kante von A. Nach Definition gilt $sup(z) = sup(\iota) + \sigma_z \cdot abs \bmod (b+1)$ und $sup(z') = sup(\iota) + \sigma_{z'} \cdot abs \bmod (b+1)$. Wenn t keine Sehne ist, das heißt $t \in T$, dann gibt es genau einen Pfad von ι nach z beziehungsweise z' in T. Somit gilt $\sigma_{z'}(e) = \sigma_z(e) + 1$. Damit erhalten wir $sup(z') = sup(\iota) + \sigma_z \cdot abs + abs(e) \bmod (b+1)$, das heißt $sup(z') = sup(z) + abs(e) \bmod (b+1)$.

Wenn t eine Sehne ist, das heißt, $t \notin T$, dann gelten mit $\sigma_t(e) = (\sigma_z(e) - \sigma_{z'}(e) + 1)$ die folgenden Äquivalenzen, wobei alle Gleichungen modulo $(b+1)$ zu lesen sind:

$$
\begin{aligned}
0 &= \sigma_t \cdot abs && \iff \\
0 &= \sum_{i=1}^{n} ((\sigma_z(e_i) - \sigma_{z'}(e_i)) \cdot abs(e_i) + abs(e) && \iff \\
0 &= \sum_{i=1}^{n} \sigma_z(e_i) \cdot abs(e_i) - \sum_{i=1}^{n} \sigma_{z'}(e_i) \cdot abs(e_i) + abs(e) && \iff \\
\sigma_{z'} \cdot abs &= \sigma_z \cdot abs + abs(e) && \iff \\
sup(\iota) + \sigma_{z'} \cdot abs &= sup(\iota) + \sigma_z \cdot abs + abs(e) && \iff \\
sup(z') &= sup(z) + abs(e)
\end{aligned}
$$

Dies zeigt, dass R eine (abstrakte) τ-Region von A ist.

Für die andere Richtung haben wir bereits oben diskutiert, dass $\sigma_t \cdot abs = 0$ für jede abstrakte τ-Region $R = (sup, abs)$ und Sehne t gilt. Dies beweist die erste Aussage.

Für die zweite Aussage, sei T' ein von T verschiedener Spannbaum mit Sehnen $t'_1, \ldots, t'_k$, wobei wir beachten, dass je zwei Spannbäume dieselbe Anzahl von Kanten haben. Mit der ersten Aussage erhalten wir, dass genau dann $\sigma_{t_i} \cdot abs = 0$ für alle $i \in \{1, \ldots, k\}$ gilt,

wenn *abs* eine abstrakte Region ist und dies ist genau dann der Fall, wenn $\sigma'_{t_i} \cdot abs = 0$ für alle $i \in \{1, \ldots, k\}$. Dies beweist das Lemma. □

Um die Lösbarkeit von ZTP zu zeigen, wollen wir uns auf abstrakte Regionen beschränken. Das folgende Lemma beweist, dass es zu jeder τ-Region von A eine *induzierte* abstrakte τ-Region von A mit dem gleichen Support gibt:

Lemma 5.16 (Induzierte Region)**.** *Wenn $R = (sup, sig)$ eine τ-Region von A ist und die Abbildung $abs : E \to \{0, \ldots, b\}$ für alle $e \in E$ definiert ist durch*

$$abs(e) = \begin{cases} sig(e), & \text{falls } sig(e) \in \{0, \ldots, b\} \\ -m + n, & \text{falls } -m + n \geq 0 \\ b + 1 - m + n, & \text{falls } -m + n < 0 \end{cases}$$

dann ist $R' = (sup, abs)$ eine abstrakte τ-Region von A. Wir nennen R' die von R induzierte *abstrakte τ-Region.*

Beweis. Sei $R = (sup, sig)$ eine τ-Region von A und $z \xrightarrow{e} z'$ eine beliebige aber feste Kante von A. Wenn $sig(e) \in \{0, \ldots, b\}$, dann folgt sicherlich $sup(z') = sup(z) + abs(e) \bmod (b + 1)$.

Nehmen wir an, dass $sig(e) = (m, n)$. Mit der Definition von τ impliziert dies $|-m + n| \leq b$, denn andernfalls erhielten wir einen Widerspruch zu $q \in \{0, \ldots, b\}$ für alle Zustände q von τ. Sei weiterhin $abs(e) = -m + n$, falls $-m + n \geq 0$ und $abs(e) = b + 1 - m + n$, falls $-m + n < 0$, wobei jeder Fall $abs(e) \in \{0, \ldots, b\}$ impliziert.

Wenn $-m + n \geq 0$, dann gilt $sup(z') - sup(z) = -m + n = abs(e)$ und somit insbesondere $sup(z') = sup(z) + abs(e) \bmod (b + 1)$. Wenn $-m + n < 0$, dann erhalten wir $sup(z') - sup(z) = -m + n = b + 1 - m + n \bmod (b + 1)$, denn offenbar gilt $-x = b + 1 - x \bmod b + 1$ für alle $x \in \{0, \ldots, b\}$. Somit gilt auch in diesem Fall $sup(z') = sup(z) + abs(e) \bmod (b + 1)$. Weil $z \xrightarrow{e} z'$ beliebig war, folgt das Lemma. □

Gemäß des folgende Lemmas ist die Lösbarkeit eines ZTP äquivalent zur Existenz bestimmter abstrakter Signaturen:

Lemma 5.17. *Ein ZTP (z, z') von A ist genau dann τ-lösbar, wenn es eine abstrakte Signatur abs von A gibt, sodass $\sigma_z \cdot abs \neq \sigma_{z'} \cdot abs$.*

Beweis. Sei $R' = (sup, sig)$ eine τ-Region, die α löst. Die durch R' induzierte Region $R = (sup, abs)$ hat den gleichen Support wie R', deshalb gilt $sup(z) \neq sup(z')$. Weil $sup(z) = sup(\iota) + \sigma_z \cdot abs$ und $sup(z') = sup(\iota) + \sigma_{z'} \cdot abs$, folgt damit $\sigma_z \cdot abs \neq \sigma_{z'} \cdot abs$.

Für die andere Richtung, sei abs eine abstrakte Signatur von A, die $\sigma_z \cdot abs \neq \sigma_{z'} \cdot abs$ erfüllt. Gemäß Bemerkung 5.5 definieren $sup(\iota)$ und abs für jedes $sup(\iota) \in \{0, \ldots, b\}$ implizit eine abstrakte τ-Region $R = (sup, sig)$ von A, wobei $sup(q) = sup(\iota) + \sigma_q \cdot abs$ für alle $q \in Z$. Mit $sup(\iota) = 0$ gilt dann insbesondere $sup(z) = \sigma_z \cdot abs \neq \sigma_{z'} \cdot abs = sup(z')$ und R löst α. □

Das folgende Lemma komplettiert den Beweis von Satz 5.2.2:

Lemma 5.18. *Es gibt einen Algorithmus, der mit Zeitaufwand $\mathcal{O}(|E|^3 \cdot |Z|^6)$ entscheidet, ob A die τ-ZTE hat.*

Beweis. Sei $\alpha = (z, z')$ ein beliebiges aber festes ZTP von A. Gemäß Lemma 5.17 ist α genau dann τ-lösbar, wenn es eine abstrakte Signatur abs von A mit $\sigma_z \cdot abs \neq \sigma_{z'} \cdot abs$ gibt. Man beachte, dass genau dann $\sigma_z \cdot abs \neq \sigma_{z'} \cdot abs$ gilt, wenn $(\sigma_z - \sigma_{z'}) \cdot abs \neq 0$ und dies ist genau dann der Fall, wenn $(\sigma_z - \sigma_{z'}) \cdot abs \in \{1, \ldots, b\}$. Um die τ-Lösbarkeit von α zu entscheiden, gehen wir nun wie folgt vor: Für jedes $y \in \{1, \ldots, b\}$ konstruieren wir das Gleichungssystem, das definiert ist durch (1) die Gleichungen $t_i \cdot \mathbf{x} = 0$ für alle $i \in \{0, \ldots, k\}$, die durch die Sehnen von A definiert sind und (2) die Gleichung $(\sigma_z - \sigma_{z'}) \cdot x = y$. Nach Lemma 5.15 ist jede Lösung $\mathbf{x}$ des Systems eine abstrakte τ-Region von A, die, wegen $(\sigma_z - \sigma_{z'}) \cdot \mathbf{x} = y$, das ZTP α löst. Im Gegenzug, wenn es eine abstrakte Region $R = (sup, sig)$ gibt, die α löst, dann muss es ein $y \in \{1, \ldots, b\}$ geben, sodass das entsprechende Gleichungssystem mit der Gleichung $(\sigma_z - \sigma_{z'}) \cdot x = y$ lösbar ist. Deshalb ist α genau dann lösbar, wenn eines der b Gleichungssyteme lösbar ist.

Aus diesen Überlegungen erhalten wir folgende Abschätzungen des Zeitaufwands: Ein Spannbaum A' kann in Zeit $\mathcal{O}(|E| \cdot |Z|^3)$ konstruiert werden [Tar77]. Es gibt höchstens $|Z|^2 \cdot |E| - 1$ Sehnen für A'. Somit hat ein Gleichungssystem (für eine festes ZTP und einen festen Wert y) höchstens $|Z|^2 \cdot |E|$ Gleichungen, das heißt, die Koeffizientenmatrix hat Dimension $|Z|^2 \cdot |E| \times |E|$. Für ein festes ZTP müssen wir höchstens b solcher Systeme lösen, wobei b nicht Teil der Eingabe ist. Deshalb benötigt dies gemäß Lemma 5.14 höchstens den Zeitaufwand $\mathcal{O}(|Z|^4 \cdot |E|^3)$. Weil A höchsten $|Z|^2$ ZTP hat, folgt die Behauptung. □

Wegen der Kongruenz zwischen $\tau^1_{\mathbb{Z}PPT}$ und dem booleschen Netztyp $\{\mathsf{nop}, \mathsf{inp}, \mathsf{out}, \mathsf{swap}\}$ folgt aus Satz 2.2 und Lemma 5.18 das Lemma 4.4 aus Abschnitt 4.1.2.

5.2.3 Der Sonderfall $\mathbb{Z}_2$-erweiterter 1-beschränkter Netztypen

In diesem Abschnitt werden wir zunächst Satz 5.2.3 beweisen. Darüber hinaus werden wir den in Abschnitt 4.1.2 schuldig gebliebenen Beweis von Lemma 4.6 nachliefern. Solange nicht ausdrücklich anders vereinbart, sei im Folgenden $A = (Z, E, \delta, \iota)$ ein Transitionssystem und $\tau \in \{\tau^1_{\mathbb{Z}PPT}, \tau^1_{\mathbb{Z}PT}\}$, jeweils beliebig aber fest.

Um die angekündigten Ziele zu erreichen, gehen wir wie folgt vor: Zunächst zeigen wir, dass es für die τ-Lösbarkeit eines EZTP $\alpha = (a, q)$ von A ausreichend ist, die Existenz von τ-Regionen $R = (sup, sig)$ zu untersuchen, die $sup(q) = 0$ und $sig(a) \in \{(0, 1), (1, 1)\}$ sowie $sig(e) \in \{0, 1\}$ für alle $e \in E \backslash \{a\}$ erfüllen. Weiterhin zeigen wir, dass in Polynomialzeit entschieden werden kann, ob es eine solche τ-Region gibt. Anschließend argumentieren wir, dass die Existenz so einer Region mit $sig(a) = (1, 0)$ beziehungsweise $sig(a) = (1, 1)$ äquivalent zur Existenz einer $\{\mathsf{nop}, \mathsf{inp}, \mathsf{swap}\}$- beziehungsweise $\{\mathsf{nop}, \mathsf{swap}, \mathsf{used}\}$-Region von A ist, die α löst. Dies impliziert bereits Lemma 4.6, da sich der entsprechende Zeitaufwand mit dem Lemma verträgt. Weil A höchstens $|Z| \cdot |E|$ EZTP hat und wir bereits

im vorherigen Abschnitt 5.2.2 gezeigt haben, dass in Polynomialzeit entschieden werden kann, ob A die τ-ZTE hat, folgt schließlich auch Satz 5.2.3.

Lemma 5.19. *Sei (a, q) ein EZTP von A.*

1. *Wenn es eine τ-Region $R = (sup, sig)$ mit $sup(q) = 1$ und $sig(a) = (0, 1)$ für A gibt, dann gibt es eine τ-Region $R' = (sup', sig')$ mit $sup(q) = 0$ und $sig(a) = (1, 0)$ von A.*

2. *Wenn $R = (sup, sig)$ eine τ-Region ist, die α löst, dann gibt es eine τ-Region $R' = (sup, sig')$ mit demselben Support, sodass $sig'(a) = sig(a)$ und $sig'(e) \in \{0, 1\}$ für alle $e \in E \setminus \{a\}$.*

Beweis. (1): Wir erhalten R' wie folgt: $sup'(z) = 1 - sup(z)$ für alle $z \in Z$ und für alle $e \in E$, wenn $sig(e) = (1, 0)$, dann $sig'(e) = (0, 1)$; wenn $sig(e) = (0, 1)$, dann $sig'(e) = (1, 0)$, andernfalls (das heißt $sig(e) \in \{0, (1, 1)\}$) sei $sig'(e) = 0$.

(2): Wir erhalten R' wie folgt: $sup'(z) = sup(z)$ für alle $z \in Z$ und für alle $e \in E$, wenn $e = a$ oder $sig(e) = 0$, dann $sig'(e) = sig(e)$, andernfalls $sig'(e) = 1$. □

Eine τ-Region $R = (sup, sig)$, die ein EZTP (a, q) löst, erfüllt offenbar $sig(a) \in \{(0, 1), (1, 0), (1, 1)\}$. Daraus folgt mit Lemma 5.19, dass wir uns für die τ-Lösbarkeit eines EZTP $\alpha = (a, q)$ auf die Suche von τ-Regionen beschränken dürfen, die $sup(q) = 0$ und $sig(a) \in \{(1, 0), (1, 1)\}$ und $sig(e) \in \{0, 1\}$ für alle $e \in E \setminus \{a\}$ erfüllen. Gemäß des folgenden Lemmas können wir in Polynomialzeit entscheiden, ob es eine solche Region:

Lemma 5.20. *Es gibt einen Algorithmus der mit Zeitaufwand $\mathcal{O}(|Z|^4 \cdot |E|^3)$ für ein EZTP $\alpha = (a, q)$ von A entscheidet, ob es eine τ-Region $R = (sup, sig)$ mit $sup(q) = 0$ und $sig(a) = (1, 0)$ beziehungsweise $sig(a) = (1, 1)$ sowie $sig(e) \in \{0, 1\}$ für alle $e \in E \setminus \{a\}$ gibt.*

Beweis. Es seien $p_1, \ldots, p_m \in Z$ genau die Ursprünge und $q_1, \ldots, q_\ell$ genau die Ziele von a in A und sei A' ein beliebiger aber fester Spannbaum von A und $t_1, \ldots, t_k$ die Fundamentalkreise für A'.

Für $sig(a) = (1, 0)$: Wir bilden wie folgt ein lineares Gleichungssystem: Für jedes $i \in \{1, \ldots, k\}$ fügen wir die Gleichung $\sigma_{t_i} \cdot abs$ für den Fundamentalkreis t_i hinzu. Für jedes $i \in \{1, \ldots, m\}$ und jedes $j \in \{1, \ldots, \ell\}$ fügen wir die Gleichungen $(\sigma_{p_i} - \sigma_q) \cdot abs = 1$ und $(\sigma_{q_j} - \sigma_q) \cdot abs = 0$ hinzu. (Dabei sind σ_{p_i} und σ_{q_j} sowie σ_q die Parikh-Vektoren der eindeutigen Pfade von ι nach p_i und q_j sowie q in A'.)

Wir argumentieren nun, dass das resultierende Gleichungssystem $K \cdot \mathbf{x} = c$ genau dann lösbar ist, wenn es eine τ-Region $R = (sup, sig)$ von A gibt, die $sig(a) = (1, 0)$ und $sup(q) = 0$ und $sig'(e) \in \{0, 1\}$ für alle $e \in E \setminus \{a\}$ erfüllt:

Sei abs eine Lösung des resultierenden Systems. Gemäß Lemma 5.15 ist abs eine abstrakte Signatur von A. Wegen $(\sigma_{p_i} - \sigma_q) \cdot abs = 1$ für alle $i \in \{1, \ldots, m\}$ und $(\sigma_{q_i} - \sigma_q) \cdot abs = 0$ für alle $i \in \{1, \ldots, \ell\}$, gilt $\sigma_{p_1} \cdot abs = \cdots = \sigma_{p_m} \cdot abs \neq \sigma_q \cdot abs = \sigma_{q_1} \cdot abs = \cdots = \sigma_{q_\ell} \cdot abs$. Wenn $\sigma_q \cdot abs = 0$ beziehungsweise wenn $\sigma_q \cdot abs = 1$, dann erfüllt

die (implizit) durch $sup(\iota) = 0$ beziehungsweise $sup(\iota) = 1$ und abs definierte (abstrakte) τ-Region $R = (sup, abs)$ von A, dass $sup(q) = 0$ und $sup(p_i) = 1$ und $sup(q_j) = 0$ für alle $i \in \{1, \ldots, m\}$ und alle $j \in \{1, \ldots, \ell\}$. Die folgende Region $R' = (sup, sig)$ mit demselben Support wie R ist daher wohldefiniert: Für alle $e \in E$, wenn $e = a$, dann $sig(a) = (1, 0)$, andernfalls $sig(e) = abs(e)$.

Für die andere Richtung, sei $R = (sup, sig)$ eine τ-Region, die $sup(q) = 0$ und $sig(a) = (1, 0)$ erfüllt. Wegen $sig(a) = (1, 0)$ gilt insbesondere $sup(p_1) = \cdots = sup(p_m) \neq sup(q) = sup(q_1) = \cdots = sup(q_\ell)$. Daraus folgt, dass die gemäß Lemma 5.16 durch R induzierte abstrakte Region $R' = (sup, abs)$ wegen $sup(z) = sup(\iota) + \sigma_z \cdot abs$ für alle $z \in Z$, das oben definierte Gleichungssystem löst.

Es bleibt, den Zeitaufwand zur Berechnung des angegebene Systems abzschätzen: Weil A höchstens $|Z|^2 \cdot |E|$ Sehnen hat und a höchstens $|Z|$ Ursprünge beziehungsweise Ziele hat, besteht das das resultierende System aus höchstens $3 \cdot |Z|^2 \cdot |E|$ Gleichungen. Die Behauptung folgt deshalb aus Lemma 5.14.

Für $sig(a) = (1, 1)$: Zunächst bilden wir folgendes lineare Gleichungssystem: Für jedes $i \in \{1, \ldots, k\}$ fügen wir die Gleichung $\sigma_{t_i} \cdot abs$ für den Fundamentalkreis t_i hinzu. Für jedes $i \in \{1, \ldots, m\}$ und jedes $j \in \{1, \ldots, \ell\}$ fügen wir die Gleichungen $(\sigma_{p_i} - \sigma_q) \cdot abs = 1$ und $(\sigma_{q_i} - \sigma_q) \cdot abs = 1$ hinzu. Diese stellen sicher, dass eine Lösung abs stets $\sigma_q \cdot abs \neq \sigma_z \cdot abs = \sigma_{z'} \cdot abs$ für alle a-beschrifteten Kanten $z \xrightarrow{a} z'$ von A impliziert. Wenn $\sigma_q \cdot abs = 0$, dann folgt daraus für die implizit durch $sup(\iota) = 0$ und abs definierte abstrakte Region (sup, sig), dass $sup(q) = 0$ und $sup(q_1) = \cdots = sup(q_\ell) = sup(p_1) = \cdots = sup(p_\ell) = 1$, sodass dieser Support eine Signatur mit $sig(a) = (1, 1)$ erlaubt. Andernfalls, wenn $\sigma_q \cdot abs = 1$, dann erhalten wir durch $sup(\iota) = 1$ und abs einen Support, der $sup(q) = 0$ erfüllt und eine Signatur mit $sig(a) = (1, 1)$ erlaubt.

Analog zu (1) argumentiert man nun im Gegenzug: Ist $R = (sup, sig)$ eine τ-Region mit $sig(a) = (1, 1)$ und $sup(q) = 0$, dann ist die abstrakte Signatur der durch R induzierten Region (sup, abs) eine Lösung des genannten Gleichungssystems. Die Abschätzung des Zeitaufwands, um über die Existenz einer Lösung zu entscheiden, ist analog zu (1). □

Wie man leicht sieht, erhält man durch eine einfache Umbenennung von nop zu 0 und inp zu $(1, 0)$ und swap zu 1 sowie used zu $(1, 1)$, zwei (boolesche) Netztypen $\{(0, 0), (1, 0), 1\}$ beziehungsweise $\{(0, 0), (1, 1), 1\}$, die zu $\{\mathsf{nop}, \mathsf{inp}, \mathsf{swap}\}$ beziehungsweise $\{\mathsf{nop}, \mathsf{swap}, \mathsf{used}\}$ kongruent sind. Lemma 4.6 ist deshalb eine einfache Folgerung aus Satz 2.2 und Lemma 5.20.

Weil wir bereits im vorherigen Abschnitt 5.2.2 gezeigt haben, dass in Polynomialzeit entschieden werden kann, ob A die τ-ZTE hat, vervollständigt das folgende Lemma den Beweis von Satz 5.2.

Lemma 5.21. *Es gibt einen Algorithmus, der mit Zeitaufwand $\mathcal{O}(|Z|^5 \cdot |E|^4)$ entscheidet, ob A die τ-EZTE hat.*

Beweis. Das Transitionssystem A hat höchstens $|Z| \cdot |E|$ EZTP. Wenn $R = (sup, sig)$ eine τ-Region ist, die ein EZTP $\alpha = (a, q)$ löst, dann gilt $sig(a) \in \{(0, 1), (1, 0), (1, 1)\}$ und

$\neg sup(q) \xrightarrow{sig(a)}$. Das Lemmas folgt damit aus Lemma 5.19 und Lemma 5.20. □

5.3 Zusammenfassung, Konsequenzen und Ausblick

In diesem Kapitel haben wir die Komplexität der klassischen Syntheseprobleme für alle (puren) b-beschränkten P/T-Netze und ihrer jeweiligen Erweiterungen um die Gruppe $\mathbb{Z}_{b+1}$ für alle $b \in \mathbb{N}$ charakterisiert. Während die Synthese für alle Implementierungen und alle $b \in \mathbb{N}$ für die (puren) b-beschränkten P/T-Netze NP-vollständig ist, gibt es bei den gruppenerweiterten Netztypen sowohl bezüglich der im Voraus fest gewählten Schranke b als auch bezüglich der angestrebten Implementierung überraschende Unterschiede: Zum einen ist Synthese für alle Implementierungen polynomiell, wenn $b \leq 1$; im Gegensatz dazu ist sie NP-vollständig, wenn wir eine Sprachsimulation beziehungsweise eine Realisierung anstreben und $b \geq 2$. Zum anderen ist die Suche nach einer Einbettung hier für alle $b \in \mathbb{N}$ polynomiell. Nach besten Wissen des Autors der vorliegenden Arbeit definieren $\tau^b_{\mathbb{Z}PPT}$ und $\tau^b_{\mathbb{Z}PT}$ somit die ersten Netztypen überhaupt, sodass der rechnerische Aufwand von Sprachsimulation und Realisierung nachweislich größer ist, als der von Einbettung.

Zwei offene Fragen im Bereich der Netz-Synthese betreffen die Komplexität der Synthese b-beschränkter Netze, die auf *Sprachäquivalenz* abzielt sowie Synthese b-beschränkter Netze, die von *modalen Transitionssystemen* ausgeht.

τ-Sprachäquivalenz ist das Problem, für ein gegebenes Transitionssystem A zu entscheiden, ob es ein τ-Netz N gibt, sodass A und der Erreichbarkeitsgraph A_N von N die gleiche Sprache haben: $L(A) = L(A_N)$. Badouel, Bernardinello und Darondeau [BBD15, p. 164] haben die Sprache $L(A)$ von A mit einem Transitionssystem $L_A = (L(A), E, \delta_L, \varepsilon)$ assoziiert, wobei genau dann $\delta_L(w, e) = we$, wenn $we \in L(A)$. Gemäß Theorem 5.12 aus [BBD15, p. 164] gibt es genau dann eine gesuchtes τ-Netz für A, wenn L_A die τ-EZTE hat. Weil es im Allgemeinen exponentiell viele (beziehungsweise sogar unendlich viele) Pfade in A geben kann, ist der Ansatz, zunächst L_A zu berechnen und dann zu testen, ob L_A die τ-EZTE hat, wenigstens von exponentiellem Aufwand. Nach bestem Wissen des Autors der vorliegenden Arbeit, war bisher für τ-Sprachäquivalenz weder die genaue Komplexität noch eine untere Schranke bekannt, wenn $\tau \in \{\tau^b_{PPT}, \tau^b_{PT}\}$. Die Reduktion aus Abschnitt 5.1 und die daraus folgenden Resultate implizieren jedoch eine untere Schranke für τ-Sprachäquivalenz für diese Netztypen. Dies sieht man so: Wenn $A = z_0 \xrightarrow{e_1} z_1 \xrightarrow{e_2} z_2 \xrightarrow{e_3} \dots \xrightarrow{e_n} z_n$ ein lineares Transitionssystem ist, dann ist $L_A = \varepsilon \xrightarrow{e_1} e_1 \xrightarrow{e_2} e_1e_2 \xrightarrow{e_3} \dots \xrightarrow{e_n} e_1 \dots e_n$, das heißt, $\varepsilon, e_1, e_1e_2, \dots, e_1 \dots e_n$ sind die Zustände von L_A. Wie man leicht sieht, sind A und L_A isomorph. Daher ist ein lineares Transitionssystem genau dann durch ein τ-Netz bis auf Sprachäquivalenz implementierbar, wenn es die τ-EZTE hat. Die Reduktion aus Abschnitt 5.1 liefert ein lineares Transitionssystem. Dies impliziert, dass τ-Sprachäquivalenz wenigstens NP-schwer ist.

Schlachter und Wimmel [SW17] stellten die Frage nach der Komplexität der Synthese (purer) b-beschränkter P/T-Netze, die im vorliegenden Kapitel umfassend beantwortet

wurde. Die Frage der Autoren geht jedoch über die in dieser Arbeit untersuchten Syntheseprobleme hinaus und adressiert für $\tau \in \{\tau^b_{PPT}, \tau^b_{PT}\}$ die Komplexität des Problems MODALE τ-SYNTHESE, die für ein *modales Transitionssystem* fragt, ob es durch ein (pures) b-beschränktes P/T-Netz implementiert werden kann. Ein *modales Transitionssystem* $A = (Z, E, \delta_{must}, \delta_{may}, \iota)$ (wobei *must* und *may* engl. für *verpflichtend* und *möglicherweise*) besteht aus einer Menge von Zuständen Z, Ereignissen E und zwei partiellen Funktionen $\delta_{must}, \delta_{may} : Z \times E \to Z$, welche sogenannte *must*-Kanten und *may*-Kanten definieren. Dabei gilt, dass jede *must*-Kante eine *may*-Kante ist, aber nicht jede *may*-Kante notwendigerweise eine *must*-Kante. Ein Transitionssystem B mit denselben Ereignissen wie A *implementiert* A, wenn es eine Relation $R \subseteq Z \times Z(B)$ gibt, sodass $(\iota, \iota_B) \in R$ und für alle $(z, q) \in R$ und alle $e \in E = E(B)$ gilt:

1. Wenn $\delta_{must}(z, e) = z'$, dann gibt es ein $q' \in Z(B)$, sodass $\delta(B)(q, e) = q'$ und $(z', q') \in R$.

2. Wenn $\delta(b)(q, e) = q'$, dann gibt es ein $z' \in Z$, sodass $\delta_{may}(z, e) = z'$ und $(z', q') \in R$.

Während die genaue Charakterisierung von MODALE τ-SYNTHESE weiterhin ungeklärt bleibt, liefern die Ergebnisse dieses Kapitels zumindest eine untere Schranke. Dies sieht man wie folgt: Jedes Transitionssystem kann als modales Transitionssystem interpretiert werden, für das die *must*- und *may*-Kanten zusammenfallen. Für solche Transitionssysteme entspricht die gerade definierte Implementierungs-Relation der wohlbekannten Relation *Bisimulation* [Gor17, p. 22]. Darüber hinaus ist bekannt, dass zwei (deterministische) Transitionssysteme A_0 und A_1 genau dann bisimilar sind, wenn sie sprachäquivalent sind (in diesem Zusammenhang oft auch *trace-äquivalent* genannt) [Gor17, p. 26]. Insgesamt folgt mit obigen Überlegungen, dass ein lineares Transitionssystem genau dann die τ-EZTE hat, wenn es bist auf Sprachäquivalenz implementierbar ist, genau dann wenn es –interpretiert als modales Transitionssystem– gemäß der gerade definierten (Bisimulations-) Implementierungsrelation implementiert werden kann. Der folgende Satz fasst die diskutierten Konsequenzen der Resultate dieses Kapitels zusammen:

Satz 5.3. *Wenn* $\tau \in \{\tau^b_{PPT}, \tau^b_{PT}\}$*, dann sind* τ-SPRACHÄQUIVALENZ *und* MODALE τ-SYNTHESE *NP-schwer.*

Die genaue Charakterisierung von τ-SPRACHÄQUIVALENZ und MODALE τ-SYNTHESE bleibt eine Aufgabe zukünftiger Forschungsarbeit.

Zukünftige Forschung könnte auch die Frage untersuchen, ob die als schwer identifizierten Syntheseprobleme dieses Abschnitts von anderer Komplexität sind, wenn das Eingabe-Transitionssystem beziehungsweise das angestrebte Netz eingeschränkt wird. Bezüglich des Transitionssystems sind der Knotengrad, die Vielfachheit von Ereignissen oder auch die Existenz von Kreisen naheliegende Eigenschaften, die eingeschränkt werden könnten. Allerdings resultieren die Reduktionen dieses Kapitel schon in erheblich eingeschränkten kreisfreien Transitionssystemen, die sogar linear (Abschnitt 5.1) beziehungsweise von Grad Zwei sind (Abschnitt 5.2). Es wäre interessant zu wissen, ob τ-SPRACHSIMULATION

und τ-Realisierung für die gruppenerweiterten Netztypen auch dann schwer bleiben, wenn die Eingaben lineare Transitionssysteme sind, so wie es für die zu Grunde liegenden Netztypen ohne Gruppenerweiterung der Fall ist. Des Weiteren hat der Autor der vorliegenden Arbeit gezeigt [Tre19a], dass τ-Sprachsimulation und τ-Realisierung in Polynomialzeit entscheidbar sind, wenn die Vielfachheit der Ereignisse des Eingabe-Transitionssystems *a priori* durch eine beliebige aber feste Zahl beschränkt ist. Insbesondere sind die Probleme fest-Parameter-berechenbar, wenn die Vielfachheit als Parameter gewählt wird. Für die (puren) b-gebundenen Netze sind die Auswirkungen dieses Parameters auf die Komplexität für τ^1_{PPT} untersucht worden [TRW18, TR18]: selbst wenn jedes Ereignis höchstens zweimal stattfindet, bleibt Synthese für alle Implementationen schwer. Zukünftige Forschung könnte untersuchen, ob dies auch für $b \geq 2$ der Fall ist.

Einschränkungen des angestrebten Netzes können die Umgebung von Plätzen beziehungsweise Transitionen betreffen. Für die (puren) b-beschränkten Netze sieht man relativ leicht: für jedes im Voraus fest gewählte k, kann in Polynomialzeit entschieden werden kann, ob ein Transitionssystem durch ein (pures) b-gebundenes P/T-Netz implementiert werden kann, sodass sich in der *Umgebung jedes Platzes* höchstens k Transitionen befinden, das heißt, auf jeden Platz können höchstens k Transitionen produzieren beziehungsweise davon konsumieren [Tre19c]. Der Beweis basiert grob gesagt darauf, dass eine Region $R = (sup, sig)$ vollständig durch $sup(\iota)$ und sig beschrieben ist und die Netztypen der b-beschränkten P/T-Netze nur polynomiell viele Ereignisse zur Auswahl für einen Signaturwert $sig(e)$ haben. Dieser Ansatz lässt sich auch für die gruppenerweiterten Netze verwenden. Im Kontrast dazu haben wir in [Tre19c] ebenso gezeigt, dass τ-Sprachsimulation und τ-Realisierung auch dann schwer sind, wenn die *Umgebung jeder Transition* im gesuchten Netz höchstens zwei Plätze enthalten darf, das heißt, jede Transition darf von höchstens zwei Plätzen konsumieren beziehungsweise darauf produzieren. Es wäre interessant zu wissen, ob diese Probleme auch dann schwer bleiben, wenn jede Transition von höchstens *einem* Platz konsumieren beziehungsweise darauf produzieren darf. Darüber hinaus bleibt bezüglich dieser Netztypen und Einschränkung die Komplexität von τ-Einbettung zu klären. Ebenso kann zukünftige Arbeit untersuchen, wie sich diese Einschränkungen der Umgebungen von Transitionen auf die Komplexität der Synthese gruppenerweiterter Netze auswirkt.

Dass die Synthese (purer, gruppenerweiterter) b-beschränkter Netze mit k-beschränkten Platz-Umgebungen für festes k in Polynomial-Zeit entscheidbar ist, impliziert die Zugehörigkeit zu XP (*slice-wise polynomial*). Dies gibt Anlass zur Frage, ob diese Instanz der Synthese fest-Parameter-berechenbar ist, wenn k als Parameter gewählt wird. In [Tre20c] haben wir gezeigt, dass dies für b-gebundene P/T-Netze nicht der Fall ist, wenn eine Realisierung gesucht wird: das resultierende parametrisierte Problem ist $\mathsf{W}[1]$-schwer. Für die verbleibenden Netzklassen (dieses Kapitels) sowie die verbleibenden Implementierungen, ist die parametrisierte Komplexität der resultierenden Syntheseprobleme von Netzen mit beschränkter Platz-Umgebung jedoch noch ungeklärt und kann deshalb Gegenstand zukünftiger Forschung sein.

Teil II

Parametrisierte Syntheseprobleme

Kapitel 6

Boolesche Netze mit begrenzten Platzumgebungen

In diesem Kapitel untersuchen wir die Komplexität der Synthese von booleschen Netzen, die gewissen Restriktionen genügen sollen. Diese Restriktionen schränken *strukturelle* Aspekte des Netzes ein, nämlich die *Umgebung von Plätzen*. Dabei verstehen wir die Umgebung eines Platzes als die Menge der Transitionen, von denen der Platz nicht unabhängig ist, das heißt, jene Transitionen, deren Schalten die Markierung des Platzes (potentiell) beeinflussen kann beziehungsweise deren Schalten durch die Markierung eines Platzes (potentiell) verhindert werden kann. *Synthese von Netzen mit begrenzter Platzumgebung* adressiert die Aufgabe, für ein Transitionssystem A ein implementierendes Netz zu finden, sodass die Anzahl der Transitionen in der Umgebung jedes Platzes begrenzt ist. Die Plätze beziehungsweise die Transitionen eines Netzes entsprechen den Ressourcen beziehungsweise den Akteuren/Agenten des Systems, das durch das Netz modelliert wird. Die Formulierung dieses Syntheseproblems berücksichtigt daher nicht nur den Aspekt der Verteilbarkeit des Systems, sondern nimmt auch Beschränkungen in den Blick, die modellieren, dass nicht „zu viele“ Akteure/Agenten um eine (beliebige aber feste) Ressource konkurrieren sollen. Wie wir sehen werden, wird es erheblichen Einfluss auf die Komplexität der Synthese solcher Netze haben, ob diese Anzahl im Voraus festgelegt oder ein Teil der Eingabe ist. Darüber hinaus ist sofort klar, dass nur solche Netztypen eine Rolle spielen können, deren Fluss überhaupt potentielle Unabhängigkeit zwischen Plätzen und Transitionen zulässt, das heißt, die nop erlauben. Andernfalls ist einfach jede Transition des Netzes in der Umgebung jedes Platzes.

Definition 6.1 (begrenzte (Platz-) Umgebung)**.** Sei τ ein boolescher Netztyp mit nop und $N = (P, T, f, \mathfrak{m}_0)$ ein τ-Netz und $d \in \mathbb{N}$. Für jeden Platz $p \in P$ heißt die Menge ${}^\bullet p^\bullet = \{t \in T \mid f(p,t) \neq \mathsf{nop}\}$ die *(Platz-) Umgebung* von p. Wir sagen, die Umgebung von p ist *d-begrenzt*, wenn $|{}^\bullet p^\bullet| \leq d$. Die *Platz-Umgebungszahl* von N is definiert als $d_N = \max\{|{}^\bullet p^\bullet| \mid p \in P\}$

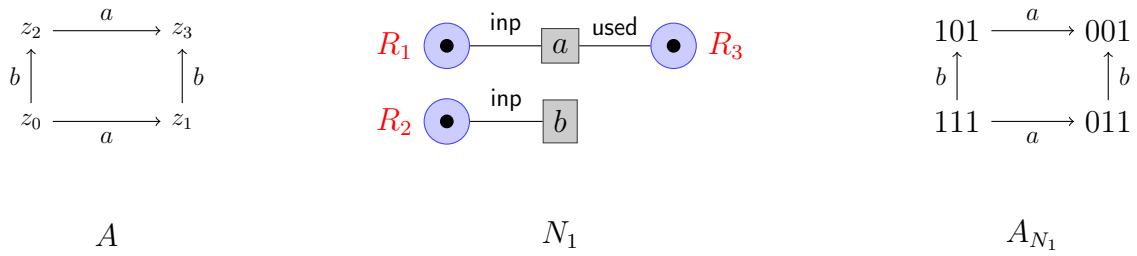


Abbildung 6.1: Das Transitionssystem A, das $\{\mathsf{nop}, \mathsf{inp}, \mathsf{used}\}$-Netz N_1 mit 1-begrenzten Plätzen R_1, R_2, R_3 (nop-gewichtete Kanten sind ausgelassen) und initialer Markierung $\mathfrak{m}_0(R_1)\mathfrak{m}_0(R_2)\mathfrak{m}_0(R_3) = 111$ sowie der Erreichbarkeitsgraph A_{N_1} von N_1.

Beispiel 6.1. Abbildung 6.1 zeigt ein Transitionssystem A und ein $\{\mathsf{nop}, \mathsf{inp}, \mathsf{used}\}$-Netz N_1, welches A realisiert und dessen Plätze 1-begrenzte Umgebungen haben.

In diesem Kapitel untersuchen wir die parametrisierte Komplexität des Problems, für eine Zahl d und ein Transitionssystem A zu entscheiden, ob A durch ein boolesches Netz implementiert werden kann, dessen Platz-Umgebungszahl durch d beschränkt ist, wobei die Umgebungszahl der Parameter ist. Dieses Kapitel basiert auf den Veröffentlichungen [TE21, TE20c, TE20d].

Solange nicht ausdrücklich anders vereinbart, sei τ im weiteren Verlauf dieses Kapitels ein boolescher Netztyp mit nop. Wenn $N = (P, T, f, \mathfrak{m}_0)$ ein τ-Netz ist, dann ist die Umgebungszahl eines Platzes $p \in P$ gemäß Definition 6.1 definiert als die Anzahl der Transitionen t, bezüglich denen der Platz nicht unabhängig ist, das heißt $f(p,t) \neq \mathsf{nop}$. Wir wissen, dass ein Transitionssystem $A = (Z, E, \delta, \iota)$ genau dann durch ein τ-Netz implementierbar ist, wenn es eine Menge $\mathcal{R}$ von τ-Regionen gibt, welche die zu der Implementierung gehörenden Trennungseigenschaften nachweist. Darüber hinaus ist in diesem Fall das synthetisierte Netz $N_A^{\mathcal{R}} = (\mathcal{R}, E, f, \mathfrak{m}_0)$ bereits eine gesuchte Implementierung. Jede Region $R = (sup, sig)$ von $\mathcal{R}$ ist ein Platz in $N_A^{\mathcal{R}}$ und es gilt $f(R,e) = sig(e)$ für alle $e \in E$. Insbesondere ist e genau dann in der Umgebung von R, wenn $sig(e) \neq \mathsf{nop}$. Das bedeutet, dass $N_A^{\mathcal{R}}$ genau dann ein gesuchtes Netz mit durch d beschränkter Platz-Umgebungszahl ist, wenn für jede Regionen aus $\mathcal{R}$ gilt, dass ihre *Umgebung* höchstens d Ereignisse enthält, das heißt, wenn sie *d-beschränkt* ist:

Definition 6.2 (Umgebung boolescher Region, d-beschränkt)**.** Sei τ ein boolescher Netztyp mit nop und $A = (Z, E, \delta, \iota)$ ein Transitionssystem sowie $d \in \mathbb{N}$. Wenn $R = (sup, sig)$ eine τ-Region von A ist, dann nennen wir ${}^\bullet R^\bullet = \{e \in E \mid sig(e) \neq \mathsf{nop}\}$ die *Umgebung* von R; des Weiteren heißt R *d-beschränkt*, wenn $|{}^\bullet R^\bullet| \leq d$.

Diese Überlegungen ermöglichen es uns, die sich ergebenden parametrisierten Syntheseprobleme auf Basis von Zeugen für Trennungseigenschaften zu formulieren, wobei wir insbesondere die Implementierungen Sprachsimulation sowie Realisierung adressieren:

Boolesche τ-Sprachsimulation mit begrenzter Platzumgebung (Bτ-SMBP)	
Eingabe:	Ein Transitionssystem $A = (Z, E, \delta, \iota)$ und eine natürliche Zahl d.
Parameter:	d.
Frage:	Gibt es einen Zeugen $\mathcal{R}$ d-beschränkter Regionen für die τ-EZTE von A?

Boolesche τ-Realisierung mit begrenzter Platzumgebung (Bτ-RMBP)	
Eingabe:	Ein Transitionssystem $A = (Z, E, \delta, \iota)$ und eine natürliche Zahl d.
Parameter:	d.
Frage:	Gibt es einen Zeugen $\mathcal{R}$ d-beschränkter Regionen für die τ-ZTE und die τ-EZTE von A?

In diesem Kapitel zeigen wir einerseits, dass Bτ-SMBP und Bτ-RMBP in XP sind. Andererseits zeigen wir mittels parametrisierter Reduktionen für 99 der 128 booleschen Netztypen mit nop, dass diese Probleme nicht fest-Parameter-berechenbar sind, da sie wenigstens W[1]-schwer sind. Abbildung 6.2 gibt einen Überblick über die bisher ermittelten unteren Schranken für Bτ-SMBP und Bτ-RMBP.

Diskutieren wir kurz die „klassische“ Komplexität (der nicht-parametrisierten Versionen) dieser Probleme: Wie man leicht sieht, sind Bτ-SMBP und Bτ-RMBP NP-vollständig, wenn die unterliegenden unbeschränkten Probleme τ-Sprachsimulation und τ-Realisierung NP-vollständig sind. Während man nämlich für die Zugehörigkeit zu NP analog zu Lemma 4.1 argumentiert, folgt die NP-Schwere durch die triviale Reduktion, die ein Transitionssystem $A = (Z, E, \delta, \iota)$ auf ein Paar $(A, |E|)$ abbildet. Offenbar erlaubt A genau dann eine positive Entscheidung, wenn $(A, |E|)$ diese erlaubt, da höchstens alle Ereignisse von A in der Umgebung einer Region sein können. Damit ist die NP-Vollständigkeit der Probleme für die Netztypen C2 und C3 sowie C6 - C9 aus Abbildung 6.2 eine Konsequenz der in Kapitel 4 diskutierten Resultate. Anders verhält es sich mit den Netztypen C4 und C5, für die τ-Sprachsimulation und τ-Realisierung gemäß Satz 4.5 in P sind. Die NP-Vollständigkeit der nicht-parametrisierten Varianten von Bτ-SMBP und Bτ-RMBP folgt für diese Netztypen aus den angekündigten parametrisierten Reduktionen, weil diese insbesondere polynomiell sind und die reduzierten Probleme NP-vollständig sind.

Das folgende Lemma adressiert die Zugehörigkeit zur Komplexitätsklasse XP wobei der sich anschließende Beweis insbesondere für alle Implementierungen impliziert, dass für jedes fest gewählte $d \in \mathbb{N}$ in Polynomialzeit entschieden werden kann, ob ein Transitionssystem durch ein τ-Netz mit d-beschränkter Umgebungszahl implementiert werden kann:

Lemma 6.1. *Die parametrisierten Probleme* Bτ-SMBP *und* Bτ-RMBP *sind in XP.*

Beweis. Sei $A = (Z, E, \delta, \iota)$ ein Transitionssystem und $d \in \mathbb{N}$ sowie $|A|$ die Größe von A. Wenn $R = (sup, sig)$ eine τ-Region ist, dann ist R implizit bereits vollständig durch $sup(\iota)$

C	Netztyp τ	Schwere	Anzahl	Satz
1	$\{\mathsf{nop}\} \cup \omega$ und $\omega \subseteq \{\mathsf{used}, \mathsf{free}\}$	trivial	4	
2	$\{\mathsf{nop}, \mathsf{inp}, \mathsf{free}\}$, $\{\mathsf{nop}, \mathsf{inp}, \mathsf{used}, \mathsf{free}\}$, $\{\mathsf{nop}, \mathsf{out}, \mathsf{used}\}$, $\{\mathsf{nop}, \mathsf{out}, \mathsf{used}, \mathsf{free}\}$	W[1]	4	6.2.1
3	$\{\mathsf{nop}, \mathsf{inp}, \mathsf{out}\} \cup \omega$ und $\omega \subseteq \{\mathsf{used}, \mathsf{free}\}$	W[1]	4	6.2.2
4	$\{\mathsf{nop}, \mathsf{swap}\} \cup \omega$ und $\omega \subseteq \{\mathsf{inp}, \mathsf{out}, \mathsf{used}, \mathsf{free}\}$ und $\omega \cap \{\mathsf{inp}, \mathsf{out}\} \neq \emptyset$	W[1]	12	6.3.1
5	$\{\mathsf{nop}, \mathsf{swap}\} \cup \omega$ und $\emptyset \neq \omega \subseteq \{\mathsf{used}, \mathsf{free}\}$	W[1]	3	6.3.2
6	$\{\mathsf{nop}, \mathsf{inp}, \mathsf{set}\} \cup \omega$ und $\omega \subseteq \{\mathsf{out}, \mathsf{res}, \mathsf{swap}, \mathsf{used}, \mathsf{free}\}$ $\{\mathsf{nop}, \mathsf{out}, \mathsf{res}\} \cup \omega$ und $\omega \subseteq \{\mathsf{inp}, \mathsf{set}, \mathsf{swap}, \mathsf{used}, \mathsf{free}\}$	W[2]	56	6.4.1
7	$\{\mathsf{nop}, \mathsf{set}, \mathsf{res}\} \cup \omega$ und $\{\mathsf{nop}, \mathsf{set}, \mathsf{res}, \mathsf{swap}\} \cup \omega$ und $\emptyset \neq \omega \subseteq \{\mathsf{used}, \mathsf{free}\}$	W[2]	6	6.4.2
8	$\{\mathsf{nop}, \mathsf{set}, \mathsf{swap}\} \cup \omega$ und $\{\mathsf{nop}, \mathsf{out}, \mathsf{set}, \mathsf{swap}\} \cup \omega$ und $\{\mathsf{nop}, \mathsf{res}, \mathsf{swap}\} \cup \omega$ und $\{\mathsf{nop}, \mathsf{inp}, \mathsf{res}, \mathsf{swap}\} \cup \omega$ und $\emptyset \neq \omega \subseteq \{\mathsf{used}, \mathsf{free}\}$	W[2]	12	6.4.3
9	$\{\mathsf{nop}, \mathsf{inp}, \mathsf{res}, \mathsf{swap}\}$ und $\{\mathsf{nop}, \mathsf{out}, \mathsf{set}, \mathsf{swap}\}$	W[2]	2	6.4.4
10	$\{\mathsf{nop}, \mathsf{res}\} \cup \omega$ und $\omega \subseteq \{\mathsf{inp}, \mathsf{used}, \mathsf{free}\}$ und $\{\mathsf{nop}, \mathsf{set}\} \cup \omega$ und $\omega \subseteq \{\mathsf{out}, \mathsf{used}, \mathsf{free}\}$	offen	16	
11	$\{\mathsf{nop}, \mathsf{swap}\}$ und $\{\mathsf{nop}, \mathsf{inp}\}$, $\{\mathsf{nop}, \mathsf{inp}, \mathsf{used}\}$, $\{\mathsf{nop}, \mathsf{out}\}$, $\{\mathsf{nop}, \mathsf{out}, \mathsf{free}\}$, $\{\mathsf{nop}, \mathsf{set}, \mathsf{swap}\}$, $\{\mathsf{nop}, \mathsf{res}, \mathsf{swap}\}$, $\{\mathsf{nop}, \mathsf{set}, \mathsf{res}\}$, $\{\mathsf{nop}, \mathsf{set}, \mathsf{res}, \mathsf{swap}\}$	offen	9	

Abbildung 6.2: Ein Überblick über die aktuell ermittelten unteren Schranken für die parametrisierte Komplexität der Probleme Bτ-SMBP und Bτ-RMBP in Abhängigkeit des Netztyps τ.

und sig definiert Wir interessieren uns nur für Regionen, für die es ein $i \in \{0, \ldots, d\}$ gibt, sodass $|\{e \in E \mid sig(e) \neq \mathsf{nop}\}| = i$. Dabei enthält τ höchstens $|\tau| - 1 \leq 7$ boolesche Interaktionen, die von nop verschieden sind. Mit $sup(\iota) \in \{0, 1\}$ impliziert dies, dass A überhaupt höchstens $2 \cdot 7^d \cdot \sum_{i=0}^{d} \binom{|E|}{i}$, das heißt $\mathcal{O}(7^d |A|^d)$, passend beschränkte Regionen haben kann. Um zu verifizieren, ob eine gewählte Abbildung sig tatsächlich die Signatur einer Region $R = (sup, sig)$ ist und ob diese Region Trennungsprobleme löst, müssen wir sup explizit berechnen. Dazu berechnen wir zunächst (einmalig) einen Spannbaum A' von A. Mit dem Algorithmus von Tarjan [Tar77] beträgt der Zeitaufwand dafür $\mathcal{O}(|A|^2)$. In A' gibt es für jeden Zustand $z \in Z$ genau einen Pfad von ι nach z und A' genau $|S| - 1$ Kanten. Das heißt, wenn der Spannbaum vorhanden ist und $sup(\iota)$ und sig gewählt sind, dann können wir sup in Zeit $\mathcal{O}(|A|)$ berechnen. Folglich beträgt der Aufwand alle *potentiell* in Frage kommenden Regionen zu berechnen höchstens $\mathcal{O}(7^d |A|^{d+1})$. Im Anschluss müssen wir für jede *potentielle* Region überprüfen, ob sie *tatsächlich* eine Region ist, dass heißt, ob $s \xrightarrow{e} s' \in A$ stets $sup(s) \xrightarrow{sig(e)} sup(s') \in \tau$ impliziert. Dies ist für ein festes Paar (sup, sig) offenbar in Zeit $\mathcal{O}(|A|)$ möglich. Insgesamt

folgt aus diesen Überlegungen, dass mit Zeitaufwand $\mathcal{O}(7^d|A|^{d+2})$ alle d-beschränkten Regionen von A berechnet werden können.

Es bleibt noch zu überprüfen, ob A die τ-ZTE beziehungsweise die τ-EZTE hat. Für ein ZTP (z, z') beziehungsweise ein EZTP (z, e) von A müssen wir schauen, ob $sup(z) \neq sup(z')$ beziehungsweise ob $\neg sup(z) \xrightarrow{sig(e)} sup(z')$. Dies ist in Zeit $\mathcal{O}(|A|)$ möglich. Weil A höchstens $|A|^2$ Trennungsprobleme und höchstens $\mathcal{O}(7^d|A|^d)$ Regionen hat, kann – auch unter Berücksichtigung der Berechnung des Spannbaums und der Regionen– in Zeit $\mathcal{O}(7^d|A|^{d+2})$ entschieden werden, ob A die τ-ZTE oder die τ-EZTE hat. □

Wenn zwei boolesche Netztypen kongruent sind, dann genügt es, sich auf einen der beiden Netztypen zu konzentrieren, um (für beide Netztypen) die Frage nach der Implementierbarkeit eines Transitionssystems durch ein entsprechendes Netz mit d-beschränkter Umgebungszahl zu beantworten:

Satz 6.1. *Seien $\tau_0 = (Z_{\tau_0}, E_{\tau_0}, \delta_{\tau_0})$ und $\tau_1 = (Z_{\tau_1}, E_{\tau_1}, \delta_{\tau_1})$ zwei kongruente boolesche Netztypen, sodass für die bijektiven Abbildungen $\varphi : Z_{\tau_0} \to Z_{\tau_1}$ und $\psi : E_{\tau_0} \to E_{\tau_1}$, welche die Kongruenz gemäß Definition 2.24 rechtfertigen, $\psi(\mathsf{nop}) = \mathsf{nop}$ gilt. Darüber hinaus sei $A = (Z, E, \delta, \iota)$ ein Transitionssystem und d eine natürliche Zahl. Es gibt genau dann einen Zeugen $\mathcal{R}_0$ d-beschränkter τ_0-Regionen für die τ_0-ZTE beziehungsweise die τ_0-EZTE von A, wenn es einen Zeugen $\mathcal{R}_1$ d-beschränkter τ_1-Regionen für die τ_1-ZTE beziehungsweise die τ_1-EZTE von A gibt.*

Beweis. „⇒“: Sei α ein beliebiges aber festes EZTP beziehungsweise ZTP von A und $R_0 = (sup_0, sig_0)$ eine τ_0-Region, die α löst. Analog zum Beweis von Satz 2.2 erhalten wir eine τ_1-Region $R_1 = (sup_1, sig_1)$ die α löst durch $sup_1(z) = \varphi(sup_0(z))$ für alle $z \in Z$ und $sig_1(e) = \psi(sig_0(e))$ für alle $e \in E$. Wegen $\psi(\mathsf{nop}) = \mathsf{nop}$, folgt aus $|\{e \in E \mid sig_0(e) \neq \mathsf{nop}\}| \leq d$ auch $|\{e \in E \mid sig_1(e) = \psi(sig_0(e)) \neq \mathsf{nop}\}| \leq d$. Entsprechend argumentiert man für die andere Richtung. Weil α beliebig war, folgt die Behauptung. □

Die weitere Struktur dieses Kapitels ist wie folgt: Durch parametrisierte Reduktionen des Problems Regular Independent Set wird im nächsten Abschnitt 6.1 die W[1]-Schwere von Bτ-SMBP und Bτ-RMBP für die Netztypen C2 und C3 aus Abbildung 6.2 gezeigt; im Anschluss wird in Abschnitt 6.2 die W[1]-Schwere dieser Probleme für die Netztypen C4 und C5 aus Abbildung 6.2 durch Reduktionen des Problems Odd Set gezeigt; schließlich wird in Abschnitt 6.3 durch Reduktionen des Problems Hitting Set für die Netztypen C6 bis C9 aus Abbildung 6.2 bewiesen, dass Bτ-SMBP und Bτ-RMBP W[2]-schwer sind.

6.1 W[1]-Schwere durch Reduktion von Regular Independent Set

Der folgende Satz adressiert die Netztypen C2 und C3 aus Abbildung 6.2 und impliziert, dass Bτ-SMBP und Bτ-RMBP nicht fest-Parameter-berechenbar sind:

Satz 6.2. Bτ-SMBP *und* Bτ-RMBP *sind* W[1]*-schwer, wenn*

1. $\tau = \{\mathsf{nop}, \mathsf{inp}, \mathsf{free}\}$ *oder* $\tau = \{\mathsf{nop}, \mathsf{inp}, \mathsf{used}, \mathsf{free}\}$ *oder* $\tau = \{\mathsf{nop}, \mathsf{out}, \mathsf{used}\}$ *beziehungsweise* $\tau = \{\mathsf{nop}, \mathsf{out}, \mathsf{used}, \mathsf{free}\}$ *oder*
2. $\tau = \{\mathsf{nop}, \mathsf{inp}, \mathsf{out}\} \cup \omega$ *mit* $\omega \subseteq \{\mathsf{used}, \mathsf{free}\}$.

Wir beweisen Satz 6.2 durch parametrisierte Reduktionen des Problems Regular Independent Set, welches wie folgt definiert ist:

Regular Independent Set (RIS)	
Eingabe:	Ein Tripel $(\mathfrak{U}, M, \kappa)$ mit einer endlichen Menge $\mathfrak{U}$ und einem Mengensystem $M = \{M_0, \ldots, M_{m-1}\}$ mit $M_i = \{X_{i_0}, X_{i_1}\} \subseteq \mathfrak{U}$ für alle $i \in \{0, \ldots, m-1\}$ und einer natürlichen Zahl κ. Es gibt ein $r \in \mathbb{N}$, sodass $\lvert\{a \in M \mid X \in a\}\rvert = r$ für alle $X \in \mathfrak{U}$.
Parameter:	κ
Frage:	Gibt es eine unabhängige Menge S für $(\mathfrak{U}, M)$, dass heißt, $S \subseteq \mathfrak{U}$ und $\{X, Y\} \notin M$ für alle $X, Y \in S$, die $\lvert S\rvert \geq \kappa$ erfüllt?

Beispiel 6.2. Die Instanz $(\mathfrak{U}, M, 2)$ mit $\mathfrak{U} = \{X_0, X_1, X_2, X_3\}$ und $M = \{M_0, \ldots, M_3\}$, wobei $M_0 = \{X_0, X_1\}$, $M_1 = \{X_0, X_3\}$, $M_2 = \{X_1, X_2\}$ sowie $M_3 = \{X_2, X_3\}$, erlaubt eine positive Entscheidung: es ist $S = \{X_0, X_2\}$ eine unabhängige Menge für $(\mathfrak{U}, M)$.

Es ist RIS ein geläufiges W[1]-vollständiges Problem, siehe zum Beispiel [CFK+15]. Der Beweis von Satz 6.2.1 und Satz 6.2.2 erfolgt jeweils mittels einer parametrisierten Reduktion von RIS auf Bτ-SMBP beziehungsweise Bτ-RMBP. Dabei werden wir uns bei der Reduktion für den Beweis von Satz 6.2.1 auf $\tau \in \{\{\mathsf{nop}, \mathsf{inp}, \mathsf{free}\}, \{\mathsf{nop}, \mathsf{inp}, \mathsf{used}, \mathsf{free}\}\}$ beschränken. Die Aussage für die verbleibenden Netztypen von Satz 6.2.1 folgt dann offensichtlich aus Satz 6.1. Im Detail weichen die Reduktionen für den Beweis von Satz 6.2.1 beziehungsweise von Satz 6.2.2 voneinander ab. Dennoch folgen sie einer gemeinsamen Idee, die wir im Folgenden zunächst vorstellen.

Die Strategie der Reduktionen von Regular Independent Set

Solange nicht ausdrücklich anders vereinbart, sei im weiteren Verlauf dieses Abschnitts $(\mathfrak{U}, M, \kappa)$ eine beliebige aber feste Eingabe für RIS mit $\mathfrak{U} = \{X_0, \ldots, X_{n-1}\}$ und $M = \{M_0, \ldots, M_{m-1}\}$, wobei wir ohne Beeinträchtigung der Allgemeinheit $i_0 < i_1$ für die Elemente der Menge $M_i = \{X_{i_0}, X_{i_1}\}$ für alle $i \in \{0, \ldots, m-1\}$ annehmen. Darüber hinaus sei $r \in \mathbb{N}$ die natürliche Zahl, sodass $\lvert\{a \in M \mid X \in a\}\rvert = r$ für alle $X \in \mathfrak{U}$.

Für einen betreffenden Netztypen τ liefern die angekündigten Reduktionen jeweils ein Paar (A_τ, d_τ) mit Transitionssystem A_τ und natürlicher Zahl d_τ, sodass es genau dann eine unabhängige Menge S mit wenigstens κ Elementen für $(\mathfrak{U}, M)$ gibt, wenn es einen Zeugen $\mathcal{R}$ d_τ-beschränkter Regionen für die τ-EZTE und die τ-ZTE von A_τ gibt. Dabei gilt für den Parameter $d_\tau = f(\kappa)$, wobei f eine lineare Funktion ist. Das Transitionssystem

A_τ hat für jedes $i \in \{0, \ldots, m-1\}$ folgenden gerichteten Pfad P_i auf dem die Elemente von $M_i = \{X_{i_0}, X_{i_1}\}$ als Ereignisse stattfinden:

$$P_i = t_{i,0} \xrightarrow{X_{i_0}} t_{i,1} \xrightarrow{X_{i_1}} t_{i,2}$$

Darüber hinaus hat A_τ ein EZTP α, sodass die folgende Eigenschaft erfüllt ist: Wenn $R = (sup, sig)$ eine τ-Region von A_τ ist, die α löst, dann gibt es genau $r\kappa$ Indizes $i_0, \ldots, i_{r\kappa-1} \in \{0, \ldots, m-1\}$, sodass $sup(t_{i_\ell,0}) = 1$ und $sup(t_{i_\ell,2}) = 0$ für alle $\ell \in \{0, \ldots, r\kappa - 1\}$. Dies bedeutet für alle $\ell \in \{0, \ldots, r\kappa - 1\}$, dass das Bild $P_{i_\ell}^R$ des Pfades P_{i_ℓ} unter R ein Pfad von 1 nach 0 in τ ist. Folglich muss es ein $X \in \{X_{i_{\ell 0}}, X_{i_{\ell 1}}\}$ mit $sig(X) = \mathsf{inp}$ geben. Wenn nun $X, Y \in \mathfrak{U}$ zwei Ereignisse mit $sig(X) = sig(Y) = \mathsf{inp}$ sind, dann gilt $\{X, Y\} \notin M$. Andernfalls gäbe es ein $i \in \{0, \ldots, m-1\}$ mit $M_i = \{X, Y\}$ und das Bild P_i^R des Pfades P_i, auf dem X und Y als Ereignisse stattfinden, wäre ein Pfad $sup(t_{i,0}) \xrightarrow{\mathsf{inp}} sup(t_{i,1}) \xrightarrow{\mathsf{inp}} sup(t_{i,2})$ in τ. Dies ist ein Widerspruch, denn $sup(t_{i,0}) \xrightarrow{\mathsf{inp}} sup(t_{i,1})$ impliziert $sup(t_{i,1}) = 0$ und inp ist nicht an 0 definiert. Daraus folgt $\{X, Y\} \notin M$ für je zwei Elemente X, Y der Menge $S = \{X \in \mathfrak{U} \mid sig(X) = \mathsf{inp}\}$. Weil nun jedes $X \in \mathfrak{U}$ ein Element von genau r Mengen aus M ist und auf jedem der $r\kappa$ Pfade $P_{i_0}, \ldots, P_{i_{r\kappa-1}}$, welche die $r\kappa$ Mengen $M_{i_0}, \ldots, M_{i_{r\kappa-1}}$ von M repräsentieren, ein Ereignis mit Signatur inp stattfindet, gilt außerdem $|S| \geq \kappa$. Somit ist S eine gesuchte unabhängige Menge. Wenn also A_τ einen Zeugen $\mathcal{R}$ d_τ-beschränkter Regionen für die τ-EZTE hat, dann gibt es eine unabhängige Menge mit κ Elementen für $(\mathfrak{U}, M)$, denn $\mathcal{R}$ enthält eine Region, die α löst.

Für die andere Richtung sichert die Konstruktion von (A_τ, d_τ) zu, dass es einen Zeugen $\mathcal{R}$ d_τ-beschränkter Regionen für die τ-EZTE und die τ-ZTE von A_τ gibt, wenn es eine unabhängige Menge mit κ Elementen für $(\mathfrak{U}, M)$ gibt.

In den folgenden beiden Abschnitten stellen wir die genaue Konstruktion von (A_τ, d_τ) in Abhängigkeit der untersuchten Netztypen vor.

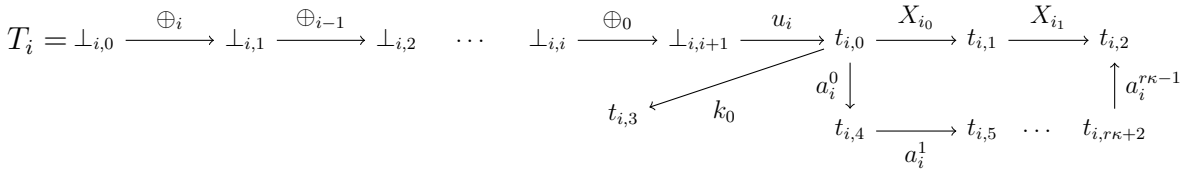
Der Beweis von Satz 6.2.1

Solange nicht ausdrücklich anders vereinbart, sei im Folgenden $\tau = \{\mathsf{nop}, \mathsf{inp}, \mathsf{free}\}$ oder $\tau = \{\mathsf{nop}, \mathsf{inp}, \mathsf{used}, \mathsf{free}\}$ beliebig aber fest. Wir beweisen, dass Bτ-SMBP und Bτ-RMBP W[1]-schwer sind. Die Aussage von Satz 6.2.1 für die verbleibenden Netztypen folgt dann aus der Beliebigkeit von τ und Satz 6.1.

Die Konstruktion von (A_τ, d_τ) ist wie folgt: Zunächst definieren wir für den Parameter $d_\tau = \kappa \cdot (r+1) + 2$. Das Transitionssystem A_τ hat das folgende Gadget H, welches das angekündigte EZTP $\alpha = (k_1, h_0)$ liefert:

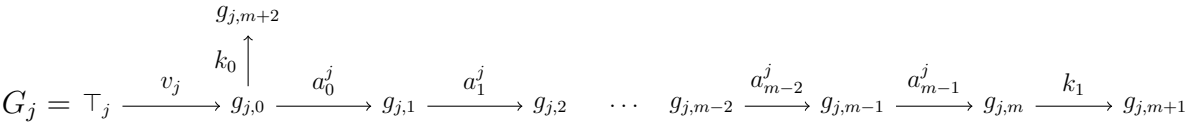
$$H = \quad h_0 \xrightarrow{k_0} h_1 \xrightarrow{k_1} h_2$$

Darüber hinaus hat A_τ für alle $i \in \{0, \ldots, m-1\}$ das folgende Gadget T_i, welches insbesondere den angekündigten Pfad P_i zur Verfügung stellt, auf dem die Elemente von $M_i = \{X_{i_0}, X_{i_1}\}$ als Ereignisse stattfinden.



Außerdem hat T_i die $r\kappa$ Ereignisse $a_i^0, \ldots, a_i^{r\kappa-1}$ die hintereinander auf einem gerichteten Pfad stattfinden. Für alle $i \in \{0, \ldots, m-1\}$ und alle $j \in \{0, \ldots, r\kappa - 1\}$ bezeichnen wir a_i^j als das j-te mit M_i assoziierte a-Ereignis.

Weiterhin hat das Transitionssystem A_τ für jedes $j \in \{0, \ldots, r\kappa - 1\}$ das folgende Gadget G_j:



Man beachte, dass auf G_j für jedes $i \in \{0, \ldots, m-1\}$ das j-te mit M_i assoziierte a-Ereignis stattfindet, wobei die Ereignisse $a_0^j, \ldots, a_{m-1}^j$ sukzessive hintereinander stattfinden.

Um die eingeführten Gadgets miteinander zu verbinden, fügen wir zunächst für alle $i \in \{1, \ldots, m-1\}$ die Kante $\perp_{i,0} \xrightarrow{\ominus_i} \perp_{i-1,0}$ sowie die Kante $\perp_{0,0} \xrightarrow{\ominus_0} h_0$ hinzu. Des Weiteren fügen wir für alle $j \in \{0, \ldots, r\kappa - 2\}$ die Kante $\perp_{m-1,0} \xrightarrow{\odot_0} \top_0$ sowie die Kante $\top_j \xrightarrow{\odot_{j+1}} \top_{j+1}$ hinzu. Das resultierende Transitionssystem A_τ hat den Initialzustand $\perp_{m-1,0}$ und es ist leicht zu sehen, dass (A_τ, d_τ) aus einer parametrisierten Reduktion resultiert. Abbildung 6.3 zeigt das Transitionssystem A_τ, das sich aus Beispiel 6.2 ergibt.

Lemma 6.2. *Wenn A_τ die τ-EZTE hat, dann gibt es eine unabhängige Menge mit κ Elementen für $(\mathfrak{U}, M)$.*

Beweis. Weil A_τ die τ-EZTE hat, gibt es eine τ-Region, die $\alpha = (k_1, h_0)$ löst. Sei $R = (sup, sig)$ so eine Region. Für alle $sig(k_0) \in \{\mathsf{nop}, \mathsf{inp}, \mathsf{free}, \mathsf{used}\}$ gilt, dass $0 \xrightarrow{sig(k_0)} 1$ keine Kante in τ ist. Deshalb ist $sig(k_1) \in \{\mathsf{inp}, \mathsf{used}\}$ und $sup(h_0) = 0$ offensichtlich nicht möglich. Somit gilt $sig(k_1) = \mathsf{free}$ und $sup(h_0) = 1$ und wir erhalten $sup(h_1) = 0$ und $sig(k_0) = \mathsf{inp}$.

Wir zeigen nun, dass es $r\kappa$ Indizes $i_0, \ldots, i_{r\kappa-1} \in \{0, \ldots, m-1\}$ gibt, sodass $sup(t_{i_\ell,0}) = 1$ und $sup(t_{i_\ell,2}) = 0$ für alle $\ell \in \{0, \ldots, r\kappa - 1\}$: Sei $j \in \{0, \ldots, m-1\}$ beliebig aber fest. Aus $sig(k_0) = \mathsf{inp}$ und $sig(k_1) = \mathsf{free}$ erhalten wir $sup(g_{j,0}) = 1$ und $sup(g_{j,m}) = 0$. Insbesondere ist das Bild Q_j^R des Pfades $Q_j = g_{j,0} \xrightarrow{a_0^j} \ldots \xrightarrow{a_{m-1}^j} g_{j,m}$ unter R ein Pfad von 1 nach 0 in τ. Deshalb gibt es ein $i \in \{0, \ldots, m-1\}$ mit $sig(a_i^j) = \mathsf{inp}$. Dieser Index ist eindeutig, denn es gibt keinen Pfad in τ, auf dem inp zweimal stattfindet. Wegen $sig(k_0) = \mathsf{inp}$, gilt $sup(t_{i,0}) = 1$ und aus $sig(a_i^j) = \mathsf{inp}$ folgt $sup(t_{i,j+1}) = 0$. Weil es keinen Pfad von 0 nach 1 in τ gibt, impliziert dies $sup(t_{i,2}) = 0$. Aus der Beliebigkeit

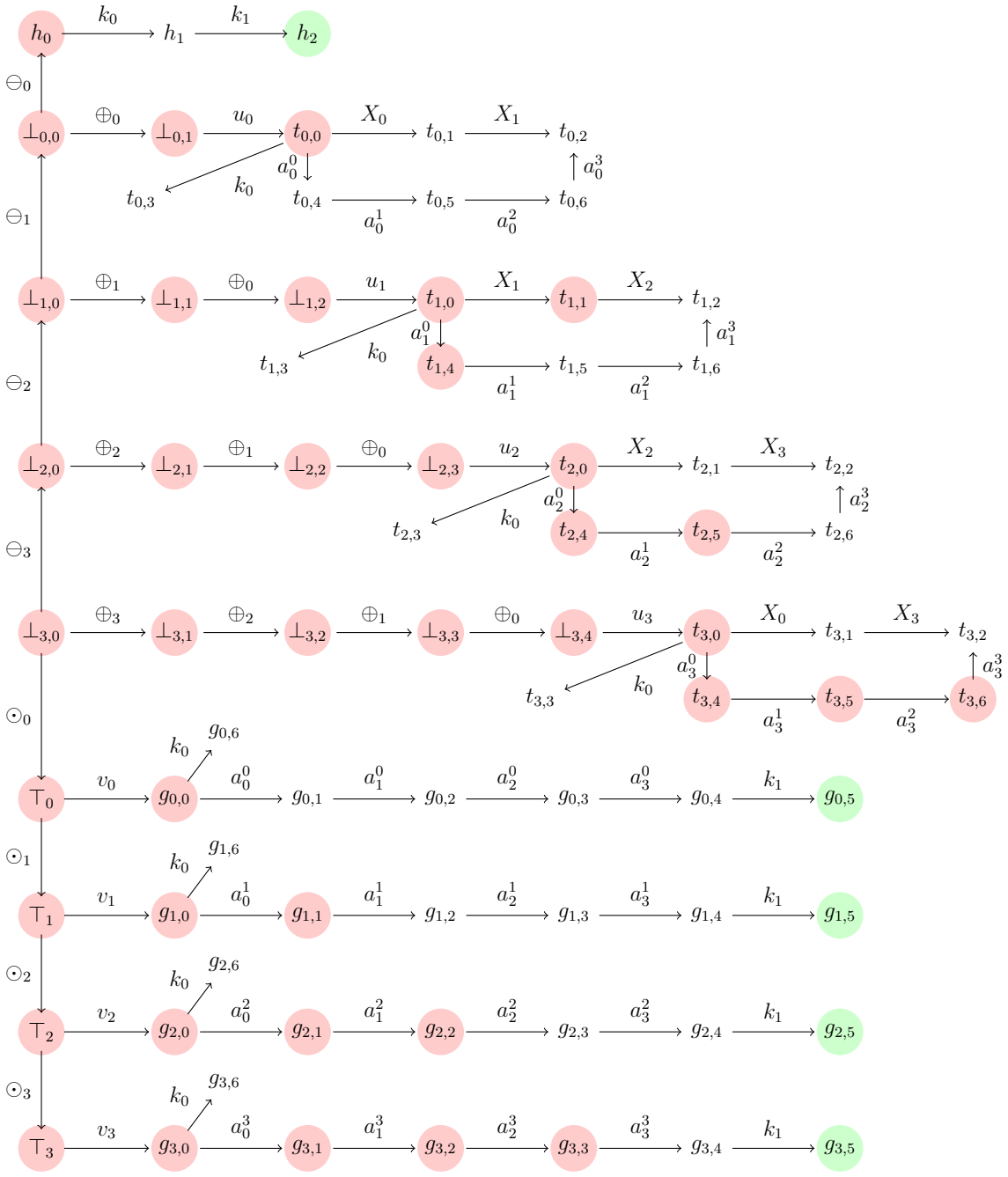


Abbildung 6.3: Das Transitionssystem A_τ, welches sich aus der Reduktion für den Beweis von Satz 6.2.1 und der Instanz von Beispiel 6.2 ergibt. Der rote Bereich skizziert die Region R_1 aus Fakt 6.2, die α löst. Für den grünen Bereich siehe Abbildung 6.4.

von j und der Eindeutigkeit des Ereignisses mit der Signatur inp auf Q_j folgt, dass es $r\kappa$ paarweise voneinander verschiedene Ereignisse $a^0_{i_0}, a^1_{i_1}, \ldots, a^{r\kappa-1}_{i_{r\kappa-1}}$ gibt, welche die $r\kappa$ verschiedenen Gadgets $T_{i_0}, \ldots, T_{i_{r\kappa-1}}$ auswählen, sodass $sup(t_{i_\ell,0}) = 1$ und $sup(t_{i_\ell,2}) = 0$ für alle $\ell \in \{0, \ldots, r\kappa - 1\}$. Wie bereits in der Einführung unseres Reduktionsansatzes argumentiert, impliziert dies, dass $S = \{X \in \mathfrak{U} \mid sig(X) = \mathsf{inp}\}$ eine unabhängige Menge für $(\mathfrak{U}, M)$ mit wenigstens κ Elementen definiert. □

Solange nicht ausdrücklich anders vereinbart, sei S im Folgenden eine beliebige aber feste unabhängige Menge mit κ Elementen für $(\mathfrak{U}, M)$. Um den Beweis von Satz 6.2.1 zu vervollständigen, zeigen wir nun, dass die Existenz von S einen Zeugen $\mathcal{R}$ d_τ-beschränkter Regionen für die τ-EZTE und die τ-ZTE von A_τ impliziert. Der Beweis der Existenz von $\mathcal{R}$ ist technisch aufwendig. Der besseren Lesbarkeit wegen, behandeln wir deshalb Ereignisse „gleicher Art“ jeweils individuell und beweisen einen entsprechenden Fakt, der ihre Lösbarkeit konstatiert. Die (Gesamtaussage der) Existenz von $\mathcal{R}$ ergibt sich dann als Summe der einzelnen Fakten. Alle präsentierten Regionen $R = (sup, sig)$ werden –wie bisher auch– implizit durch den Supportwert $sup(\bot_{m-1,0})$ für den Initialzustand und sig definiert. Weil alle diese Regionen $sup(\bot_{m-1,0}) = 1$ erfüllen, beschränken wir uns bei der Präsentation auf die Definition von sig und setzen $sup(\bot_{m-1,0}) = 1$ stets als bekannt voraus. Für eine noch kompaktere und übersichtlichere Darstellung fassen wir einerseits Ereignisse mit gleichem Signaturwert $e_\tau \in \tau$ wie gehabt in einer entsprechenden Menge $\mathcal{E}^R_{e_\tau} = \{e \in E(A_\tau) \mid sig(e) = e_\tau\}$ zusammen, wenn dies möglich ist. Darüber hinaus gilt genau für jene Ereignisse $e \in E(A_\tau)$, die nicht explizit in einer Menge $\mathcal{E}^R_{e_\tau}$ mit $e_\tau \neq \mathsf{nop}$ aufgelistet sind, dass $sig(e) = \mathsf{nop}$. Wir werden die Menge $\mathcal{E}^R_{\mathsf{nop}}$ deshalb nicht explizit definieren.

Fakt 6.1. *Das Ereignis k_0 ist durch d_τ-beschränkte τ-Regionen lösbar.*

Beweis. Sei $i \in \{0, \ldots m-1\}$ beliebig aber fest und seien $i_1, \ldots, i_{r-1} \in \{0, \ldots, m-1\} \setminus \{i\}$ die paarweise verschiedenen Indizes, sodass $X_{i_0} \in M_{i_\ell}$ für alle $\ell \in \{1, \ldots, r-1\}$. Die folgende Region $R_1 = (sup_1, sig_1)$ löst (k_0, z) für alle $z \in \{t_{i,1}, \ldots, t_{i,r\kappa+2}\}$ und alle $z \in \{h_1, h_2\}$: $\mathcal{E}^{R_1}_{\mathsf{inp}} = \{k_0, X_{i_0}, a^0_i, a^1_{i_1}, a^2_{i_2}, \ldots, a^{r-1}_{i_{r-1}}\}$.

Die folgende Region $R_2 = (sup_2, sig_2)$ löst (k_0, z) für alle $z \in \{\bot_{i,0}, \ldots, \bot_{i,i+1}\}$: $\mathcal{E}^{R_2}_{\mathsf{free}} = \{k_0\}$ und $\mathcal{E}^{R_2}_{\mathsf{inp}} = \{u_i, \ominus_i, \oplus_{i+1}, \odot_0\}$. Dabei ist $\oplus_{i+1}$ natürlich zu ignorieren, wenn $i = m-1$.

Aus der Beliebigkeit von i folgt, dass (k_0, z) für alle relevanten Zustände $z \in \bigcup_{i=0}^{m-1} Z(T_i)$ und $z \in Z(H)$ durch d_τ-beschränkte τ-Regionen lösbar ist.

Die folgende Region $R_3 = (sup_3, sig_3)$ löst (k_0, z) für alle $z \in \{\top_0, \ldots, \top_{r\kappa-1}\}$: $\mathcal{E}^{R_3}_{\mathsf{free}} = \{k_0\}$ und $\mathcal{E}^{R_3}_{\mathsf{inp}} = \{\oplus_{m-1}, \ominus_{m-1}, v_0, \ldots, v_{r\kappa-1}\}$. Man beachte, dass für diese Region $|\{e \in E(A_\tau) \mid sig_3(e) \neq \mathsf{nop}\}| = r\kappa + 3$ gilt.

Sei $j \in \{0, \ldots, r\kappa - 1\}$ beliebig aber fest und seien $i_1, \ldots, i_{r-1} \in \{1, \ldots, m-1\}$ die paarweise voneinander (und von 0) verschiedenen Indizes, sodass $X_{0_0} \in M_{i_\ell}$ für alle $\ell \in \{1, \ldots, r-1\}$. Die folgende Region $R_4 = (sup_4, sig_4)$ löst (k_0, z) für alle $z \in \{g_{j,1}, \ldots, g_{j,m+1}\}$. Aus technischen Gründen unterscheiden wir die drei Fälle (1) $j = 0$ und (2) $0 < j < r-1$ und (3) $r-1 \leq j$:

(1): $\mathcal{E}^{R_4}_{\mathsf{inp}} = \{k_0, X_{0_0}, a^0_0, a^1_{i_1}, a^2_{i_2}, \ldots, a^{r-1}_{i_{r-1}}\}$;
(2): $\mathcal{E}^{R_4}_{\mathsf{inp}} = \{k_0, X_{0_0}, a^0_{i_1}, \ldots, a^{j-1}_{i_j}, a^j_0, a^{j+1}_{i_j}, \ldots, a^r_{i_{r-1}}\}$;
(3): $\mathcal{E}^{R_4}_{\mathsf{inp}} = \{k_0, X_{0_0}, a^0_{i_1}, \ldots, a^{r-2}_{i_{r-1}}, a^j_0\}$. □

Fakt 6.2. *Das Ereignis k_1 ist durch d_τ-beschränkte τ-Regionen lösbar.*

Beweis. Wir benötigen die Existenz der unabhängigen Menge S mit κ Elementen. Jedes Element $X \in S$ ist ein Element von genau r verschiedenen Mengen von M. Deshalb gibt es genau $r\kappa$ paarweise verschiedene Indizes $i_0, \ldots, i_{r\kappa-1} \in \{0, \ldots, m-1\}$, sodass $S \cap M_{i_\ell} \neq \emptyset$ für alle $\ell \in \{0, \ldots, r\kappa\}$. Die folgende Region $R_1 = (sup_1, sig_1)$ löst α: $\mathcal{E}^{R_1}_{\mathsf{free}} = \{k_1\}$ und $\mathcal{E}^{R_1}_{\mathsf{inp}} = \{k_1\} \cup S \cup \{a^\ell_{i_\ell} \mid \ell \in \{0, \ldots, r\kappa - 1\}\}$. Man beachte, dass $|\{e \in E(A_\tau) \mid sig_1(e) \neq \mathsf{nop}\}| = 2 + \kappa + r\kappa = d$.

Die folgende Region $R_2 = (sup_2, sig_2)$ löst (k_1, z) für alle $z \in \{q \in Z(A_\tau) \mid \xrightarrow{k} q\}$: $\mathcal{E}^{R_2}_{\mathsf{inp}} = \{k_1\}$.

Die folgende Region $R_3 = (sup_3, sig_3)$ löst (k_1, z) für alle $z \in \bigcup_{i=0}^{m-1} Z(T_i)$: $\mathcal{E}^{R_3}_{\mathsf{free}} = \{k_1\}$ und $\mathcal{E}^{R_3}_{\mathsf{inp}} = \{\ominus_0, \odot_0\}$.

Es bleibt die Lösbarkeit der EZTP zu zeigen, an denen Zustände aus $G_0, \ldots, G_{r\kappa-1}$ beteiligt sind. Sei $j \in \{0, \ldots, r\kappa - 1\}$ beliebig aber fest. Das Ereignis k_1 findet an $g_{j,m}$ statt und das EZTP $(k_1, g_{j,m+1})$ ist bereits durch R_2 gelöst. Wir definieren nun die Region $R_4 = (sup_4, sig_4)$, die (k_1, z) für alle verbleibenden $z \in Z(G_j) \setminus \{g_{j,m}, g_{j,m+1}\}$ löst: Seien $i_0, \ldots, i_{r-2} \in \{0, \ldots, m-2\}$ die $r-1$ paarweise verschiedenen Indizes, sodass $X_{m-1_0} \in M_{i_\ell}$ für alle $\ell \in \{0, \ldots, r-2\}$. (Das heißt, $M_{i_0}, \ldots, M_{i_{r-2}}, M_{m-1}$ sind genau die Mengen von M, die das Element X_{m-1_0} enthalten.) Für die korrekte Definition von R_4 unterscheiden wir die drei Fälle (1) $j = 0$ und (2) $0 < j < r-1$ und (3) $r-1 \leq j$. Für jeden Fall definieren wir $\mathcal{E}^{R_4}_{\mathsf{free}} = \{k_1\}$ und dann je nach Fall
(1): $\mathcal{E}^{R_4}_{\mathsf{inp}} = \{\ominus_0, X_{m-1_0}, a^0_{m-1}, a^1_{i_1}, a^2_{i_2}, \ldots, a^{r-1}_{i_{r-1}}\} \cup \{v_i \mid i \in \{r, \ldots, r\kappa - 1\}\}$;
(2): $\mathcal{E}^{R_4}_{\mathsf{inp}} = \{\ominus_0, X_{m-1_0}, a^0_{i_1}, \ldots, a^{j-1}_{i_j}, a^j_{m-1}, a^{j+1}_{i_{j+1}}, \ldots, a^{r-1}_{i_{r-1}}\} \cup \{v_i \mid i \in \{r, \ldots, r\kappa - 1\}\}$;
(3): $\mathcal{E}^{R_4}_{\mathsf{inp}} = \{\ominus_0, X_{m-1_0}, a^0_{i_1}, \ldots, a^{r-2}_{i_{r-1}}, a^j_{m-1}\} \cup \{v_i \mid i \in \{r-1, \ldots, r\kappa - 1\} \setminus \{j\}\}$.
Man beachte, dass R_4 stets $|\{e \in E(A_\tau) \mid sig_4(e) \neq \mathsf{nop}\}| \leq r\kappa + 3 \leq d$ erfüllt. □

Fakt 6.3. *Für alle $i \in \{0, \ldots, m-1\}$ und alle $j \in \{0, \ldots, r\kappa - 1\}$ ist das Ereignis a^j_i durch d_τ-beschränkte τ-Regionen lösbar.*

Beweis. Seien $i \in \{0, \ldots, m-1\}$ und $j \in \{0, \ldots, r\kappa - 1\}$ beliebig aber fest. Es ist leicht zu sehen, dass (a^j_i, z) gelöst werden kann, wenn z ein „$\bot$“- beziehungsweise ein „$\top$“-Zustand ist. Wir werden die entsprechenden EZTP deshalb nicht explizit diskutieren.

Die folgende Region $R_1 = (sup_1, sig_1)$ löst (a^j_i, z) für alle $z \in Z(A_\tau) \setminus (\{t_{i,0}, \ldots, t_{r\kappa+2}\} \cup \{g_{j,0}, \ldots, g_{j,m+1}\})$: $\mathcal{E}^{R_1}_{\mathsf{free}} = \{a^j_i\}$ und $\mathcal{E}^{R_1}_{\mathsf{inp}} = \{u_i, v_j\}$.

Wir zeigen nun, dass (a^j_i, z) für alle relevanten $z \in \{t_{i,0}, \ldots, t_{i,r\kappa+2}\}$ τ-lösbar ist. Wenn $j = 0$, dann sind wir bereits fertig, denn die Region R_1 aus Fakt 6.1 löst die entsprechenden EZTP. Schauen wir uns deshalb den Fall $j \geq 1$ an: Seien $i_0, \ldots, i_{r-2} \in \{0, \ldots, m-1\} \setminus \{i\}$ die $r-1$ paarweise voneinander verschiedenen Indizes (ungleich i), sodass $X_{i_0} \in M_{i_\ell}$ für

alle $\ell \in \{0, \ldots, r-2\}$. Seien außerdem $j_0, \ldots, j_{r-2} \in \{0, \ldots, r\kappa - 1\} \setminus \{j-1, j\}$ paarweise voneinander verschieden.

Die folgende Region $R_2 = (sup_2, sig_2)$ löst (a_i^j, z) für alle $z \in \{t_{i,0}, t_{i,3}, \ldots, t_{i,j+2}\}$: $\mathcal{E}_{\mathsf{free}}^{R_2} = \{a_i^j\}$ und $\mathcal{E}_{\mathsf{inp}}^{R_2} = \{k_0, X_{i_0}, v_j, a_i^{j-1}, a_{i_0}^{j_0}, \ldots, a_{i_{r-2}}^{j_{r-2}}\}$.
Man beachte, dass R_2 auch $(a_i^j, g_{j,m+3})$ löst.

Die folgende Region $R_3 = (sup, sig)$ löst (a_i^j, z) für alle $z \in \{t_{i,1}, t_{i,2}, t_{i,j+4}, \ldots, t_{i,r\kappa+2}\}$: für alle $e \in E$, wenn $e \in \{X_{i_0}, a_i^j, a_{i_0}^{j_0}, \ldots, a_{i_{r-2}}^{j_{r-2}}\}$, dann $sig(e) = \mathsf{inp}$, sonst $sig(e) = \mathsf{nop}$.

Um den Beweis dieses Fakts zu vervollständigen, bleibt zu zeigen, dass (a_i^j, z) für alle relevanten $z \in \{g_{j,0}, \ldots, g_{j,m+2}\}$ passend τ-lösbar ist. Man beachte, dass die (gerade definierte) Region R_3 das EZTP $(a_i^j, g_{j,\ell})$ für alle $\ell \in \{i+1, \ldots, m+2\}$ löst. Deshalb sind wir für den Fall $i = 0$ bereits fertig. Betrachten wir nun den Fall $i > 0$. Sei $X \in M_{i-1}$ ein beliebiges aber festes Element, welches nicht in M_i ist. (Man beachte, dass X wegen $M_{i-1} \neq M_i$ existiert.) Seien $i_0, \ldots, i_{r-2} \in \{0, \ldots, m-1\} \setminus \{i-1\}$ die $r-1$ paarweise verschiedenen Indizes (ungleich $i-1$), sodass $X \in M_{i_\ell}$ für alle $\ell \in \{0, \ldots, r-2\}$. Weiterhin seien $j_0, \ldots, j_{r-2} \in \{0, \ldots, r\kappa - 1\} \setminus \{j\}$ paarweise verschieden.

Die folgende Region $R_4 = (sup_4, sig_4)$ löst (a_i^j, z) für alle $z \in \{g_{j,0}, \ldots, g_{j,i-1}\}$: $\mathcal{E}_{\mathsf{free}}^{R_4} = \{a_i^j\}$ und $\mathcal{E}_{\mathsf{inp}}^{R_4} = \{X, u_i, a_{i-1}^j, a_{i_0}^{j_0}, \ldots, a_{i_{r-2}}^{j-2}\}$.

Dies beweist den Fakt. □

Fakt 6.4. *Wenn $X \in \mathfrak{U}$, dann ist X durch d_τ-beschränkte τ-Regionen lösbar.*

Beweis. Sei $X \in \mathfrak{U}$ beliebig aber fest und seien $i_0, \ldots, i_{r-1} \in \{0, \ldots, m-1\}$ die paarweise verschiedenen Indizes, sodass $X \in M_{i_\ell}$ für alle $\ell \in \{0, \ldots, r-1\}$.

Die folgende Region $R_1 = (sup_1, sig_1)$ löst (X, z) bereits für alle Zustände $z \in Z(A_\tau) \setminus (\bigcup_{\ell=0}^{r-1} \{t_{i_\ell,0}, \ldots, t_{i_\ell,r\kappa+2}\})$: $\mathcal{E}_{\mathsf{free}}^{R_1} = \{X\}$ und $\mathcal{E}_{\mathsf{inp}}^{R_1} = \{u_{i_0}, \ldots, u_{i_{r-1}}\}$.

Sei $\ell \in \{0, \ldots, r-1\}$ beliebig aber fest. Um den Beweis dieses Fakts zu vervollständigen, müssen wir zeigen, dass (X, z) für alle relevanten Zustände $z \in \{t_{i_\ell,0}, \ldots, t_{i_\ell,r\kappa+2}\}$ passend τ-lösbar ist.

Sei $j_\ell = 0$ und seien $j_0, \ldots, j_{\ell-1}, j_{\ell+1}, \ldots, j_{r-1} \in \{1, \ldots, r\kappa - 1\}$ paarweise verschieden. Die folgende Region $R_2 = (sup_2, sig_2)$ löst (X, z) für alle $z \in \{t_{i_\ell,2}, \ldots, t_{i_\ell,r\kappa+2}\}$ und wenn $X = X_{i_{\ell 0}}$, dann auch für $z = t_{i_\ell,1}$: $\mathcal{E}_{\mathsf{inp}}^{R_2} = \{X, k_0, a_{i_0}^{j_0}, \ldots, a_{i_{r-1}}^{j_{r-1}}\}$.

Man beachte, dass wir für den Fall $X = X_{i_{\ell 0}}$ bereits fertig sind. Wenden wir uns nun dem Fall $X = X_{i_{\ell 1}}$ zu, das heißt, wir müssen noch $(X, t_{i_\ell,0})$ lösen. Seien $j_0, \ldots, j_{r-2} \in \{0, \ldots, m-1\} \setminus \{i_\ell\}$ die paarweise verschiedenen Indizes (ungleich i_ℓ), sodass $X_{i_{\ell 0}} \in M_{j_n}$. Man beachte, dass dies $X_{i_{\ell 1}} \notin M_{j_n}$ für alle $n \in \{0, \ldots, r-2\}$ impliziert. Seien außerdem $p_0, \ldots, p_{r-2} \in \{1, \ldots, r\kappa - 1\}$ paarweise verschieden. Die folgende Region $R_3 = (sup_3, sig_3)$ löst $(X, t_{i_\ell,0})$: $\mathcal{E}_{\mathsf{free}}^{R_3} = \{X\}$ und $\mathcal{E}_{\mathsf{inp}}^{R_3} = \{u_{i_0}, \ldots, u_{i_{\ell-1}}, u_{i_{\ell+1}}, \ldots, u_{i_{r-1}}\} \cup \{X_{i_{\ell 0}}, a_{i_\ell}^0, a_{j_0}^{p_0}, \ldots, a_{j_{r-2}}^{p_{r-2}}\}$. □

Fakt 6.5. *Wenn $e \in \{u_0, \ldots, u_{m-1}\}$, dann ist e durch d_τ-beschränkte τ-Regionen lösbar.*

Beweis. Sei $i \in \{0, \ldots, m-1\}$ beliebig aber fest. Die folgende Region $R_1 = (sup_1, sig_1)$ löst (u_i, z) für alle $z \in \bigcup_{j=0}^{r\kappa-1} Z(G_j)$ und $z \in \bigcup_{\ell=0}^{m-1} \{\bot_{\ell,0}, \ldots, \bot_{\ell,\ell}\} \cup \{h_0, h_1, h_2\}$: $\mathcal{E}_{\mathsf{free}}^{R_1} = \{u_i\}$ und $\mathcal{E}_{\mathsf{inp}}^{R_1} = \{\oplus_0\}$.

Die folgende Region $R_2 = (sup_2, sig_2)$ löst (u_i, z) für alle relevanten $z \in \bigcup_{\ell=0}^{m-1} Z(T_\ell)$ und alle $z \in (\bigcup_{\ell=0}^{m-1}\{\bot_{\ell,\ell+1}\}) \setminus \{\bot_{i,i+1}\})$: $\mathcal{E}_{\mathsf{inp}}^{R_2} = \{u_i, \ominus_i, \oplus_{i+1}, \odot_0\}$, wobei wir $\oplus_{i+1}$ ignorieren, wenn $i = m-1$. □

Fakt 6.6. *Wenn $e \in \{\oplus_0, \ldots, \oplus_{m-1}\}$, dann ist e durch d_τ-beschränkte τ-Regionen lösbar.*

Beweis. Die folgende Region $R_1 = (sup_1, sig_1)$ löst $(\oplus_{m-1}, z)$ für alle relevanten $z \in Z(A_\tau)$: $\mathcal{E}_{\mathsf{inp}}^{R_1} = \{\oplus_{m-1}, \ominus_{m-1}, \odot_0\}$.

Sei $i \in \{0, \ldots, m-2\}$ beliebig aber fest. Die folgende Region $R_2 = (sup_2, sig_2)$ löst $(\oplus_i, z)$ für alle $z \in Z(A_\tau) \setminus (\bigcup_{j=i+1}^{m-1}\{\bot_{j,0}, \ldots, \bot_{j,j-1}\})$: $\mathcal{E}_{\mathsf{inp}}^{R_2} = \{\oplus_i, \ominus_i, \odot_0\}$.

Die folgende Region $R_3 = (sup_3, sig_3)$ löst $(\oplus_i, z)$ für die verbleibenden relevanten Zuständen, das heißt, für alle $z \in \bigcup_{j=i+1}^{m-1}\{\bot_{j,0}, \ldots, \bot_{j,j-1}\}$: $\mathcal{E}_{\mathsf{free}}^{R_3} = \{\oplus_i\}$ und $\mathcal{E}_{\mathsf{inp}}^{R_3} = \{\oplus_{i+1}, \ominus_{i+1}, \odot_0\}$. □

Die Aussage des folgenden Fakts ist leicht zu sehen, sodass wir auf einen expliziten Beweis verzichten:

Fakt 6.7 (Ohne Beweis)**.** *Wenn $e \in \{\ominus_0, \ldots, \ominus_{m-1}, \odot_0, \ldots, \odot_{r\kappa-1}\}$, dann ist e durch d_τ-beschränkte τ-Regionen lösbar.*

Das folgende Lemma ist eine einfache Folgerung der präsentierten Fakten und komplettiert den Beweis von Satz 6.2.1:

Lemma 6.3. *Wenn es eine unabhängige Menge mit κ Elementen für $(\mathfrak{U}, M)$ gibt, dann gibt es einen Zeugen τ-beschränkter Regionen für die τ-EZTE und die τ-ZTE von A_τ.*

Beweis. Aus den Aussagen von Fakt 6.1 bis Fakt 6.7 folgt, dass es einen Zeugen $\mathcal{R}$ d_τ-beschränkter Regionen für die τ-EZTE von A_τ gibt. Darüber hinaus ist leicht zu sehen, dass die entsprechenden Regionen ebenso die τ-ZTE von A_τ bezeugen. □

Der Beweis von Satz 6.2.2

Solange nicht ausdrücklich anders vereinbart, sei im Folgenden $\tau = \{\mathsf{nop}, \mathsf{inp}, \mathsf{out}\} \cup \omega$ mit $\omega \subseteq \{\mathsf{used}, \mathsf{free}\}$ beliebig aber fest. Die folgende Reduktion für den Beweis von Satz 6.2.2 ist eine Erweiterung der Reduktion für den Beweis von Satz 6.2.1, die wir gerade diskutiert haben: Für das Paar (A_τ, d_τ) definieren wir zunächst wieder $d_\tau = \kappa \cdot (r+1) + 2$. Des Weiteren hat das aktuelle Transitionssystem A_τ alle Zustände, Ereignisse und Kanten sowie den gleichen Initialzustand $\bot_{m-1,0}$ wie das Transitionssystem aus der Reduktion für den Beweis von Satz 6.2.1. Darüber hinaus hat A_τ für alle $i \in \{0, \ldots, m-1\}$ und alle $j \in \{0, \ldots, r\kappa - 1\}$ das folgende Gadget $F_{i,j}$, auf dem die Ereignisse k_0 und a_i^j stattfinden:

$$F_{i,j} = \vdash_{i,j} \xrightarrow{w_{i,j}} f_{i,j,0} \xrightarrow{a_i^j} f_{i,j,1}, \quad f_{i,j,2} \overset{k_0}{\uparrow} f_{i,j,0}$$

Diese Gadgets sind mit dem Zustand $\top_{r\kappa-1}$ sowie untereinander über die folgenden Kanten verbunden:

$$\top_{r\kappa-1} \xrightarrow{\otimes_{0,0}} \vdash_{0,0} \xrightarrow{\otimes_{0,1}} \cdots \xrightarrow{\otimes_{0,r\kappa-1}} \vdash_{0,r\kappa-1} \xrightarrow{\otimes_{1,0}} \vdash_{1,0} \xrightarrow{\otimes_{1,1}} \cdots \xrightarrow{\otimes_{m-1,r\kappa-1}} \vdash_{m-1,r\kappa-1}$$

Es ist klar, dass (A_τ, d_τ) das Ergebnis einer parametrisierten Reduktion sind. Abbildung 6.4 zeigt die Gadgets $F_{0,0}, \ldots, F_{3,3}$, die sich aus Beispiel 6.2 ergeben, sowie jene Kanten, die sie miteinander verbinden. Wir definieren $U = \{u_0, \ldots, u_{m-1}\}$ und $\oplus = \{\oplus_0, \ldots, \oplus_{m-1}\}$ und $\ominus = \{\ominus_0, \ldots, \ominus_{m-1}\}$ und $\odot = \{\odot_0, \ldots, \odot_{r\kappa-1}\}$ und außerdem $V = \{v_0, \ldots, v_{r\kappa-1}\}$ und $\oplus = \{\oplus_{0,0}, \ldots, \oplus_{m-1,r\kappa-1}\}$ und $W = \{w_{0,0}, \ldots, w_{m-1,r\kappa-1}\}$.

Lemma 6.4. *Wenn es einen Zeugen für die τ-EZTE von A_τ gibt, dann gibt es eine unabhängige Menge mit κ Elementen für $(\mathfrak{U}, M)$.*

Beweis. Weil es einen Zeugen für die τ-EZTE von A_τ gibt, gibt es eine τ-Region, die $\alpha = (k_1, h_0)$ löst. Sei $R = (sup, sig)$ so eine Region. Weil R das EZTP α löst, gilt entweder $sig(k_1) \in \{\mathsf{out}, \mathsf{free}\}$ und $sup(h_0) = 1$ oder $sig(k_1) \in \{\mathsf{inp}, \mathsf{used}\}$ und $sup(h_0) = 0$. Wir beschränken uns im Folgenden auf den ersten Fall, denn der zweite ist zu diesem symmetrisch. Sei also $sig(k_1) \in \{\mathsf{out}, \mathsf{free}\}$ und $sup(h_0) = 1$. Dies impliziert sofort $sup(h_1) = 0$ und $sig(k_0) = \mathsf{inp}$. Für alle $i \in \{0, \ldots, m-1\}$ und alle $j \in \{0, \ldots, r\kappa - 1\}$ erhalten wir aus $sig(k_0) = \mathsf{inp}$ und $sig(k_1) \in \{\mathsf{out}, \mathsf{free}\}$ die folgenden Support- beziehungsweise Signaturwerte: $sup(t_{i,0}) = sup(g_{j,0}) = sup(f_{i,j,0}) = 1$ sowie $g_{j,m} = 0$ und $sig(a_i^j) \neq \mathsf{out}$ (wegen $sup(f_{i,j,0}) = 1$). Offensichtlich gilt, dass zwischen je zwei mit inp beschrifteten Kanten eines Pfades in τ, stets eine mit out beschriftete Kante liegen muss. Weil nun alle a-Ereignisse eine von out verschiedene Signatur haben, folgt ähnlich zu dem Beweis von Lemma 6.2, dass es genau $r\kappa$ paarweise verschiedene $i_0, \ldots, i_{r\kappa-1} \in \{0, \ldots, m-1\}$ Indizes gibt, sodass $sig(a_{i_\ell}^\ell) = \mathsf{inp}$ und folglich $sup(t_{i_\ell,2}) = 0$ für alle $\ell \in \{0, \ldots, r\kappa - 1\}$. Somit ist $S = \{X \in \mathfrak{U} \mid sig(X) = \mathsf{inp}\}$ erneut eine unabhängige Menge mit wenigstens κ Elementen von $(\mathfrak{U}, M)$. □

Für die andere Richtung sei S eine beliebige aber feste unabhängige Menge für $(\mathfrak{U}, M)$. Wir präsentieren wieder eine Reihe von Fakten, welche die τ-Lösbarkeit der einzelnen (Arten von) Ereignissen individuell diskutieren. Die Gesamtheit der diskutierten Regionen wird dann die Existenz eines Zeugen $\mathcal{R}$ d_τ-beschränkter Regionen für die τ-EZTE von A_τ belegen. Darüber hinaus wird $\mathcal{R}$ ebenso die τ-ZTE bezeugen. Abgesehen von der Definition von $sup(\perp_{m-1,0})$ verwenden wir für die Darstellung und Definition von τ-Regionen alle Vereinfachungen, die wir bereits gerade für den Beweis von Satz 6.2.1 verwendet haben.

Fakt 6.8. *Das Ereignis k_0 ist durch d_τ-beschränkte τ-Regionen lösbar.*

Beweis. Sei $i \in \{0, \ldots, m-1\}$ beliebig aber fest. Die Region R_1 von Fakt 6.1 ist auch eine τ-Region. Deshalb können wir (k_0, z) für alle $z \in \{h_1, h_2, t_{i,1}, \ldots, t_{i,r\kappa+2}\}$ als gelöst ansehen.

Die folgende Region $R_2 = (sup_2, sig_2)$ löst (k_0, z) für alle $z \in \{\perp_{i,0}, \ldots, \perp_{i,i+1}\}$: wenn $i = m-1$, dann $sup_2(\perp_{m-1,0}) = 0$, andernfalls $sup_2(\perp_{m-1,0}) = 1$; für alle $e \in E(A_\tau)$,

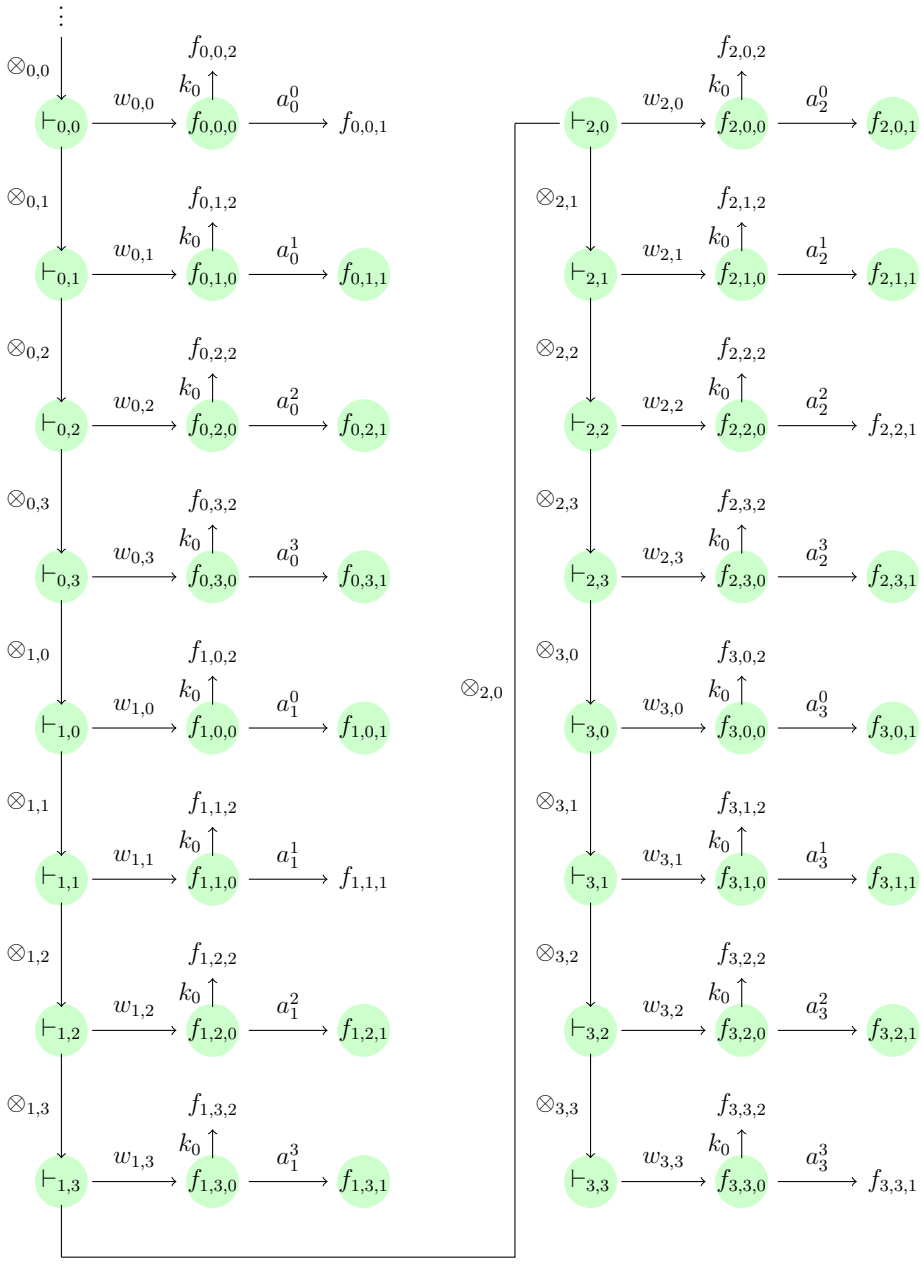


Abbildung 6.4: Die Gadgets $F_{0,0}, \ldots, F_{3,3}$ mit den Verbindungskanten die das Transitionssystem aus Abbildung 6.3 erweitern, sodass wir A_τ für den Beweis von Satz 6.2.2 erhalten, das aus Beispiel 6.2 hervorgeht. Der grüne Bereich sowie der rote *und* grüne Bereich der Abbildung 6.3 skizzieren die Region R_1 aus dem Beweis von Fakt 6.9, die α löst.

wenn $e \in \{k_0, \ominus_{i+1}\}$, dann $sig_2(e) = \mathsf{inp}$; wenn $e \in \{u_i, \ominus_i\}$, dann $sig_2(e) = \mathsf{out}$; wenn $i = m-1$ und $e = \odot_0$, dann $sig_2(e) = \mathsf{out}$; andernfalls $sig_2(e) = \mathsf{nop}$. Natürlich ignorieren wir $\ominus_{i+1}$, wenn $i = m-1$.

Sei $j \in \{0, \ldots, r\kappa - 1\}$ beliebig aber fest. Seien $i_0, \ldots, i_{r-2} \in \{1, \ldots, m-1\}$ die $r-1$ paarweise verschiedenen Indizes (ungleich 0), sodass $X_{0_0} \in M_{i_\ell}$ für alle $\ell \in \{0, \ldots, r-2\}$. Außerdem seien $j_0, \ldots, j_{r-2} \in \{0, \ldots, r\kappa - 1\} \setminus \{j\}$ paarweise verschieden. Die folgende Region $R_3 = (sup_3, sig_3)$ löst (k_0, z) für alle $z \in \{\top_j, g_{j,1}, \ldots, g_{j,m+2}\}$: $sup_3(\bot_{m-1,0}) = 1$; für alle $e \in E(A_\tau)$, wenn $e \in \{k_0, \odot_j, X_{0_0}, a_0^j, a_{i_0}^{j_0}, \ldots, a_{i_{r-2}}^{j_{r-2}}\}$, dann $sig_3(e) = \mathsf{inp}$; wenn $e \in \{v_j, \odot_{j+1}\}$, dann $sig_3(e) = \mathsf{out}$; wenn $j = r\kappa - 1$ und $e = \otimes_{0,0}$, dann $sig_3(e) = \mathsf{out}$; andernfalls $sig_3(e) = \mathsf{nop}$.

Wie man leicht sieht, kann (k_0, z) für alle $z \in \{\vdash_{i,j}, f_{i,j,1}, f_{i,j,2}\}$ und alle $i \in \{0, \ldots, m-1\}$ sowie alle $j \in \{0, \ldots, r\kappa - 1\}$ durch d_τ-beschränkte τ-Regionen gelöst werden, die zu R_3 ähnlich sind. Dies beweist den Fakt. □

Fakt 6.9. *Das Ereignis k_1 ist durch d_τ-beschränkte τ-Regionen lösbar.*

Beweis. Wir benötigen die unabhängige Menge S mit κ Elementen für $(\mathfrak{U}, M)$. Seien $i_0, \ldots, i_{r\kappa-1} \in \{0, \ldots, m-1\}$ die $r\kappa$ paarweise verschiedenen Indizes, sodass sets $S \cap M_{i_\ell} \neq \emptyset$ für alle $\ell \in \{0, \ldots, r\kappa - 1\}$. Die folgende Region $R_1 = (sup_1, sig_1)$ löst α: $sup_1(\bot_{m-1,0}) = 1$ und $\mathcal{E}^{R_1}_{\mathsf{out}} = \{k_1\}$ und $\mathcal{E}^{R_1}_{\mathsf{inp}} = \{k_0\} \cup S \cup \{a^0_{i_0}, \ldots, a^{r\kappa-1}_{i_{r\kappa-1}}\}$. Man beachte, dass $|\{e \in E(A_\tau) \mid sig_1(e) \neq \mathsf{nop}\} = d_\tau$.

Es ist leicht zu sehen, dass (k_1, z) für alle $z \in \{h_2\} \cup (\bigcup_{i=0}^{m-1}(Z(T_i)) \cup (\bigcup_{i=0}^{m-1}(Z(F_{i,0}) \cup \cdots \cup Z(F_{i,r\kappa-1})))$ passend τ-lösbar ist. (Insbesondere weil k_1 für alle $i \in \{0, \ldots, m-1\}$ und alle $j \in \{0, \ldots, r\kappa - 1\}$ nicht in T_i und $F_{i,j}$ stattfindet).

Sei $j \in \{0, \ldots, r\kappa - 1\}$ beliebig aber fest. Seien weiterhin $i_0, \ldots, i_{r-2} \in \{0, \ldots, m-2\}$ die $r-1$ paarweise verschiedenen Indizes (ungleich $m-1$), sodass $X_{m-1_0} \in M_{i_\ell}$ für alle $\ell \in \{0, \ldots, r-2\}$. Außerdem seien $j_0, \ldots, j_{r-2} \in \{0, \ldots, r\kappa - 1\} \setminus \{j\}$ paarweise verschieden. Die folgende Region $R_2 = (sup_2, sig_2)$ löst (k_1, z) für $z = \top_j$ und für alle $z \in \{g_{j,0}, \ldots, g_{j,m-1}, g_{j,m+2}, g_{j,m+3}\}$: $sup(\bot_{m-1,0}) = 0$ und $\mathcal{E}^{R_2}_{\mathsf{inp}} = \{k_1\}$ und $\mathcal{E}^{R_2}_{\mathsf{out}} = \{X_{m-1_0}, \ominus_0\} \cup \{a^j_{m-1}, a^{j_0}_{i_0}, \ldots, a^{j_{r-2}}_{i_{r-2}}\} \cup \{v_0, \ldots, v_{r\kappa-1}\} \setminus \{v_j, v_{j_0}, \ldots, v_{j_{r-2}}\}$. Man beachte, dass $|\{e \in E(A_\tau) \mid sig_2(e) \neq \mathsf{nop}\}| = r\kappa + 3 \leq d_\tau$. □

Fakt 6.10. *Seien $i \in \{0, \ldots, m-1\}$ und $j \in \{0, \ldots, r\kappa - 1\}$ beliebig aber fest. Das Ereignis a_i^j ist durch d_τ-beschränkte τ-Regionen lösbar.*

Beweis. Im Folgenden diskutieren wir zunächst die EZTP, die durch Zustände des Gadgets G_j begründet werden. Man beachte, dass die Region R_3 aus dem Beweis von Fakt 6.8 das EZTP (a_0^j, z) für alle relevanten Zustände von G_j löst. Deshalb nehmen wir im Folgenden $i > 0$ an.

Die folgende Region $R_1 = (sup_1, sig_1)$ löst (a_i^j, z) für alle $z \in \{\top_j, g_{j,0}, \ldots, g_{j,m+2}\} \setminus \{g_{j,i}\}$, wobei wir die drei Fälle (1) $0 < j < r\kappa - 1$ (das heißt, $j-1$ und $j+1$ existieren) und (2) $j = 0$ und (3) $j = r\kappa - 1$ unterscheiden: $sup_1(\bot_{m-1,0}) = 0$ und

(1): $\mathcal{E}^{R_1}_{\mathsf{inp}} = \{e_i^j\} \cup \{a_{i-1}^{j+1}\}$ und $\mathcal{E}^{R_1}_{\mathsf{out}} = \{a_{i-1}^j, w_{i,j}\} \cup \{a_i^{j-1}, v_{j+1}, w_{i-1,j+1}\}$;

(2): $\mathcal{E}^{R_1}_{\mathsf{inp}} = \{e_i^j\} \cup \{a_{i-1}^{r\kappa-2}\}$ und $\mathcal{E}^{R_1}_{\mathsf{out}} = \{a_{i-1}^j, w_{i,j}\} \cup \{a_i^{r\kappa-1}, u_i, v_{r\kappa-2}, w_{i-1,r\kappa-2}\}$;

(3): $\mathcal{E}^{R_1}_{\mathsf{inp}} = \{e_i^j\} \cup \{a_{i-1}^{j-2}\}$ und $\mathcal{E}^{R_1}_{\mathsf{out}} = \{a_{i-1}^j, w_{i,j}\} \cup \{a_i^{j-1}, w_{i-1,j-2}, v_{j-2}, u_{i-1}\}$.

Die folgenden Regionen vervollständigen den Beweis der τ-Lösbarkeit von (a_i^j, z) für alle verbleibenden Zustände $z \in \bigcup_{\ell=0}^{r\kappa-1} Z(G_\ell)$ (das heißt $z \notin Z(G_j)$). Dabei müssen wir erneut die drei Fälle (1) $j = 0$ und (2) $1 \leq j < r\kappa - 1$ und (3) $j = r\kappa - 1$ unterscheiden:

(1): Die folgende Region $R_2 = (sup_2, sig_2)$ löst (a_i^0, z) für alle Zustände $z \in \bigcup_{\ell=2}^{r\kappa-1} Z(G_\ell)$: $sup_2(\bot_{m-1,0}) = 0$ und $\mathcal{E}_{\mathsf{inp}}^{R_2} = \{a_i^0\}$ und $\mathcal{E}_{\mathsf{out}}^{R_2} = \{u_i, v_0, w_{i,0}, a_i^1\}$.

Um (a_i^0, z) für alle $z \in Z(G_1)$ zu lösen, nutzen wir eine Region R_3, welche ähnlich wie R_2 definiert ist, aber a_i^2 auf out und a_i^1 auf nop abbildet.

(2): Die folgende Region $R_4 = (sup_4, sig_4)$ löst (a_i^j, z) für alle $z \in \bigcup_{\ell=0}^{r\kappa-1} Z(G_\ell) \setminus (Z(G_j) \cup Z(G_{j+1}))$: $sup_4(\bot_{m-1,0}) = 0$ und $\mathcal{E}_{\mathsf{inp}}^{R_4} = \{a_i^j\}$ und $\mathcal{E}_{\mathsf{out}}^{R_4} = \{u_i, v_j, w_{i,j}, a_i^{j+1}\}$.

Die folgende Region $R_5 = (sup_5, sig_5)$ löst (a_i^j, z) für alle $z \in Z(G_{j+1})$: $sup_5(\bot_{m-1,0}) = 0$ und $\mathcal{E}_{\mathsf{inp}}^{R_5} = \{a_i^j\}$ und $\mathcal{E}_{\mathsf{out}}^{R_5} = \{v_j, w_{i,j}, a_i^{j-1}\}$.

(3): Die folgende Region $R_6 = (sup_6, sig_6)$ löst (a_i^j, z) für alle $z \in \bigcup_{\ell=0}^{r\kappa-1} Z(G_\ell) \setminus (Z(G_j) \cup Z(G_{j-1}))$: $sup_6(\bot_{m-1,0}) = 0$ und $\mathcal{E}_{\mathsf{inp}}^{R_6} = \{a_i^j\}$ und $\mathcal{E}_{\mathsf{out}}^{R_6} = \{v_j, w_{i,j}, a_i^{j-1}\}$.

Wir erhalten eine Region R_7, die (a_i^j, z) für alle $z \in Z(G_{j-1})$ löst und ähnlich wie R_6 definiert ist, aber a_i^{j-2} auf out und a_i^{j-1} auf nop abbildet.

Diskutieren wir nun die EZTP (a_i^j, z), sodass $z \in \bigcup_{i=0}^{m-1} Z(T_i)$. Wir unterscheiden den einfachen Fall (1) $j > 1$ von dem komplizierteren Fall (2) $j = 0$:

(1): Die folgende Region $R_8 = (sup_8, sig_8)$ löst (a_i^j, z) für alle Zustände $z \in \bigcup_{i=0}^{m-1} Z(T_i)$: $sup_8(\bot_{m-1,0}) = 0$ und $\mathcal{E}_{\mathsf{inp}}^{R_8} = \{a_i^j\}$ und $\mathcal{E}_{\mathsf{out}}^{R_8} = \{a_i^{j-1}, v_j, w_{i,j}\}$.

(2): Die folgende Region $R_9 = (sup_9, sig_9)$ löst (a_i^0, z) für alle $z \in (\bigcup_{i=0}^{m-1} Z(T_i)) \setminus \{t_{i,1}, t_{i,2}, t_{i,3}\}$: $sup_9(\bot_{m-1,0}) = 0$ und $\mathcal{E}_{\mathsf{inp}}^{R_9} = \{a_i^0\}$ und $\mathcal{E}_{\mathsf{out}}^{R_9} = \{u_i, a_i^{r\kappa-1}, w_{i,0}\}$.

Seien $i_0, \ldots, i_{r-2} \in \{0, \ldots, m-1\} \setminus \{i\}$ die $r-1$ paarweise verschiedenen Indizes (ungleich i), sodass $X_{i_0} \in M_{i_\ell}$ für alle $\ell \in \{0, \ldots, r-2\}$. Seien außerdem $j_0, \ldots, j_{r-2} \in \{1, \ldots, r\kappa-1\}$ paarweise verschieden. Die folgende Region $R_{10} = (sup_{10}, sig_{10})$ löst (a_i^0, z) für alle $z \in \{t_{i,1}, t_{i,2}, t_{i,3}\}$: $sup_{10}(\bot_{m-1,0}) = 1$ und $\mathcal{E}_{\mathsf{inp}}^{R_{10}} = \{k_0, X_{i_0}, a_i^z, a_{i_0}^{j_0}, \ldots, a_{i_{r-2}}^{j_{r-2}}\}$.

Wie man leicht sieht, gibt es d_τ-beschränkte τ-Regionen, die den hier präsentierten ähnlich sind und insgesamt (a_i^j, z) für alle $z \in \bigcup_{i=0}^{m-1}(\bigcup_{j=0}^{r\kappa-1} S(F_{i,j})) \setminus \{f_{i,j,0}\}$ lösen. □

Fakt 6.11. *Wenn $X \in \mathfrak{U}$, dann ist X durch d_τ-beschränkte τ-Regionen lösbar.*

Beweis. Seien $i_0, \ldots, i_{r-1} \in \{0, \ldots, m-1\}$ die paarweise verschiedenen Indizes, sodass $X \in M_{i_\ell}$ für alle $\ell \in \{0, \ldots, r-1\}$.

Die folgende Region $R_1 = (sup_1, sig_1)$ löst das EZTP (X, z) für alle Zustände $z \in (\bigcup_{i=0}^{m-1} Z(T_i)) \setminus (\bigcup_{\ell=0}^{r-1}\{t_{i_\ell,0}, t_{i_\ell,1}, t_{i_\ell,2}\})$: $sup_1(\bot_{m-1,0}) = 0$ und $\mathcal{E}_{\mathsf{inp}}^{R_2} = \{X, a_{i_0}^0, \ldots, a_{i_{r-1}}^{r-1}\}$ und $\mathcal{E}_{\mathsf{out}}^{R_2} = \{v_0, \ldots, v_{r-1}\} \cup \{w_{i_0,0}, \ldots, w_{i_{r-1},r-1}\}$. Offenbar gilt $|\{e \in E(A_\tau) \mid sig_1(e) \neq \mathsf{nop}\}| = 3r + 1 \leq d_\tau$.

Man beachte, dass diese Region R_1 auch (X, z) für alle $z \in \bigcup_{j=r}^{r\kappa-1} Z(G_j)$ löst. Wie man leicht sieht, kann R_1 darüber hinaus so zu einer Region R_1' modifiziert werden, dass $a_{i_0}^0, \ldots, a_{i_{r-1}}^{r-1}$ auf nop und $a_{i_0}^r, \ldots, a_{i_{r-1}}^{2r-1}$ auf inp abgebildet werden (und die entsprechenden Modifizierungen für die v- und w-Ereignisse vorgenommen werden). Dies zeigt, dass (X, z) für alle $z \in \bigcup_{j=0}^{r\kappa-1} Z(G_j)$ gelöst werden kann. Ähnlich argumentiert man, dass (X, z) für alle $z \in \bigcup_{i=0}^{m-1}(\bigcup_{j=0}^{r\kappa-1} Z(F_{i,j}))$ gelöst werden kann.

Es bleibt zu zeigen, dass (X, z) für alle relevanten $z \in \bigcup_{\ell=0}^{r-1}\{t_{i_\ell,0}, t_{i_\ell,1}, t_{i_\ell,2}, t_{i_\ell,3}\}$ gelöst werden kann. Sei $\ell \in \{0, \ldots, r-1\}$ beliebig aber fest. Die folgende Region $R_2 = (sup_2, sig_2)$ löst (X, z) für alle $z \in \{t_{i_\ell,2}, t_{i_\ell,3}\}$ und wenn $X = X_{i_{\ell 0}}$, dann auch für $z = t_{i_\ell,1}$: $sup_2(\bot_{m-1,0}) = 1$ und $\mathcal{E}^{R_2}_{\mathsf{inp}} = \{X, k_0, a^0_{i_0}, \ldots, a^{r-1}_{i_{r-1}}\}$.

Zuletzt ist der Fall $X = X_{i_{\ell 1}}$ zu diskutieren und die Lösbarkeit von $(X, t_{i_\ell,0})$ zu zeigen: Seien $j_0, \ldots, j_{r-2} \in \{0, \ldots, m-1\} \setminus \{i_\ell\}$ die paarweise verschiedenen Indizes (ungleich i_ℓ), sodass $X_{i_{\ell 0}}$ (das heißt, das „erste“ Element von M_{i_ℓ}) für alle $k \in \{0, \ldots, r-2\}$ in M_{j_k} ist. Die folgende Region $R_3 = (sup_3, sig_3)$ löst $(X, t_{i_\ell,0})$: $sup_3(\bot_{m-1,0}) = 0$ und $\mathcal{E}^{R_3}_{\mathsf{inp}} = \{X_{i_{\ell 1}}, a^0_{i_0}, \ldots, a^{\ell-1}_{i_{\ell-1}}, a^{\ell+1}_{i_{\ell+1}}, \ldots, a^{r-1}_{i_{r-1}}\}$ und außerdem $\mathcal{E}^{R_3}_{\mathsf{out}} = \{X_{i_{\ell 0}}, a^r_{j_0}, \ldots, a^{2r-2}_{j_{r-2}}\} \cup \{u_{i_0}, \ldots, u_{i_{\ell-1}}\} \cup \{u_{i_{\ell+1}}, \ldots, u_{i_{r-1}}\} \cup \{v_0, \ldots, v_{\ell-1}, v_{\ell+1}, \ldots, v_{r-1}\} \cup \{w_{i_0,0}, \ldots, w_{i_{\ell-1},\ell-1}\} \cup \{w_{i_{\ell+1},\ell+1}, \ldots, w_{i_{r-1},r-1}\}$. Man beachte, dass für R_3 per Definition $|\{e \in E(A_\tau) \mid sig(e)\}| = 5r-3$ gilt. Wir dürfen jedoch $\kappa \geq 5$ und daher $5r-3 \leq 5\cdot(r+1) \leq \kappa\cdot(r+1)+2$ annehmen. Dies ist der Fall, weil RIS für jedes im Voraus fest gewählte κ in Polynomialzeit lösbar ist. Dies beweist den Fakt. □

Wie man leicht sieht, sind auch die verbleibenden Ereignisse von A_τ durch d_τ-beschränkte τ-Regionen lösbar. Letzteres ist die Aussage des folgenden Fakts, den wir deshalb ohne expliziten Beweis präsentieren:

Fakt 6.12 (Ohne Beweis)**.** *Wenn $e \in U \cup \oplus \cup \ominus \cup \odot \cup V \cup \oplus \cup W$, dann ist e durch d_τ-beschränkte τ-Regionen lösbar.*

Darüber hinaus ist leicht zu sehen, dass ein ZTP (z, z') von A_τ durch d_τ-beschränkte τ-Regionen lösbar ist, wenn z und z' nicht Zustände desselben Gadgets sind. Wenn dagegen z und z' zu dem gleichen Gadget von A_τ gehören, dann ist das entsprechende ZTP (z, z') bereits durch eine der präsentierten Regionen gelöst, die zum Zeugen $\mathcal{R}$ für die τ-EZTE von A_τ gehören. Insgesamt haben wir deshalb das folgende Lemma bewiesen, welches den Beweis von Satz 6.2 komplettiert:

Lemma 6.5. *Wenn es eine unabhängige Menge mit κ Elementen für $(\mathfrak{U}, M)$ gibt, dann gibt es einen Zeugen d_τ-beschränkter Regionen für die τ-EZTE und die τ-ZTE von A_τ.*

6.2 W[1]-Schwere durch Reduktion von Odd Set

Der folgende Satz adressiert die parametrisierte Komplexität von Bτ-SMBP und Bτ-RMBP für die booleschen Netztypen C4 und C5 aus Abbildung 6.2:

Satz 6.3. Bτ-SMBP *und* Bτ-RMBP *sind* W[1]*-schwer, wenn*

1. $\tau = \{\mathsf{nop}, \mathsf{swap}\} \cup \omega$ *mit* $\omega \subseteq \{\mathsf{inp}, \mathsf{out}, \mathsf{used}, \mathsf{free}\}$ *und* $\omega \cap \{\mathsf{inp}, \mathsf{out}\} \neq \emptyset$ *oder*
2. $\tau = \{\mathsf{nop}, \mathsf{swap}\} \cup \omega$ *mit* $\omega \subseteq \{\mathsf{used}, \mathsf{free}\}$ *und* $\omega \neq \emptyset$.

Der Beweis von Satz 6.3 erfolgt durch parametrisierte Reduktionen des folgenden Problems Odd Set, das als W[1]-vollständig bekannt ist [CFK+15]:

Odd Set (OS)	
Eingabe:	Eine endliche Menge $\mathfrak{U}$ und eine Menge $M = \{M_0, \ldots, M_{m-1}\}$ von Teilmengen von $\mathfrak{U}$, eine natürliche Zahl κ.
Parameter:	κ.
Frage:	Gibt es eine ungerade Menge S für $(\mathfrak{U}, M)$, das heißt, $S \subseteq \mathfrak{U}$ und $\lvert S \cap M_i \rvert$ ist ungerade für alle $i \in \{0, \ldots, m-1\}$, die $\lvert S \rvert \leq \kappa$ erfüllt?

Beispiel 6.3. Es ist $(\mathfrak{U}, M, 2)$ mit $M = \{M_0, \ldots, M_3\}$, wobei $M_0 = \{X_0, X_1\}$, $M_1 = \{X_0, X_3\}$, $M_2 = \{X_1, X_2\}$ sowie $M_3 = \{X_2, X_3\}$, eine Ja-Instanz von OS, denn $S = \{X_0, X_2\}$ ist eine passende ungerade Menge.

Für den Beweis von Satz 6.3.1 beziehungsweise von Satz 6.3.2 benötigen wir jeweils unterschiedliche Reduktionen. Dies liegt zum Beispiel daran, dass für die Netztypen aus Satz 6.3.2 eine EZTP (e, z) nicht gelöst werden kann, wenn z ein Ziel von e ist, das heißt, $\xrightarrow{e} z$. Folglich muss jedes Ziel von e auch ein Ursprung von e sein. Dies führt aber zum Beispiel für den Netztyp $\tau = \{\mathsf{nop}, \mathsf{inp}, \mathsf{swap}\}$ aus Satz 6.3.1 zu unlösbaren Trennungsproblemen. Folglich müssen die Reduktion unterschiedlich sein. Dennoch folgen beide Reduktionen wieder einer gemeinsamen Idee, welche nun kurz skizziert wird.

Die Strategie der Reduktionen von Odd Set

Solange nicht ausdrücklich anders vereinbart, sei $(\mathfrak{U}, M, \kappa)$ im Folgenden eine beliebige aber feste Eingabe von OS mit $\mathfrak{U} = \{X_0, \ldots, X_{n-1}\}$ und $M = \{M_0, \ldots, M_{m-1}\}$, wobei $M_i = \{X_{i_0}, \ldots, X_{i_{m_i-1}}\}$ (und somit $\lvert M_i \rvert = m_i$) für alle $i \in \{0, \ldots, m-1\}$. Wir reduzieren $(\mathfrak{U}, M, \kappa)$ auf eine Eingabe (A_τ, d_τ) für Bτ-SMBP beziehungsweise für Bτ-RMBP sodass es genau dann eine gesuchte ungerade Menge für $(\mathfrak{U}, M)$ gibt, wenn es für A_τ einen Zeugen d_τ-beschränkter Regionen für die τ-EZTE und die τ-ZTE von A_τ gibt. Die Konstruktion sichert $d_\tau = f(\kappa)$ für eine lineare Funktion f zu. Darüber hinaus hat das Transitionssystem A_τ für jedes $i \in \{0, \ldots, m-1\}$ einen Pfad P_i wie in Abbildung 6.5, der die Menge $M_i = \{X_{i_0}, \ldots, X_{m_i-1}\}$ repräsentiert.

$$P_i = \; z_{i,0} \xrightarrow{X_{i_0}} \cdots \xrightarrow{X_{i_{\ell-1}}} z_{i,i_\ell} \xrightarrow{X_{i_\ell}} z_{i,i_\ell+1} \xrightarrow{X_{i_{\ell+1}}} \cdots \xrightarrow{X_{i_{m_i-1}}} z_{i,i_{m_i}}$$

Abbildung 6.5: Skizze des Pfades P_i, auf dem die Elemente von M_i als Ereignisse stattfinden.

Sei $i \in \{0, \ldots, m-1\}$ beliebig aber fest. Das Transitionssystem A_τ hat des Weiteren ein EZTP α, sodass die folgende Eigenschaft erfüllt ist: wenn $R = (sup, sig)$ eine

d_τ-beschränkte τ-Region ist, die α löst, dann gilt $sup(z_{i,0}) \neq sup(z_{i,m_i})$. Folglich ist das Bild P_i^R von P_i unter R ein Pfad von 0 nach 1 oder ein Pfad von 1 nach 0 in τ. Einerseits muss es deshalb ein Element $X \in \{X_{i_0}, \ldots, X_{m_i-1}\}$ geben, dessen Signatur $sig(X)$ diesen Zustandswechsel in τ verursacht. Für die hier untersuchten Netztypen gilt, dass $sig(X)$ genau dann einen solchen Zustandswechsel vollzieht, wenn $sig(X) \in \{\mathsf{inp}, \mathsf{out}, \mathsf{swap}\}$. Insbesondere folgt aus $sig(X) \in \{\mathsf{inp}, \mathsf{out}, \mathsf{swap}\}$, dass $sup(z) \neq sup(z')$ für jede Kante $z \xrightarrow{X} z'$ von A_τ. Andererseits muss die Anzahl der Zustandswechsel zwischen 0 und 1 auf diesem Pfad P_i^R ungerade sein. Mit anderen Worten, die Zahl $|\{X \in M_i \mid sig(X) \subseteq \{\mathsf{inp}, \mathsf{out}, \mathsf{swap}\}\}|$ ist ungerade. Abbildung 6.6 skizziert die Situation für $sup(z_{i,0}) = 0$ und $sup(z_{i,m_i}) = 1$ und $sig(X_{i_\ell}) = sig(X_{i_\ell+1}) = sig(X_{i_{m_i}}) = \mathsf{swap}$ und $sig(e) = \mathsf{nop}$ für alle $e \in M_i \setminus \{X_{i_\ell}, X_{i_\ell+1}, X_{i_{m_i}}\}$.

$$P_i^R = \underset{0}{sup(z_{i,0})} \xrightarrow[\mathsf{nop}]{sig(X_{i_0})} \cdots \xrightarrow[\mathsf{nop}]{sig(X_{i_{\ell-1}})} \underset{0}{sup(z_{i,i_\ell})} \xrightarrow[\mathsf{swap}]{sig(X_{i_\ell})} \underset{1}{sup(z_{i,i_{\ell+1}})} \xrightarrow[\mathsf{swap}]{sig(X_{i_{\ell+1}})} \cdots \xrightarrow[\mathsf{swap}]{sig(X_{i_{m_i-1}})} \underset{1}{sup(z_{i,i_{m_i}})}$$

Abbildung 6.6: Skizze des Pfades P_i^R des Bildes von P_i unter einer Region R, die α löst.

Weil i beliebig war, gilt dies gleichzeitig für alle Pfade $P_0, \ldots, P_{m-1}$, welche die Mengen $M_0, \ldots, M_{m-1}$ repräsentieren. Daraus folgt, dass $S = \{X \in \mathfrak{U} \mid sig(X) \subseteq \{\mathsf{inp}, \mathsf{out}, \mathsf{swap}\}\}$ eine ungerade Menge für $(\mathfrak{U}, M)$ ist. Darüber hinaus ist R d_τ-beschränkt und die Konstruktion sichert über den Parameter $|S| \leq \kappa$ zu. Deshalb definiert S eine gesuchte ungerade Menge für $(\mathfrak{U}, M)$.

Für die andere Richtung sichert die Konstruktion von (A_τ, d_τ) zu, dass es einen Zeugen d_τ-beschränkter Regionen für die τ-EZTE und die τ-ZTE von A_τ gibt, wenn es eine ungerade Menge mit höchstens κ Elementen für $(\mathfrak{U}, M)$ gibt. Die folgenden beiden Unterabschnitte präsentieren die jeweiligen Reduktionen und beweisen ihren Funktionalität.

Der Beweis von Satz 6.3.1

Solange nicht ausdrücklich anders vereinbart, sei τ ein beliebiger aber fester boolescher Netztyp gemäß Satz 6.3.1. Wir definieren zunächst für den Parameter $d_\tau = \kappa + 2$. Das Transitionssystem A_τ hat das folgende Gadget H, welches das EZTP $\alpha = (k, h_2)$ liefert:

$$H = \quad \perp_m \xrightarrow{u_m} h_0 \xrightarrow{k} h_1 \xrightarrow{w} h_2 \xrightarrow{o} h_3 \xrightarrow{k} h_4$$

Darüber hinaus hat A_τ für alle $i \in \{0, \ldots, m-1\}$ das folgende Gadget T_i, welches die Menge $M_i = \{X_{i_0}, \ldots, X_{i_{m_i-1}}\}$ repräsentiert, indem die Elemente von M_i als Ereignisse verwendet werden:

$$T_i = t_{i,0} \xrightarrow{k} t_{i,1} \xrightarrow{X_{i_0}} t_{i,2} \xrightarrow{X_{i_1}} t_{i,3} \quad \cdots \quad t_{i,m_i} \xrightarrow{X_{i_{m_i-1}}} t_{i,m_i+1} \xrightarrow{w} t_{i,m_i+2} \xrightarrow{k} t_{i,m_i+3}$$

Wir erhalten das angekündigte Transitionssystem A_τ schließlich, indem die einzelnen Gadgets geeignet miteinander verbunden werden. Da wir zusichern wollen, dass alle EZTP von A_τ durch d_τ-beschränkte τ-Regionen lösbar sind, ist dies –wie wir sehen werden– für die vorliegenden Netztypen mit erheblichem Aufwand verbunden. Wir wollen jedoch die Funktionalität der Gadgets an dieser Stelle nicht aus dem Blick zu verlieren. Deshalb zeigen wir zunächst, dass sie bereits ausreichen, um aus der Lösbarkeit von α durch eine A_τ-beschränkte τ-Region, auf eine ungerade Menge mit höchstens κ Elementen für $(\mathfrak{U}, M)$ zu schließen. Dies ist im Wesentlichen der Beitrag des folgenden Lemmas:

Lemma 6.6. *Sei A ein Transitionssystem, welches die Gadgets H und $T_0, \ldots, T_{m-1}$ hat. Wenn es einen Zeugen A_τ-beschränkter Regionen für die τ-EZTE von A gibt, dann gibt es eine ungerade Menge mit höchstens κ Elementen für $(\mathfrak{U}, M)$.*

Beweis. Weil es einen Zeugen d_τ-beschränkter Regionen für die τ-EZTE von A gibt, gibt es eine d_τ-beschränkte τ-Region, die $\alpha = (k, h_2)$ löst. Sei $R = (sup, sig)$ ein solche Region, das heißt, es gilt entweder $sig(k) \in \{\mathsf{inp}, \mathsf{used}\}$ und $sup(h_2) = 0$ oder $sig(k) \in \{\mathsf{out}, \mathsf{free}\}$ und $sup(h_2) = 1$. Wir werden im Folgenden nur den Fall $sig(k) \in \{\mathsf{inp}, \mathsf{used}\}$ und $sup(h_2) = 0$ untersuchen, denn der andere Fall ist zu diesem symmetrisch.

Untersuchen wir zunächst den Fall $sig(k) = \mathsf{inp}$ und $sup(h_2) = 0$. Sei $i \in \{0, \ldots, m-1\}$ beliebig aber fest. Wegen $sig(k) = \mathsf{inp}$ erhalten wir unverzüglich $sup(h_1) = sup(t_{i,1}) = 0$ und $sup(t_{i,m_i+2}) = 1$. Aus $sup(h_1) = sup(h_2) = 0$ schließen wir $sup(w) \in \{\mathsf{nop}, \mathsf{free}\}$ und $sig(o) \in \{\mathsf{out}, \mathsf{swap}\}$. Wegen $sup(t_{i,m_i+2}) = 1$ impliziert dies $sup(w) = \mathsf{nop}$ und damit insbesondere $sup(t_{i,m_i+1}) = 1$. Das Bild P_i^R des Pfades $P_i = t_{i,0} \xrightarrow{X_{i_0}} \ldots \xrightarrow{X_{i_{m_i-1}}} t_{i,m_i+1}$ unter R ist damit ein Pfad von 0 nach 1 in τ. Dies impliziert, dass die Anzahl der Zustandswechsel zwischen 0 und 1 entlang des Pfades P_i^R ungerade ist. Darüber hinaus verursacht der Signaturwert $sig(X_{i_j})$ von X_{i_j} für jedes $j \in \{0, \ldots, m_i - 1\}$ genau dann einen solchen Zustandswechsel, wenn $sig(X_{i_j}) \in \{\mathsf{inp}, \mathsf{out}, \mathsf{swap}\}$. Daraus folgt sofort, dass die Anzahl $|\{j \in \{0, \ldots, m_i - 1\} \mid sig(X_{i_j}) \in \{\mathsf{inp}, \mathsf{out}, \mathsf{swap}\}\}|$ ungerade ist. Weil i beliebig war, gilt für die Menge $S = \{X \in \mathfrak{U} \mid sig(X) \in \{\mathsf{inp}, \mathsf{out}, \mathsf{swap}\}\}$, dass $|S \cap M_i|$ für alle $i \in \{0, \ldots, m-1\}$ ungerade ist. Darüber hinaus gilt wegen $sig(k) \neq \mathsf{nop}$ und $sig(o) \neq \mathsf{nop}$ sowie $|\{e \in E \mid sig(e) \neq \mathsf{nop}\}| \leq d_\tau = \kappa + 2$, dass $|S| \leq \kappa$. Folglich ist impliziert S eine gesuchte ungerade Menge für $(\mathfrak{U}, M)$.

Argumentieren wir nun für den Fall $sig(k) = \mathsf{used}$ und $sup(h_2) = 0$. Sei $i \in \{0, \ldots, m-1\}$ beliebig aber fest. Mit $sig(k) = \mathsf{used}$ erhalten wir $sup(h_1) = sup(h_3) = sup(t_{i,1}) = sup(t_{i,m_i+2}) = 1$. Aus $sup(h_1) = 1$ und $sup(h_2) = 0$ schließen wir $sup(w) \in \{\mathsf{inp}, \mathsf{swap}\}$. Wegen $sup(t_{i,m_i+2}) = 1$ impliziert dies $sup(w) = \mathsf{swap}$, sodass $sup(t_{i,m_i+1}) = 0$. Somit ist das Bild P_i^R des Pfades $P_i = t_{i,0} \xrightarrow{X_{i_0}} \ldots \xrightarrow{X_{i_{m_i-1}}} t_{i,m_i+1}$ unter R ein Pfad von 1 nach 0 in τ. Analog zu der vorherigen Argumentation halten wir, dass die Menge $S = \{X \in \mathfrak{U} \mid$

$sig(X) \in \{\mathsf{inp}, \mathsf{out}, \mathsf{swap}\}\}$ eine ungerade Menge für $(\mathfrak{U}, M)$ mit höchstens κ Elementen definiert. □

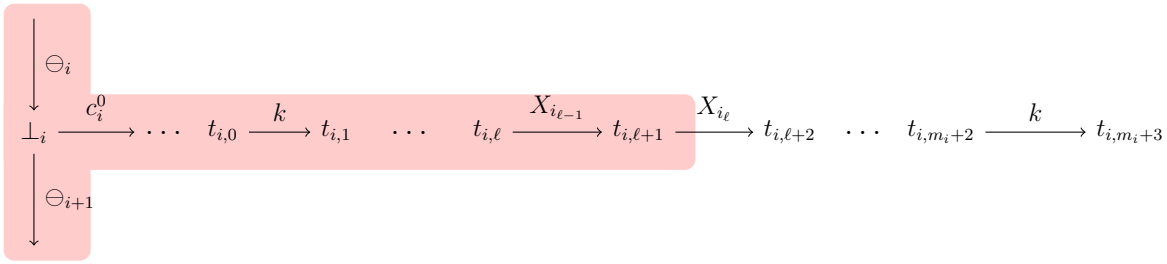
Wir werden nun einführen, wie die einzelnen Gadgets H und $T_0, \ldots, T_{m-1}$ zu dem angekündigten Transitionssystem A_τ verbunden werden. Da wir sicherstellen müssen, dass alle EZTP von A_τ durch d_τ-beschränkte Regionen gelöst werden können, ist diese Konstruktion technisch aufwendig und benötigt (aber nur polynomiell) viele zusätzliche Kanten beziehungsweise Pfade. Wir werden deshalb die notwendigen „Zutaten“ schrittweise entwickeln, um dem Leser die Intuition hinter der Konstruktion möglichst einfach verständlich zu machen.

Genau wie bei vielen bisherigen Konstruktionen möchten wir die Gadgets im Wesentlichen via der Initialzustände $\bot_m$ und $t_{0,0}, \ldots, t_{m-1,0}$ der Gadgets entlang eines einfachen gerichteten Pfades miteinander verbinden, dessen Ereignisse dann jeweils genau einmal in A_τ vorkommen. Jedoch gilt es die Existenz der bereits erwähnten d_τ-beschränkten Regionen zuzusichern. Daher wird das Gadget T_i zunächst für alle $i \in \{0, \ldots, m-1\}$ zu einem Gadget $G_i = \bot_i \xrightarrow{c_i^0} P_i \xrightarrow{u_i} T_i$ erweitert. Dabei ist P_i ein gerichteter Pfad, der sich wiederum selbst aus sogenannten *relevanten* Pfaden zusammensetzt. Diese relevanten Pfade rühren aus der Notwendigkeit her, dass die Ereignisse $X_{i_1}, \ldots, X_{i_{m_i-1}}$ sowie w von allen Zuständen von T_i „getrennt“ werden müssen, die (entlang von T_i) vor ihrem eindeutigen Ursprung in T_i stattfinden. (Es sei bemerkt, dass wir bei der Auflistung der Ereignisse absichtlich mit dem Index i_1 und nicht mit dem Index i_0 beginnen.) Anschließend wird die Kante $\bot_i \xrightarrow{\ominus_i} \bot_{i+1}$ für alle $i \in \{0, \ldots, m-1\}$ hinzugefügt und somit die Pfade $G_0, \ldots, G_{m-1}$ und H zu A_τ verbunden. Abbildung 6.7 skizziert das resultierende Transitionssystem A_τ mit Initialzustand $\bot_0$.

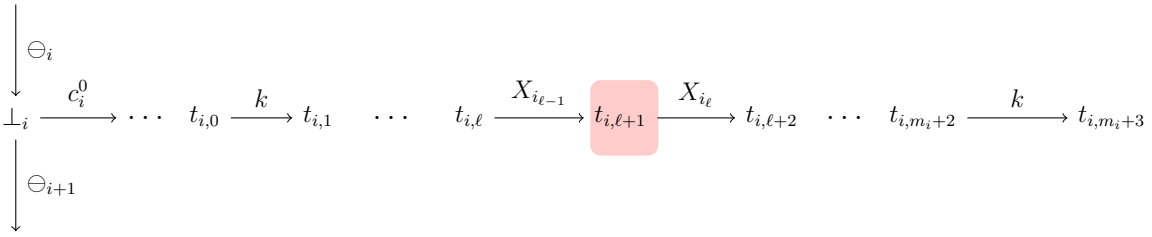
Abbildung 6.7: Skizze des Transitionssystems A_τ der Reduktion für die Netztypen C4.

Um bezüglich der Intuition hinter den relevanten Pfaden etwas genauer werden zu können, seien $\ell \in \{1, \ldots, m_i - 1\}$ und $z \in \{t_{i,0}, \ldots, t_{i,\ell}, t_{i,\ell+2}, \ldots, t_{i,m_i+3}\}$ beliebig aber fest. Es ist (X_{i_ℓ}, z) ein gültiges EZTP, das gelöst werden muss. Unser Ziel ist es, eine d_τ-beschränkte τ-Region $R = (sup, sig)$ zu definieren, sodass $sig(X_{i_\ell}) = \mathsf{inp}$ und $sup(z) = 0$. Wenn $z \in \{t_{i,\ell+2}, \ldots, t_{i,m_i+3}\}$, das heißt, z findet nach dem Ursprung von X_{i_ℓ} in T_i statt,

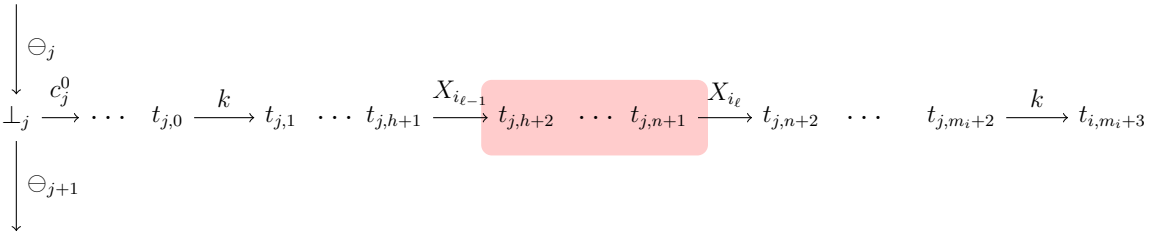
dann erhalten wir eine passende Region wie folgt: $sup(\bot_0) = 1$ und $sig(X_{i_\ell}) = \mathsf{inp}$ und $sig(e) = \mathsf{nop}$ für alle $e \in E \setminus \{X_{i_\ell}\}$. Das heißt, R bildet genau die Nachfolger-Zustände der Ursprünge von X_{i_ℓ} in A_τ auf 0 und alle anderen auf 1 ab. Da X_{i_ℓ} niemals zweimal auf demselben Pfad stattfindet, ist R wohldefiniert. Reduziert auf das Gadget G_i, lässt sich die Situation wie folgt skizzieren:



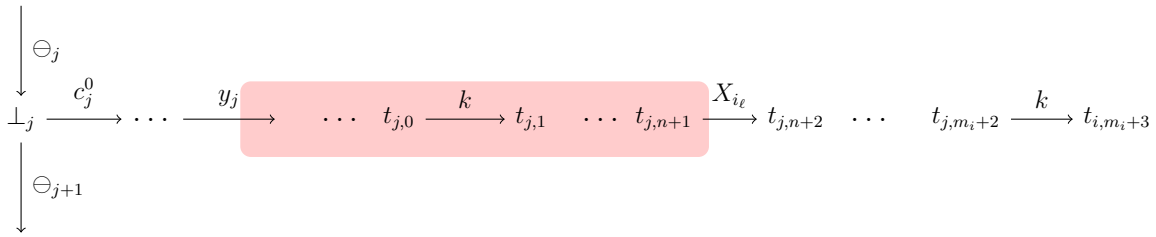
Wenn $z \in \{t_{i,0}, \ldots, t_{i,\ell}\}$, das heißt, z findet vor dem Ursprung $t_{i,\ell+1}$ von X_{i_ℓ} in T_i statt, dann wollen wir $R = (sup, sig)$ mit $sup(\bot_0) = 0$ so definieren, dass –eingeschränkt auf T_i beziehungsweise G_i– nur der Ursprung $t_{i,\ell+1}$ von X_{i_ℓ} in T_i auf 1 abgebildet wird und $sig(X_{i_\ell}) = \mathsf{inp}$ sowie $sig(X_{i_{\ell-1}}) = \mathsf{swap}$ gilt. Reduziert auf das Gadget G_i, lässt sich die Situation wie folgt skizzieren:



Insbesondere sollen die Ereignisse $e \in E(A_\tau) \setminus \{X_{i_\ell}\}$ mit $sig(e) \neq \mathsf{nop}$ nur dazu dienen, dafür zu sorgen, dass $sup(q) = 1$ für jeden Ursprung von X_{i_ℓ} in A_τ gilt. Natürlich soll die Anzahl solcher Ereignisse klein sein, das heißt den Parameter d_τ berücksichtigen. Sei nun $j \in \{0, \ldots m-1\} \setminus \{i\}$ beliebig aber fest, sodass $X_{i_\ell} \in M_j$. Das heißt, es gibt ein $n \in \{0, \ldots, m_j - 1\}$, sodass $X_{j_n} = X_{i_\ell}$. Für die Definition von R sind nun zwei Fälle zu unterscheiden: erstens den Fall $X_{i_{\ell-1}} \in M_j$, das heißt, es gibt ein $h \in \{0, \ldots, m_j - 1\}$ mit $X_{j_h} = X_{i_{\ell-1}}$, und zweitens den Fall $X_{i_{\ell-1}} \notin M_j$. Der erste Fall ist angenehm, denn wegen $i_{\ell-1} < i_\ell$ gilt natürlich $j_h < j_n$ und durch $sig(X_{i_{\ell-1}}) = \mathsf{swap}$ sichern wir bereits $sup(t_{j,n+1}) = 1$ für den Ursprung von X_{i_ℓ} in T_j zu. Mit Blick auf das Gadget G_i, lässt sich die Situation wie folgt skizzieren:



Wenn $X_{i_{\ell-1}} \notin M_j$, dann benötigen wir entlang des Teil-Pfades von G_j, der bei $\perp_j$ startet und $t_{j,n+1}$ endet ein Ereignis e, dessen Signatur dafür sorgt, dass $sup(t_{j,n+1}) = 1$. Natürlich wollen wir sichergehen, dass e (und dessen Signatur) nicht die oben skizzierte Situation in T_i durcheinander bringt, das heißt, $e \notin \{k\} \cup \mathfrak{U}$. Somit sollte e ein „neues“ Ereignis sein, welches auf dem Pfad zwischen $\perp_j$ und $t_{j,n+1}$ stattfindet. Eingeschränkt auf das Gadget G_i, lässt sich die Situation für $e = y_i$ und $sig(y_i) = \mathsf{swap}$ wie folgt skizzieren:



Folgendes ist nun zu beachten: Es kann X_{i_ℓ} in *vielen* verschiedenen Mengen stattfinden, in denen $X_{i_{\ell-1}}$ nicht stattfindet, sagen wir für $j_0, \dots, j_h \in \{0, \dots, m-1\}$ seien $M_{j_0}, \dots, M_{j_h}$ genau diese Mengen. Die oben skizzierte Situation (bezüglich der Ursprüngen von X_{i_ℓ}) tritt nun in allen $T_{j_0}, \dots, T_{j_h}$. Fügen wir nun für jedes T_{j_k} ein eigenes y_{j_k} hinzu, welches dann auf swap abgebildet wird, um $sup(q) = 1$ für den Ursprung q von X_{i_ℓ} in T_{j_k} sicherzustellen, dann gilt $|\{e \in E \mid sig(e) \neq \mathsf{nop}\}| \geq |\{j_0, \dots, j_h\}|$ für $R = (sup, sig)$. Die Anzahl der Vorkommen von X_{i_ℓ} in M ist jedoch nicht durch den Parameter κ gebunden sondern hängt schlimmstenfalls von der Größe m von M ab. In diesem Fall würde R nicht den Parameter d_τ berücksichtigen. (Diese Überlegungen offenbaren vor allem auch, warum wir nicht einfach die jeweils einmaligen Ereignisse $c_{j_0}^0, \dots, c_{j_h}^0$ benutzen.) Wenn wir nun andererseits für alle $T_{j_0}, \dots, T_{j_h}$ *dasselbe* Ereignis y verwenden, welches die beschriebene Funktion für die Ursprünge von X_{i_ℓ} hat, dann überträgt sich die gesamte skizzierte Problematik auf y und die zugehörigen EZTP.

Die Lösung liegt in der Verwendung von für X_{i_ℓ} *relevanter Pfade* (statt einzelner Ereignisse), welche einerseits die Funktion übernehmen, $sup(q) = 1$ für die Ursprünge von X_{i_ℓ} zu gewährleisten und andererseits die Berücksichtigung des Parameters d_τ erlauben.

Sei $i \in \{0, \dots, m-1\}$ beliebig aber fest. Um einheitlich auf die Ereignisse $X_{i_0}, \dots, X_{i_{m_i-1}}$ und w referenzieren zu können, definieren wir $e_i^\ell = X_{i_\ell}$ für alle $\ell \in \{0, \dots, m_i - 1\}$ sowie $e_i^{m_i} = w$. Sei nun $j \in \{1, \dots, m_i\}$ beliebig aber fest und seien $\mathfrak{i}_0 < \dots < \mathfrak{i}_\mathfrak{j} \in \{0, \dots, m-1\} \setminus \{i\}$ genau die von i verschiedenen Indizes, sodass gilt $e_i^j \in E(T_{\mathfrak{i}_\ell})$ aber $e_i^{j-1} \notin E(T_{\mathfrak{i}_\ell})$ für alle $\ell \in \{0, \dots, \mathfrak{j}\}$. (Beachte, dass möglicherweise $j \neq \mathfrak{j}$.) Für alle $\ell \in \{0, \dots, \mathfrak{j}\}$ sagen wir, dass (das Ereignis) e_i^j relevant für $G_{\mathfrak{i}_\ell}$ ist. Darüber hinaus ist

$$P_{\mathfrak{i}_\ell}^{i,j} = s_{\mathfrak{i}_\ell,0}^{i,j} \xrightarrow{v_{\mathfrak{i}_\ell}^{i,j}} s_{\mathfrak{i}_\ell,1}^{i,j} \xrightarrow{\oplus_\ell^{i,j}} s_{\mathfrak{i}_\ell,2}^{i,j} \xrightarrow{\oplus_{\ell-1}^{i,j}} \cdots \xrightarrow{\oplus_0^{i,j}} s_{\mathfrak{i}_\ell,\ell+2}^{i,j}$$

der relevante Pfad für $G_{\mathfrak{i}_\ell}$, der von e_j^i *und seines Vorkommens an j-ter Position in T_i* herrührt. Wir sagen auch, dass $P_{\mathfrak{i}_0}^{i,j}, \dots, P_{\mathfrak{i}_\mathfrak{j}}^{i,j}$ die von e_j^i und seines Vorkommens an j-ter Position in T_i verursachten relevanten Pfade (von A_τ) sind.

Wir können nun für beliebiges aber festes $i \in \{0, \ldots, m-1\}$ das Gadget G_i definieren: Wenn es keine für G_i relevanten Ereignisse gibt, dann ist $G_i = \bot_i \xrightarrow{c_i^0} q_i \xrightarrow{u_i} T_i$. Andernfalls seien $e_{i_0}^{j_0}, \ldots, e_{i_\ell}^{j_\ell}$ mit $i_0 \leq \cdots \leq i_\ell$ und $j_0 \leq \cdots \leq j_\ell$ die –bezüglich ihrer Vorkommen beziehungsweise Positionen in den jeweiligen T-Gadgets paarweise verschiedenen– Ereignisse, die für G_i relevant sind und $P_i^{i_0,j_0}, \ldots, P_i^{i_\ell,j_\ell}$ die entsprechenden für G_i relevanten Pfade. Durch die Verbindung dieser relevanten Pfade ergibt sich G_i wie folgt:

$$G_i = \bot_i \xrightarrow{c_i^0} P_i^{i_0,j_0} \xrightarrow{c_i^1} P_i^{i_1,j_1} \xrightarrow{c_i^2} \ldots \xrightarrow{c_i^\ell} P_i^{i_\ell,j_\ell} \xrightarrow{u_i} T_i$$

Wir bezeichnen $c_i^0, \ldots, c_i^\ell$ als die *c-Ereignisse* von G_i und fassen diese durch die Menge $C_i = \{c_i^0, \ldots, c_i^\ell\}$ zusammen.

Das folgende Beispiel entwickelt vollständig das Transitionssystem A_τ, welches aus der Eingabe aus Beispiel 6.3 resultiert, das definiert war durch $(\mathfrak{U}, M, 2)$ mit $M = \{M_0, \ldots, M_3\}$, wobei $M_0 = \{X_0, X_1\}$, $M_1 = \{X_0, X_3\}$, $M_2 = \{X_1, X_2\}$ sowie $M_3 = \{X_2, X_3\}$:

Beispiel 6.4. Die Gadgets $T_0, \ldots, T_3$ zur Instanz aus Beispiel 6.3 sind wie folgt definiert:

$$\begin{aligned}
T_0 &= t_{0,0} \xrightarrow{k} t_{0,1} \xrightarrow{X_0} t_{0,2} \xrightarrow{X_1} t_{0,3} \xrightarrow{w} t_{0,4} \xrightarrow{k} t_{0,5} \\
T_1 &= t_{1,0} \xrightarrow{k} t_{1,1} \xrightarrow{X_0} t_{1,2} \xrightarrow{X_3} t_{1,3} \xrightarrow{w} t_{1,4} \xrightarrow{k} t_{1,5} \\
T_2 &= t_{2,0} \xrightarrow{k} t_{2,1} \xrightarrow{X_1} t_{2,2} \xrightarrow{X_2} t_{2,3} \xrightarrow{w} t_{2,4} \xrightarrow{k} t_{2,5} \\
T_3 &= t_{3,0} \xrightarrow{k} t_{3,1} \xrightarrow{X_2} t_{3,2} \xrightarrow{X_3} t_{3,3} \xrightarrow{w} t_{3,4} \xrightarrow{k} t_{3,5}
\end{aligned}$$

Im Folgenden wollen wir sukzessive alle relevanten Pfade für A bestimmen:

1. $e_0^1 = X_1$ ist (nur) relevant für G_2, denn $X_0 \notin E(T_2)$. Der relevante Pfad ist

$$P_2^{0,1} = s_{2,0}^{0,1} \xrightarrow{v_2^{0,1}} s_{2,1}^{0,1} \xrightarrow{\oplus_0^{0,1}} s_{2,2}^{0,1}$$

2. $e_0^2 = w$ is relevant für G_1 und G_3, denn $X_1 \notin E(T_1) \cup E(T_3)$. Es ergeben sich die von w und seines Vorkommens an zweiter Position in T_0 verursachten relevanten Pfade

$$P_1^{0,2} = s_{1,0}^{0,2} \xrightarrow{v_1^{0,2}} s_{2,1}^{0,2} \xrightarrow{\oplus_0^{0,2}} s_{1,2}^{0,2} \text{ und } P_3^{0,2} = s_{3,0}^{0,2} \xrightarrow{v_3^{0,2}} s_{3,1}^{0,2} \xrightarrow{\oplus_1^{0,2}} s_{3,2}^{0,2} \xrightarrow{\oplus_0^{0,2}} s_{3,3}^{0,2}$$

3. $e_1^1 = X_3$ ist (nur) relevant für G_3, denn $X_0 \notin E(T_3)$. Der relevante Pfad ist

$$P_3^{1,1} = s_{3,0}^{1,1} \xrightarrow{v_3^{1,1}} s_{3,1}^{1,1} \xrightarrow{\oplus_0^{1,1}} s_{3,2}^{1,1}$$

4. $e_1^2 = w$ is relevant für G_0 und G_2, denn $X_3 \notin E(T_0) \cup E(T_2)$. Es ergeben sich die von w und seines Vorkommens an zweiter Position in T_1 verursachten relevanten Pfade

$$P_0^{1,2} = s_{0,0}^{1,2} \xrightarrow{v_0^{1,2}} s_{0,1}^{1,2} \xrightarrow{\oplus_0^{1,2}} s_{0,2}^{1,2} \text{ und } P_2^{1,2} = s_{2,0}^{1,2} \xrightarrow{v_3^{1,2}} s_{2,1}^{1,2} \xrightarrow{\oplus_1^{1,2}} s_{2,2}^{1,2} \xrightarrow{\oplus_0^{1,2}} s_{2,3}^{1,2}$$

   Man beachte vor allem, dass $e_0^2 = e_1^2 = w$, aber die von w wegen seines Vorkommens an zweiter Position in T_0 verursachten relevanten Pfade sind verschieden von den von w wegen seines Vorkommens an zweiter Position in T_1 verursachten relevanten Pfade.

5. $e_2^1 = X_2$ ist (nur) relevant für G_3, denn $X_1 \notin E(T_3)$. Der relevante Pfad ist

$$P_3^{2,1} = s_{3,0}^{2,1} \xrightarrow{v_3^{2,1}} s_{3,1}^{2,1} \xrightarrow{\oplus_0^{2,1}} s_{3,2}^{2,1}$$

6. $e_2^2 = w$ is relevant für G_0 und G_1, denn $X_2 \notin E(T_0) \cup E(T_1)$. Es ergeben sich die relevanten Pfade

$$P_0^{2,2} = s_{0,0}^{2,2} \xrightarrow{v_0^{2,2}} s_{0,1}^{2,2} \xrightarrow{\oplus_0^{2,2}} s_{0,2}^{2,2} \text{ und } P_1^{2,2} = s_{1,0}^{2,2} \xrightarrow{v_1^{2,2}} s_{1,1}^{2,2} \xrightarrow{\oplus_1^{2,2}} s_{1,2}^{2,2} \xrightarrow{\oplus_0^{2,2}} s_{1,3}^{2,2}$$

7. $e_3^1 = X_3$ ist (nur) relevant für G_1, denn $X_2 \notin E(T_1)$. Der relevante Pfad ist

$$P_1^{3,1} = s_{1,0}^{3,1} \xrightarrow{v_1^{3,1}} s_{1,1}^{3,1} \xrightarrow{\oplus_0^{2,1}} s_{1,2}^{3,1}$$

8. $e_3^2 = w$ is relevant für G_0 und G_2, denn $X_3 \notin E(T_0) \cup E(T_2)$. Es ergeben sich die relevanten Pfade

$$P_0^{3,2} = s_{0,0}^{3,2} \xrightarrow{v_0^{3,2}} s_{0,1}^{3,2} \xrightarrow{\oplus_0^{3,2}} s_{0,2}^{3,2} \text{ und } P_2^{3,2} = s_{2,0}^{3,2} \xrightarrow{v_3^{3,2}} s_{2,1}^{3,2} \xrightarrow{\oplus_1^{3,2}} s_{2,2}^{3,2} \xrightarrow{\oplus_0^{3,2}} s_{2,3}^{3,2}$$

Das Transitionssystem A_τ, das auf der Eingabe von Beispiel 6.3 basiert, lässt damit wie folgt skizzieren, wobei wir die relevanten Pfade durch ihre Namen abkürzen:

$$\begin{array}{l}
\perp_0 \xrightarrow{c_0^0} P_0^{1,2} \xrightarrow{c_0^1} P_0^{2,2} \xrightarrow{c_0^2} P_0^{3,2} \xrightarrow{u_0} t_{0,0} \xrightarrow{k} t_{0,1} \xrightarrow{X_0} t_{0,2} \xrightarrow{X_1} t_{0,3} \xrightarrow{w} t_{0,4} \xrightarrow{k} t_{0,5} \\
\ominus_1 \downarrow \\
\perp_1 \xrightarrow{c_1^0} P_1^{0,2} \xrightarrow{c_1^1} P_1^{2,2} \xrightarrow{c_1^2} P_1^{3,1} \xrightarrow{u_1} t_{1,0} \xrightarrow{k} t_{1,1} \xrightarrow{X_0} t_{1,2} \xrightarrow{X_3} t_{1,3} \xrightarrow{w} t_{1,4} \xrightarrow{k} t_{1,5} \\
\ominus_2 \downarrow \\
\perp_2 \xrightarrow{c_2^0} P_2^{0,1} \xrightarrow{c_2^1} P_2^{1,2} \xrightarrow{c_2^2} P_2^{3,2} \xrightarrow{u_2} t_{2,0} \xrightarrow{k} t_{2,1} \xrightarrow{X_1} t_{2,2} \xrightarrow{X_2} t_{2,3} \xrightarrow{w} t_{2,4} \xrightarrow{k} t_{2,5} \\
\ominus_3 \downarrow \\
\perp_3 \xrightarrow{c_3^0} P_3^{0,2} \xrightarrow{c_3^1} P_3^{1,1} \xrightarrow{c_3^2} P_3^{2,1} \xrightarrow{u_3} t_{3,0} \xrightarrow{k} t_{3,1} \xrightarrow{X_2} t_{3,2} \xrightarrow{X_3} t_{3,3} \xrightarrow{w} t_{3,4} \xrightarrow{k} t_{3,5} \\
\ominus_4 \downarrow \\
\perp_4 \xrightarrow{u_4} h_0 \xrightarrow{k} h_1 \xrightarrow{w} h_2 \xrightarrow{o} h_3 \xrightarrow{k} h_4
\end{array}$$

Wir werden nun zeigen, dass A_τ die τ-EZTE beziehungsweise die τ-ZTE hat, wenn $(\mathfrak{U}, M)$ ein OS hat. Solange nicht ausdrücklich anders vereinbart, sei S im Folgenden eine beliebige aber feste ungerade Menge mit höchstens κ Elementen für $(\mathfrak{U}, M)$. Wie gehabt definieren wir eine τ-Region $R = (sup, sig)$ implizit mittels $sig(\bot_0)$ und sig und fassen Ereignisse gleicher Signatur in einer Menge $\mathcal{E}^R_{e_\tau} = \{e \in E(A_\tau) \mid sig(e) = e_\tau\}$ zusammen. Darüber hinaus gilt für ein Ereignis $e \in E(A_\tau)$, dass sein Signaturwert nop ist, das heißt $e \in E^R_{\mathsf{nop}}$, wenn es nicht explizit als Element einer Menge $\mathcal{E}^R_{e_\tau}$ mit $e_\tau \neq \mathsf{nop}$ aufgeführt ist. Wir werden deshalb der Einfachheit halber die Menge $\mathcal{E}^R_{\mathsf{nop}}$ nicht explizit definieren.

Die Aussage des folgenden Fakts erleichtert uns die Arbeit, indem es im Wesentlichen erlaubt, für viele Ereignisse a von A_τ nur Zustände zu betrachten, die erstens auf Pfaden in A_τ „vor“ a stattfinden und zweitens innerhalb der Gadgets auftreten, in denen auch a stattfindet:

Fakt 6.13. *Sei $a \in E(A_\tau)$ ein Ereignis von A_τ und $\bot_0 \xrightarrow{e_1} z_1 \xrightarrow{e_2} \dots \xrightarrow{a} z_\ell \xrightarrow{e_{\ell+1}} \dots \xrightarrow{e_n} z_n$ ein Pfad von A_τ auf dem a stattfindet und $G \in \{G_0, \dots, G_{m-1}, H\}$ ein Gadget von A_τ, in dem a nicht stattfindet, das heißt $a \notin E(G)$. Wenn a auf allen Pfaden von A_τ jeweils höchstens einmal stattfindet, dann ist das EZTP (a, q) für alle $q \in \{z_\ell, \dots, z_n\}$ sowie für alle $q \in Z(G)$ durch eine d_τ-beschränkte τ-Region lösbar.*

Beweis. Die folgende τ-Region $R_1 = (sup_1, sig_1)$ löst (a, q) für alle $q \in \{z_\ell, \dots, z_n\}$: $sup_1(\bot_0) = 1$ und $\mathcal{E}^{R_1}_{\mathsf{inp}} = \{a\}$.

Die folgende τ-Region $R_2 = (sup_2, sig_2)$ löst (a, q) für alle $q \in Z(G)$ für den Fall, dass $G = G_i$ für ein $i \in \{0, \dots, m-1\}$ ist: wenn $i = 0$, dann $sup_2(\bot_0) = 0$, sonst $sup_2(\bot_0) = 1$; außerdem $\mathcal{E}^{R_2}_{\mathsf{inp}} = \{a\}$ und $\mathcal{E}^{R_2}_{\mathsf{swap}} = \{\ominus_i, \ominus_{i+1}\}$. Analog definiert man eine entsprechende Region für den Fall $G = H$. □

Man beachte, dass –mit Ausnahme von k– alle Ereignisse von A_τ die in Fakt 6.13 genannte Voraussetzung erfüllen. Die folgenden Fakten behandeln die Ereignisse (gleicher Art) von A_τ im Einzelnen und beweisen ihre Lösbarkeit.

Fakt 6.14. *Das Ereignis k ist durch d_τ-beschränkte τ-Regionen lösbar.*

Beweis. Wir benötigen die ungerade Menge S mit $|S| \leq \kappa$ für $(\mathfrak{U}, M)$.

Die folgende Region $R_1 = (sup_1, sig_1)$ löst (k, q) $q = h_2$ und für alle $q \in \{z \in Z \mid \xrightarrow{k} z\}$: $sup_1(\bot_0) = 0$ und $\mathcal{E}^{R_1}_{\mathsf{inp}} = \{k\}$ und $\mathcal{E}^{R_1}_{\mathsf{swap}} = \{o\} \cup S$. Wegen $|S| \leq \kappa$, erfüllt R_1 auch $|\{e \in E(A_\tau) \mid sig_1(e) \neq \mathsf{nop}\}| \leq \kappa + 2 = d_\tau$.

Sei $i \in \{0, \dots, m-1\}$ beliebig aber fest. Die folgende Region $R_2 = (sup_2, sig_2)$ löst (k, q) für alle relevanten $q \in Z(G_i)$: wenn $i = 0$, dann $sup_2(\bot_0) = 0$, andernfalls $sup_2(\bot_0) = 1$; außerdem $\mathcal{E}^{R_2}_{\mathsf{inp}} = \{k\}$ und $\mathcal{E}^{R_2}_{\mathsf{swap}} = \{\ominus_i, \ominus_{i+1}, u_i\}$.

Analog beweist man, dass $(k, \bot_m)$ passend lösbar ist. Weil i beliebig war, folgt die τ-Lösbarkeit von k. □

Fakt 6.15. *Das Ereignis w ist durch d_τ-beschränkte τ-Regionen lösbar.*

$$
\begin{aligned}
P^{i,j}_{\mathsf{i}_0} &= s^{i,j}_{\mathsf{i}_0,0} \xrightarrow{v^{i,j}_{\mathsf{i}_0}} s^{i,j}_{\mathsf{i}_0,1} \xrightarrow{\oplus^{i,j}_0} s^{i,j}_{\mathsf{i}_0,2}\\
P^{i,j}_{\mathsf{i}_1} &= s^{i,j}_{\mathsf{i}_1,0} \xrightarrow{v^{i,j}_{\mathsf{i}_1}} s^{i,j}_{\mathsf{i}_1,1} \xrightarrow{\oplus^{i,j}_1} s^{i,j}_{\mathsf{i}_1,2} \xrightarrow{\oplus^{i,j}_0} s^{i,j}_{\mathsf{i}_1,3}\\
P^{i,j}_{\mathsf{i}_2} &= s^{i,j}_{\mathsf{i}_2,0} \xrightarrow{v^{i,j}_{\mathsf{i}_2}} s^{i,j}_{\mathsf{i}_2,1} \xrightarrow{\oplus^{i,j}_2} s^{i,j}_{\mathsf{i}_2,2} \xrightarrow{\oplus^{i,j}_1} s^{i,j}_{\mathsf{i}_2,3} \xrightarrow{\oplus^{i,j}_0} s^{i,j}_{\mathsf{i}_2,4}\\
P^{i,j}_{\mathsf{i}_3} &= s^{i,j}_{\mathsf{i}_3,0} \xrightarrow{v^{i,j}_{\mathsf{i}_3}} s^{i,j}_{\mathsf{i}_3,1} \xrightarrow{\oplus^{i,j}_3} s^{i,j}_{\mathsf{i}_3,2} \xrightarrow{\oplus^{i,j}_2} s^{i,j}_{\mathsf{i}_3,3} \xrightarrow{\oplus^{i,j}_1} s^{i,j}_{\mathsf{i}_3,4} \xrightarrow{\oplus^{i,j}_0} s^{i,j}_{\mathsf{i}_3,5}\\
&\vdots\\
P^{i,j}_{\mathsf{i}_\mathsf{j}} &= s^{i,j}_{\mathsf{i}_\mathsf{j},0} \xrightarrow{v^{i,j}_{\mathsf{i}_\mathsf{j}}} s^{i,j}_{\mathsf{i}_\mathsf{j},1} \xrightarrow{\oplus^{i,j}_{\mathsf{j}}} s^{i,j}_{\mathsf{i}_\mathsf{j},2} \xrightarrow{\oplus^{i,j}_{\mathsf{j}-1}} s^{i,j}_{\mathsf{i}_\mathsf{j},3} \xrightarrow{\oplus^{i,j}_{\mathsf{j}-2}} s^{i,j}_{\mathsf{i}_\mathsf{j},4} \xrightarrow{\oplus^{i,j}_{\mathsf{j}-3}} s^{i,j}_{\mathsf{i}_\mathsf{j},5} \xrightarrow{\oplus^{i,j}_{\mathsf{j}-4}} \cdots \xrightarrow{\oplus^{i,j}_2} s^{i,j}_{\mathsf{i}_\mathsf{j},\mathsf{j}} \xrightarrow{\oplus^{i,j}_1} s^{i,j}_{\mathsf{i}_\mathsf{j},\mathsf{j}+1} \xrightarrow{\oplus^{i,j}_0} s^{i,j}_{\mathsf{i}_\mathsf{j},\mathsf{j}+2}
\end{aligned}
$$

Abbildung 6.8: Eine Skizze der Region R_6 aus dem Beweis von Fakt 6.17 für den Fall $\ell = 1$, die das Ereigniss $\oplus^{i,j}_1$ löst, eingeschränkt auf die relevanten Pfade $P^{i,j}_{\mathsf{i}_0}, \ldots, P^{i,j}_{\mathsf{i}_\mathsf{j}}$. Der rote Bereich korrespondiert genau mit den Zuständen von A_τ, deren Support positiv ist. Es ist leicht zu sehen, dass sich diese Einschränkung auf A_τ fortsetzten lässt, ohne die Anzahl der Ereignisse zu erhöhen, deren Signatur ungleich nop ist.

Beweis. Sei $i \in \{0, \ldots, m-1\}$ beliebig aber fest. Die folgende Region $R_3 = (sup_3, sig_3)$ löst (w, z) für alle $z \in \{\bot_4, h_0\}$ und alle $z \in Z(G_i) \setminus \{t_{i,1}, \ldots, t_{i,m_i+3}\}$: $sup_3(\bot_0) = 0$ und $\mathcal{E}^{R_3}_{\mathsf{inp}} = \{w\}$ und $\mathcal{E}^{R_3}_{\mathsf{swap}} = \{k\}$.

Die folgende Region $R_4 = (sup_4, sig_4)$ löst (w, z) für alle $z \in \{t_{i,1}, \ldots, t_{i,m_i}\}$ und benötigt die für $e^{m_i}_i = w$ relevanten Pfade (wenn es denn solche gibt), genauer gesagt das zugehörige Ereignis $\oplus^{i,m_i}_0$ (wenn es existiert): $sup(\bot_0) = 0$ und $\mathcal{E}^{R_4}_{\mathsf{inp}} = \{w\}$ und $\mathcal{E}^{R_4}_{\mathsf{swap}} = \{X_{i_{m_i-1}}, \oplus^{i,m_i}_0\}$.

Aus der Beliebigkeit von i sowie der Aussage von Lemma 6.13 folgt die τ-Lösbarkeit von w durch d_τ-beschränkte Regionen. □

Fakt 6.16. *Wenn $X \in \mathfrak{U}$, dann ist X durch d_τ-beschränkte τ-Regionen lösbar.*

Beweis. Seien $i \in \{0, \ldots, m-1\}$ und $j \in \{0, \ldots, m_i - 1\}$ beliebig aber fest. Die Lösbarkeit von (X_{i_j}, z) für alle $z \in Z(G_i) \setminus \{t_{i,1}, \ldots, t_{i,m_i+3}\}$ folgt aus einer Region, die analog zu R_3 aus Fakt 6.15 definiert ist. Die Lösbarkeit von (X_{i_j}, z) für alle $z \in \{t_{i,1}, \ldots, t_{i,j}\}$, wobei insbesondere $j > 0$ impliziert ist, folgt aus einer Region, die analog zu R_4 aus Fakt 6.15 definiert ist und die für $e^j_i = X_{i_j}$ relevanten Pfade beziehungsweise das Ereignis $\oplus^{i,j}_0$ benötigt (wenn es diese denn gibt). Lemma 6.13 komplettiert die τ-Lösbarkeit von X_{i_j} durch d_τ-beschränkte Regionen. Aus der Beliebigkeit von i und j folgt, dass alle $X \in \mathfrak{U}$ passend lösbar sind. □

Wenden wir uns nun den Ereignissen der relevanten Pfade zu:

Fakt 6.17. *Wenn $e \in \{v_i^{j,\ell}, \oplus_i^{j,\ell} \mid i,j,\ell \in \mathbb{N}\} \cap E(A_\tau)$ ein Ereignis von A_τ ist, dann ist e durch d_τ-beschränkte τ-Regionen lösbar.*

Beweis. Seien $i \in \{0,\ldots,m-1\}$ und $j \in \{1,\ldots,m_i\}$ beliebig aber fest und seien $\mathfrak{i}_0 < \cdots < \mathfrak{i}_\mathfrak{j} \in \{0,\ldots,m-1\}$ genau die paarweise verschiedenen Indizes, sodass $P_{\mathfrak{i}_0}^{i,j},\ldots,P_{\mathfrak{i}_\mathfrak{j}}^{i,j}$ die von e_j^i und seines Vorkommens an j-ter Position in T_i verursachten relevanten Pfade (von A_τ) sind. Das bedeutet, wir müssen uns um die τ-Lösbarkeit der Ereignisse $v_{\mathfrak{i}_0}^{i,j},\ldots,v_{\mathfrak{i}_\mathfrak{j}}^{i,j}$ und der Ereignisse $\oplus_0^{i,j},\ldots,\oplus_\mathfrak{j}^{i,j}$ kümmern.

Sei $\ell \in \{0,\ldots,\mathfrak{j}\}$ beliebig aber fest. Es ist leicht zu sehen, dass $v_{\mathfrak{i}_\ell}^{i,j}$ durch eine 2-beschränkte τ-Region lösbar ist: Sowohl $v_{\mathfrak{i}_\ell}^{i,j}$ als auch das eindeutige c-Ereignis $c \in C_{\mathfrak{i}_\ell}$, welches der direkte Vorgänger von $v_{\mathfrak{i}_\ell}^{i,j}$ ist, kommen genau einmal in A vor. Deshalb ist die Region $R_5 = (sup_5, sig_5)$ mit $sup_5(\bot_0) = 0$ und $sig(v_{\mathfrak{i}_\ell}^{i,j}) = \mathsf{inp}$ sowie $sig(c) = \mathsf{swap}$ und $sig(e) = \mathsf{nop}$ für alle $e \in E \setminus \{v_{\mathfrak{i}_\ell}^{i,j}, c\}$ wohldefiniert und löst $v_{\mathfrak{i}_\ell}^{i,j}$.

Damit bleibt, die τ-Lösbarkeit der Ereignisse $\oplus_0^{i,j},\ldots,\oplus_\mathfrak{j}^{i,j}$ zu diskutieren. Sei $\ell \in \{0,\ldots,\mathfrak{j}\}$ beliebig aber fest. Es ist $v_{\mathfrak{i}_\ell}^{i,j}$ der direkte Vorgänger von $\oplus_\ell^{i,j}$ in $G_{\mathfrak{i}_\ell}$. Darüber hinaus ist $\oplus_{\ell+1}^{i,j}$ der direkte Vorgänger von $\oplus_\ell^{i,j}$ in allen Gadgets $G_{\mathfrak{i}_{\ell+1}},\ldots,G_{\mathfrak{i}_\mathfrak{j}}$, wobei dies natürlich nur für den Fall $\ell < \mathfrak{j}$ interessant ist. Andernfalls kommt $\oplus_\ell^{i,j} = \oplus_\mathfrak{j}^{i,j}$ genau einmal in A_τ vor, nämlich in $G_{\mathfrak{i}_\mathfrak{j}}$. Die folgende Region $R_6 = (sup_6, sig_6)$ löst $\oplus_\ell^{i,j}$: $sup(\bot_0) = 0$; für alle $e \in E$, wenn $e = \oplus_\ell^{i,j}$, dann $sig(e) = \mathsf{inp}$; wenn $e \in \{v_{\mathfrak{i}_\ell}^{i,j}, \oplus_{\ell+1}^{i,j}\}$, dann $sig(e) = \mathsf{swap}$, sonst $sig(e) = \mathsf{nop}$. Abbildung 6.8 skizziert R_6 für $\ell = 1$ eingeschränkt auf $P_{\mathfrak{i}_0}^{i,j},\ldots,P_{\mathfrak{i}_\mathfrak{j}}^{i,j}$. □

Das folgende Lemma komplettiert den Beweis von Satz 6.3.1:

Lemma 6.7. *Wenn es eine ungerade Menge mit höchstens κ Elementen für $(\mathfrak{U}, M)$ gibt, dann gibt es einen Zeugen d_τ-beschränkter Regionen für die τ-EZTE und die τ-ZTE von A_τ.*

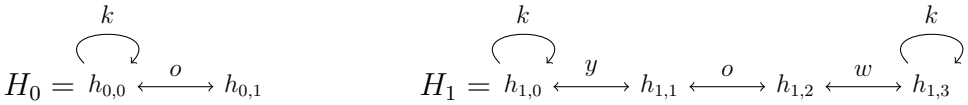
Beweis. Es ist leicht einzusehen, dass die (Verbindungs-) Ereignisse $c_i^0,\ldots,c_i^\ell$ sowie u_i des Gadgets G_i (wie oben definiert) für alle $i \in \{0,\ldots,m-1\}$ durch d_τ-beschränkte Regionen lösbar sind. Schließlich stellt man dies auch für die (Verbindungs-) Ereignisse $\ominus_1,\ldots,\ominus_m$ fest. Die τ-Lösbarkeit aller anderen Ereignisse von A_τ folgt aus Fakt 6.14 bis Fakt 6.17. Des Weiteren lösen die (besprochenen beziehungsweise angedeuteten) Regionen, welche die τ-EZTE von A_τ bezeugen, auch alle ZTP von A_τ. □

Der Beweis von Satz 6.3.2

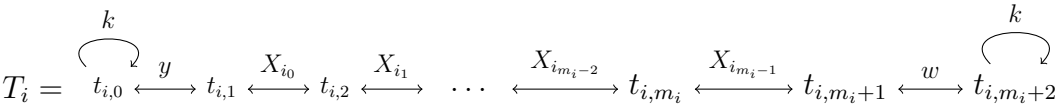
Der Beweis von Satz 6.3.2 erfolgt gemäß des eingangs skizzierten Vorgehens mittels einer parametrisierten Reduktion von OS. Sei also $(\mathfrak{U}, M, \kappa)$ eine beliebige aber feste Eingabe für OS, die wir wie folgt auf eine Eingabe (A_τ, d_τ) für Bτ-SMBP beziehungsweise Bτ-RMBP reduzieren: Wir definieren zunächst $d_\tau = \kappa + 3$ und fahren mit der Konstruktion von A_τ fort. Ein wesentlicher Aspekt von τ ist, dass ein EZTP (e, z') nicht gelöst werden

kann, wenn z' ein Ziel von e ist, das heißt, wenn für einen Zustand z die Kante $z \xrightarrow{e} z'$ in A_τ ist. Dies liegt daran, dass $sig(e) \in \{\mathsf{used}, \mathsf{free}\}$ stets $sup(z) = sup(z')$ impliziert, sodass $sig(e) \in \{\mathsf{used}, \mathsf{free}\}$ und $\neg sup(z') \xrightarrow{sig(e)}$ nicht gleichzeitig möglich sind. Per Konstruktion sichern wir deshalb zu, dass Ereignisse nur an bi-direktionalen Kanten $z \xleftrightarrow{e} z'$ stattfinden, dass heißt, $z \xrightarrow{e} z' \in A_\tau$ impliziert stets $z' \xrightarrow{e} z \in A_\tau$.

Das Transitionssystem A_τ hat die folgenden Gadgets H_0 und H_1, welche das EZTP $\alpha = (k, h_{0,1})$ zur Verfügung stellen:



Darüber hinaus hat A_τ für alle $i \in \{0, \ldots, m-1\}$ das folgende Gadget T_i, das die Menge $M_i = \{X_{i_0}, \ldots, X_{m_i-1}\}$ durch Verwendung ihrer Elemente als Ereignisse repräsentiert:



Des Weiteren wird T_i erneut für alle $i \in \{0, \ldots, m-1\}$ zu einem Gadget $G_i = \bot_i \xleftrightarrow{c_i^0} P_i \xleftrightarrow{u_i} T_i$ erweitert. Es ist P_i die Verbindung relevanter Pfade $P_i^{i_0,j_0}, \ldots, P_i^{i_\ell,j_\ell}$, die analog zu den für den Beweis von Satz 6.3.1 eingeführten Begrifflichkeiten definiert sind. Einzig die folgende Ausnahme ist zu beachten: Es finden alle Ereignisse e an bi-direktionalen Kanten statt, das heißt, wo vorher nur $z \xrightarrow{e} z'$ hinzugefügt wurde, wird nun auch noch $z' \xrightarrow{e} z$ hinzugefügt. Schließlich wird für alle $i \in \{0, \ldots, m\}$ die bi-direktionale Kante $\bot_i \xleftrightarrow{\ominus_{i+1}} \bot_{i+1}$ sowie $\bot_m \xleftrightarrow{u_m} h_{0,0}$ und $\bot_{m+1} \xleftrightarrow{u_{m+1}} h_{1,0}$ ergänzt. Abbildung 6.9 skizziert das resultierende Transitionssystem A_τ mit Initialzustand $\bot_0$.

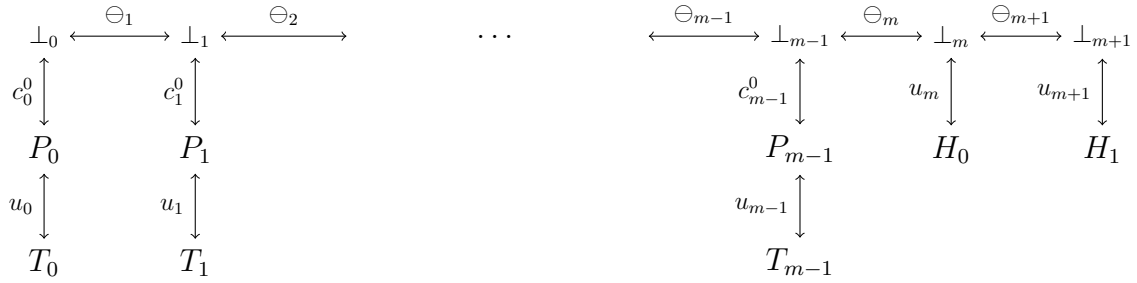


Abbildung 6.9: Skizze des Transitionssystems A_τ der Reduktion für die Netztypen C5.

Gemäß des folgenden Lemmas erlaubt $(\mathfrak{U}, M)$ eine positive Entscheidung, wenn (A_τ, d_τ) diese erlaubt:

Lemma 6.8. *Wenn es einen Zeugen $\mathcal{R}$ d_τ-beschränkter Region für die τ-EZTE von A_τ gibt, dann gibt es eine ungerade Menge mit höchstens κ Elementen für $(\mathfrak{U}, M)$.*

Beweis. Weil es einen Zeugen $\mathcal{R}$ d_τ-beschränkter Regionen für die τ-EZTE von A_τ gibt, gibt es eine A_τ-beschränkte τ-Region, die α löst. Sei $R = (sup, sig)$ eine solche Region, das heißt, es ist entweder $sig(k) = \mathsf{used}$ und $sup(h_{0,1}) = 0$ oder $sig(k) = \mathsf{free}$ und $sup(h_{0,1}) = 1$. Wir diskutieren im Folgenden nur den ersten Fall, das heißt $sig(k) = \mathsf{used}$ und $sup(h_{0,1}) = 0$, denn der zweite Fall ist dazu symmetrisch.

Sei $i \in \{0, \ldots, m-1\}$ beliebig aber fest. Wegen $sig(k) = \mathsf{used}$ erhalten wir $sup(z) = sup(z') = 1$ für jede Kante $z \xrightarrow{k} z'$ von A. Mit $sup(h_{0,0}) = 1$ und $sup(h_{0,1}) = 0$ erhalten wir sofort $sig(o) = \mathsf{swap}$. Des Weiteren schließen wir aus $sup(h_{1,0}) = sup(h_{1,3}) = 1$, dass das Bild R^P des Pfades $P = h_{1,0} \xrightarrow{y} \ldots \xrightarrow{w} h_{1,3}$ ein Pfad von 1 nach 1 in τ ist. Dies impliziert das die Anzahl der Zustandswechsel auf diesem Pfad P^R gerade ist. Das heißt, die Zahl $|\{e \in \{o, y, w\} \mid sig(e) = \mathsf{swap}\}|$ ist gerade. Aus $sig(o) = \mathsf{swap}$ folgt, dass es genau ein weiteres Ereignis $e \in \{y, w\}$ gibt, sodass $sig(e) = \mathsf{swap}$. Wir diskutieren im Folgenden nur den Fall $sig(w) = \mathsf{swap}$ und $sig(y) \neq \mathsf{swap}$, denn der Fall $sig(w) \neq \mathsf{swap}$ und $sig(y) = \mathsf{swap}$ ist dazu symmetrisch.

Sei also $sig(w) = \mathsf{swap}$ und $sig(y) \neq \mathsf{swap}$, sodass $z \xrightarrow{w} z'$ stets $sup(z) \neq sup(z')$ und $q \xrightarrow{y} q'$ stets $sup(q) = sup(q')$ impliziert. Wegen $sup(t_{i,0}) = sup(t_{i,m_i+2}) = 1$ erhalten wir daraus $sup(t_{i,1}) = 1$ und $sup(t_{i,m_i+1}) = 0$. Insbesondere ist das Bild P_i^R des Pfades $P_i = t_{i,1} \xrightarrow{X_{i_0}} \ldots \xrightarrow{X_{i_{m_i-1}}} t_{i,m_i+1}$ ein Pfad von 1 nach 0 in τ. Daraus folgt, das Anzahl der Zustandswechsel auf diesem Pfad –und damit insbesondere die Anzahl $|\{X \in M_i \mid sig(X) = \mathsf{swap}\}$– ungerade ist. Aus der Beliebigkeit von i folgt nun, dass für die Menge $S = \{X \in \mathfrak{U} \mid sig(X) = \mathsf{swap}\}$ die Kardinalität $|S \cap M_i|$ für alle $i \in \{0, \ldots, m-1\}$ ungerade ist. Darüber hinaus gilt $|S| \leq \kappa$, denn R ist $\kappa + 3$-beschränkt und es gilt $sig(e) \neq \mathsf{nop}$ für alle $e \in \{k, o, w\}$. Dies beweist das Lemma. □

Um den Beweis von Satz 6.3.2 abzuschließen, bleibt zu zeigen, dass es einen Zeugen d_τ-beschränkter Regionen für die τ-EZTE und die τ-ZTE von A_τ gibt, wenn $(\mathfrak{U}, M)$ eine positive Entscheidung erlaubt. Solange nicht ausdrücklich anders vereinbart, sei S deshalb im Folgenden eine beliebige aber feste ungerade Menge mit höchstens κ Elementen für $(\mathfrak{U}, M)$.

Wir gehen schrittweise vor und behandeln die jeweiligen Ereignisse von A_τ individuell. Der Einfachheit halber diskutieren nur den Fall $\mathsf{used} \in \tau$ und kommen mit Regionen aus, die nur nop und swap sowie used benutzen. Wegen $\{\mathsf{nop}, \mathsf{swap}, \mathsf{used}\} \equiv \{\mathsf{nop}, \mathsf{swap}, \mathsf{free}\}$ und Satz 6.1, folgt dann, dass es auch einen entsprechenden Zeugen für den Fall $\mathsf{used} \notin \tau$ gibt, denn dieser impliziert $\mathsf{free} \in \tau$. Folgende Beobachtung erlaubt uns eine weitere Vereinfachung: Wie man leicht sieht, ist ein EZTP (e, z) von A_τ wegen $\mathsf{used} \in \tau$ passend lösbar, wenn e und z in unterschiedlichen Gadgets stattfinden: Man definiert die offensichtliche Region, die e auf used abbildet und alle Zustände, bis auf jene des Gadgets in denen z stattfindet, auf 1 abbildet. Deshalb werden wir im Folgenden für ein Ereignis e

von A_τ nur jene EZTP (e, z) betrachten, für die gilt, dass e und z im gleichen Gadget stattfinden.

Wie gehabt wird eine τ-Region $R = (sup, sig)$ implizit mittels $sup(\bot_0)$ und sig definiert. Ereignisse gleicher Signatur $e_\tau \in \tau$ werden in einer Menge $\mathcal{E}^R_{e_\tau} = \{e \in E(A_\tau) \mid sig(e) = e_\tau\}$ zusammengefasst. Darüber hinaus gilt stets $\mathcal{E}^R_{\mathsf{nop}} = E(A_\tau) \setminus (\mathcal{E}^R_{\mathsf{used}} \cup \mathcal{E}^R_{\mathsf{swap}})$, sodass wir die Menge $\mathcal{E}^R_{\mathsf{nop}}$ jeweils nicht explizit definieren werden.

Fakt 6.18. *Das Ereignis k ist durch d_τ-beschränkte τ-Regionen lösbar.*

Beweis. Wir benötigen die ungerade Menge S mit höchstens κ Elementen für $(\mathfrak{U}, M)$.

Die folgende Region $R_1 = (sup_1, sig_1)$ löst $\alpha = (k, h_{0,1})$: $sup_1(\bot_0) = 1$ und $\mathcal{E}^{R_1}_{\mathsf{used}} = \{k\}$ und $\mathcal{E}^{R_1}_{\mathsf{swap}} = \{o, w\} \cup S$. Weil $|S| \leq \kappa$, ist R_1 offenbar $\kappa + 3$-beschränkt.

Sei $i \in \{0, \ldots, m-1\}$ beliebig aber fest. Die folgende Region $R_2 = (sup_2, sig_2)$ löst (k, z) für alle $z \in \{h_{1,1}, h_{1,2}\}$ und alle relevanten $z \in Z(G_i)$: Wenn $i = 0$, dann $sup_2(\bot_0) = 0$, sonst $sup_2(\bot_0) = 1$; außerdem $\mathcal{E}^{R_2}_{\mathsf{used}} = \{k\}$ und $\mathcal{E}^{R_2}_{\mathsf{swap}} = \{\ominus_i, \ominus_{i+1}, u_i, y, w\}$.

Es ist klar, dass (k, z) für alle $z \in \{\bot_m, \bot_{m+1}\}$ passend lösbar ist. Weil i beliebig war, folgt die Behauptung. □

Fakt 6.19. *Das Ereignis y ist durch d_τ-beschränkte τ-Regionen lösbar.*

Beweis. Sei $i \in \{0, \ldots, m-1\}$ beliebig aber fest.

Die folgende Region $R_1 = (sup_1, sig_1)$ löst (y, z) für alle $z \in \{h_{1,2}, h_{1,3}\}$ und alle $z \in \{t_{i,2}, \ldots, t_{i,m_i+2}\}$: $sup_1(\bot_0) = 1$ und $\mathcal{E}^{R_1}_{\mathsf{used}} = \{y\}$ und $\mathcal{E}^{R_1}_{\mathsf{swap}} = \{o, X_{i_0}\}$.

Die folgende Region $R_2 = (sup_2, sig_2)$ löst (y, z) für alle relevanten $z \in Z(G_i) \setminus \{t_{i,2}, \ldots, t_{i,m_i+2}\}$: wenn $i = 0$, dann $sup_2(\bot_0) = 0$, sonst $sup_2(\bot_0) = 1$; außerdem $\mathcal{E}^{R_2}_{\mathsf{used}} = \{y\}$ und $\mathcal{E}^{R_2}_{\mathsf{swap}} = \{\ominus_i, \ominus_{i+1}, u_i\}$. Weil i beliebig war, folgt die Behauptung. □

Das folgenden Lemma komplettiert den Beweis von Satz 6.3.2:

Lemma 6.9. *Wenn es eine ungerade Menge mit höchstens κ Elementen für $(\mathfrak{U}, M)$ gibt, dann gibt es einen Zeugen d_τ-beschränkter τ-Regionen für die τ-EZTE und die τ-ZTE von A_τ.*

Beweis. Es ist klar, dass o und e für alle $e \in \ominus$ durch d_τ-beschränkte τ-Regionen lösbar sind. Wegen Fakt 6.18 und Fakt 6.19 gilt dies auch für k und y. Bezüglich der τ-EZTE von A_τ bleibt e noch für alle $e \in \{w\} \cup \mathfrak{U} \cup C \cup V \cup \oplus$ zu diskutieren. Es ist leicht zu sehen, dass alle Ereignisse aus $C \cup U$ passend lösbar sind. Für die anderen Ereignisse erhält man die entsprechenden Regionen analog zu denen von Fakt 6.15 für w, Fakt 6.16 für $\mathfrak{U}$ sowie Fakt 6.17 für $\oplus \cup V$. Dabei gilt es zu beachten, dass das zu lösende Ereignis, auf used statt auf inp abgebildet wird: Zum Beispiel kann man für die Region R_4 aus Fakt 6.15 einfach $sig(w) = \mathsf{inp}$ durch $sig(w) = \mathsf{used}$ ersetzen und erhält eine Region des (hier konstruierten) Transitionssystems A_τ, die (w, z) für alle $z \in \{t_{i,1}, \ldots, t_{i,m_i}\}$ löst. In anderen Fällen ist mitunter etwas mehr zu tun, aber die notwendigen Anpassungen liegen stets auf der Hand. Wie man darüber hinaus leicht sieht, bezeugen die Regionen, welche die τ-EZTE nachweisen, auch die τ-ZTE von A_τ. □

6.3 W[2]-Schwere durch Reduktion von Hitting Set

In diesem Abschnitt bestimmen wir eine untere Schranke der parametrisierten Komplexität von Bτ-SMBP und Bτ-RMBP für die Netztypen aus Abbildung 6.2 C6 - C9. Genauer gesagt werden wir argumentieren, dass Bτ-SMBP und Bτ-RMBP für diese Netztypen wenigstens W[2]-schwer sind. Der folgende Satz fasst die Ergebnisse zusammen:

Satz 6.4. Bτ-SMBP *und* Bτ-RMBP *sind* W[2]*-schwer, wenn*

1. $\tau \supseteq \{\mathsf{nop}, \mathsf{inp}, \mathsf{set}\}$ *oder* $\tau \supseteq \{\mathsf{nop}, \mathsf{out}, \mathsf{res}\}$,
2. $\tau = \{\mathsf{nop}, \mathsf{set}, \mathsf{res}\} \cup \omega, \{\mathsf{nop}, \mathsf{set}, \mathsf{res}, \mathsf{swap}\} \cup \omega$ *mit* $\omega \subseteq \{\mathsf{used}, \mathsf{free}\}$ *und* $\omega \neq \emptyset$,
3. $\tau = \{\mathsf{nop}, \mathsf{set}, \mathsf{swap}\} \cup \omega$ *oder* $\tau = \{\mathsf{nop}, \mathsf{out}, \mathsf{set}, \mathsf{swap}\} \cup \omega$ *oder* $\tau = \{\mathsf{nop}, \mathsf{res}, \mathsf{swap}\} \cup \omega$ *oder* $\tau = \{\mathsf{nop}, \mathsf{inp}, \mathsf{res}, \mathsf{swap}\} \cup \omega$ *und* $\emptyset \neq \omega \subseteq \{\mathsf{used}, \mathsf{free}\}$,
4. $\tau = \{\mathsf{nop}, \mathsf{inp}, \mathsf{res}, \mathsf{swap}\}$ *oder* $\tau = \{\mathsf{nop}, \mathsf{out}, \mathsf{set}, \mathsf{swap}\}$.

Der Beweis von Satz 6.4 basiert auf parametrisierten Reduktionen des allgemein als W[2]-vollständig bekannten Problems Hitting Set [CFK+15], das wie folgt definiert ist:

Hitting Set (HS)	
Eingabe:	Ein Tripel $(\mathfrak{U}, M, \kappa)$ mit einer endlichen Menge $\mathfrak{U}$, einer Menge $M = \{M_0, \ldots, M_{m-1}\}$ mit Teilmengen von $\mathfrak{U}$ und einer natürlichen Zahl κ.
Parameter:	κ
Frage:	Gibt es eine treffende Menge S für $(\mathfrak{U}, M)$, das heißt, $S \subseteq \mathfrak{U}$ und $S \cap M_i \neq \emptyset$ für alle $i \in \{0, \ldots, m-1\}$, die $\lvert S\rvert \leq \kappa$ erfüllt?

Beispiel 6.5. Die Eingabe $(\mathfrak{U}, M, 2)$ mit $\mathfrak{U} = \{X_0, X_1, X_2, X_3\}$ und $M = \{M_0, M_1, M_2, M_3\}$, wobei $M_0 = \{X_0, X_1\}$, $M_1 = \{X_1, X_2\}$, $M_2 = \{X_0, X_3\}$ und $M_3 = \{X_0, X_2, X_3\}$, erlaubt eine positive Entscheidung, denn $S = \{X_0, X_2\}$ definiert eine treffende Menge.

Die Netztypen jedes Aufzählungspunktes von Satz 6.4 benötigen jeweils eine eigene Reduktion. Alle diese Reduktionen folgen wieder einer gemeinsamen Strategie.

Die Strategie der Reduktionen von Hitting Set

Die Grundidee dieses Vorgehens folgt zunächst der in Abschnitt 6.2 vorgestellten Strategie, bei der Odd Set reduziert wurde. Wir reduzieren $(\mathfrak{U}, M, \kappa)$ zu einem Paar (A_τ, d_τ), sodass es genau dann einen Zeugen $\mathcal{R}$ d_τ-beschränkter Regionen für die τ-ZTE und die τ-EZTE von A_τ gibt, wenn es eine treffende Menge mit höchstens κ Elementen für $(\mathfrak{U}, M)$ gibt. Um dies zu erreichen, konstruieren wir erstens für jedes $i \in \{0, \ldots, m-1\}$ einen Pfad P_i, der die Menge $M_i = \{X_{i_0}, \ldots, X_{m_i-1}\}$ repräsentiert:

$$P_i = \quad z_{i,0} \xrightarrow{X_{i_0}} \cdots \xrightarrow{X_{i_{\ell-1}}} z_{i,i_\ell} \xrightarrow{X_{i_\ell}} z_{i,i_\ell+1} \xrightarrow{X_{i_{\ell+1}}} \cdots \xrightarrow{X_{i_{m_i-1}}} z_{i,i_{m_i}}$$

Zweitens konstruieren wir ein EZTP α, sodass eine d_τ-beschränkte τ-Region $R = (sup, sig)$, die α löst, $sup(z_{i,0}) \neq sup(z_{i,m_i})$ für alle $\in \{0, \ldots, m-1\}$ impliziert. So wird zugesichert, dass es ein Element $X \in \{X_{i_0}, \ldots, X_{m_i-1}\}$ mit $sig(X) \in \{\mathsf{inp}, \mathsf{out}, \mathsf{res}, \mathsf{set}, \mathsf{swap}\}$ geben muss, denn das Bild P_i^R von P_i unter R ist ein Pfad von 0 nach 1 oder ein Pfad von 1 nach 0 in τ. Die folgende Skizze zeigt die Situation für $sup(z_{i,0}) = 0$ und $sup(z_{i,m_i}) = 1$ und $sig(X_{i_\ell}) = sig(X_{i_\ell+1}) = \mathsf{set}$ und $sig(e) = \mathsf{nop}$ für alle $e \in M_i \setminus \{X_{i_\ell}, X_{i_\ell+1}\}$ (für beliebiges aber festes $i \in \{0, \ldots, m-1\}$), wobei der rote Bereich mit Zuständen positiven Support korrespondiert:

$$P_i^R = \underset{0}{sup(z_{i,0})} \xrightarrow[\mathsf{nop}]{sig(X_{i_0})} \cdots \xrightarrow[\mathsf{nop}]{sig(X_{i_{\ell-1}})} \underset{0}{sup(z_{i,i_\ell})} \xrightarrow[\mathsf{set}]{sig(X_{i_\ell})} \underset{1}{sup(z_{i,i_{\ell+1}})} \xrightarrow[\mathsf{set}]{sig(X_{i_{\ell+1}})} \cdots \xrightarrow[\mathsf{nop}]{sig(X_{i_{m_i-1}})} \underset{1}{sup(z_{i,i_{m_i}})}$$

Die Menge $S = \{X \in \mathfrak{U} \mid sig(X) \neq \mathsf{nop}\}$ erfüllt dann $S \cap M_i \neq \emptyset$ für alle $i \in \{0, \ldots, m-1\}$. Drittens stellen wir über die Eigenschaft $|\{e \in E \mid sig(e) \neq \mathsf{nop}\}| \leq d_\tau$ sicher, dass $|S| \leq \kappa$ erfüllt ist. Soweit gleicht der Ansatz jenem in Abschnitt 6.2. Der wesentliche Unterschied liegt nun in der Tatsache, dass die Signatur $sig(X)$ in τ potentiell gleichzeitig zustandsverändernd und zustandserhaltend sein kann, nämlich wenn $sig(X) \in \{\mathsf{set}, \mathsf{res}\}$. Zum Beispiel können in der oben skizzierten Situation beliebig viele Ereignisse aus $\{X_{i_0}, \ldots, X_{i_{m_i-1}}\}$ eine set-Signatur haben, solange es nur wenigstens eines davon gibt: es ist $0 \xrightarrow{\mathsf{set}} 1 \xrightarrow{\mathsf{set}} 1 \ldots \xrightarrow{\mathsf{set}} 1$ ein Pfad in τ. Das heißt, einerseits können wir –wie vorher für RIS oder OS– eine gesuchte Menge S über die Elemente von $\mathfrak{U}$ erhalten, deren Signaturen Zustandswechsel in τ verursachen (können). Andererseits ist die Menge $\{X \in M_i \mid sig(X) \in \{\mathsf{inp}, \mathsf{out}, \mathsf{res}, \mathsf{set}, \mathsf{swap}\}\}$ zunächst keinen weiteren Restriktionen unterworfen wie zum Beispiel einelementig (siehe RIS) oder ungerade (siehe OS) zu sein.

Die Eigenschaft, dass potentiell beliebig viele Ereignisse aus $\{X_{i_0}, \ldots, X_{i_{m_i-1}}\}$ eine set-Signatur haben können (oder res-Signatur, je nachdem), ist vor allem auch für die Rückrichtung des Beweises wesentlich. Die τ-Lösbarkeit von α durch eine d_τ-beschränkte τ-Region $R = (sup, sig)$, und damit die Existenz der oben skizzierten Pfade in τ, hängt wieder von der Existenz einer treffenden Menge $S \subseteq \mathfrak{U}$ ab, deren Elemente über ihre Signaturen die notwendigen Zustandswechsel in τ verursachen. Da set gleichzeitig zustandsverändernd (von 0 nach 1) und zustandserhaltend (von 1 nach 1) sein kann, muss S keine weiteren Bedingungen erfüllen, als $S \cap M_i \neq \emptyset$ für alle $i \in \{0, \ldots, m-1\}$. Es sind dann nämlich genau die Ereignisse aus S, die auf set abgebildet werden und zusichern, dass P_i^R ein Pfad von 0 nach 1 in τ ist. (Die Situation für res ist symmetrisch.) Diese Überlegungen verdeutlichen, wie sich die $\mathsf{W}[2]$-Schwere von Bτ-SMBP und Bτ-RMBP für die betreffenden Netztypen quasi in natürlicher Weise aus dem Vorhandensein von set (beziehungsweise res) ergibt. Insbesondere ist es nicht leicht zu sehen, wie dieses Vorgehen auf die Netztypen ohne set und res übertragen ließe, obwohl wir die $\mathsf{W}[2]$-Schwere deren Bτ-SMBP und Bτ-RMBP nicht ausschließen können. Die genaue Positionierung des Problems in XP ist eben eine bisher ungelöste Frage. Insgesamt stellt die Konstruktion von A_τ darüber hinaus sicher, dass es einen Zeugen $\mathcal{R}$ d_τ-beschränkter Regionen für

die τ-EZTE und die τ-ZTE von A_τ gibt, wenn es eine treffende Menge mit höchstens κ Elementen für $(\mathfrak{U}, M)$ gibt.

Der Beweis von Satz 6.4.1

Solange nicht ausdrücklich anders vereinbart, sei $\tau \supseteq \{\mathsf{nop}, \mathsf{inp}, \mathsf{set}\}$ im Rest dieses Abschnitts beliebig aber fest. Wir zeigen die Aussage von Satz 6.4.1 für τ. Aus Satz 6.1 folgt die Aussage dann auch für die Netztypen mit $\tau \supseteq \{\mathsf{nop}, \mathsf{out}, \mathsf{res}\}$.

Es folgt die Reduktion von (A_τ, d_τ) gemäß der eingangs angekündigten Strategie. Für den Parameter wird $d_\tau = \kappa + 2$ definiert. Das Transitionssystem A_τ hat zunächst die gleichen Gadgets wie das Transitionssystem für den Beweis von Satz 6.3.1, die wir der besseren Lesbarkeit wegen noch einmal kurz präsentieren: A_τ hat das folgende Gadget H, welches das angekündigte EZTP $\alpha = (k, h_2)$ liefert:

$$H = \quad h_0 \xrightarrow{k} h_1 \xrightarrow{w} h_2 \xrightarrow{o} h_3 \xrightarrow{k} h_4$$

Darüber hinaus hat A_τ für alle $i \in \{0, \ldots, m-1\}$ das folgende Gadget T_i, auf dem die Elemente der Menge $M_i = \{X_{i_0}, \ldots, M_{i_{m_i-1}}\}$ als Ereignisse stattfinden:

$$T_i = t_{i,0} \xrightarrow{k} t_{i,1} \xrightarrow{X_{i_0}} t_{i,2} \xrightarrow{X_{i_1}} t_{i,3} \xrightarrow{X_{i_2}} \cdots \xrightarrow{X_{i_{m_i-2}}} t_{i,m_i} \xrightarrow{X_{i_{m_i-1}}} t_{i,m_i+1} \xrightarrow{w} t_{i,m_i+2} \xrightarrow{k} t_{i,m_i+3}$$

Anders als für das Transitionssystem für den Beweis von Satz 6.3.1, ist die Verbindung der Gadgets $H, T_0, \ldots, T_{m-1}$ hier wesentlich einfacher. Insbesondere benötigen wir die Konstruktion der relevanten Pfade nicht. Letztere benötigten wir im betreffenden Abschnitt für die „Rückrichtung“, das heißt, für die Konstruktion eines Zeugen d_τ-beschränkter τ-Regionen für AA_τ. Die Besonderheiten der Interaktion set beziehungsweise res, je nachdem, erlauben es jedoch so einen Zeugen auch ohne die relevanten Pfade nachzuweisen, wenn es denn eine treffende Menge mit höchstens κ Elementen gibt. Wenn wir uns tatsächlich mit diesem Teil des Beweises der Konstruktion auseinandersetzen, das heißt in Fakt 6.22, dann werden auf diesen Aspekt noch einmal genauer eingehen. An dieser Stelle fügen wir jedenfalls für jedes $i \in \{0, \ldots, m-1\}$ die Kanten $\bot_i \xrightarrow{\ominus_{i+1}} \bot_{i+1}$ und $\bot_i \xrightarrow{u_i} t_{i,0}$ sowie die Kante $\bot_m \xrightarrow{u_m} h_0$ hinzu. Wir erhalten das Transitionssystem $A = (Z, E, \delta, \bot_0)$ mit Initialzustand $\bot_0$. Abbildung 6.10 zeigt das Transitionssystem A_τ welches aus Beispiel 6.5 resultiert.

Wir zeigen nun die angekündigte Funktionalität von A_τ.

Lemma 6.10. *Wenn es einen Zeugen $\mathcal{R}$ d_τ-beschränkter Regionen für die τ-EZTE von A_τ gibt, dann gibt es eine treffende Menge mit höchstens κ Elementen für $(\mathfrak{U}, M)$.*

Beweis. Nach Voraussetzung gibt es eine d_τ-beschränkte τ-Region von A_τ, die α löst. Sei $R = (sup, sig)$ eine solche Region. Weil R das EZTP (k, h_2) löst, gilt entweder $sig(k) \in \{\mathsf{inp}, \mathsf{used}\}$ und $sup(h_2) = 0$ oder $sig(k) \in \{\mathsf{out}, \mathsf{free}\}$ und $sup(h_2) = 1$. Wir werden im

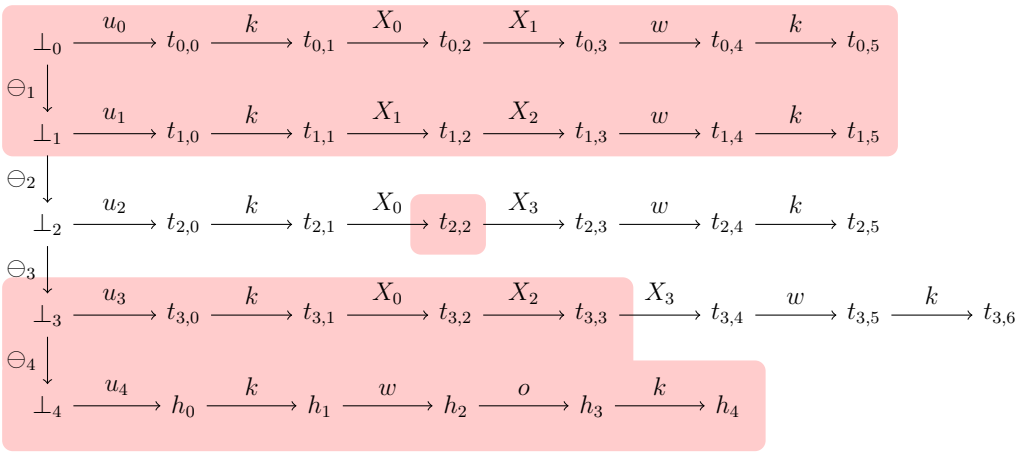


Abbildung 6.10: Das Transitionssystem A_τ der Konstruktion für den Beweis von Satz 6.4.1, welches aus Beispiel 6.5 resultiert. Der rote Bereich skizziert (die Zustände mit positivem Support der) Region R_1 aus Fakt 6.22 für $i = 2$ und $j = 1$.

Folgenden nur den ersten Fall diskutieren, denn der zweite ist zu diesem symmetrisch. Gelte also $sig(k) \in \{\mathsf{inp}, \mathsf{used}\}$ und $sup(h_2) = 0$.

Wir untersuchen zunächst den Fall $sig(k) = \mathsf{inp}$ und $sup(h_2) = 0$. Sei $i \in \{0, \ldots, m-1\}$ beliebig aber fest. Aus $sig(k) = \mathsf{inp}$ erhalten wir $sup(z) = 1$ und $sup(z') = 0$ für alle Kanten $z \xrightarrow{e} z'$ von A_τ. Mit $sup(h_1) = sup(h_2) = 0$ und $sup(h_3) = 1$ erhalten wir $sig(w) \in \{\mathsf{nop}, \mathsf{free}, \mathsf{res}\}$ und $sig(o) \neq \mathsf{nop}$. Mit $sup(t_{i,m_i+2}) = 1$ impliziert dies $sig(w) = \mathsf{nop}$ und daher $sup(t_{i,m_i+1}) = 1$. Insbesondere ist das Bild P_i^R des Pfades $P_i = t_{i,0} \xrightarrow{X_{i_0}} \ldots \xrightarrow{X_{i_{m_i-1}}} t_{i,m_i+1}$ ein Pfad von 0 nach 1 in τ. Deshalb ist $\{X \in M_i \mid sig(X) \in \{\mathsf{out}, \mathsf{set}, \mathsf{swap}\}\} \neq \emptyset$. Weil i beliebig war, gilt für die Menge $S = \{X \in \mathfrak{U} \mid sig(X) \neq \mathsf{nop}\}$ einerseits $S \cap M_i \neq \emptyset$ für alle $i \in \{0, \ldots, m-1\}$. Andererseits ist $|S| \leq \kappa$ erfüllt, denn $sig(k) \neq \mathsf{nop}$ und $sig(o) \neq \mathsf{nop}$ und R ist $\kappa + 2$-beschränkt.

Wir untersuchen wir nun den Fall $sig(k) = \mathsf{used}$ und $sup(h_2) = 0$. Sei $i \in \{0, \ldots, m-1\}$ beliebig aber fest. Aus $sig(k) = \mathsf{used}$ erhalten wir $sup(z) = sup(z') = 1$ für alle Kanten $z \xrightarrow{e} z'$ von A_τ. Mit $sup(h_1) = sup(h_3) = 1$ und $sup(h_2) = 0$ erhalten wir $sig(w) \in \{\mathsf{inp}, \mathsf{res}, \mathsf{swap}\}$ und $sig(o) \neq \mathsf{nop}$. Aus $sup(t_{i,m_i+2}) = 1$ und $sig(w) \in \{\mathsf{inp}, \mathsf{res}, \mathsf{swap}\}$ erhalten wir $sig(w) = \mathsf{swap}$ und daher $sup(t_{i,m_i+1}) = 0$. Wegen $sup(t_{i,1}) = 1$ ist das Bild P_i^R des Pfades P_i ein Pfad von 1 nach 0 in τ. Analog zur Diskussion des vorherigen Falles folgt aus der Beliebigkeit von i, dass $S = \{X \in \mathfrak{U} \mid sig(X) \neq \mathsf{nop}\}$ eine treffende Menge mit höchstens κ Elementen für $(\mathfrak{U}, M)$ ist. □

Für die andere Richtung werden wir nun argumentieren, dass (A_τ, d_τ) eine positive Entscheidung erlaubt, wenn es eine gesuchte treffende Menge für $(\mathfrak{U}, M)$ gibt. Solange nicht ausdrücklich anders vereinbart, sei S also im Folgenden eine beliebige aber feste treffende Menge mit höchstens κ Elementen für $(\mathfrak{U}, M)$.

Wir diskutieren zunächst sukzessive die τ-Lösbarkeit der Ereignisse von A_τ. Regionen $R = (sup, sig)$ werden mittels $sup(\bot_0)$ und sig definiert. Dabei kommt jede präsentierte Region mit den Signaturwerten nop und inp sowie set aus. Die Menge $\mathcal{E}^R_{e_\tau} = \{e \in E(A_\tau)\}$ fasst Ereignisse gleicher Signatur zusammen. Darüber hinaus gilt stets $E^R_{\mathsf{nop}} = E(A_\tau) \setminus (E^R_{\mathsf{inp}} \cup \mathcal{E}^R_{\mathsf{set}})$, sodass wir die Menge $\mathcal{E}^R_{\mathsf{nop}}$ jeweils nicht explizit definieren.

Fakt 6.20. *Das Ereignis k ist durch d_τ-beschränkte τ-Regionen lösbar.*

Beweis. Wir benötigen die treffende Menge S von $(\mathfrak{U}, M)$, die $|S| \leq \kappa$ erfüllt.

Die folgende Region $R_1 = (sup_1, sig_1)$ löst (k, z) für alle $z \in \{h_2\} \cup \{z \in Z \mid \xrightarrow{k} z\}$: $sup_1(\bot_0) = 1$ und $\mathcal{E}^{R_1}_{\mathsf{inp}} = \{k\}$ und $\mathcal{E}^{R_1}_{\mathsf{set}} = \{o\} \cup S$. Offenbar ist R_1 d_τ-beschränkt, weil $|S| \leq \kappa$.

Die folgende Region $R_2 = (sup_2, sig_2)$ löst (k, z) für alle relevanten $z \in \bigcup_{i=0}^{m-2} Z(T_i)$: $sup_2(\bot_0) = 1$ und $\mathcal{E}^{R_2}_{\mathsf{inp}} = \{k\}$ und $\mathcal{E}^{R_2}_{\mathsf{set}} = \{w\} \cup S$.

Es ist klar, dass (k, z) für alle $z \in \bot$ durch d_τ-beschränkte τ-Regionen lösbar ist. Das beweist den Fakt. □

Fakt 6.21. *Das Ereignis w ist durch d_τ-beschränkte τ-Regionen lösbar.*

Beweis. Sei $i \in \{0, \ldots, m-1\}$ beliebig aber fest. Die folgende Region $R = (sup, sig)$ löst (w, z) für alle $z \in \{\bot_i, t_{i,0}, \ldots, t_{i,m_i}\}$ und alle $z \in \{h_2, h_3, h_4\}$: Wenn $i = 0$, dann $sup(\bot_0) = 0$, sonst $sup(\bot_0) = 1$; außerdem $\mathcal{E}^R_{\mathsf{inp}} = \{w, \ominus_i\}$ und $\mathcal{E}^R_{\mathsf{set}} = \{\ominus_{i+1}, X_{i_{m_i-1}}\}$.

Wie man leicht sieht, ist (w, z) für alle $z \in \{t_{i,m_i+2}, t_{i,m_i+3}\}$ und alle $z \in \{\bot_m, h_0\}$ durch d_τ-beschränkte τ-Regionen lösbar. Aus der Beliebigkeit von i folgt der Fakt. □

Fakt 6.22. *Wenn $X \in \mathfrak{U}$, dann ist X durch d_τ-beschränkte τ-Regionen lösbar.*

Beweis. Seien $i \in \{0, \ldots, m-1\}$ und $j \in \{0, \ldots, m_i - 1\}$ beliebig aber fest.

Die folgende Region $R_1 = (sup_1, sig_1)$, löst (X_{i_j}, z) für alle $z \in \{\bot_i, t_{i,0}, \ldots, t_{i,j}\}$: Wenn $i = 0$, dann $sup_1(\bot_0) = 0$, sonst $sup_1(\bot_0) = 1$; für alle $e \in E(A_\tau)$, wenn $e \in \{X_{i_j}, \ominus_i\}$, dann $sig_1(e) = \mathsf{inp}$; wenn $j = 0$ und $e = k$ oder $j > 0$ und $e = X_{i_{j-1}}$ dann $sig_1(e) = \mathsf{set}$; wenn $e = \ominus_{i+1}$, dann $sig_1(e) = \mathsf{set}$, sonst $sig_1(e) = \mathsf{nop}$.

An dieser Stelle wird deutlich, dass wir wegen der Möglichkeit, $X_{i_{j-1}}$ auf set abzubilden, die Konstruktion der relevanten Pfade nicht benötigen: Wenn $X_{i_{j-1}}$ und X_{i_j} beide in einem T_ℓ mit $i \neq \ell$ stattfinden, dann findet $X_{i_{j-1}}$ wegen $i_{j-1} < i_j$ „vor“ X_{i_j} statt. Nun gilt nach Definition von R_1 war $sup_1(z) = 1$ für den eindeutigen Ursprung des Ereignisses $X_{i_{j-1}}$ in T_ℓ, aber set erzwingt –im Gegensatz zu swap– eben keinen Zustandswechsel. Deshalb gilt $sup_1(z') = 1$ für den eindeutigen Ursprung von X_{i_j} in T_ℓ, sodass R_1 wohldefiniert ist.

Abbildung 6.10 skizziert R_1 für X_{2_1}, das heißt $i = 2$ und $j = 1$, wobei der rote Bereich mit den Zuständen korrespondiert, die einen positiven Support haben. Es findet X_0 auch in T_3 „vor“ X_3 statt.

Wie man leicht sieht, ist (X_{i_j}, z) für alle $z \in \{t_{i,j+2}, \ldots, t_{i,m_i+3}\}$ und alle $z \in \{\bot_m\} \cup Z(H)$ durch d-beschränkte τ-Regionen lösbar. Es waren i, j beliebig, so folgt der Fakt. □

Lemma 6.11. *Wenn es eine treffende Menge mit höchstens κ Elementen für $(\mathfrak{U}, M)$ gibt, dann gibt es einen Zeugen d_τ-beschränkter Regionen für die τ-EZTE und die τ-ZTE von A_τ.*

Beweis. Wie man leicht sieht, ist e für alle $e \in \ominus \cup \{o\} \cup U$ durch d_τ-beschränkte τ-Regionen lösbar. Die Aussage für die τ-EZTE folgt dann aus den Fakt 6.20 bis Fakt 6.22. Ebenso ist leicht zu sehen, alle ZTP von A_τ durch d_τ-beschränkte τ-Regionen lösbar sind. Die beweist das Lemma. □

Der Beweis von Satz 6.4.2

Solange nicht ausdrücklich anders vereinbart, sei τ im Folgenden ein beliebiger aber fester Netztyp gemäß Satz 6.4.2. Zu Beginn definieren wir für den Parameter $d_\tau = \kappa + 2$. Das Transitionssystem A_τ hat zunächst das folgende Gadget H, welches das EZTP $\alpha = (k, h_1)$ zu Verfügung stellt:

$$H = \quad \overset{k}{\circlearrowright}\ h_0 \xrightarrow{\ o\ } h_1 \ \overset{o}{\circlearrowright}$$

Darüber hinaus hat das Transitionssystem A_τ für alle $i \in \{0, \ldots, m-1\}$ das folgende Gadget T_i, welches die Menge $M_i = \{X_{i_0}, \ldots, X_{i_{m_i-1}}\}$ repräsentiert, indem ihre Elemente als Ereignisse verwendet werden:

$$T_i = \quad \overset{k}{\circlearrowright}\ t_{i,0} \xrightarrow{X_{i_0}} \overset{X_{i_0}}{\circlearrowright}\ t_{i,1} \xrightarrow{X_{i_1}} \overset{X_{i_1}}{\circlearrowright}\ t_{i,2} \xrightarrow{X_{i_2}} \overset{X_{i_2}}{\circlearrowright} \quad \cdots \quad \xrightarrow{X_{i_{m_i-2}}} \overset{X_{i_{m_i-2}}}{\circlearrowright}\ t_{i,m_i-1} \xrightarrow{X_{i_{m_i-1}}} \overset{X_{i_{m_i-1}}}{\circlearrowright}\ t_{i,m_i} \ \underset{o}{\circlearrowright}$$

Schließlich werden die Gadgets $T_0, \ldots, T_{m-1}$ und H mittels weiterer Kanten miteinander verbunden. Dazu fügen wir zunächst für alle $i \in \{0, \ldots, m-1\}$ die Kanten $\bot_i \xrightarrow{\ominus_{i+1}} \bot_{i+1}$ und $\bot_{i+1} \xrightarrow{\ominus_{i+1}} \bot_{i+1}$ hinzu. Ebenso fügen wir für alle $i \in \{0, \ldots, m-1\}$ die Kanten $\bot_i \xrightarrow{u_i} t_{i,0}$ und $t_{i,0} \xrightarrow{u_i} t_{i,0}$ hinzu. Schließlich werden noch die Kanten $\bot_m \xrightarrow{u_m} h_0$ und $h_0 \xrightarrow{u_m} h_0$ hinzugefügt. Das resultierende Transitionssystem mit Initialzustand $\bot_0$ ist A_τ. Abbildung 6.11 zeigt die konkrete Instanz A_τ, die sich aus der Eingabe für HS aus Beispiel 6.5 ergibt.

Wir beweisen nun die angekündigte Funktionalität von (A_τ, d_τ). Gemäß des folgenden Lemmas, erlaubt $(\mathfrak{U}, M)$ eine positive Entscheidung, wenn es einen Zeugen d_τ-beschränkter Regionen für die τ-EZTE von A_τ gibt:

Fakt 6.23. *Wenn es einen Zeugen $\mathcal{R}$ d_τ-beschränkter Regionen für die τ-EZTE von A_τ gibt, dann gibt es eine treffende Menge mit höchstens κ Elementen für $(\mathfrak{U}, M)$.*

Beweis. Weil es einen Zeugen $\mathcal{R}$ d_τ-beschränkter Regionen für die τ-EZTE von A_τ gibt, gibt es eine d_τ-beschränkte τ-Region, die $\alpha = (k, h_1)$ löst. Sei $R = (sup, sig)$ eine solche $\kappa + 2$-beschränkte Region. Weil R das EZTP α löst, gilt entweder $sig(k) = \mathsf{used}$ und $sup(h_1) = 0$ oder $sig(k) = \mathsf{free}$ und $sup(h_1) = 1$. Wir werden im Folgenden nur den ersten Fall diskutieren, denn der zweite ist dazu symmetrisch. Des Weiteren beachte man, dass jedes Ereignis e von A_τ an wenigstens einer Schleife $z \xrightarrow{e} z$ stattfindet. Für diese gilt offensichtlich, dass der Support von Ursprung und Ziel identisch sind. Deshalb gilt $sig(e) \neq \mathsf{swap}$ für alle Ereignisse e von A_τ. Sei also $sig(k) = \mathsf{used}$ und $sup(h_1) = 0$ und sei $i \in \{0, \ldots, m-1\}$ beliebig aber fest. Es impliziert $sig(k) = \mathsf{used}$ sofort $sup(z) = sup(z') = 1$ für jede Kante $z \xrightarrow{k} z'$ von A_τ. Wegen $sup(h_0) = 1$ und $sup(h_1) = 0$, erhalten wir $sig(o) = \mathsf{res}$ und damit $sup(z') = 0$ für jede Kante $z \xrightarrow{o} z'$ von A_τ. Ingesamt erhalten wir offenbar $sup(t_{i,0}) = 1$ und $sup(t_{i,m_i}) = 0$. Deshalb ist das Bild P_i^R des Pfades $t_{i,0} \xrightarrow{X_{i_0}} \ldots \xrightarrow{X_{i_{m_i-1}}} t_{i,m_i}$ ein Pfad von 1 nach 0 in τ. Es gibt also ein $X \in \{X_{i_0}, \ldots, X_{i_{m_i-1}}\}$ dessen Signatur den Zustandswechsel von 1 nach 0 verursacht, das heißt $sig(X) = \mathsf{res}$. Wegen der Beliebigkeit von i und $sig(k) \neq \mathsf{nop}$ sowie $sig(o) \neq \mathsf{nop}$ und weil R $\kappa+2$-beschränkt ist, folgt, dass die

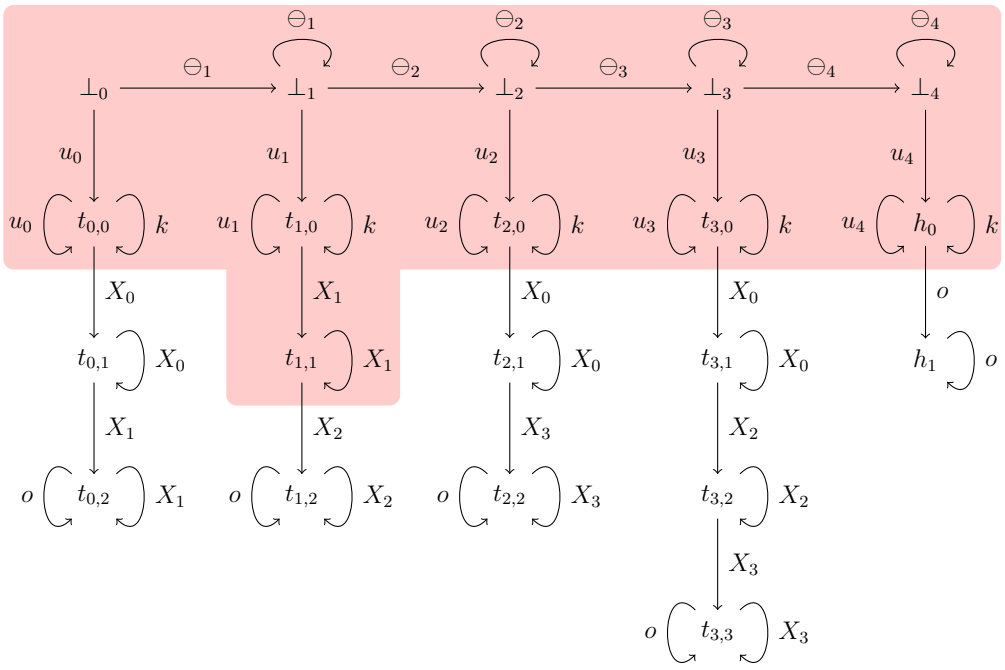


Abbildung 6.11: Das Transitionssystem A_τ der Konstruktion für Satz 6.4.2, das aus Beispiel 6.5 resultiert. Der rote Bereich skizziert die Region R_1 aus dem Beweis von Fakt 6.24, die das EZTP $\alpha = (k, h_1)$ löst und auf dem Hitting Set $S = \{X_0, X_2\}$ basiert. Dabei korrespondiert der rote Bereich mit den Zuständen, die den Supportwert 1 haben.

Menge $S = \{X \in \mathfrak{U} \mid sig(X) \neq \mathsf{nop}\}$ eine treffende Menge mit $|S| \leq \kappa$ für $(\mathfrak{U}, M)$ ist. □

Für die entgegengesetzte Richtung beweisen wir, dass alle EZTP und ZTP von A_τ durch d_τ-beschränkte τ-Regionen lösbar sind, wenn es eine treffende Menge mit höchstens κ Elementen für $(\mathfrak{U}, M)$ gibt. Sei also im Folgenden S eine beliebige aber feste solche treffenden Menge für $(\mathfrak{U}, M)$. Wir diskutieren die Ereignisse von A_τ wieder individuell. Darüber hinaus gehen wir im Weiteren jeweils davon aus, dass $\mathsf{used} \in \tau$. Alle Regionen kommen mit den Signaturwerten nop, set, res und used aus. Wenn möglich, wird jede lösende Region $R = (sup, sig)$ implizit durch $sup(\bot_0)$ und sig definiert, wobei $\mathcal{E}^R_{e_\tau} = \{e \in E(A_\tau) \mid sig(e) = e_\tau\}$ Ereignisse gleicher Signatur zusammenfasst. Insbesondere gilt stets $\mathcal{E}^R_{\mathsf{nop}} = E(A_\tau) \setminus (\mathcal{E}^R_{\mathsf{set}} \cup \mathcal{E}^R_{\mathsf{res}} \cup \mathcal{E}^R_{\mathsf{used}})$, sodass wir die Menge $\mathcal{E}^R_{\mathsf{nop}}$ jeweils nicht explizit definieren.

Die diskutierten Regionen zeigen auch die Lösbarkeit der Trennungsprobleme für den Fall, dass $\mathsf{used} \notin \tau$, denn dieser impliziert $\mathsf{free} \in \tau$: eine zu R passende Region $R' = (Z(A_\tau) \setminus sup, sig')$, welche dieselben Trennungsprobleme löst, erhält man für alle $e \in E(A_\tau)$ durch $sig'(e) = \mathsf{set}$, wenn $sig(e) = \mathsf{res}$ und $sig'(e) = \mathsf{res}$, wenn $sig(e) = \mathsf{set}$ und $sig'(e) = \mathsf{free}$, wenn $sig(e) = \mathsf{used}$ und andernfalls $sig(e) = \mathsf{nop}$.

Fakt 6.24. *Das Ereignis k ist durch d_τ-beschränkte τ-Regionen lösbar.*

Beweis. Wir benötigen die treffende Menge S mit höchstens κ Elementen für $(\mathfrak{U}, M)$.

Die folgende Region $R_1 = (sup_1, sig_1)$ löst α: $sup_1(\bot_0) = 1$ und $\mathcal{E}^{R_1}_{\mathsf{used}} = \{k\}$ und $\mathcal{E}^{R_1}_{\mathsf{res}} = \{o\} \cup S$. Weil $|S| \leq \kappa$, ist R_1 $\kappa + 2$-beschränkt.

Sei $i \in \{0, \ldots, m-1\}$ beliebig aber fest. Die folgende Region $R_2 = (sup_2, sig_2)$ löst (k, z) für alle $z \in \{\bot_i, t_{i,1}, \ldots, t_{i,m_i}\}$: wenn $i = 0$, dann $sup_2(\bot_0) = 0$, sonst $sup_2(\bot_0) = 1$; außerdem $\mathcal{E}^{R_2}_{\mathsf{used}} = \{k\}$ und $\mathcal{E}^{R_2}_{\mathsf{res}} = \{\ominus_i, X_{i_0}\}$ und $\mathcal{E}^{R_2}_{\mathsf{set}} = \{u_i, \ominus_{i+1}\}$.

Es ist klar, dass $(k, \bot_m)$ passend lösbar ist und aus der Beliebigkeit von i folgt die passende τ-Lösbarkeit von k. □

Fakt 6.25. *Das Ereignis o ist durch d_τ-beschränkte τ-Regionen lösbar.*

Beweis. Sei $i \in \{0, \ldots, m-1\}$ beliebig aber fest. Die folgende Region $R_3 = (sup_3, sig_3)$ löst (o, z) für alle $z \in \{\bot_i, t_{i,0}, \ldots, t_{i,m_i-1}\}$: wenn $i = 0$, dann $sup_3(\bot_0) = 0$, sonst $sup_3(\bot_0) = 1$; außerdem $\mathcal{E}^{R_3}_{\mathsf{used}} = \{o\}$ und $\mathcal{E}^{R_3}_{\mathsf{res}} = \{\ominus_i\}$ und $\mathcal{E}^{R_3}_{\mathsf{set}} = \{\ominus_{i+1}, X_{i_{m_i-1}}\}$.

Es ist klar, dass $(o, \bot_m)$ passend lösbar ist. Aus der Beliebigkeit von i folgt damit die τ-Lösbarkeit von o durch $\kappa + 2$-beschränkte τ-Regionen. □

Fakt 6.26. *Wenn $X \in \mathfrak{U}$, dann ist X durch d_τ-beschränkte τ-Regionen lösbar.*

Beweis. Seien $i \in \{0, \ldots, m-1\}$ und $j \in \{0, \ldots, m_i - 1\}$ beliebig aber fest. Wenn $j < m_i - 1$, dann löst die folgende Region $R_4 = (sup_4, sig_4)$ das EZTP (X_{i_j}, z) für alle $z \in \{t_{i,j+2}, \ldots, t_{i,m_i}\}$: $sup_4(\bot_0) = 1$ und $\mathcal{E}^{R_4}_{\mathsf{used}} = \{X_{i_j}\}$ und $\mathcal{E}^{R_4}_{\mathsf{res}} = \{X_{i_{j+1}}\}$. Man beachte, dass $\ell_0 < \ldots \ell_{m_\ell - 1}$ für jede Menge $M_\ell = \{X_{\ell_0}, \ldots, X_{\ell_{m_\ell-1}}\}$ von M gilt. Deshalb ist R_4 wohldefiniert.

Wenn $j \geq 1$, dann löst die folgende Region $R_5 = (sup_5, sig_5)$ das EZTP (X_{i_j}, z) für alle $z \in \{t_{i,0}, \ldots, t_{i,j-1}\}$: wenn $i = 0$, dann $sup_5(\bot_0) = 0$, sonst $sup_5(\bot_0) = 1$; außerdem $\mathcal{E}^{R_5}_{\mathsf{used}} = \{X_{i_j}\}$ und $\mathcal{E}^{R_5}_{\mathsf{res}} = \{\ominus_i\}$ und $\mathcal{E}^{R_5}_{\mathsf{set}} = \{\ominus_{i+1}, X_{i_{j-1}}\}$.

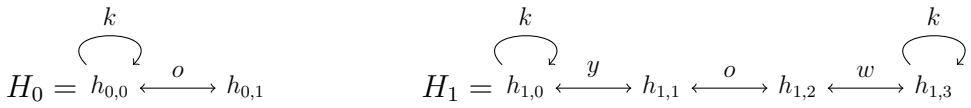
Es ist klar, dass $(X_{i_0}, \bot_i)$ durch eine $\kappa + 2$ beschränkte τ-Region lösbar ist. Aus der Beliebigkeit von i und j folgt die τ-Lösbarkeit für alle $X \in \mathfrak{U}$. □

Man sieht leicht, dass alle weiteren EZTP –das heißt jene, an denen Ereignisse aus $\ominus$ und $\{u_0, \ldots, u_m\}$ beteiligt sind– durch geeignete d_τ-beschränkte Regionen lösbar sind. Ebenso ist leicht zu sehen, dass die ZTP von A_τ entweder bereits durch die diskutierten Regionen gelöst sind oder entsprechende Regionen definiert werden können. Wir werden die zugehörigen Regionen deshalb nicht explizit diskutieren. Insgesamt erhalten wir das folgende Lemma, welches den Beweis von Satz 6.4.2 komplettiert:

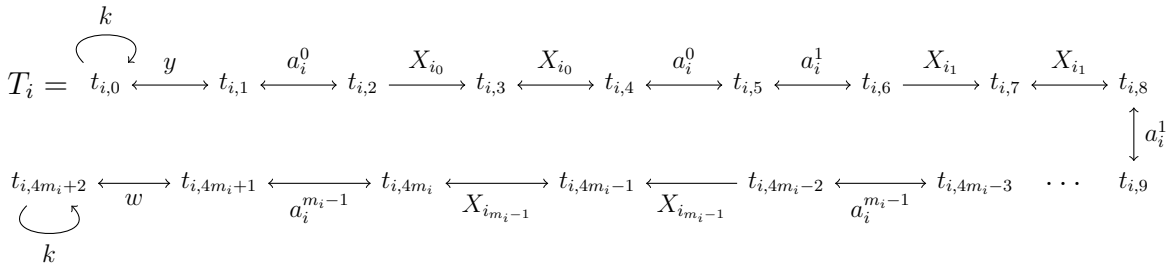
Lemma 6.12. *Wenn es eine treffende Menge mit höchstens κ Elementen für $(\mathfrak{U}, M)$ gibt, dann gibt es einen Zeugen d_τ-beschränkter Regionen für die τ-EZTE und die τ-ZTE von A_τ.*

Der Beweis von Satz 6.4.3

Wir präsentieren eine Reduktion von HS unter der Annahme, dass $\tau = \{\mathsf{nop}, \mathsf{set}, \mathsf{swap}\} \cup \omega$ oder $\tau = \{\mathsf{nop}, \mathsf{out}, \mathsf{set}, \mathsf{swap}\} \cup \omega$ und $\emptyset \neq \omega \subseteq \{\mathsf{used}, \mathsf{free}\}$. Die Aussage für die anderen Netztypen von Satz 6.4.3 folgt dann mittels Kongruenz aus Satz 6.1. Zu Beginn definieren wir für den Parameter $d_\tau = \kappa + 3$. Das Transitionssystem A_τ hat zunächst die beiden folgenden Gadgets H_0 und H_1, wobei insbesondere H_0 das EZTP $\alpha = (k, h_{0,1})$ zur Verfügung stellt:



Darüber hinaus hat das Transitionssystem A_τ für jedes $i \in \{0, \ldots, m-1\}$ das folgende Gadget T_i, auf dem die Elemente der Menge $M_i = \{X_{i_0}, \ldots, X_{i_{m_i-1}}\}$ als Ereignisse stattfinden:



Man beachte, dass jedes Ereignis $e \in M_i$ stets auf einer einfachen Kante gefolgt von einer bi-direktionalen Kante stattfindet: $z \xrightarrow{e} z' \xleftarrow{e} z''$. Darüber hinaus ist das Ereignis

a_i^j, welches das Auftreten von X_{i_j} in T_i umschließt, für alle $j \in \{0, \ldots, m_i - 1\}$ an das Gadget T_i gebunden. Wenn also X_{i_j} in einem anderen Gadget T_ℓ ebenso stattfindet, sagen wir als X_{ℓ_n} dann wird es dort von a_ℓ^n umschlossen und letzteres ist verschieden von a_i^j.

Um die eingeführten Gadgets zu verbinden, wird für jedes $i \in \{0, \ldots, m\}$ die bidirektionale Kante $\perp_i \overset{\ominus_{i+1}}{\longleftrightarrow} \perp_{i+1}$ hinzugefügt. Außerdem wird wird für jedes $i \in \{0, \ldots, m-1\}$ die Kante $\perp_i \overset{u_i}{\longleftrightarrow} t_{i,0}$ hinzugefügt sowie die Kanten $\perp_m \overset{u_m}{\longleftrightarrow} h_{0,0}$ und $\perp_{m+1} \overset{u_{m+1}}{\longleftrightarrow} h_{1,0}$. Das resultierende Transitionssystem A_τ hat den Initialzustand $\perp_0$. Abbildung 6.12 zeigt das Transitionssystem A_τ, das aus (der Eingabe für HS aus) Beispiel 6.5 resultiert.

Lemma 6.13. *Wenn es einen Zeugen $\mathcal{R}$ d_τ-beschränkter Regionen für die τ-EZTE von*

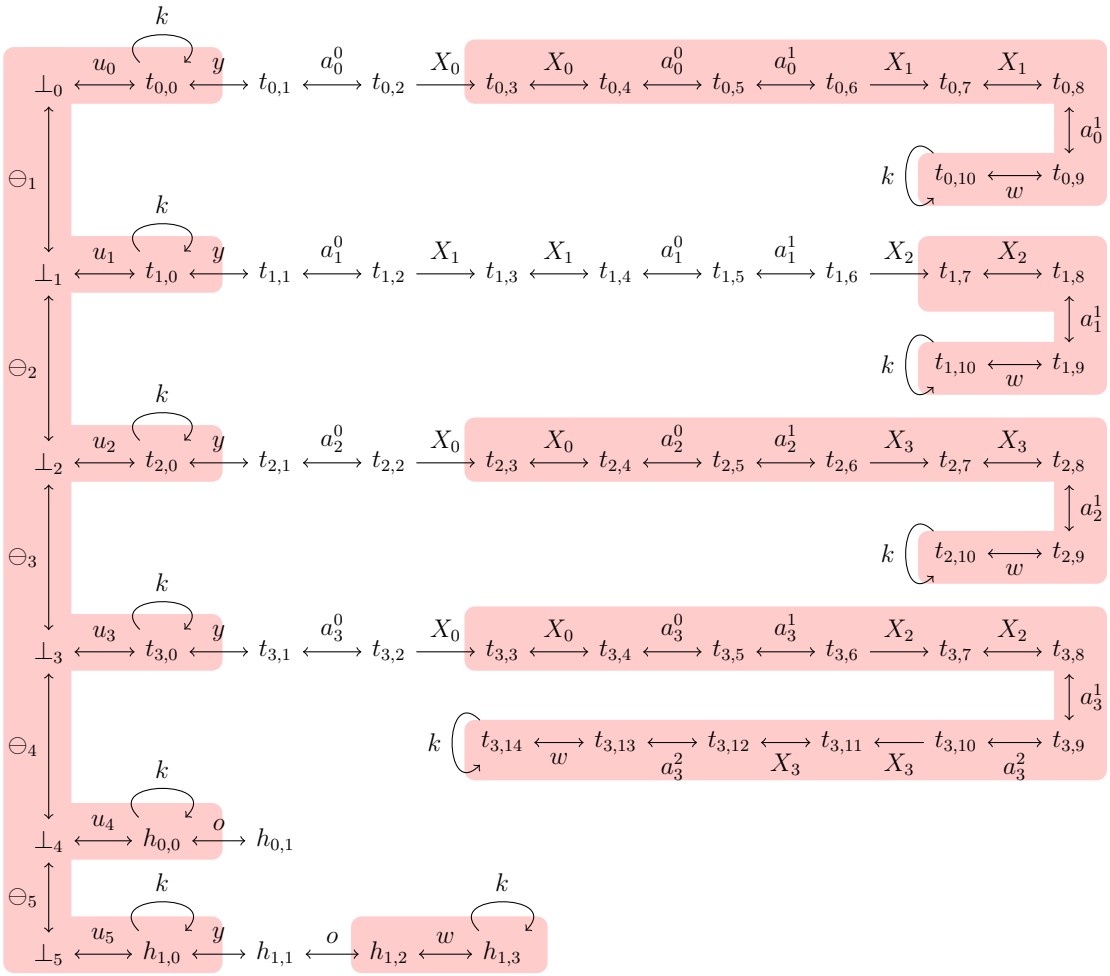


Abbildung 6.12: Das Transitionssystem A_τ der Reduktion für Satz 6.4.3, das aus Beispiel 6.5 resultiert. Der rote Bereich skizziert (die Zustände mit positivem Support der) Region R_1 aus Fakt 6.27, die $\alpha = (k, h_{0,1})$ löst.

A_τ gibt, dann gibt es ein Hitting Set mit höchstens κ Elementen für $(\mathfrak{U}, M)$.

Beweis. Sei $R = (sup, sig)$ eine d_τ-beschränkte τ-Region, die $\alpha = (k, h_{0,1})$ löst, das heißt, entweder $sig(k) = \mathsf{used}$ und $sup(h_{0,1}) = 0$ oder $sig(k) = \mathsf{free}$ und $sup(h_{0,1}) = 1$. Im Folgenden werden wir nur den ersten Fall explizit diskutieren, denn der zweite ist zu diesem symmetrisch. Sei also $sig(k) = \mathsf{used}$ und $sup(h_{0,1}) = 0$. Dies impliziert $sup(z) = sup(z') = 1$ für jede Kante $z \xrightarrow{k} z'$ von A_τ. Aus $sup(h_{0,1}) = 0$ und $sup(h_{0,0}) = 0$ schließen wir $sig(o) = \mathsf{swap}$. Darüber hinaus folgt aus $sup(h_{1,0}) = sup(h_{1,3}) = 1$, dass das Bild P^R des Pfades $P = h_{1,0} \xrightarrow{y} \dots \xrightarrow{w} h_{1,3}$ ein Pfad von 1 nach 1 in τ ist. Daher muss die Anzahl der Zustandswechsel zwischen 0 und 1 auf diesem Pfad P^R gerade sein. Man beachte, dass für bi-direktionale Kanten $z \xleftrightarrow{e} z'$ genau dann $sup(z) \neq sup(z')$ gilt, wenn $sig(e) = \mathsf{swap}$. Dies bedeutet insbesondere, dass die Zahl $|\{e \in \{y, o, w\} \mid sig(e) = \mathsf{swap}\}|$ gerade sein muss. Weil $sig(o) = \mathsf{swap}$, gibt es somit genau ein Ereignis $e \in \{y, w\}$, sodass $sig(e) = \mathsf{swap}$. Im Folgenden argumentieren wir nur für den Fall $e = y$, der $sig(w) \neq \mathsf{swap}$ impliziert, denn der Fall $e = w$ ist dazu ähnlich.

Sei $i \in \{0, \dots, m-1\}$ beliebig aber fest. Aus $sup(t_{i,0}) = sup(t_{i,4m_i+2}) = 1$ und $sig(y) = \mathsf{swap}$ sowie $sig(w) \neq \mathsf{swap}$ erhalten wir $sup(t_{i,1}) = 0$ und $sup(t_{i,4m_i+1}) = 1$. Insbesondere ist das Bild P_i^R des Pfades $P_i = t_{i,1} \xrightarrow{a_i^0} \dots \xrightarrow{a_i^{m_i-1}} t_{i,4m_i+1}$ ein Pfad von 0 nach 1 in τ. Wir werden nun argumentieren, dass dies $sig(X) \neq \mathsf{nop}$ für wenigstens ein Ereignis $X \in \{X_{i_0}, \dots, X_{i_{m_i-1}}\}$ impliziert: Angenommen es gilt $sig(X) = \mathsf{nop}$ für alle $X \in \{X_{i_0}, \dots, X_{i_{m_i-1}}\}$. Sei $j \in \{0, \dots, m_i - 1\}$ beliebig aber fest. Es findet a_i^j nur an bi-direktionalen Kanten statt. Wenn $sig(a_i^j) \neq \mathsf{swap}$, dann erhalten wir deshalb mit $sig(X_{i_j}) = \mathsf{nop}$, dass $sup(t_{i,4j+1}) = sup(t_{i,4j+2}) = sup(t_{i,4j+3}) = sup(t_{i,4j+4})$. Andernfalls erhalten wir $sup(t_{i,4j+1}) \neq sup(t_{i,4j+2}) = sup(t_{i,4j+3}) \neq sup(t_{i,4j+4})$, wegen $sig(a_i^j) = \mathsf{swap}$. Insbesondere gilt in jedem Fall $sup(t_{i,4j+1}) = sup(t_{i,4j+4})$. Weil j beliebig war, impliziert dies den Widerspruch $sup(t_{i,1}) = sup(t_{i,4m_i+1})$. Mithin war unsere Annahme falsch und es folgt $\{X \in M_i \mid sig(X) \neq \mathsf{nop}\} \neq \emptyset$. Weil i beliebig war, gilt $S \cap M_i$ für die Menge $S = \{X \in \mathfrak{U} \mid sig(X) \neq \mathsf{nop}\}$ für alle $i \in \{0, \dots, m-1\}$. Darüber hinaus gilt $|S| \leq \kappa$, denn $sig(e) \neq \mathsf{nop}$ ist wahr für alle $e \in \{k, o, y\}$ und R ist $\kappa + 3$-beschränkt. □

Solange nicht ausdrücklich anders vereinbart sei S im Folgenden eine beliebige aber feste treffende Menge für $(\mathfrak{U}, M)$. Für die Rückrichtung zeigen wir zunächst die τ-Lösbarkeit der Ereignisse von A_τ durch d_τ-beschränkte τ-Regionen. Die präsentierten Regionen implizieren dann bereits die Lösung der meisten ZTP von A_τ. Dabei beschränken wir uns im Folgenden auf den Fall $\mathsf{used} \in \tau$, denn die Regionen beziehungsweise Argumente für den Fall $\mathsf{used} \notin \tau$, der $\mathsf{free} \in \tau$ impliziert, sind dazu symmetrisch. Darüber hinaus ist für alle EZTP (e, z) von A_τ leicht zu sehen, dass diese (wegen $\mathsf{used} \in \tau$) stets passend τ-lösbar sind, wenn e und z nicht im gleichen Gadget stattfinden. Wir werden uns deshalb im Folgenden nur EZTP diskutieren, deren Ereignis und Zustand im gleichen Gadget stattfinden. Der Bequemlichkeit halber definieren wir $\perp = \{\perp_0, \dots, \perp_{m+1}\}$. Darüber hinaus definieren wir Regionen $R = (sup, sig)$ implizit mittels $sup(\perp)$ und sig und fassen

Ereignisse gleicher Signatur in einer Menge $\mathcal{E}^R_{e_\tau} = \{e \in E(A_\tau) \mid sig(e) = e_\tau\}$. Die Menge E^R_{nop} enthält alle Ereignisse von A_τ, die nicht explizit einer anderen Signatur zugeordnet worden sind und wird deshalb nicht explizit definiert.

Fakt 6.27. *Das Ereignis k ist durch d_τ-beschränkte τ-Regionen lösbar.*

Beweis. Wir benötigen die treffende Menge S mit κ Elementen für $(\mathfrak{U}, M)$.

Die folgende Region $R_1 = (sup_1, sig_1)$ löst α: $sup_1(\perp_0) = 1$ und $\mathcal{E}^{R_1}_{\mathsf{used}} = \{k\}$ und $\mathcal{E}^{R_1}_{\mathsf{swap}} = \{o, y\}$ und $\mathcal{E}^{R_1}_{\mathsf{set}} = S$. R_1 ist $\kappa + 3$-beschränkt, denn es gilt $|S| \leq \kappa$.

Die folgende Region $R_2 = (sup_2, sig_2)$ löst (k, z) für alle $z \in \bigcup_{i=0}^{m-1}\{t_{i,0}, \ldots, t_{i,4m_i+1}\}$ und alle $z \in \{h_{1,1}, h_{1,2}\}$: $sup_2(\perp_0) = 1$ und $\mathcal{E}^{R_2}_{\mathsf{used}} = \{k\}$ und $\mathcal{E}^{R_2}_{\mathsf{swap}} = \{y, w\}$.

Wir man leicht sieht, kann (k, z) passend für alle $z \in \perp$ gelöst werden kann. □

Fakt 6.28. *Das Ereignis y ist durch d_τ-beschränkte τ-Regionen lösbar.*

Beweis. Sei $i \in \{0, \ldots, m-1\}$ beliebig aber fest.

Die folgende Region $R_3 = (sup_3, sig_3)$ löst (y, z) für alle $z \in \{\perp_i, t_{i,2}, t_{i,3}, t_{i,4}\}$: wenn $i = 0$, dann $sup_3(\perp_0) = 0$, sonst $sup_3(\perp_0) = 1$; außerdem $\mathcal{E}^{R_3}_{\mathsf{used}} = \{y\}$ und $\mathcal{E}^{R_3}_{\mathsf{swap}} = \{\ominus_i, \ominus_{i+1}, u_i, a^0_i\}$.

Die folgende Region $R_4 = (sup_4, sig_4)$ löst (y, z) für alle $z \in \{t_{i,5}, t_{i,3}, t_{i,4m_i+2}\}$: wenn $i = 0$, dann $sup_4(\perp_0) = 0$, sonst $sup_4(\perp_0) = 1$; außerdem $\mathcal{E}^{R_4}_{\mathsf{used}} = \{y\}$ und $\mathcal{E}^{R_4}_{\mathsf{swap}} = \{\ominus_i, \ominus_{i+1}, u_i, a^0_i\}$.

Es ist leicht zu sehen, dass (y, z) für alle $z \in \{\perp_{m+1}, h_{1,2}, h_{1,3}\}$ passend τ-lösbar ist. Aus der Beliebigkeit von i folgt die τ-Lösbarkeit von y. □

Fakt 6.29. *Für alle $i \in \{0, \ldots, m-1\}$ und alle $j \in \{0, \ldots, m_i - 1\}$ ist das Ereignis a^j_i durch d_τ-beschränkte τ-Regionen lösbar.*

Beweis. Seien $i \in \{0, \ldots, m-1\}$ und $j \in \{0, \ldots, m_i - 1\}$ jeweils beliebig aber fest. Die folgende Region $R_5 = (sup_5, sig_5)$ löst (a^j_i, z) für alle $z \in \{\perp_i, t_{i,0}, t_{i,4j+3}, t_{i,4m_i+2}\}$: $sup_5(\perp_0) = 0$ und $\mathcal{E}^{R_5}_{\mathsf{used}} = \{a^j_i\}$ und $\mathcal{E}^{R_5}_{\mathsf{swap}} = \{y, w, X_{i_j}\}$.

Beachte, dass wir uns wegen R_5 nun –bezüglich T_i– nur noch um (einige) Zustände „links“ von a^j_i, nämlich $t_{i,1}, \ldots t_{i,4j}$, beziehungsweise und „rechts“ von a^j_i, nämlich $t_{i,4j+6}, \ldots t_{i,4m_i+1}$, kümmern müssen. Genauer hängt dies davon ab, ob $j > 0$ beziehungsweise $j < m_i - 1$. Wir unterscheiden nun diese beiden Fälle.

Sei also zunächst $j > 0$. Die folgende Region $R_6 = (sup_6, sig_6)$ löst (a^j_i, z) für alle $z \in \{t_{i,4j-2}, t_{i,4j-1}, t_{i,4j}\}$: $sup_6(\perp_0) = 0$ und $\mathcal{E}^{R_6}_{\mathsf{used}} = \{a^j_i\}$ und $\mathcal{E}^{R_6}_{\mathsf{swap}} = \{a^{j-1}_i\}$.

Die folgende Region $R_7 = (sup_7, sig_7)$ löst (a^j_i, z) für alle $z \in \{t_{i,1}, \ldots, t_{i,4j-3}\}$: $sup_7(\perp_0) = 0$ und $\mathcal{E}^{R_7}_{\mathsf{used}} = \{a^j_i\}$ und $\mathcal{E}^{R_7}_{\mathsf{set}} = \{X_{i_{j-1}}\}$.

Sei nun $j < m_i - 1$. Die folgende Region $R_8 = (sup_8, sig_8)$ löst (a^j_i, z) für alle $z \in \{t_{i,4(j+1)+2}, \ldots, t_{i,4(j+1)+4}\}$: $sup_8(\perp_0) = 1$ und $\mathcal{E}^{R_8}_{\mathsf{used}} = \{a^j_i\}$ und $\mathcal{E}^{R_8}_{\mathsf{swap}} = \{a^{j+1}_i\}$.

Die folgende Region $R_9 = (sup_9, sig_9))$ löst (a^j_i, z) für alle $z \in \{t_{i,4(j+1)+5}, \ldots, t_{4m_i+1}\}$: $sup_9(\perp_0) = 0$ und $\mathcal{E}^{R_9}_{\mathsf{used}} = \{a^j_i\}$ und $\mathcal{E}^{R_9}_{\mathsf{swap}} = \{a^{j+1}_i\}$ und $\mathcal{E}^{R_9}_{\mathsf{set}} = \{X_{i_{j+1}}\}$.

Aus der Beliebigkeit von i und j folgt der Fakt. □

Fakt 6.30. *Wenn $X \in \mathfrak{U}$, dann ist X durch d_τ-beschränkte τ-Regionen lösbar.*

Beweis. Seien $i \in \{0, \ldots, m-1\}$ und $j \in \{0, \ldots, m_i - 1\}$ beliebig aber fest. Die folgende Region $R_{10} = (sup_{10}, sig_{10})$ löst (X_{i_j}, z) für alle Zustände $z \in Z(T_i) \cup \{\bot_i\}$, an denen X_{i_j} nicht stattfindet: wenn $i = 0$, dann $sup_{10}(\bot_0) = 0$, sonst $sup_{10}(\bot_0) = 1$ und $\mathcal{E}^{R_{10}}_{\mathsf{used}} = \{X_{i_j}\}$ und $\mathcal{E}^{R_{10}}_{\mathsf{swap}} = \{\ominus_i, \ominus_{i+1}, a_i^j\}$. Aus der Beliebigkeit von i und j folgt die τ-Lösbarkeit für alle $e \in \mathfrak{U}$. □

Es ist leicht zu sehen, dass jedes $e \in \{\ominus_1, \ldots, \ominus_{m+1}\} \cup \{u_0, \ldots, u_{m+1}\} \cup \{o\}$ durch d-beschränkte τ-Regionen lösbar ist. Ebenso sieht man leicht, dass es für jedes ZTP von A eine 3-beschränkte τ-Region gibt, die es löst. Zusammen mit den gerade präsentierten Fakten dürfen wir deshalb das folgende Lemma als bewiesen betrachten. Inbesondere ist Satz 6.4.3 damit bewiesen.

Lemma 6.14. *Wenn es eine treffende Menge mit höchstens κ Elementen für $(\mathfrak{U}, M)$ gibt, dann gibt es einen Zeugen d_τ-beschränkter Region für die τ-EZTE und die τ-ZTE von d_τ.*

Der Beweis von Satz 6.4.4

Solange nicht ausdrücklich anders vereinbart, sei im Folgenden $\tau = \{\mathsf{nop}, \mathsf{inp}, \mathsf{res}, \mathsf{swap}\}$. Wir zeigen die Behauptung von Satz 6.4.4 für τ, sodass die Aussage des Satzes für den Netztypen $\{\mathsf{nop}, \mathsf{out}, \mathsf{set}, \mathsf{swap}\}$ dann mittels Kongruenz und Satz 6.1 folgt. Zunächst definieren wir für den Parameter $d_\tau = \kappa + 2$. Das Transitionssystem A_τ hat die folgenden Gadgets H_0 und H_1, wobei H_0 das angekündigte EZTP $\alpha = (k, h_{0,2})$ liefert:

$$H_0 = \bot_m \xrightarrow{u_m} h_{0,0} \xrightarrow{k} h_{0,1} \xrightarrow{w} h_{0,2} \xrightarrow{o} h_{0,3} \xrightarrow{k} h_{0,4}$$

$$H_1 = \bot_{m+1} \xrightarrow{u_{m+1}} h_{1,0} \xrightarrow{w} h_{1,1} \xrightarrow{k} h_{1,2}$$

Darüber hinaus hat A_τ für alle $i \in \{0, \ldots, m-1\}$ das folgende Gadget T_i, welches die Menge $M_i = \{X_{i_0}, \ldots, X_{i_{m_i-1}}\}$ repräsentiert, indem ihre Elemente als Ereignisse verwendet werden:

$$T_i = t_{i,0} \xrightarrow{k} t_{i,1} \xrightarrow{o} t_{i,2} \xrightarrow{X_{i_0}} \cdots \xrightarrow{X_{i_{m_i-1}}} t_{i,m_i+2} \xrightarrow{w} t_{i,m_i+3} \xrightarrow{o} t_{i,m_i+4} \xrightarrow{k} t_{i,m_i+5}$$

Um die eingeführten Gadgets zu dem Transitionssystem A_τ zusammenzufügen, nutzen wir wieder die relevanten Pfade: Für beliebiges aber festes $i \in \{0, \ldots, m-1\}$ referenzieren wir durch $e_i^0, e_i^1, \ldots, e_i^{m_i-1}, e_i^{m_i}$ wieder auf die Ereignisse $X_{i_0}, \ldots, X_{i_{m_i-1}}, w$ und (auf naheliegende Weise) auf ihre entsprechende „Position“ im Gadget T_i. Völlig analog zu der Konstruktion für den Beweis von Satz 6.3.1 erweitern wir erstens T_i zu $G_i = \bot_i \xrightarrow{c_i^0} P_i \xrightarrow{u_i} T_i$, wobei P_i wieder der entsprechende Pfad der für T_i relevanten Pfade ist. Zweitens fügen

wir die verbindenden Kanten hinzu. Abbildung 6.13 skizziert das Transitionssystem A_τ mit dem Initialzustand $\bot_0$.

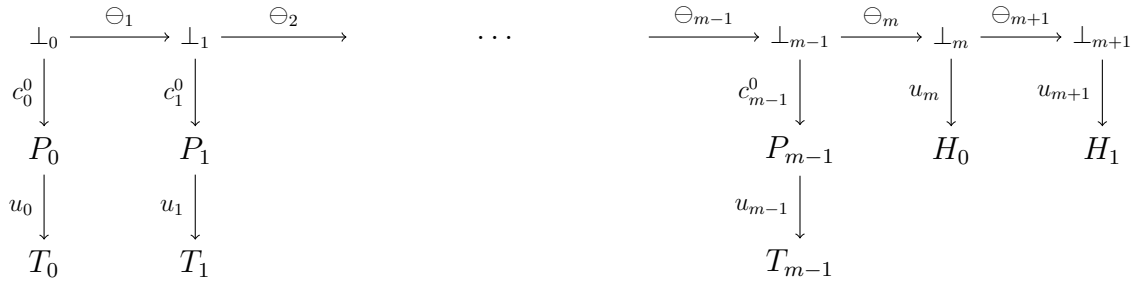


Abbildung 6.13: Skizze des Transitionssystems A_τ der Reduktion für die Netztypen C9.

Lemma 6.15. *Wenn es einen Zeugen d_τ-beschränkter Regionen für die τ-EZTE von A_τ gibt, dann gibt es eine treffende Menge mit höchstens κ Elementen für $(\mathfrak{U}, M)$.*

Beweis. Wenn es einen Zeugen d_τ-beschränkter Regionen für die τ-EZTE von A_τ gibt, dann gibt es eine d_τ-beschränkte τ-Region, die α löst. Sei $R = (sup, sig)$ eine solche Region und sei $i \in \{0, \ldots, m-1\}$ beliebig aber fest. Wir erhalten sofort $sig(k) = \mathsf{inp}$ und $sup(h_{0,2}) = 0$. Wegen $sig(k) = \mathsf{inp}$ erhalten wir außerdem $sup(z) = 1$ und $sup(z') = 0$ für alle Kanten $z \xrightarrow{k} z'$ in A_τ. Mit $sup(h_{0,1}) = sup(h_{0,2}) = 0$ und mit $sup(h_{1,1})$ folgt $sig(w) = \mathsf{nop}$. Auch folgt $sig(o) = \mathsf{swap}$ aus $sup(h_{0,2}) = 0$ und $sup(h_{0,3}) = 1$. Mit $sup(t_{i,1}) = 0$ und $sup(t_{i,m_i+4}) = 1$ sowie $sig(o) = \mathsf{swap}$ und $sig(w) = \mathsf{nop}$ folgen also $sup(t_{i,2}) = 1$ und $sup(t_{i,m_i+2}) = 0$. Insbesondere ist das Bild P_i^R des Pfades $P_i = t_{i,2} \xrightarrow{X_{i_0}} \ldots \xrightarrow{X_{m_i-1}} t_{i,m_i+2}$ ein Pfad von 1 nach 0 in τ. Es gilt also $\{X \in M_i \mid sig(X) \in \{\mathsf{res}, \mathsf{swap}\}\} \neq \emptyset$. Weil R d_τ-beschränkt ist, folgt aus der Beliebigkeit von i sowie $sig(k) \neq \mathsf{nop}$ und $sig(o) \neq \mathsf{nop}$, dass die Menge $S = \{X \in \mathfrak{U} \mid sig(X) \neq \mathsf{nop}\}$ höchstens κ Elemente enthält und $S \cap M_i \neq \emptyset$ für alle $i \in \{0, \ldots, m-1\}$ erfüllt. Dies beweist das Lemma. □

Für die Rückrichtung sei S eine beliebige aber feste treffende Menge mit höchstens κ Elementen für $(\mathfrak{U}, M)$. Wir diskutieren zunächst die Lösbarkeit der Ereignisse k und o durch d_τ-beschränkte τ-Regionen. Regionen $R = (sup, sig)$ werden implizit mittels $sup(\bot_0)$ und sig definiert; Ereignisse gleicher Signatur e_τ werden in $\mathcal{E}_{e_\tau}^R$ zusammengefasst, wobei wir die Menge $\mathcal{E}_{\mathsf{nop}}^R$ wie gehabt nicht explizit definieren.

Fakt 6.31. *Das Ereignis k ist durch d_τ-beschränkte τ-Regionen lösbar.*

Beweis. Wir benötigen die treffende Menge S mit höchstens κ Elementen für $(\mathfrak{U}, M)$.

Die folgende Region $R_1 = (sup_1, sig_1)$ löst $(k, h_{0,2})$, das heißt α, und (k, z) für alle $z \in \{t_{i,m_i+3} \mid i \in \{0, \ldots, m-1\}\} \cup \{z \in Z \mid \xrightarrow{k} z\}$: $sup_1(\bot_0) = 1$ und $\mathcal{E}_{\mathsf{inp}}^{R_1} = \{k\}$ und $\mathcal{E}_{\mathsf{swap}}^{R_1} = \{o\}$ $\mathcal{E}_{\mathsf{res}}^{R_1} = S$. Offenbar ist R_1 d_τ-beschränkt, denn $|S| \leq \kappa$.

Die folgende Region $R_2 = (sup_2, sig_2)$ löst (k, z) für alle verbleibenden relevanten Zustände $z \in \bigcup_{i=0}^{m-1} Z(T_i)$ sowie $(k, h_{1,0})$: $sup_2(\bot_0) = 1$ und $\mathcal{E}_{\mathsf{inp}} = \{k\}$ und $\mathcal{E}_{\mathsf{swap}} = \{w, \ominus_{m+1}\}$.

Analog zu der Argumentation in Fakt 6.14 zeigt man, das (k, z) für die verbleibenden Zustände von A_τ, das heißt vor allem für die Zustände der relevanten Pfade, lösbar ist. □

Fakt 6.32. *Das Ereignis o ist durch d_τ-beschränkte τ-Regionen lösbar.*

Beweis. Die folgende Region $R_1 = (sup_1, sig_1)$ löst (o, z) für alle relevanten $z \in Z(A_\tau) \setminus (\bigcup_{i=0}^{m-1}\{t_{i,m_i+5}\} \cup \{h_{0,0}, h_{0,4}, h_{1,1}\})$: $sup_1(\bot_0) = 0$ und $\mathcal{E}_{\mathsf{inp}}^{R_1} = \{o\}$ und $\mathcal{E}_{\mathsf{swap}}^{R_1} = \{k, w, u_m\}$.

Die folgende Region $R_2 = (sup_2, sig_2)$ löst (o, z) für alle $z \in \bigcup_{i=0}^{m-1}\{t_{i,m_i+5}\}$ und für alle $z \in \{h_{0,0}, h_{0,4}, h_{1,1}\}$: $sup_2(\bot_0) = 1$ und $\mathcal{E}_{\mathsf{inp}}^{R_2} = \{o\}$ und $\mathcal{E}_{\mathsf{swap}}^{R_2} = \{w\}$. □

Um (die Rückrichtung für) den Beweis zu vervollständigen, bleibt erstens zu zeigen, dass auch alle weiteren Ereignisse von A_τ durch d_τ-beschränkte τ-Regionen lösbar sind. Zweitens gilt es zu argumentieren, dass auch alle ZTP von A_τ durch solche Regionen lösbar sind. Man sieht leicht, dass dies durch Regionen möglich ist, die analog zu denen für den Beweis von 6.4.1 beziehungsweise genauer gesagt Lemma 6.7 definiert sind. Dabei beachte man vor allem, dass die betreffenden Regionen mit den Interaktionen $\mathsf{nop}, \mathsf{inp}, \mathsf{swap}$ auskommen, das heißt, res wird nicht gebraucht. Aus dieser Tatsache und Fakt 6.31 sowie Fakt 6.32 erhalten wir das folgende Lemma, welches den Beweis von Satz 6.4.4 komplettiert:

Lemma 6.16. *Wenn es es eine treffende Menge mit höchstens κ Elementen für $(\mathfrak{U}, M)$ gibt, dann gibt es einen Zeugen d_τ-beschränkter Regionen für die τ-EZTE und die τ-ZTE von A_τ.*

6.4 Zusammenfassung und Ausblick

In diesem Kapitel haben wir das Problem für ein gegebenes Paar (A, d) mit Transitionssystem A und Zahl d zu entscheiden, ob A durch ein boolesches Netz mit d-beschränkter Platz-Umgebungszahl bis auf Sprachsimulation beziehungsweise Realisation implementiert werden kann, aus komplexitäts-theoretischer Sicht untersucht. Zunächst zeigt sich, dass dieses Entscheidungsproblem NP-vollständig ist, wenn das unterliegende unbeschränkte Syntheseproblem NP-vollständig ist. Mit den Ergebnissen aus Kapitel 4 trifft dies bereits für 84 von 128 möglichen Netzen mit nop zu. Im Kontrast zur polynomiellen (unbeschränkten) Synthese für $\tau = \{\mathsf{nop}, \mathsf{swap}\} \cup \omega$ mit $\emptyset \neq \omega \subseteq \{\mathsf{inp}, \mathsf{out}, \mathsf{used}, \mathsf{free}\}$, haben wir darüber hinaus gezeigt, dass das Problem beschränkter Synthese auch für diese Netztypen NP-vollständig ist. Die vollständige Klassifizierung der Synthese von booleschen Netzen mit beschränkter Platz-Umgebung bezüglich Sprachsimulation beziehungsweise Realisation für die verbleibenden Netztypen sowie für Einbettung generell bleibt Arbeit für zukünftige Forschung.

Der natürliche Parameter des Problems der Synthese von Netzen mit beschränkter Platz-Umgebung ist die Platz-Umgebungszahl d. Je nachdem welcher Art die gesuchte Implementierung ist, führt dies auf natürliche Weise zu den parametrisierten Problemen Bτ-SMBPsowie Bτ-RMBP. In diesem Kapitel haben wir die parametrisierte Komplexität von Bτ-SMBP und Bτ-RMBP untersucht: zunächst sind beide Probleme in XP. Darüber hinaus haben wir für 23 beziehungsweise 75 der 128 booleschen Netztypen mit nop gezeigt, dass Bτ-SMBP und Bτ-RMBP W[1]-schwer beziehungsweise W[2]-schwer sind. Insbesondere sind die Probleme nicht fest-Parameter-berechenbar für Parameter d.

Natürlich stellt sich die Frage nach der genauen Positionierung von Bτ-SMBP und Bτ-RMBP in der sogenannten W-Hierarchie. Dies zu beantworten ist Gegenstand zukünftiger Arbeit.

Kapitel 7

Platz/Transitions-Netze mit begrenzten Platzbereichen

In diesem Kapitel werden wir die Synthese von Netzen, deren Plätze nur eingeschränkt von Transitionen abhängen, für die Netztypen der P/T-Netze adressieren, die wie folgt definiert sind:

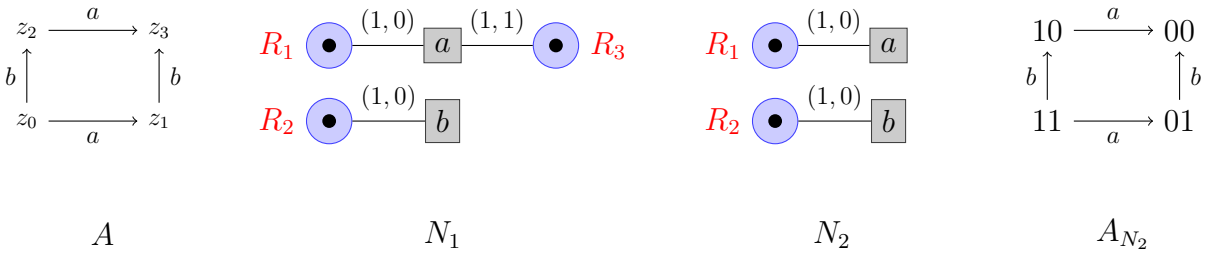
Definition 7.1 (τ_{PT})**.** Der Netztyp der *P/T-Netze* ist definiert durch $\tau_{PT} = (\mathbb{N}, \mathbb{N} \times \mathbb{N}, \delta_{\tau_{PT}}, 0)$ mit Übergangsfunktion $\delta_{\tau_{PT}}$, die für alle $z \in \mathbb{N}$ und alle $(m, n) \in \mathbb{N} \times \mathbb{N}$ genau dann definiert ist, wenn $z \geq m$, wobei in diesem Fall $\delta_{\tau_{PT}}(z, (m, n)) = z - m + n$ gilt.

Die Klasse der P/T-Netze enthält die Teilklasse der *puren* P/T-Netze, die durch den folgenden Netztyp beschrieben werden:

Definition 7.2 (τ_{PPT})**.** Der Netztyp der *puren P/T-Netze* ist definiert durch $\tau_{PPT} = (\mathbb{N}, E_{\tau_{PPT}}, \delta_{\tau_{PPT}}, 0)$ mit Ereignismenge $E_{\tau_{PPT}} = E_{\tau_{PT}} \setminus \{(m, n) \mid 1 \leq m, n\}$, wobei für alle $z \in \mathbb{N}$ und alle $e \in E_{\tau_{PPT}}$ gilt, dass $\delta_{\tau_{PPT}}(z, e) = \delta_{\tau_{PT}}(z, e)$.

Ebenso wie bei den (puren) b-beschränkten P/T-Netzen ist $(0, 0)$ hier der zu nop äquivalente Flusswert, der Unabhängigkeit zwischen Plätzen und Transitionen modelliert. Darüber hinaus werden Plätze von (puren) P/T-Netzen ebenso als Zähler interpretiert, die Ressourcen des modellierten Systems abbilden. Die Anzahl der Marken auf einem Platz entspricht dann der Anzahl der vorhandenen Einheiten der durch den Platz modellierten Ressource des Systems. Der wesentliche Unterschied zu den b-beschränkten P/T-Netzen ist, dass diese Anzahl keiner im Voraus fest gewählten Schranke unterliegt. Da wir als Eingaben nur endliche Transitionssysteme betrachten, gilt unsere Aufmerksamkeit dennoch nur *beschränkten* (puren) P/T-Netzen, das heißt, es gibt eine (nicht zwingend im Voraus bekannte) Zahl $b \in \mathbb{N}$, sodass jeder Platz in jeder erreichbaren Markierung höchstens b Marken enthält.

Um begriffliche Verwechslungen mit der Platz-Umgebung (boolescher Netze) zu vermeiden, wollen wir im Kontext der P/T-Netze statt von der Umgebung eines Platzes lieber von dem *Bereich* eines Platzes sprechen. Insbesondere ist es für Netze der Typen



A N_1 N_2 A_{N_2}

Abbildung 7.1: Das Transitionssystem A und die τ_{PT}-Netze N_1 mit $(1,1)$-begrenzten Plätzen R_1, R_2, R_3 beziehungsweise N_2 mit $(1,0)$-begrenzten Plätzen R_1, R_2, die beide A realisieren. Der Erreichbarkeitsgraph A_{N_2} ist rechts abgebildet.

τ_{PPT} und τ_{PT} sinnvoll –und in der Literatur üblich– den *Bereich* eines Platzes in dem Sinne zu verfeinern, dass zwischen dem *produzierenden* und *konsumierenden Effekt* von Transitionen unterschieden wird. Dies führt zu den Begriffen des *Vor-* beziehungsweise *Nachbereichs* eines Platzes:

Definition 7.3 (Effekt)**.** Sei $\tau \in \{\tau_{PPT}, \tau_{PT}\}$ und $N = (P, T, f, M_0)$ ein τ-Netz. Wenn $p \in P$ ein Platz und $t \in T$ eine Transition von N mit Flusswert $f(p,t) = (m,n)$ sind, dann nennen wir $f^-(p,t) = m$ den *konsumierenden Effekt* von t und $f^+(p,t) = n$ den *produzierenden Effekt* von t (bezüglich p).

Definition 7.4 (Vorbereich, Nachbereich, Bereich)**.** Sei $\tau \in \{\tau_{PPT}, \tau_{PT}\}$ sowie $N = (P, T, f, M_0)$ ein τ-Netz. Für einen Platz $p \in P$ nennen wir die Menge (der potentiell auf p produzierenden Transitionen) ${}^\bullet p = \{t \in T \mid f^+(t,p) \geq 1\}$ den *Vorbereich* von p; darüber hinaus nennen wir die Menge (der potentiell von p konsumierenden Transitionen) $p^\bullet = \{t \in T \mid f^-(t,p) \geq 1\}$ den *Nachbereich* von p. Der *Bereich* von p ist definiert durch ${}^\bullet p^\bullet = {}^\bullet p \cup p^\bullet$.

Definition 7.5 ((ϱ,κ)-begrenzter Platzbereich)**.** Seien $\varrho, \kappa \in \mathbb{N}$ und $\tau \in \{\tau_{PPT}, \tau_{PT}\}$ sowie $N = (P, T, f, M_0)$ ein τ-Netz. Für einen Platz $p \in P$ sagen wir, der Bereich von p ist (ϱ,κ)-*begrenzt*, wenn $|{}^\bullet p| \leq \varrho$ und $|p^\bullet| \leq \kappa$.

Beispiel 7.1. Abbildung 7.1 zeigt die τ_{PT}-Netze N_1 beziehungsweise N_2, deren Plätze $(1,1)$-begrenzte beziehungsweise $(1,0)$-begrenzte Bereiche haben und die beide das Transitionssystem A aus Abbildung 7.1 realisieren.

Bemerkung 7.1. Für die booleschen Netztypen ist diese Unterscheidung zwischen Vor- und Nachbereich nicht einheitlich –und daher nicht sinnvoll– möglich, wie die folgenden Überlegungen begründen: Den Interaktionen inp und res beziehungsweise out und set kann man sicherlich einen konsumierenden beziehungsweise produzierenden Effekt zuordnen. Ebenso könnte man used (analog zur Entsprechung von $(1,1)$ der P/T-Netze) als *stets* sowohl produzierend als auch konsumierend verstehen. Letzteres wird aber schon bei swap

schwierig: *machmal* hat diese Interaktion einen konsumierenden Effekt, *manchmal* einen produzierenden je nachdem, ob der entsprechende Platz eine Marke enthält oder nicht. Spätestens bei free ist jedoch klar: diese Interaktion kann zwar das Schalten einer Transition verhindern, kodiert aber weder einen produzierenden noch einen konsumierenden Effekt.

Sei $\tau \in \{\tau_{PT}, \tau_{PPT}\}$ beliebig aber fest. In [BBD95, BD96a] wurde gezeigt, dass die Synthese von τ-Netzen für alle Implementierungen polynomiell ist. Die verwendeten Methoden basieren auf dem Lösen linearer Ungleichungssysteme über den rationalen Zahlen. Seitdem gab es viele erfolgreiche Bestrebungen, die bekannten Algorithmen für Teilklassen der τ-Netze zu verbessern zum Beispiel indem die Größe der Ungleichungssysteme verringert wurde. Darüber hinaus wurden für Teilklassen auch gänzlich neue Methoden entwickelt, die ohne solche Systeme auskommen und allein auf strukturellen Eigenschaften des Transitionssystems basieren. Zu den am häufigsten untersuchten Teilklassen gehören jene, welche die Kardinalität der Vor- und Nachbereiche ihrer Plätze durch (in der Regel kleine) *im Voraus fest gewählte* Zahlen einschränken. Darunter sind insbesondere die sogenannten *(weighted) marked graphs* [CHEP71] (im Vor- und Nachbereich jedes Platzes ist jeweils genau eine Transition), die *(weighted) T-systems* [Bes86] (im Vor- und Nachbereich jedes Platzes ist jeweils höchstens eine Transition) und, als eine Verallgemeinerung dieser Netzklassen, die *(weighted) choice-free* nets [TCCS92, TCS97] (im Nachbereich jedes Platzes ist höchstens eine Transition).

Ursprünglich sind diese Restriktionen dadurch motiviert, dass die resultierenden Netzklassen eine reichhaltige und elegante Theorie aus Sicht der *Analyse* von Netzen –dem Gegenstück der Synthese– ermöglichen und, damit einhergehend, sehr effiziente Analyse-Algorithmen [Bes86, DE95, HDK14, TCS97] erlauben.

Andererseits sind solche Netzklassen seit Jahren eben auch Gegenstand der Forschung zur Netz-Synthese [BD09, BD15c, BDS18, BHW18, DEH19, DH19]. Es hat sich gezeigt, dass diese Netzklassen mitunter nützliche Eigenschaften haben, die im Sinne der rechnerischen Komplexität verbesserte (Entscheidungs-) Algorithmen ermöglichen, die nicht auf dem Lösen von Trennungsproblemen basieren, sondern auf struktureller Analyse der Eingabe-Transitionssysteme.

In diesem Kapitel werden wir das Problem, zu entscheiden, ob es für zwei natürliche Zahlen ϱ und κ sowie einem Transitionssystem A ein τ_{PT}-Netz gibt, dessen Plätze (ϱ, κ)-begrenzte Bereiche haben und das A bis auf Sprachsimulation beziehungsweise Realisierung implementiert, aus Sicht der klassischen sowie der parametrisierten Komplexität untersuchen. Methodisch gehen wir wie folgt vor: Zunächst definieren wir die zugehörigen nicht-parametrisierten Entscheidungsprobleme, die auf natürliche Weise aus der skizzierten Fragestellung abgeleitet werden können. Danach zeigen wir in Abschnitt 7.1, dass diese Probleme in NP sind. Anschließend zeigen wir in Abschnitt 7.2 mittels einer Reduktion von Hitting Set die NP-Vollständigkeit der Entscheidungsprobleme. Schließlich werden wir in Abschnitt 7.3 kurz argumentieren, dass die verwendeten Methoden ebenso Aussagen über die parametrisierte Komplexität der parametrisierten Versionen dieser Probleme implizieren, wobei natürlicherweise $\varrho + \kappa$ als Parameter gewählt wird: Genauer gesagt

führt diese Parametrisierung zwar zur Zugehörigkeit zu XP, jedoch sind die resultierenden Probleme W[2]-schwer und daher nicht fest-Parameter-berechenbar.

Solange nicht ausdrücklich anders vereinbart, sei im Rest dieses Kapitels $\tau = \tau_{PT}$. Gemäß Definition 7.5 ist ein Platz $p \in P$ eines τ-Netzes $N = (P, T, f, M_0)$ (ϱ, κ)-begrenzt, wenn $|{}^\bullet p| \leq \varrho$ und $|p^\bullet| \leq \kappa$ gilt. Das heißt, die Anzahl der Transitionen $t \in T$ mit $f(p,t) = (m,n)$, sodass $0 < m$ beziehungsweise $0 < n$ ist durch ϱ beziehungsweise κ begrenzt. Wir wissen aus Satz 2.1, dass es für ein Transitionssystem $A = (Z, E, \delta, \iota)$ genau dann ein τ-Netz gibt, welches A implementiert, wenn es einen Zeugen $\mathcal{R}$ von τ-Regionen von A gibt, welcher die Trennungseigenschaften für A nachweist, die der Implementierung entsprechen. Darüber hinaus ist dann das synthetisierte Netz $N = N_A^{\mathcal{R}}$ bereits eine gesuchte Implementierung. Die Ereignisse aus E sind die Transitionen des Netzes und jede Region $R = (sup, sig)$ aus $\mathcal{R}$ ist ein Platz von N. Darüber hinaus gilt für jede Transition $e \in E$, dass der Fluss $f(R,e)$ durch den Signaturwert $sig(e) = (m,n)$ definiert ist. Dabei nutzen wir sig^- und sig^+ wie in Definition 5.5 aus Kapitel 5 um auf m beziehungsweise n zu referenzieren, das heißt, $sig^-(e) = m$ und $sig^+(e) = n$. Es folgt, dass es genau dann ein gesuchtes Netz für A gibt, wenn es eine Menge (ϱ, κ)-begrenzter τ-Regionen von A gibt, welche, je nachdem, die τ-EZTE beziehungsweise die τ-ZTE und die τ-EZTE von A nachweisen:

Definition 7.6 ((ϱ, κ)-begrenzte τ-Region)**.** Sei $A = (Z, E, \delta, \iota)$ ein Transitionssystem und $R = (sup, sig)$ eine τ-Region von A. Der *Vorbereich* von R ist definiert durch ${}^\bullet R = \{e \in E \mid sig^+(e) > 0\}$ und der *Nachbereich* von R durch $R^\bullet = \{e \in E \mid sig^-(e) > 0\}$. Wir sagen R ist (ϱ, κ)-*begrenzt*, wenn $|{}^\bullet R| \leq \varrho$ und $|R^\bullet| \leq \kappa$.

Beispiel 7.2. Das Transitionssystem A aus Abbildung 7.1 hat die $(1,0)$-begrenzten Regionen $R_1 = (sup_1, sig_1)$ und $R_2 = (sup_2, sig_2)$, wobei $sup_1(z_0) = sup_1(z_2) = 1$, $sup_1(z_1) = sup_1(z_3) = 0$, $sig_1(a) = (1,0)$ und $sig_1(b) = (0,0)$ beziehungsweise $sup_2(z_0) = sup_1(z_1) = 1$, $sup_1(z_2) = sup_1(z_3) = 0$, $sig_2(a) = (0,0)$ und $sig_2(b) = (1,0)$. Außerdem hat A die $(1,1)$-beschränkte Region $R_3 = (sup_3, sig_3)$ mit $sup_3(z_0) = \cdots = sup_3(z_3) = 1$ und $sig_3(a) = (1,1)$ sowie $sig_3(b) = (0,0)$. Wenn $\mathcal{R}_1 = \{R_1, R_2, R_3\}$ und $\mathcal{R}_2 = \{R_1, R_2\}$, dann zeigt Abbildung 7.1 die synthetisierten Netze $N_1 = N_A^{\mathcal{R}_1}$ und $N_2 = N_A^{\mathcal{R}_2}$.

Wir erhalten die folgenden Entscheidungsprobleme, deren Komplexität in diesem Kapitel der Gegenstand unserer Untersuchungen ist:

τ-Sprachsimulation mit begrenzten Platz-Bereichen (τ-SMBPB)	
Eingabe:	Ein Transitionssystem $A = (Z, E, \delta, \iota)$, zwei natürliche Zahlen ϱ und κ.
Frage:	Gibt es einen Zeugen (ϱ, κ)-begrenzter Regionen für die τ-EZTE von A?

τ-Realisierung mit begrenzten Platz-Bereichen (τ-RMBPB)	
Eingabe:	Ein Transitionssystem $A = (Z, E, \delta, \iota)$, zwei natürliche Zahlen ϱ und κ.
Frage:	Gibt es einen Zeugen (ϱ, κ)-begrenzter Regionen für die τ-ZTE und die τ-EZTE von A?

Der folgende Satz adressiert die klassische Komplexität von τ-SMBPB und τ-RMBPB:

Satz 7.1. *τ-SMBPB und τ-RMBPB sind NP-vollständig.*

Um Satz 7.1 zu beweisen, zeigen wir einerseits im folgenden Abschnitt 7.1, dass τ-SMBPB und τ-RMBPB in NP sind. Weil die Support- und Signaturwerte von τ_{PT}-Regionen zunächst beliebig große natürliche Zahlen annehmen können, reicht die einfache Argumentation nicht, dass ein Transitionssystem nur polynomiell viele Trennungsprobleme haben kann. Wir müssen zusätzlich (und technisch aufwendig) argumentieren, dass es stets eine Zeugen mit Regionen polynomieller Größe gibt, wenn es überhaupt einen Zeugen gibt.

Andererseits zeigen wir danach in Abschnitt 7.2 mittels einer polynomiellen Reduktion des Problems Hitting Set, dass τ-SMBPB und τ-RMBPB NP-schwer sind.

7.1 τ_{PT}-SMBPB und τ_{PT}-RMBPB sind in NP

In diesem Abschnitt zeigen wir, dass τ-SMBPB und τ-RMBPB in NP sind. Dazu erweitern wir den deterministischen Polynomialzeit-Algorithmus für die Synthese von τ-Netzen, deren Platzbereiche *a priori* keinen Begrenzungen unterliegen. Dieser Algorithmus wird zum Beispiel in [BBD15] umfassend diskutiert. Er basiert auf der Tatsache, dass sich die τ-Lösbarkeit eines einzelnen Trennungsproblems α eines Transitionssystems $A = (Z, E, \delta, \iota)$ auf die Lösbarkeit eines Systems $K \cdot \mathbf{x} = c$ aus Gleichungen und Ungleichungen mit rationalen Variablen reduzieren lässt. Ein solches System kann gemäß Khachiyans Methode [Raj90, pp. 168-170] in Polynomial-Zeit gelöst werden. A hat höchstens $|E| \cdot |Z| + |Z|^2$ Trennungsprobleme. Weil über die Lösbarkeit jedes einzelnen Trennungsproblems durch die Lösbarkeit des zugehörigen Systems $K \cdot \mathbf{x} = c$ entschieden werden kann, erhält man einen polynomiellen Algorithmus, der über die Implementierbarkeit von A entscheidet. Unsere Methode erweitert nun für ein gegebenes Trennungsproblem α das eben erwähnte zugehörige System von (Un-) Gleichungen nicht-deterministisch um weitere Gleichungen. Diese zusätzlichen Gleichungen formulieren Bedingungen, welche die durch ϱ und κ geforderten Beschränkungen kodieren. Das resultierende System ist genau dann lösbar, wenn es eine (ϱ, κ)-begrenzte τ-Region von A gibt, die α löst. Die Anzahl der zusätzlichen Gleichungen –und damit die Größe des resultierenden Systems– ist polynomiell beschränkt in der Größe der Eingabe (A, ϱ, κ) und kann mit dem Algorithmus von Khachiyan gelöst werden. Weil A insgesamt höchstens $|E| \cdot |Z| + |Z|^2$ Trennungsprobleme hat, impliziert dies, dass τ-SMBPB und τ-RMBPB in NP sind.

Solange nicht ausdrücklich anders vereinbart, sei (A, ϱ, κ) im Folgenden eine beliebige aber feste Eingabe für τ-SMBPB beziehungsweise τ-RMBPB mit Transitionssystem $A = (Z, \{e_1, \ldots, e_n\}, \delta, \iota)$ sowie natürlichen Zahlen ϱ und κ und sei α ein Trennungsproblem von A. Um das angekündigte Verfahren herleiten zu können, werden wir zunächst knapp die erwähnte deterministische Methode zur Entscheidung über die Lösbarkeit von α durch eine nicht zwingend begrenzte Region herleiten. Dabei werden wir die Intention und Funktionalität eher informell beschreiben. Die Beweise der einzelnen Aussagen können in [BBD15] nachgelesen werden. Danach wird gezeigt, wie dieses Verfahren durch

nicht-deterministisches Hinzufügen weiterer Gleichungen zu einem entsprechenden Entscheidungsverfahren für die Lösbarkeit von α durch passend begrenzte Regionen modifiziert werden kann.

Wenn $R = (sup, sig)$ eine τ-Region ist, dann gilt gemäß Bemerkung 2.7, dass R bereits vollständig durch $sup(\iota)$ und sig bestimmt ist. Insbesondere kann R mit dem Element $\mathbf{y} = (sup(\iota), sig^-(e_1), \ldots, sig^-(e_n), sig^+(e_1), \ldots, sig^+(e_n))$ aus $\mathbb{Q}^{2n+1}$ identifiziert werden. Die Reduktion der Lösbarkeit von α auf die Lösbarkeit eines Systems $K \cdot \mathbf{x} = c$ von Gleichungen und Ungleichungen mit rationalen Variablen lässt sich wie folgt skizzieren: Wenn α durch R gelöst wird, dann wird $K \cdot \mathbf{x} = c$ durch $\mathbf{y}$ gelöst. Andererseits, wenn $\mathbf{x}' \in \mathbb{Q}^{2n+1}$ eine rationale Lösung des (Un-) Gleichungssystems ist, dann gibt es auch eine ganzzahlige Lösung $\mathbf{x} = (x_0, x_1, \ldots, x_n, x_{n+1}, \ldots, x_{2n}) \in \mathbb{Z}^{2n+1}$, die implizit eine Region $R = (sup, sig)$ durch $sup(\iota) = x_0$ und $sig^-(e_i) = x_i$ und $sig^+(e_i) = x_{i+n}$ für alle $i \in \{1, \ldots, n\}$ definiert, die α löst. Dabei erhalten wir $\mathbf{x}$ durch die Multiplikation von $\mathbf{x}'$ mit dem kleinsten gemeinsamen Vielfachen der Nenner seiner Einträge. Daraus ergibt sich, dass α genau dann τ-lösbar ist, wenn das (zu α gehörende) Gleichungssystem $K \cdot \mathbf{x} = c$ eine rationale Lösung besitzt.

Wie bei der Herangehensweise für die gruppenerweiterten Netze in Abschnitt 5.2.2, besteht das erwähnte (Un-) Gleichungssystem im Wesentlichen aus zwei Teilen: Der eine Teil besteht aus Gleichungen und Ungleichungen, die zusichern, dass eine Lösung überhaupt als Region von A interpretiert werden kann. Dieser Teil umfasst unter anderem (wieder) Gleichungen die aus *Fundamentalkreisen* hervorgehen, die durch die Sehnen eines Spannbaums A' von A definiert werden. Der zweite Teil besteht aus einer Ungleichung, die zusichert, dass die Region α tatsächlich löst. Die entsprechenden Begrifflichkeiten werden dabei auf natürliche Weise von $\mathbb{Z}_{b+1}$ auf $\mathbb{Q}$ übertragen: Sei A' im Folgenden ein beliebiger aber fester Spannbaum gemäß Definition 5.14 von A. Für jeden Zustand $z \in Z$ gibt es einen eindeutigen Pfad P_z von ι nach z, wobei dieser Pfad für $z = \iota$ der leere Pfad ist. Analog zu Definition 5.15 erhalten wir für alle $z \in Z$ den *Parikh-Vektor* $\sigma_z : \{e_1, \ldots, e_n\} \to \mathbb{N}$ von z, der jeweils die Anzahl der Vorkommen der Ereignisse von A entlang P_z zählt –und zwar absolut und nicht modulo $b + 1$. Der Einfachheit wegen identifizieren wir $\sigma_z = (\sigma_z(e_1), \ldots, \sigma_z(e_n))$. Ebenso analog zu Definition 5.16 ist der *Fundamentalkreis* $\sigma_t : \{e_1, \ldots, e_n\} \to \mathbb{Z}$ mit $\sigma_t = \sigma_z + \vec{e_i} - \sigma_{z'}$ zu einer Sehne $t = z \xrightarrow{e_i} z'$ definiert, wobei $\vec{e_i}$ den i-ten Einheitsvektor symbolisiert. Wieder identifizieren wir $\sigma_t = (\sigma_t(e_1), \ldots, \sigma_t(e_n))$. Schließlich referenzieren wir für zwei Elemente $x = (x_1, \ldots, x_n)$ und $y = (y_1, \ldots, y_n)$ des $\mathbb{Q}$-Vektroraums $(\mathbb{Q}^n, +, \cdot)$ auf ihr Skalarprodukt durch $x \cdot y = x_1 \cdot y_1 + \cdots + x_n \cdot y_n$.

Wenn $R = (sup, sig)$ eine τ-Region ist, dann ist R vollständig definiert durch $sup(\iota)$ und sig, denn wenn $\iota \xrightarrow{a_1} s_1 \xrightarrow{a_2} \ldots \xrightarrow{a_m} z$ ein Pfad von ι zu z in A ist, dann ist $sup(z)$ definiert durch

$$sup(z) = sup(\iota) - sig^-(a_1) + sig^+(a_1) \cdots - sig^-(a_m) + sig^+(a_m) \tag{7.1}$$

In A' ist dieser Pfad P_z eindeutig. Mit $\mathbf{z} = (sig^+(e_1) - sig^-(e_1), \ldots, sig^+(e_n) - sig^-(e_n))$

lässt sich Gleichung 7.1 dann umschreiben zu

$$sup(z) = sup(\iota) + \psi_z \cdot \mathbf{z} \tag{7.2}$$

Ausgestattet mit den notwendigen Notationen und Begrifflichkeiten werden wir nun zunächst das angekündigte System $K \cdot \mathbf{x} = c$ vorstellen. Anschließend werden wir erläutern, wie dieses System nicht-deterministisch um Gleichungen zu erweitern ist, sodass eine Lösung des resultierenden Systems eine Region mit begrenztem Bereich impliziert. Dabei behandeln wir erst den Fall, dass α ein beliebiges aber festes EZTP (e_j, q) ist, wobei $j \in \{1, \ldots, n\}$. Wir werden danach argumentieren, dass die vorgestellte Methode analog für ZTP von A funktioniert.

Für jede Sehne t und zugehörigen Fundamentalkreis σ_t von A' hat das (Un-) Gleichungssystem die folgende Gleichung:

$$\sigma_t \cdot (x_{n+1} - x_1, \ldots, x_{2n} - x_n) = 0 \tag{7.3}$$

Diese aus den Fundamentalkreisen resultierenden Gleichungen sichern zu, dass eine Lösung des Systems, interpretiert als Region, $sup(z') = sup(z) - sig^-(e) + sig^+(e)$ für jede Kante $z \xrightarrow{e} z' \in A$ erfüllt.

Die folgende Ungleichung 7.4 sichert zu, dass eine Lösung, interpretiert als Region, dem Initialzustand korrekt einen nicht-negativen Supportwert zuordnet:

$$0 \leq \underbrace{x_0}_{sup(\iota)} \tag{7.4}$$

Darüber hinaus hat das System für jeden Zustand $z \in Z$ und jedes $i \in \{1, \ldots, n\}$, sodass e_i an z stattfindet, das heißt $z \xrightarrow{e_i}$, die folgende Ungleichung 7.5:

$$0 \leq \underbrace{x_0 + \sigma_z \cdot (x_{n+1} - x_1, \ldots, x_{2n} - x_n)}_{sup(z)} - \underbrace{x_i}_{sig^-(e_i)} \tag{7.5}$$

Insgesamt sichern diese Ungleichungen für alle $e \in E$ und alle $z \in Z$ zu, dass eine Lösung, interpretiert als Region, wie gefordert $sig^-(e) \leq sup(z)$ erfüllt, wenn e an z stattfindet. Weil mit Gleichung 7.4 gilt, dass $sup(\iota) \geq 0$, sorgen diese Gleichungen außerdem dafür, dass der Supportwert jedes Zustands von A nicht-negativ ist. Man beachte, dass e_i in A' nicht zwingend an z definiert sein muss.

Schließlich enthält das System noch die folgende Ungleichung, die zusichert, das eine Lösung eine Region beschreibt, für die $sig^-(e_j) > sup(q)$ gilt und damit das EZTP $\alpha = (e_j, q)$ löst:

$$\underbrace{x_0 + \sigma_q \cdot (x_{n+1} - x_1, \ldots, x_{2n} - x_n)}_{sup(q)} - \underbrace{x_j}_{sig^-(e_j)} \leq -1 \tag{7.6}$$

Wenn $R = (sup, sig)$ das EZTP α löst, dann löst der Vektor $\mathbf{y}$, definiert durch $\mathbf{y} = (sup(\iota), sig^-(e_1), \ldots, sig^-(e_n), sig^+(e_1), \ldots, sig^+(e_n))$, die eingeführten (Un-) Gleichungen. Für die Gleichungen, die aus den Fundamentalkreisen resultieren (Gleichung 7.3) folgt dies aus Proposition 6.16 aus [BBD15]. Für die Ungleichungen 7.4 und 7.5 folgt dies aus Gleichung 7.2 und Definition 2.18, wobei letztere $sup(\iota) \geq 0$ und $sig^-(e_i) \leq sup(z)$ für jeden Zustand z von A impliziert, an dem e_i stattfindet. Für Ungleichung 7.6 folgt dies schließlich aus Gleichung 7.2 und der Tatsache, dass R das EZTP α löst, denn dies impliziert $sup(q) < sig^-(e_j)$ und somit $sup(q) - sig^-(e_j) \leq -1$ nach Definition 2.22.

Andererseits zeigen die Erläuterungen zu den Bedeutungen der (Un-) Gleichungen bereits: Wenn $\mathbf{x} = (x_0, x_1, \ldots, x_n, x_{n+1}, \ldots, x_{2n})$ eine ganzzahlige Lösung dieses Systems ist, dann ist durch $\mathbf{x} = (sup(\iota), sig^-(e_1), \ldots, sig^-(e_n), sig^+(e_1), \ldots, sig^+(e_n))$ implizit eine τ_{PT}-Region von A, die α löst definiert [BBD15].

Folglich löst ein ganzzahliger Vektor $\mathbf{x}$ das vorgestellte System genau dann, wenn $\mathbf{x} = (sup(\iota), sig^-(e_1), \ldots, sig^-(e_n), sig^+(e_1), \ldots, sig^+(e_n))$ implizit eine Region $R = (sup, sig)$ von A definiert, die α löst. Des Weiteren gibt es genau dann eine rationale Lösung, wenn es eine ganzzahlige Lösung gibt: Wegen der Homogenität der (Un-) Gleichungen 7.3 und 7.4 sowie 7.5 kann eine rationale Lösung $\mathbf{x}'$ durch Multiplikation mit dem kleinsten gemeinsamen Vielfachen (der Nenner) seiner Einträge zu einer gültigen ganzzahligen Lösung umgeformt werden. Eine so erhaltene Lösung, erfüllt Ungleichung 7.6.

Wir erläutern nun, wie das eingeführte System nicht-deterministisch ergänzt werden kann, sodass eine ganzzahlige Lösung eine (ϱ, κ)-begrenzte τ-Region impliziert, die α löst.

Wenn $R = (sup, sig)$ ein solche Region ist, das heißt $|\{e \in E \mid sig^-(e) > 0\}| \leq \varrho$ und $|\{e \in E \mid sig^+(e) > 0\}| \leq \kappa$, dann gibt es höchstens ϱ Indizes $i_1, \ldots, i_\varrho \in \{1, \ldots, n\}$ beziehungsweise κ Indizes $j_1, \ldots, j_\kappa \in \{1, \ldots, n\}$, sodass $sig^-(e_{i_\ell}) > 0$ für alle $\ell \in \{1, \ldots, \varrho\}$ und $sig^+(e_{j_k}) > 0$ für alle $k \in \{1, \ldots, \kappa\}$. Deshalb erfüllt der Vektor $\mathbf{y} = (sup(\iota), sig^-(e_1), \ldots, sig^-(e_n), sig^+(e_1), \ldots, sig^+(e_n))$ für alle $i \in \{1, \ldots, n\} \setminus \{i_1, \ldots, i_\varrho\}$ und alle $j \in \{1, \ldots, n\} \setminus \{j_1, \ldots, j_\kappa\}$ die folgenden Gleichungen:

$$x_i = 0 \tag{7.7}$$

$$x_{j+n} = 0 \tag{7.8}$$

Darüber hinaus erfüllt $\mathbf{y}$ nach obigen Ausführungen auch die (Un-) Gleichungen 7.3 bis 7.8.

Umgekehrt sei nun $\mathbf{x}'$ eine rationale Lösung des Systems, das durch die (Un-) Gleichungen 7.3 bis 7.8 definiert ist. Weil die Gleichungen 7.7 und 7.8 homogen sind, erhalten wir wieder eine entsprechende ganzzahlige Lösung $\mathbf{x}$ durch Multiplikation mit dem kleinsten gemeinsamen Vielfachen. Gemäß den obigen Überlegungen definiert der Lösungsvektor $\mathbf{x} = (sup(\iota), sig^-(e_1), \ldots, sig^-(e_n), sig^+(e_1), \ldots, sig^+(e_n))$ implizit eine τ-Region $R = (sup, sig)$ von A, die α löst. Insbesondere gilt $sig^-(e_i) = 0$ für alle $i \in \{1, \ldots, n\} \setminus \{i_1, \ldots, i_\varrho\}$ und $sig^+(e_j) = 0$ für alle $j \in \{1, \ldots, n\} \setminus \{j_1, \ldots, j_\kappa\}$ und somit auch $|\{e \in E \mid sig^-(e) > 0\}| \leq \varrho$ und $|\{e \in E \mid sig^+(e) > 0\}| \leq \kappa$. Das bedeutet, R ist (ϱ, κ)-begrenzt.

Sicherlich ist die Anzahl n der Ereignisse von A jeweils eine obere Schranke für ϱ und κ. Wenn R existiert, dann kann eine Turing-Maschine T die Indizes $i_1, \ldots, i_\varrho$ und $j_1, \ldots, j_\kappa$

deshalb in einer nicht-deterministischen Berechnung mit polynomiellem Zeitaufwand raten. Anschließend kann T deterministisch das System der (Un-) Gleichungen 7.3 bis 7.8 konstruieren, eine (ganzzahlige) Lösung $\mathbf{x}$ des Systems berechnen und verifizieren, dass $\mathbf{x}$ eine Region definiert, die α löst. Jeder dieser Teilschritte ist jeweils offenbar mit polynomiellen Zeitaufwand möglich. Insgesamt folgt daraus, dass über die τ-Lösbarkeit eines EZTP α von A durch eine (ϱ, κ)-begrenzte Region nicht-deterministisch in polynomieller Zeit entschieden werden kann. Weil A höchstens $|E| \cdot |Z|$ EZTP hat, ist τ-SMBPB in NP.

Analog argumentiert man, dass dies auch zutrifft, wenn α ein ZTP (p, q) ist. Anstatt der Ungleichung 7.6 hat das initiale System dann die folgende Gleichung 7.9:

$$\sigma_p \cdot (x_{n+1} - x_1, \ldots, x_{2n} - x_n) - \sigma_q \cdot (x_{n+1} - x_1, \ldots, x_{2n} - x_n) \leq -1 \tag{7.9}$$

In [BBD15, p. 214] wird argumentiert, dass das resultierende System genau dann eine Lösung hat, wenn α τ-lösbar ist. (Insbesondere brauchen wir die analoge Ungleichung $\sigma_q \cdot (x_{n+1} - x_1, \ldots, x_{2n} - x_n) - \sigma_p \cdot (x_{n+1} - x_1, \ldots, x_{2n} - x_n) \leq -1$ nicht separat berücksichtigen.) Wie vorher erhalten wir einen nicht-deterministischen Algorithmus, der die Lösbarkeit von α durch eine begrenzte Region entscheidet. Da A höchstens $|Z|^2$ ZTP hat, haben wir nun insgesamt gezeigt, dass auch τ-RMBPB in NP ist.

7.2 τ_{PT}-SMBPB und τ_{PT}-RMBPB sind NP-schwer

Ziel dieses Abschnitts ist es, den Beweis von Satz 7.1 zu vervollständigen. Dazu zeigen wir, dass τ-SMBPB und τ-RMBPB NP-schwer sind. Die NP-Schwere resultiert aus einer Reduktion des Problems Hitting Set (HS), welches in Kapitel 6 definiert wurde: Für ein Tripel $(\mathfrak{U}, M, \lambda)$ mit Menge $\mathfrak{U}$, Mengensystem M mit Teilmengen aus $\mathfrak{U}$ und natürlicher Zahl λ, wird gefragt, ob es eine treffende Menge mit höchstens λ Elementen für $(\mathfrak{U}, M)$ gibt. Wir ergänzen den Reduktionsbeweis um ein vollständiges Beispiel (Abbildung 7.2), das auf der folgenden Instanz von HS basiert:

Beispiel 7.3. Die Instanz $(\mathfrak{U}, M, 3)$ mit $\mathfrak{U} = \{X_0, X_1, X_2, X_3\}$ und $M = \{M_0, \ldots, M_5\}$, wobei $M_0 = \{X_0, X_1\}$, $M_1 = \{X_0, X_2\}$, $M_2 = \{X_0, X_3\}$, $M_3 = \{X_1, X_2\}$, $M_4 = \{X_1, X_3\}$ und $M_5 = \{X_2, X_3\}$, erlaubt eine positive Entscheidung: $S = \{X_0, X_1, X_2\}$ ist eine passende treffende Menge für $\mathfrak{U}$.

Solange nicht ausdrücklich anders angegeben, sei $(\mathfrak{U}, M, \lambda)$ im weiteren Verlauf dieses Abschnitts eine beliebige aber feste Eingabe für HS, sodass $\mathfrak{U} = \{X_0, \ldots, X_{n-1}\}$ und $M = \{M_0, \ldots, M_{m-1}\}$, wobei $M_i = \{X_{i_0}, \ldots, X_{i_{m_i-1}}\}$ (und somit $|M_i| = m_i$) für alle $i \in \{0, \ldots, m-1\}$. Aus technischen Gründen nehmen wir ohne Beschränkung der Allgemeinheit $i_0 < \cdots < i_{m_i-1}$ für die Elemente $X_{i_0}, \ldots, X_{i_{m_i-1}}$ der Menge M_i für alle $i \in \{0, \ldots, m-1\}$ an. Ebenso setzen wir aus technischen Gründen $\lambda \geq 5$ voraus. Dies ist keine Beschränkung der Allgemeinheit, denn HS ist für jedes feste $\lambda \in \mathbb{N}$ in Polynomial-Zeit lösbar [CFK+15].

Bemerkung 7.2. Offensichtlich erfüllt die Eingabe aus Beispiel 7.3 *nicht* die Eigenschaft $\lambda \geq 5$. Um dem Leser jedoch ein vollständiges *übersichtliches* Beispiel der Reduktion anbieten zu können, ist diese Eingabe bewusst klein gewählt.

Die Reduktionsstrategie für die NP-Schwere von τ-SMBPB und τ-RMBPB

Für die Eingabe $(\mathfrak{U}, M, \lambda)$ konstruieren wir eine Eingabe (A, ϱ, κ) mit Transitionssystem A und natürlichen Zahlen ϱ und κ für τ-SMBPB und τ-RMBPB. Für alle $i \in \{0, \ldots, m-1\}$ enthält das Transitionssystem A den folgenden Pfad P_i auf dem die Elemente der Menge $M_i = \{X_{i_0}, \ldots, X_{i_{m_i-1}}\}$ als Ereignisse stattfinden und der M_i repräsentiert:

$$P_i = t_{i,0} \xrightarrow{X_{i_0}} \ldots \xrightarrow{X_{i_{m_i-1}}} t_{i,m_i+1}$$

Des Weiteren hat A hat ein EZTP α, sodass eine (ϱ, κ)-begrenzte τ-Region $R = (sup, sig)$, die α löst, die folgende Eigenschaften erfüllt: Sei $i \in \{0, \ldots, m-1\}$ beliebig aber fest. Erstens gilt $sup(t_{i,0}) < sup(t_{i,m_i+1})$. Weil das Bild P_i^R des Pfades P_i unter der Region R ein Pfad von $sup(t_{i,0})$ nach $sup(t_{i,m_i+1})$ in τ ist, folgt, dass es ein $X \in \{X_{i_0}, \ldots, X_{i_{m_i-1}}\}$ gibt, sodass $sig^+(X) > 0$. Weil i beliebig war, folgt, dass die Menge $S = \{X \in \mathfrak{U} \mid sig^+(X) > 0\}$ von jeder Menge aus M ein Element enthält. Zweitens sichert die Konstruktion zu, dass $|S| \leq \lambda$ gilt, weil R (ϱ, κ)-begrenzt ist. Folglich ist S eine treffende Menge mit höchstens λ Elementen für $(\mathfrak{U}, M)$. Wenn (A, ϱ, κ) eine positive Entscheidung erlaubt, dann gibt es einen Zeugen $\mathcal{R}$ mit (ϱ, κ)-begrenzten τ-Regionen für die τ-EZTE von A. Insbesondere enthält $\mathcal{R}$ eine solche Region, die α löst und impliziert somit, dass $(\mathfrak{U}, M, \lambda)$ eine positive Entscheidung erlaubt.

Im Gegenzug werden wir beweisen, dass ein Zeuge $\mathcal{R}$ mit (ϱ, κ)-begrenzten τ-Regionen für die τ-EZTE und die τ-ZTE von A existiert, wenn es eine gesuchte treffende Menge für $(\mathfrak{U}, M)$ gibt. Insgesamt erhalten wir, dass $(\mathfrak{U}, M, \lambda)$ genau dann eine positive Entscheidung erlaubt, wenn (A, ϱ, κ) dies tut.

Die Reduktion

Zunächst definieren wir $\varrho = 2\lambda$ und $\kappa = \lambda + 1$. Man beachte, dass dies $\varrho \geq 10$ und $\kappa \geq 6$ impliziert, da wir $\lambda \geq 5$ vorausgesetzt haben. Wir benötigen diese unteren Schranken für ϱ und κ, um zu beweisen, dass das konstruierte Transitionssystem die τ-EZTE hat, wenn es eine passende treffende Menge gibt. Dabei führt die Annahme $\varrho \geq 10$ und $\kappa \geq 6$ nicht zu einer Einschränkung der Allgemeinheit, denn die Überlegungen aus Abschnitt 7.2 implizieren, dass τ-SMBPB und τ-RMBPB für alle festen ϱ und κ in Polynomial-Zeit entscheidbar sind. Letzteren Aspekt werden wir in Abschnitt 7.4 noch genauer diskutieren. Abbildung 7.2 zeigt ein vollständiges Beispiel für das Transitionssystem A der folgenden Konstruktion, welches auf der Instanz aus Beispiel 7.3 basiert.

Das Transitionssystem A hat für jedes $i \in \{0, \ldots, m-1\}$ das folgende Gadget T_i, das die Menge $M_i = \{X_{i_0}, \ldots, X_{i_{m_i-1}}\}$ durch Verwendung ihrer Elemente als Ereignisse repräsentiert:

$$T_i = t_{i,0} \xrightarrow{\quad k \quad} t_{i,1} \xrightarrow{\quad X_{i_0} \quad} \cdots \xrightarrow{\quad X_{i_{m_i-1}} \quad} t_{i,m_i+1} \xrightarrow{\quad k \quad} t_{i,m_i+2}$$

Insbesondere stellt das Gadget T_0 das angekündigte EZTP $\alpha = (k, t_{0,1})$ zur Verfügung. Darüber hinaus sind die Zustände $t_{i,1}$ und $t_{i+1,1}$ für alle $i \in \{0, \ldots, m-2\}$ durch eine u_{i+1}-beschriftete Kante $t_{i,1} \xrightarrow{u_{i+1}} t_{i+1,1}$ miteinander verbunden.

Des Weiteren hat A für alle $i \in \{0, \ldots, \lambda - 1\}$ das folgende Gadget F_i und für alle $j \in \{1, \ldots, m-1\}$ das folgende Gadget G_j, in welchem das Ereignis u_j erneut stattfindet:

$$F_i = f_{i,0} \underset{k_i}{\overset{k \quad z_i}{\rightleftarrows}} f_{i,1} \qquad\qquad G_j = g_{j,0} \underset{v_j}{\overset{u_j}{\rightleftarrows}} g_{j,1}$$

Die bisher eingeführten Gadgets umfassen den funktionalen Teil von A. Der Initialzustand von A ist $\perp_0$. Des Weiteren fügen wir die folgenden Kanten hinzu, um die einzelnen Gadgets miteinander zu verbinden: Für alle $i \in \{0, \ldots, m-1\}$ fügen wir die Kante $\perp_i \xrightarrow{y_i} t_{i,0}$ und wenn $i < m-1$, dann auch die Kante $\perp_i \xrightarrow{w_{i+1}} \perp_{i+1}$ hinzu. Darüber hinaus wird die Kante $\perp_0 \xrightarrow{a_0} \top_0$ und für alle $i \in \{0, \ldots, \lambda - 1\}$ die Kante $\top_i \xrightarrow{b_i} f_{i,0}$ und wenn $i < \lambda - 1$, dann auch die Kante $\top_i \xrightarrow{a_{i+1}} \top_{i+1}$ hinzugefügt. Schließlich hat A die Kante $\perp_0 \xrightarrow{c_1} \triangle_1$ und für alle $i \in \{1, \ldots, m-1\}$ die Kante $\triangle_i \xrightarrow{d_i} g_{i,0}$ und wenn $i < m-1$, dann ebenso die Kante $\triangle_i \xrightarrow{c_{i+1}} \triangle_{i+1}$.

Lemma 7.1. *Wenn es einen Zeugen $\mathcal{R}$ (ϱ, κ)-begrenzter Regionen für die τ-EZTE von A gibt, dann gibt es eine treffende Menge mit höchstens λ Elementen für $(\mathfrak{U}, M)$.*

Beweis. Weil $\mathcal{R}$ die τ-EZTE von A bezeugt, gibt es eine (ϱ, κ)-begrenzte Region $R \in \mathcal{R}$, die $\alpha = (k, t_{0,1})$ löst. Sei $R = (sup, sig)$ eine solche Region, dass heißt, $sig^-(k) > sup(t_{0,1})$ und $|{}^\bullet R| \leq 2\lambda$ und $|R^\bullet| \leq \lambda + 1$. Wir werden nun argumentieren, dass $S = \{X \in \mathfrak{U} \mid sig^+(X) > 0\}$ eine gesuchte treffende Menge für $(\mathfrak{U}, M)$ definiert.

Weil k an $t_{0,0}$ stattfindet und R das EZTP α löst, gelten die folgenden (Un-) Gleichungen: (1) $sig^-(k) \leq sup(t_{0,0})$ und (2) $sup(t_{0,1}) = sup(t_{0,0}) - sig^-(k) + sig^+(k)$ und (3) $sig^-(k) > sup(t_{0,1})$. Aus (1) und (2) erhalten wir $sup(t_{0,1}) \geq sig^+(k)$. Mit $sup(t_{0,1}) \geq sig^+(k)$ und (3), erhalten wir $sig^-(k) > sig^+(k)$. Für alle $i \in \{0, \ldots, \lambda - 1\}$ schließen wir aus $sig^-(k) > sig^+(k)$ und $f_{i,0} \xrightarrow{k} f_{i,1}$ auf $sup(f_{i,0}) > sup(f_{i,1})$, denn $sup(f_{i,1}) = sup(f_{i,0}) - sig^-(k) + sig^+(k)$. Dies impliziert $sig^-(k_i) > sig^+(k_i)$ sowie $sig^-(z_i) < sig^+(z_i)$ und somit auch $k_i \in R^\bullet$ und $z_i \in {}^\bullet R$ für alle $i \in \{0, \ldots, \lambda - 1\}$. Wegen $k \in R^\bullet$ und $|R^\bullet| \leq \kappa$, erhalten wir bereits $|R^\bullet| = \lambda + 1$. Insbesondere kann kein weiteres Ereignis von A ein Element von $R^\bullet$ sein.

Sei nun $i \in \{1, \ldots, m-1\}$ beliebig aber fest. Wenn $sig^+(u_i) > sig^-(u_i)$, dann erhalten wir $sup(g_{i,0}) > sup(g_{i,1})$ und somit auch $sig^-(v_i) > sig^+(v_i)$, wegen der Kanten $g_{i,1} \xrightarrow{u_i} g_{i,0}$ und $g_{i,0} \xrightarrow{v_i} g_{i,1}$. Daraus würden $v_i \in R^\bullet$ und $|R^\bullet| \geq \lambda + 2$ folgen, im

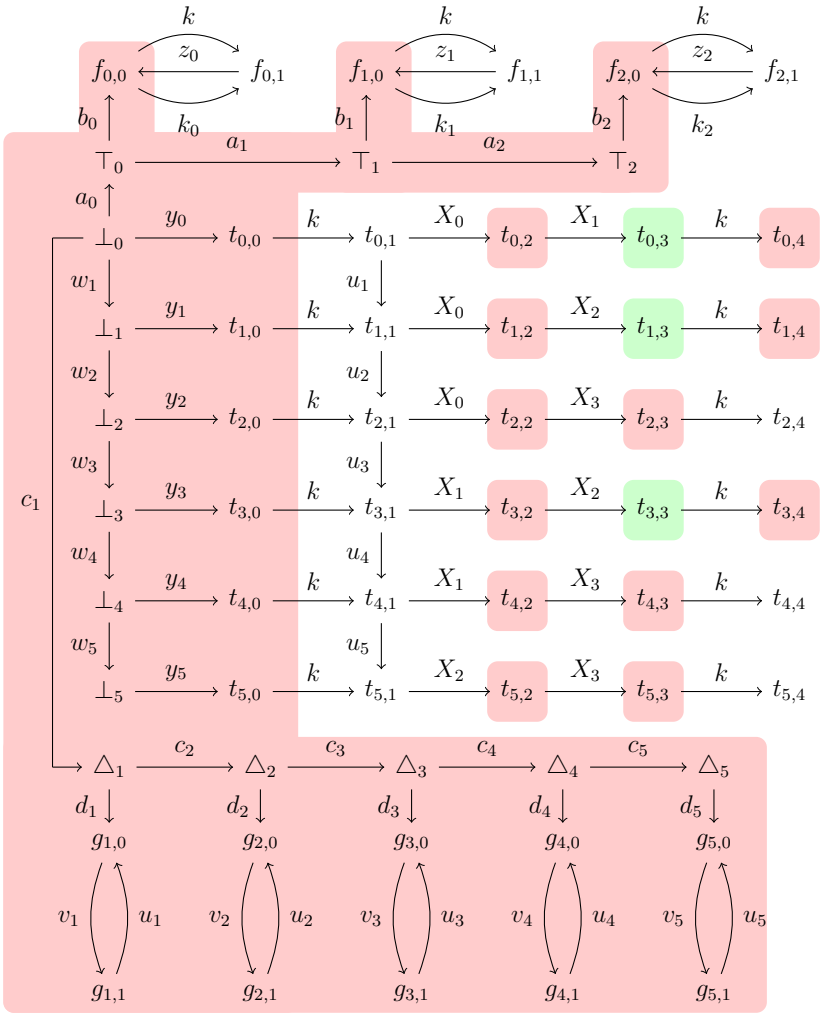


Abbildung 7.2: Das Transitionssystem A mit Initialzustand $\bot_0$, das gemäß der Reduktion aus Beispiel 7.3 resultiert. Der farbige Bereich skizziert die Region R_1 aus Fakt 7.1, die auf der treffenden Menge $S = \{X_0, X_1, X_2\}$ basiert und die das EZTP $\alpha = (k, t_{0,1})$ löst: der rote Bereich korrespondiert mit Zuständen mit Supportwert 1, der grüne mit Zuständen mit Supportwert 2 und die anderen Zustände haben den Supportwert 0.

Widerspruch dazu, dass R (ϱ, κ)-begrenzt ist. Deshalb gilt $sig^+(u_i) \leq sig^-(u_i)$. Wegen $t_{i-1,1} \xrightarrow{u_i} t_{i,1}$, impliziert dies $sup(t_{i-1,1}) \geq sup(t_{i,1})$. Weil i beliebig war, erhalten wir $sup(t_{0,1}) \geq sup(t_{1,1}) \geq \cdots \geq sup(t_{m-1,1})$. Mit $sig^-(k) > sup(t_{0,1})$ impliziert dies $sig^-(k) > sup(t_{i,1})$ für alle $i \in \{0, \ldots, m-1\}$. Andererseits schließen wir aus $t_{i,m_i+1} \xrightarrow{k}$, dass $sig^-(k) \leq sup(t_{i,m_i+1})$ und somit $sup(t_{i,0}) < sup(t_{i,m_i+1})$ für alle $i \in \{0, \ldots, m-1\}$ gilt. Folglich muss es ein Ereignis $X \in \{X_{i_0}, \ldots, X_{i_{m_i-1}}\}$ geben, sodass $sig^+(X) > 0$. Daraus folgt $S \cap M_i \neq \emptyset$ für alle $i \in \{0, \ldots, m-1\}$, wobei $S = \{X \in \mathfrak{U} \mid sig^+(X) > 0\}$.

Weil nun $z_i \in {}^\bullet R$ für alle $i \in \{0, \ldots, \lambda - 1\}$ und $S \subseteq {}^\bullet R$ sowie $|{}^\bullet R| \leq 2\lambda$ gelten, erhalten wir $|S| \leq \lambda$. Die beweist die Behauptung und somit das Lemma. □

Für die andere Richtung werden wir nun zeigen, dass alle EZTP und alle ZTP von A durch (ϱ, κ)-begrenzte τ-Regionen lösbar sind, wenn es eine treffende Menge mit höchstens λ Elementen für $(\mathfrak{U}, M)$ gibt. Dabei folgt die Lösbarkeit der ZTP im Wesentlichen bereits durch die Regionen, welche die EZTP von A lösen. Sei S im Folgenden also eine beliebige aber feste treffende Menge mit höchstens λ Elementen für $(\mathfrak{U}, M)$. Wir behandeln die einzelnen Ereignisse von A individuell. Der Einfachheit halber wird eine Region $R = (sup, sig)$ implizit mittels $sup(\bot_0)$ und sig definiert. Des Weiteren fassen wir Ereignisse gleicher Signatur in einer Menge $\mathcal{E}^R_{m,n} = \{e \in E(A) \mid sig(e) = (m, n)\}$ zusammen. Dabei gilt, dass alle Ereignisse, die nicht explizit in einer solchen Menge $\mathcal{E}^R_{m,n}$ mit $m \neq n$ aufgelistet sind, den Signaturwert $(0, 0)$ haben. Wir werden deshalb die Menge $\mathcal{E}^R_{0,0}$ jeweils nicht explizit definieren.

Fakt 7.1. *Das Ereignis k ist durch (ϱ, κ)-begrenzte τ-Regionen lösbar.*

Beweis. Die folgende Region $R_0 = (sup_0, sig_0)$ löst (k, z) für alle $z \in \bigcup_{i=1}^{m-1} Z(G_i)$ und alle $z \in \{\top_0, \ldots, \top_{\lambda-1}\}$ und alle $z \in \{\triangle_1, \ldots, \triangle_{m-1}\}$: $sup_0(\bot_0) = 1$ und $\mathcal{E}^{R_0}_{1,1} = \{k\}$ und $\mathcal{E}^{R_0}_{0,1} = \{b_0, \ldots, b_{\lambda-1}\}$ und $\mathcal{E}^{R_0}_{1,0} = \{a_0, c_1\}$. Diese Region erfüllt $|{}^\bullet R_0| = \lambda + 1 \leq 2\lambda$ und $|R_0^\bullet| = 3$.

Die folgende Region $R_1 = (sup_1, sig_1)$ löst $\alpha = (k, t_{0,1})$ sowie (k, z) für alle $z \in \{f_{i,1} \mid i \in \{0, \ldots, \lambda - 1\}\}$: $sup_1(\bot_0) = 1$; $\mathcal{E}^{R_1}_{1,0} = \{k, k_0, \ldots, k_{\lambda-1}\}$ und $\mathcal{E}^{R_1}_{0,1} = S \cup \{z_0, \ldots, z_{\lambda-1}\}$. Diese Region erfüllt $|{}^\bullet R_1| \leq 2\lambda$, weil $|S| \leq \lambda$, und $|R_1^\bullet| = \lambda + 1$.

Die folgende Region $R_2 = (sup_2, sig_2)$ löst $\alpha = (k, z)$ für alle $z \in \{t_{i,1}, t_{i,m_i+2} \mid i \in \{0, \ldots, m - 1\}\}$: $sup_2(\bot_0) = 2$ und $\mathcal{E}^{R_2}_{1,0} = \{k, k_0, \ldots, k_{\lambda-1}\}$ und $\mathcal{E}^{R_2}_{0,1} = \{z_0, \ldots, z_{\lambda-1}\}$. Diese Region erfüllt $|{}^\bullet R_2| = \lambda \leq 2\lambda$ und $|R_2^\bullet| = \lambda + 1$.

Die folgende Region $R_3 = (sup_3, sig_3)$ löst (k, z) für $z = \bot_0$ und alle Zustände $z \in \{t_{0,2}, \ldots, t_{0,m_0}\}$: $sup_3(\bot_0) = 0$ und $\mathcal{E}^{R_3}_{1,1} = \{k\}$ und $\mathcal{E}^{R_3}_{0,1} = \{y_0, u_1, X_{0_{m_0-1}}, a_0, c_1\}\}$ und $\mathcal{E}^{R_3}_{0,2} = \{w_1\}$ und $\mathcal{E}^{R_3}_{1,0} = \{X_{0_0}, v_1\}$.

Sei $i \in \{1, \ldots, m - 1\}$ beliebig aber fest. Die folgende Region $R_4 = (sup_4, sig_4)$ löst (k, z) für alle $z \in \{\bot_i, t_{i,2}, \ldots, t_{i,m_i}\}$: $sup_4(\bot_0) = 2$ und $\mathcal{E}^{R_4}_{1,1} = \{k\}$ und $\mathcal{E}^{R_4}_{2,0} = \{w_i\}$ und $\mathcal{E}^{R_4}_{0,2} = \{w_{i+1}\}$ und $\mathcal{E}^{R_4}_{0,1} = \{y_i, v_i, u_{i+1}, X_{i_{m_i-1}}\}$ und $\mathcal{E}^{R_4}_{1,0} = \{u_i, v_{i+1}, X_{i_0}\}$.
Aus der Beliebigkeit von i folgt die Behauptung. □

Fakt 7.2. *Wenn $e \in \{u_1, \ldots, u_{m-1}\}$, dann ist e durch (ϱ, κ)-begrenzte τ-Regionen lösbar.*

Beweis. Sei $i \in \{1, \ldots, m-1\}$ beliebig aber fest. Die folgende Region $R_5 = (sup_5, sig_5)$ löst (u_i, z) für alle $z \in Z(A) \setminus (Z(T_{i-1}) \cup \{g_{i-1,0}\})$ mit $\neg z \xrightarrow{u_i}$: Wenn $i = 1$, dann $sup_5(\bot_0) = 1$, sonst $sup(\bot_0) = 0$; wenn $i = 1$, dann $\mathcal{E}^{R_5}_{1,0} = \{u_i, w_i, v_{i-1}\} \cup \{a_0, c_1\}$, sonst $\mathcal{E}^{R_5}_{1,0} = \{u_i, w_i, v_{i-1}\}$; außerdem $\mathcal{E}^{R_5}_{0,1} = \{w_{i-1}, u_{i-1}, d_{i-1}, v_i\}$.

Die folgende Region $R_6 = (sup_6, sig_6)$ löst (u_i, z) für alle $z \in \{\bot_{i-1}, t_{i-1,0}\}$ und wenn $i \geq 2$, dann für $z = g_{i-1,0}$: $sup_6(\bot_0) = 0$ und $\mathcal{E}_{1,1}^{R_6} = \{u_i\}$ und $\mathcal{E}_{0,1}^{R_6} = \{d_i, k, k_0, \ldots, k_{\lambda-1}\}$ und $\mathcal{E}_{1,0}^{R_6} = \{z_0, \ldots, z_{\lambda-1}\}$. Es gilt $|{}^\bullet R_6| = \lambda + 3 \leq 2\lambda$, wegen $\lambda \geq 6$, sowie $|R_6^\bullet| = \lambda + 1$.

Die folgende Region $R_7 = (sup_7, sig_7)$ löst (u_i, z) für alle $z \in \{t_{i-1,2}, \ldots, t_{i-1,m_i+2}\}$: $sup_7(\bot_0) = 1$ und $\mathcal{E}_{1,1}^{R_7} = \{u_i\}$ und $\mathcal{E}_{1,0}^{R_7} = \{X_{i_0}\}$. Dies komplettiert den Beweis für u_i und weil i beliebig war, folgt die Behauptung. □

Fakt 7.3. *Wenn $e \in \{X_0, \ldots, X_{n-1}\}$, dann ist e durch (ϱ, κ)-begrenzte τ-Regionen lösbar.*

Beweis. Sei $i \in \{0, \ldots, n-1\}$ beliebig aber fest. Seien außerdem $j, \ell \in \{0, \ldots, m-1\}$ beliebig aber fest, sodass $X_i \notin M_j$ und $X_i \in M_\ell$.

Die folgende Region $R_8 = (sup_8, sig_8)$ löst (X_i, z) für $z = \bot_j$ und für alle Zustände $z \in \{t_{j,0}, \ldots, t_{j,m_j+2}\}$: Wenn $j = 0$, dann $sup_8(\bot_0) = 0$, sonst $sup(\bot_0) = 1$; $\mathcal{E}_{1,1}^{R_8} = \{X_i\}$ und wenn $j > 0$, dann $\mathcal{E}_{1,0}^{R_8} = \{w_j, u_j, v_{j+1}\} \cup \{a_0, c_1\}$, sonst $\mathcal{E}_{1,0}^{R_8} = \{w_j, u_j, v_{j+1}\}$; $\mathcal{E}_{0,1}^{R_8} = \{v_j, w_{j+1}, u_{j+1}, d_{j+1}\}$.

Die folgende Region $R_9 = (sup_9, sig_9)$ löst (X_i, z) für alle $z \in \{\bot_\ell, t_{\ell,0}\}$: $sup_9(\bot_0) = 0$; $\mathcal{E}_{1,1}^{R_9} = \{X_i\}$ und $\mathcal{E}_{1,0}^{R_9} = \{z_0, \ldots, z_{\lambda-1}\}$ und $\mathcal{E}_{0,1}^{R_9} = \{k, k_0, \ldots, k_{\lambda-1}\}$ und $\mathcal{E}_{0,1}^{R_9} = E \setminus (\mathcal{E}_{1,1}^{R_9} \cup \mathcal{E}_{1,0}^{R_9} \cup \mathcal{E}_{0,1}^{R_9})$. Es gelten offenbar $|R_9^\bullet| = \lambda + 1$ und $|{}^\bullet R_9| = \lambda + 1 \leq 2\lambda$.

Sei $h \in \{0, \ldots, m_\ell - 1\}$ der eindeutige Index, sodass $X_i = X_{\ell_h}$, das heißt, X_i ist das „h-te“ Element von M_ℓ. Die folgende Region $R_{10} = (sup_{10}, sig_{10})$ löst (X_i, z) für alle $z \in \{t_{\ell,h+1}, t_{\ell,m_\ell+2}\}$: $sup_{10}(\bot_0) = 1$ und $\mathcal{E}_{1,0}^{R_{10}} = \{X_i, a_0, c_1\}$.

Es verbleibt noch den Fall $X_i \neq X_{\ell_0}$ zu diskutieren, dass heißt $h \geq 1$, der erfordert, dass wir (X_i, z) für alle $z \in \{t_{\ell,1}, \ldots, t_{\ell,h}\}$ lösen. Sei also $z \in \{t_{\ell,1}, \ldots, t_{\ell,h}\}$ beliebig aber fest.

Wir unterscheiden zwischen $\ell = 0$ und $\ell \geq 1$: Wenn $\ell = 0$, dann löst die folgende Region $R_{11} = (sup_{11}, sig_{11})$ das EZTP (X_i, z): $sup_{11}(\bot_0) = 0$ und $\mathcal{E}_{1,0}^{R_{11}} = \{X_i, v_1\}$ und $\mathcal{E}_{0,1}^{R_{11}} = \{w_1, u_1, d_1, X_{0_{h-1}}\}$.

Wenn $\ell \geq 1$, dann löst die folgende Region $R_{12} = (sup_{12}, sig_{12})$ das EZTP (X_i, z): $sup_{12}(\bot_0) = 1$ und $\mathcal{E}_{1,0}^{R_{12}} = \{X_i, w_\ell, u_\ell, v_{\ell+1}, a_0, c_1\}$ und $\mathcal{E}_{0,1}^{R_{12}} = \{X_{i_{h-1}}, w_{\ell+1}, u_{\ell+1}, v_\ell, d_{\ell+1}\}$. Wegen der Beliebigkeit von h komplettiert dies die Lösbarkeit von (X_i, z) für alle $z \in Z(T_\ell)$.

Die folgende Region $R_{13} = (sup_{13}, sig_{13})$ löst (X_i, z) für alle $z \in Z(A) \setminus (\bigcup_{j=0}^{m-1}(Z(T_j) \cup \{\bot_j\})$: $sup_{13}(\bot_0) = 1$ und $\mathcal{E}_{1,1}^{R_{13}} = \{X_i\}$ und $\mathcal{E}_{1,0}^{R_{13}} = \{a_0, c_1\}$.

Aus der Beliebigkeit von i und j sowie ℓ folgt die Behauptung. □

Der folgende Fakt adressiert die verbleibenden Ereignisse von A:

Fakt 7.4. *Wenn $e \in \{k_0, \ldots, k_{\lambda-1}\}$ oder $e \in \{z_0, \ldots, z_{\lambda-1}\}$ oder $e \in \{v_1, \ldots, v_{m-1}\}$ oder $e \in \{w_1, \ldots, w_{m-1}\}$ oder $e \in \{a_0, \ldots, a_{\lambda-1}\}$ oder $e \in \{b_0, \ldots, b_{\lambda-1}\}$ oder $e \in \{y_0, y_i, w_i, c_i, d_i \mid 1 \leq i \leq m-1\}$, dann ist e durch (ϱ, κ)-begrenzte τ-Regionen lösbar.*

Beweis. Sei $i \in \{0, \ldots, \lambda - 1\}$ beliebig aber fest. Die Region R_1 von Fakt 7.1 löst $(k_i, f_{i,1})$ und die Region R_5 von Fakt 7.2 löst $(z_i, f_{i,0})$. Wie man leicht sieht, sind (k_i, z) und (z_i, z) passend für alle $z \in Z(A) \setminus \{f_{i,0}, f_{i,1}\}$ lösbar. Weil i beliebig war, beweist dies die Behauptung für alle $e \in \{k_0, \ldots, k_{\lambda-1}, z_0, \ldots, z_{\lambda-1}\}$.

Sei $i \in \{1, \ldots, m-1\}$ beliebig aber fest. Die Region R_5 oder die Region R_7 löst $(v_i, g_{i,1})$. Es ist leicht zu sehen, dass (v_i, z) passend für alle $z \in Z(A) \setminus \{g_{i,0}, g_{i,1}\}$ lösbar ist.

Da die verbleibenden Ereignisse jeweils nur einmal in A stattfinden, sieht man leicht, dass auch sie durch passend begrenzte Regionen lösbar sind. Die beweist den Fakt. □

Insgesamt beweisen die gerade präsentierten Fakten, dass es einen Zeugen (ϱ, κ)-begrenzter τ-Regionen für die τ-EZTE von A gibt, wenn es eine treffende Menge mit höchstens λ Elementen für $(\mathfrak{U}, M)$ gibt. Wenn (z, z') ein ZTP von A ist, dann gilt außerdem, dass (z, z') entweder bereits durch eine der diskutierten Regionen gelöst wird oder es ist leicht zu sehen, dass eine lösende (ϱ, κ)-begrenzte τ-Region existiert. Wir erhalten damit das folgende Lemma, welches den Beweis von Satz 7.1 vervollständigt.

Lemma 7.2. *Wenn es eine treffende Menge mit höchstens λ Elementen für $(\mathfrak{U}, M)$ gibt, dann gibt es einen Zeugen (ϱ, κ)-begrenzter Regionen für die τ-EZTE und die τ-ZTE von A.*

7.3 τ_{PT}-SMBPB und τ_{PT}-RMBPB mit Parameter $\varrho+\kappa$

Gemäß Satz 7.1 sind die Probleme τ-SMBPB und τ-RMBPB NP-vollständig und daher aus Sicht der klassischen Komplexität, das heißt unter der Annahme $\mathsf{P} \neq \mathsf{NP}$, nicht effizient lösbar. Insbesondere erwarten wir, dass es keinen Lösungsalgorithmus dieser Probleme gibt, der in Abhängigkeit von der Eingabelänge polynomielle Laufzeit hat.

In der parametrisierten Komplexität messen wir den Aufwand von Entscheidungsproblemen nicht nur in Abhängigkeit von der Eingabelänge, sondern nehmen auch einen Parameter in den Blick. Der natürliche Parameter von τ-SMBPB und τ-RMBPB ist $\varrho + \kappa$.

In Abschnitt 7.1 wurde gezeigt, wie für eine Eingabe (A, ϱ, κ) nicht-deterministisch in Polynomialzeit entschieden werden kann, ob (A, ϱ, κ) für τ-SMBPB oder τ-RMBPB eine positive Entscheidung erlauben: Für jedes Trennungsproblem α von A wird nicht-deterministisch ein (Un-) Gleichungssystem konstruiert, dessen Lösbarkeit über $\mathbb{Q}$ mit der Lösbarkeit von α durch eine (ϱ, κ)-begrenzte Region äquivalent ist. Dabei beschränkt sich die nicht-deterministische Berechnung im Wesentlichen auf das Raten der der Indizes $i_1, \ldots, i_\varrho$ und $j_1, \ldots, j_\kappa$ der Ereignisse, die (potentiell) konsumieren beziehungsweise produzieren dürfen.

Wenn $m = |A| + \log(\varrho) + \log(\kappa)$, dann gibt es höchstens $\binom{m}{\varrho}$ beziehungsweise $\binom{m}{\kappa}$ Möglichkeiten $\{e_{i_1}, \ldots, e_{i_\varrho}\}$ beziehungsweise $\{e_{j_1}, \ldots, e_{j_\kappa}\}$ zu wählen. Um die Lösbarkeit von α durch eine (ϱ, κ)-begrenzte Region zu entscheiden, müssen folglich höchstens $\mathcal{O}(m^{\varrho+\kappa})$

(Un-) Gleichungssysteme gemäß der in Abschnitt 7.1 diskutierten Bauart auf ihre Lösbarkeit überprüft werden. Die Konstruktion jedes Systems und der Test seiner Lösbarkeit kann (deterministisch) mit polynomiellen Zeitaufwand in m durchgeführt werden, das heißt mit Zeitaufwand $\mathcal{O}(m^c)$ für eine Konstante c. Weil A höchstens $|Z| \cdot |E| + |Z|^2$ Trennungsprobleme hat und diese Anzahl offenbar polynomiell in m ist, gibt es also eine Konstante c', sodass τ-SMBPB und τ-RMBPB deterministisch mit Zeitaufwand $\mathcal{O}(m^{\varrho+\kappa+c'})$ gelöst werden können. Die bedeutet, dass (die parametrisierten Version von) τ-SMBPB und τ-RMBPB mit Parameter $\varrho + \kappa$ in XP sind.

Wenn ϱ und κ darüber hinaus im Voraus fest gewählt sind, dann ist $k = \varrho + \kappa + c'$ eine Konstante. Jede Lösung eines Systems definiert bereits eine lösende Region für das zu Grunde liegende Trennungsproblem α und jeder Zeuge $\mathcal{R}$ (für die ZTP oder EZTP) definiert bereits ein gesuchtes Netz $N = N_A^{\mathcal{R}}$. Dies bedeutet einerseits, dass im Fall fest gewählter ϱ, κ in Polynomialzeit entschieden werden kann, ob A durch ein τ-Netz mit (ϱ, κ)-beschränkten Plätzen implementiert werden kann. Andererseits kann eine gesuchtes Netz in Polynomialzeit konstruiert werden, wenn es ein solches gibt.

Schließlich ist die in Abschnitt 7.2 diskutierte Reduktion eine parametrisierte Reduktion für τ-SMBPB und τ-RMBPB mit Parameter $\varrho + \kappa$, denn $\varrho + \kappa = 3\lambda + 1$. Weil Hitting Set eine bekanntes W[2]-vollständiges Problem ist, folgt, dass τ-SMBPB und τ-RMBPB mit Parameter $\varrho + \kappa$ nicht fest-Parameter-berechenbar sind.

Wir fassen diese Überlegungen in einem Satz zusammen:

Satz 7.2. *1. Wenn $\varrho, \kappa \in \mathbb{N}$ beliebig aber fest gewählt sind, dann kann für ein gegebenes Transitionssystem A in Polynomialzeit entschieden werden, ob es durch eine τ-Einbettung oder eine τ-Sprachsimulation oder eine τ-Realisierung N implementiert werden kann, deren Plätze (ϱ, κ)-begrenzt sind.*

2. τ-SMBPB und τ-RMBPB mit Parameter $\varrho + \kappa$ sind sind in XP.

3. τ-SMBPB und τ-RMBPB mit Parameter $\varrho + \kappa$ sind W[2]-schwer.

7.4 Zusammenfassung und Ausblick

In diesem Kapitel wurde zunächst gezeigt, dass τ-SMBPB und τ-RMBPB NP-vollständig sind. Für zukünftige Arbeit bleibt zu untersuchen, ob τ-SMBPB und τ-RMBPB auch für den Netztyp τ_{PPT} der puren P/T-Netze NP-vollständig sind. Analog zur Argumentation in Abschnitt 7.1 erhält man, dass τ-SMBPB und τ-RMBPB auch für τ_{PPT} in NP sind. Die Funktionalität der Reduktion in Abschnitt 7.2 basiert jedoch wesentlich auf der Möglichkeit, EZTP (e, z) mittels Regionen $R = (sup, sig)$ lösen zu können, die $sig(e) = (m, m)$ und $m > sup(z)$ erlauben. Insbesondere sind die Ereignisse aus $u_0, \ldots, u_{m-1}$ nicht (allein) durch pure Regionen lösbar. Andererseits sind diese Ereignisse notwendig, um $sup(t_{0,1}) \geq sup(t_{1,1}) \geq \cdots \geq sup(t_{m-1,1})$ für eine (ϱ, κ)-Region

$R = (sup, sig)$, die α löst, zuzusichern. Es wäre also interessant zu wissen, ob die Reduktion so modifiziert werden kann, dass τ_{PPT}-Regionen zu Lösbarkeit aller Ereignisse ausreichen.

Darüber hinaus wurde in diesem Kapitel argumentiert, dass τ-SMBPB und τ-RMBPB mit Parameter $\varrho + \kappa$ zwar in XP, jedoch nicht fest-Parameter-berechenbar sondern jeweils $\mathsf{W}[2]$-schwer sind. Es wäre natürlich interessant zu wissen, wo genau τ-SMBPB und τ-RMBPB mit Parameter $\varrho + \kappa$ in der W-Hierarchie zu positionieren sind.

Die Probleme τ-SMBPB und τ-RMBPB stellen die Frage nach der Implementierbarkeit von Transitonssystemen durch τ-Netze mit beschränkten Platz-Bereichen. Analog kann Synthese Netze adressieren, deren Transitionen begrenzte Bereiche haben. Die Charakterisierung der Komplexität der zugehörigen Entscheidungsprobleme kann Gegenstand zukünftiger Forschung sein.

Kapitel 8

Boolesche Netze mit beschränkter Belegungszahl

Während der Begriff der Platzumgebung offenbar einen strukturellen Aspekt betrifft, adressiert die *Belegungszahl* eines booleschen Netzes einen Aspekt des Verhaltens und wird erst über Entwicklung der Schaltvorgänge und der sich daraus ergebenden Markierungen interessant:

Definition 8.1 (Belegungszahl)**.** Sei τ ein boolescher Netztyp und $N = (P, T, f, M_0)$ ein τ-Netz sowie A_N der Erreichbarkeitsgraph von N. Für einen Platz $p \in P$ ist die *Belegungszahl* ϱ_p definiert als die Anzahl $|\{z \in Z(A_N) \mid z(p) = 1\}|$ der erreichbaren Markierungen von N, in denen p eine Marke enthält. Die *Belegungszahl* ϱ_N des Netzes N ist definiert als $\varrho_N = \max\{\varrho_p \mid p \in P\}$.

Plätze boolescher Netze modellieren Eigenschaften eines Systems, die genau dann erfüllt sind, wenn die Plätze Marken enthalten. Dabei kann es sich auch um Eigenschaften handeln, die grundsätzlich zwar beachtet und daher modelliert werden müssen, bestenfalls aber nur selten erfüllt sind. Aus der Perspektive der Synthese gilt es also neben der Verteilbarkeit des modellierten Systems auch die Frage in den Blick zu nehmen, ob Plätze (Eigenschaften) im Erreichbarkeitsgraphen (Systemverhalten) nicht „zu häufig“ eine Marke enthalten (erfüllt sind): Die *Synthese von booleschen Netzen mit begrenzter Belegungszahl* adressiert die Aufgabe, für ein Transitionssystem A ein implementierendes Netz zu finden, dessen Belegungszahl begrenzt ist.

In diesem Kapitel werden wir (zwar auch die klassische, aber vordergründig) die parametrisierte Komplexität dieser Formulierung des Syntheseproblems für die booleschen Netze untersuchen, die Unabhängigkeit erlauben, das heißt nop enthalten. Genauer werden wir zeigen, dass für jeden solcher Netztypen τ und jede im Voraus fest gewählte Zahl ϱ in Polynomialzeit entschieden werden kann, ob ein Transitionssystem durch ein τ-Netz implementiert werden kann, sodass $\varrho_N \leq \varrho$. Wenn ϱ ein Teil der Eingabe wird, erhalten wir für jede der klassischen Implementierungen in natürlicher Weise ein Entscheidungsproblem, das für ein gegebenes Paar (A, ϱ) fragt, ob A durch ein τ-Netz implementiert

werden kann, dessen Belegungszahl durch ϱ begrenzt ist. Man sieht relativ leicht, dass die resultierenden Entscheidungsprobleme NP-vollständig sind, wenn bereits das zu Grunde liegende unbegrenzte Syntheseproblem NP-vollständig ist. Aus Sicht der parametrisierten Komplexität erhalten wir jedoch ein positives Resultat: Wir werden zeigen, dass diese Entscheidungsprobleme (für alle Implementierungen) fest-Parameter berechenbar sind, wenn die Belegungszahl als Parameter gewählt wird. Die Inhalte dieses Abschnitts basieren auf der Publikation [ET20].

Solange nicht ausdrücklich anders vereinbart, sei τ im Folgenden ein beliebiger aber fester boolescher Netztyp mit nop. Gemäß Definition 8.1 entspricht die Belegungszahl eines τ-Netzes N der maximalen Anzahl von erreichbaren Markierungen, in denen ein Platz markiert sein kann. Wir erinnern uns, dass N genau dann eine Einbettung, Sprachsimulation oder Realisierung eines Transitionssystems $A = (Z, E, \delta, \iota)$ ist, wenn es eine Menge $\mathcal{R}$ von τ-Regionen von A gibt, welche die zu der Implementierung passende Trennungseigenschaft von A nachweist und $N = N_A^{\mathcal{R}}$ erfüllt. Jede Region $R = (sup, sig) \in \mathcal{R}$ ist gemäß Definition 2.20 ein Platz in N. Wenn φ die Simulation zwischen A und A_N ist, welche die Art der Implementierung charakterisiert, dann gilt für jeden Zustand $z \in Z$ genau dann $sup(z) = 1$, wenn $\varphi(z)(R) = 1$ für den Platz R in der Markierung $\varphi(z) \in Z(A_N)$ erfüllt ist, die dem Bild von z unter φ entspricht. Die Belegungszahl ϱ_R des Platzes $R = (sup, sig)$ in N entspricht deshalb der *Belegungszahl der Region R*, das heißt der Anzahl der Zustände von A, deren Wert unter *sup* positiv ist:

Definition 8.2 (Belegungszahl einer Region)**.** Sei $A = (Z, E, \delta, \iota)$ ein Transitionssystem und $R = (sup, sig)$ eine τ-Region von A. Die *Belegungszahl* ϱ_R von R ist $\varrho_R = |\{z \in Z \mid sup(z) = 1\}|$.

In diesem Abschnitt adressieren wir die Synthese von τ-Netzen, deren Belegungszahl eine gegebene natürliche Zahl nicht überschreiten soll. Mit den bisherigen Überlegungen erhalten wir die folgenden parametrisierten Entscheidungsprobleme, deren parametrisierte Komplexität Gegenstand unserer Untersuchungen ist:

τ-Realisierung mit beschränkter Belegungszahl (τ-RMBBZ)	
Eingabe:	Ein Transitionssystem $A = (Z, E, \delta, \iota)$ und eine natürliche Zahl ϱ.
Parameter:	ϱ.
Frage:	Gibt es einen Zeugen $\mathcal{R}$ für die τ-ZTE und die τ-EZTE von A mit $\lvert\{z \in Z \mid sup(z) = 1\}\rvert \leq \varrho$ für alle Regionen $(sup, sig) \in \mathcal{R}$?

τ-Sprachsimulation mit beschränkter Belegungszahl (τ-SMBBZ)	
Eingabe:	Ein Transitionssystem $A = (Z, E, \delta, \iota)$ und eine natürliche Zahl ϱ.
Parameter:	ϱ.
Frage:	Gibt es einen Zeugen $\mathcal{R}$ für die τ-EZTE von A mit $\lvert\{z \in Z \mid sup(z) = 1\}\rvert \leq \varrho$ für alle Regionen $(sup, sig) \in \mathcal{R}$?

τ-Einbettung mit beschränkter Belegungszahl (τ-EMBBZ)	
Eingabe:	Ein Transitionssystem $A = (Z, E, \delta, \iota)$ und eine natürliche Zahl ϱ.
Parameter:	ϱ.
Frage:	Gibt es einen Zeugen $\mathcal{R}$ für die τ-ZTE von A mit $\lvert\{z \in Z \mid sup(z) = 1\}\rvert \leq \varrho$ für alle Regionen $(sup, sig) \in \mathcal{R}$?

Es sei bemerkt, dass jedes der drei Entscheidungsprobleme τ-RMBBZ und τ-SMBBZ sowie τ-EMBBZ NP-vollständig ist, wenn das zu Grunde liegende unbeschränkte Problem NP-vollständig ist: Die Zugehörigkeit zu NP ist klar, denn ein Zeuge beschränkter Regionen für die jeweilige Trennungseigenschaft τ-ZTE oder τ-EZTE kann in Polynomialzeit geraten werden. Darüber hinaus gilt für einen Zeugen $\mathcal{R}$ für die τ-ZTE oder τ-EZTE eines Transitionssystems $A = (Z, E, \delta, \iota)$ sicherlich, dass $\varrho_R \leq |Z|$ für alle $R \in \mathcal{R}$ erfüllt ist. Deshalb kann die unbeschränkte Version eines Problems trivial auf die beschränkte reduziert werden, wobei A auf eine Eingabe $(A, |Z|)$ abgebildet wird. Für 84 der 128 booleschen Netztypen mit nop folgt die NP-Vollständigkeit von τ-RMBBZ und τ-SMBBZ damit aus den Ergebnissen aus Abschnitt 4.1 (siehe die Netztypen A1, A2, A6 - A9 aus Abbildung 4.3) und für 92 der 128 booleschen Netztypen mit nop folgt die NP-Vollständigkeit aus den Resultaten aus Abschnitt 4.3 (siehe die Netztypen B1, B2, B4 - B6 und B8 aus Abbildung 4.8).

Bessere Nachrichten erhalten wir aus Sicht ihrer parametrisierten Komplexität, die wir als Hauptresultat dieses Kapitels formulieren:

Satz 8.1. *τ-RMBBZ und τ-SMBBZ und τ-EMBBZ sind fest-Parameter-berechenbar.*

Der Rest dieses Kapitels ist dem Beweis von Satz 8.1 gewidmet. Solange nicht ausdrücklich anders vereinbart, sei $A = (Z, E, \delta, \iota)$ im weiteren Verlauf dieses Kapitels ein beliebiges aber festes Transitionssystem. Wie in Kapitel 4 werden wir im Folgenden einerseits jede Teilmenge $Q \subseteq Z$ von Zuständen von A mit ihrer charakteristischen Funktion $Q : Z \to \{0, 1\}$ identifizieren, dass heißt, es gilt genau dann $Q(z) = 1$, wenn $z \in Q$. Andererseits werden wir jede Funktion $Q : Z \to \{0, 1\}$ –und damit insbesondere jeden Support *sup* einer τ-Region (sup, sig)– mit der Menge der Zustände identifizieren, deren Bild unter Q positiv ist, das heißt $Q = \{z \in Z \mid Q(z) = 1\}$. Des Weiteren erinnern wir daran, dass für ein Ereignis $e \in E$ und eine Teilmenge $Q \subseteq Z$ das Signaturpotential von e bezüglich Q gemäß Definition 4.3 definiert ist als die Menge $S_e^Q = \{(Q(z), Q(z')) \mid z \xrightarrow{e} z' \in A\}$.

8.1 Ein parametrisierter Algorithmus

In diesem Abschnitt werden wir Satz 8.1 beweisen, indem wir einen fest-Parameter-Algorithmus für τ-RMBBZ und τ-SMBBZ sowie τ-EMBBZ vorstellen. Dieser Algorithmus berechnet für jedes Trennungsproblem α von A eine lösende τ-Region R mit durch ϱ beschränkter Belegungszahl ϱ_R, wenn es eine solche gibt. Andernfalls wird A korrekt als

nicht durch ein τ-Netz mit ϱ-beschränkter Belegungszahl implementierbar abgelehnt. Im Falle einer positiven Entscheidung, ist das Kernstück der Berechnungen ein –bezüglich gewisser Grundannahmen, die als Teilmenge Q von Zuständen von A formuliert sind– *minimaler* Support *sup*. Dieser wird dann in kanonischer Weise durch eine *normalisierte* Signatur *sig* zu einer bezüglich Q minimalen Region $R = (sup, sig)$ vervollständigt:

Definition 8.3 (Minimale Region)**.** Eine τ-Region $R = (sup, sig)$ von A heißt bezüglich einer Teilmenge $Q \subseteq Z$ von Zuständen von A *Q-minimal*, wenn $Q \subseteq sup$ und es gilt $|sup| \leq |sup'|$ für alle τ-Regionen $R' = (sup', sig')$ mit $Q \subseteq sup'$.

Bemerkung 8.1 (Existenz minimaler Region). Weil $\mathsf{nop} \in \tau$, gibt es für jede Teilmenge $Q \subseteq Z$ von $A = (Z, E, \delta, \iota)$ eine Q-minimale Region, denn durch $R = (Z, sig)$ mit $sig(e) = \mathsf{nop}$ für alle $e \in E$ ist eine τ-Region von A definiert.

Aufgrund der Definition der Lösbarkeit von Trennungsproblemen, können wir gewisse Grundannahmen über die möglichen Supporte lösender Regionen treffen. Sei zum Beispiel $\alpha = (z, z')$ ein ZTP und $R = (sup, sig)$ eine τ-Region, die α löst. Nach Definition 2.21 erfüllt R sicherlich entweder $z \in sup$ oder $z' \in sup$. In diesem Fall startet der Algorithmus von den *positiven Grundannahmen* $Q = \{z\}$ beziehungsweise $Q' = \{z'\}$ und erweitert diese jeweils zu einem Support *sup* beziehungsweise *sup'* einer minimalen τ-Region, die ϱ berücksichtigt, wenn dies möglich ist. Zunächst ist dabei nicht *a priori* klar, dass der resultierende Support, im Falle seiner Existenz, tatsächlich zu einer lösenden Region gehört. Folglich müssen solche Supporte ausgesiebt werden, welche keine lösende Region ermöglichen. Dies wird realisiert, indem überprüft wird, ob der resultierende Support gewisse *negative Grundannahmen* erfüllt. Im Falle unseres ZTP α erhalten wir zum Beispiel die zu den (positiven) Grundannahmen Q und Q' passenden (negativen) Grundannahmen $F = \{z'\}$ und $F' = \{z\}$: Wenn $Q \subseteq sup$, dann $F \cap sup = \emptyset$ und wenn $Q' \subseteq sup$, dann $F' \cap sup = \emptyset$.

Wenn $\alpha = (e, q)$ ein EZTP von A ist, dann bildet eine lösende τ-Region das Ereignis e auf eine der booleschen Interaktionen $\mathsf{inp}, \mathsf{out}, \mathsf{used}$ oder free ab, denn alle weiteren booleschen Interaktionen sind sowohl für 0 als auch für 1 definiert. Auch in diesem Fall erhalten wir entsprechende positive beziehungsweise negative Grundannahmen bezüglich Zuständen, die der Support einer lösenden Region enthält beziehungsweise nicht erhält. Wir fassen die entsprechenden Zusammenhänge in dem folgenden Lemma zusammen, dessen Aussagen direkt aus der Definition der Lösbarkeit von Trennungsproblemen folgen:

Lemma 8.1 (Ohne Beweis)**.** *Sei (e, q) ein EZTP und (z, z') ein ZTP von A und $R = (sup, sig)$ eine τ-Region von A. Wenn R das EZTP (e, q) löst, dann ist eine der folgenden Bedingungen erfüllt:*

1. *$sig(e) = \mathsf{inp}$ und mit $Q = \{z \in Z \mid z \xrightarrow{e} z' \in A\}$ und $F = \{q\} \cup \{z' \in Z \mid z \xrightarrow{e} z' \in A\}$ gilt $Q \subseteq sup$ sowie $F \cap sup = \emptyset$,*

2. *$sig(e) = \mathsf{out}$ und mit $Q = \{q\} \cup \{z' \in Z \mid z \xrightarrow{e} z' \in A\}$ und $F = \{z \in Z \mid z \xrightarrow{e} z' \in A\}$ gilt $Q \subseteq sup$ sowie $F \cap sup = \emptyset$,*

3. $sig(e) = \mathsf{used}$, $sup(q) = 0$ *und mit* $Q = \{z, z' \in Z \mid z \xrightarrow{e} z' \in A\}$ *und* $F = \{q\}$ *gilt* $Q \subseteq sup$ *sowie* $F \cap sup = \emptyset$,

4. $sig(e) = \mathsf{free}$, $sup(q) = 1$ *und mit* $Q = \{q\}$ *und* $F = \{z, z' \in Z \mid z \xrightarrow{e} z' \in A\}$ *gilt* $Q \subseteq sup$ *sowie* $F \cap sup = \emptyset$.

Wenn R das ZTP (z, z') löst, dann ist eine der folgenden Bedingungen erfüllt:

5. *Mit* $Q = \{z\}$ *und* $F = \{z'\}$ *gilt* $Q \subseteq sup$ *sowie* $F \cap sup = \emptyset$.

6. *Mit* $Q = \{z'\}$ *und* $F = \{z\}$ *gilt* $Q \subseteq sup$ *sowie* $F \cap sup = \emptyset$.

Dem Algorithmus liegt die Idee zu Grunde, eine positive Grundannahme Q solange durch Hinzufügen von Zuständen zu erweitern, bis entweder der Support einer minimalen lösenden Region resultiert oder die Existenz einer solchen Region ausgeschlossen werden kann. Dabei ist einerseits zu beachten, dass jede lösende Region wenigstens eine der in Lemma 8.1 formulierten Grundannahmen erfüllt. Andererseits gibt es genau dann eine Region $R' = (sup', sig')$, die $Q \subseteq sup'$ erfüllt und α löst, wenn es eine Q-minimale Region gibt, die α löst. Salopp gesagt: wenn es überhaupt eine lösende Region gibt, dann sicherlich auch eine, welche den wenigsten Zuständen einen positiven Supportwert zuordnet.

Um eine minimale Region berechnen zu können, suchen wir Obermengen *sup* von Q, sodass die Kanten $z \xrightarrow{e} z'$ von A in dem Sinne *konfliktfrei* (bezüglich *sup*) sind, dass sie eine passende Signatur erlauben:

Definition 8.4 (Kanten in Konflikt)**.** Sei $e \in E$ sowie $Q \subseteq Z$. Eine Kante $z \xrightarrow{e} z'$ von A ist *in Konflikt mit* Q, wenn es keine Interaktion $i \in \tau$ gibt, sodass $Q(z) \xrightarrow{i} Q(z') \in \tau$. Darüber hinaus sind zwei Kanten $z \xrightarrow{e} z'$ und $q \xrightarrow{e} q'$ von A *in Konflikt mit* Q, wenn es keine Interaktion $i \in \tau$ gibt, sodass $Q(z) \xrightarrow{i} Q(z') \in \tau$ und $Q(q) \xrightarrow{i} Q(q') \in \tau$. Kanten sind *konfliktfrei*, wenn sie nicht in Konflikt sind.

Im Wesentlichen besagt das folgende Lemma, dass eine Teilmenge $Q \subseteq Z$ genau dann eine τ-Region $R = (Q, sig)$ von A erlaubt, wenn alle gleichbeschrifteten Kanten paarweise konfliktfrei sind:

Lemma 8.2. *Eine Teilmenge* $Q \subseteq Z$ *erlaubt genau dann eine* τ*-Region* $R = (Q, sig)$ *von* A*, wenn jedes Ereignis* $e \in E$ *von* A *eine der folgenden Bedingungen erfüllt:*

1. *Es gibt genau eine e-beschriftete Kante* $z \xrightarrow{e} z'$ *in* A *und es gibt eine Interaktion* $i \in \tau$*, sodass* $Q(z) \xrightarrow{i} Q(z')$ *ein Kante in* τ *ist.*

2. *Für je zwei e-beschriftete Kanten* $z \xrightarrow{e} z'$ *und* $q \xrightarrow{e} q'$ *gibt es eine Interaktion* $i \in \tau$*, sodass* $Q(z) \xrightarrow{i} Q(z')$ *und* $Q(q) \xrightarrow{i} Q(q')$ *Kanten in* τ *sind.*

Beweis. Wenn $R = (Q, sig)$ eine τ-Region ist, dann gilt mit $i = sig(e)$ sicherlich dass $Q(z) \xrightarrow{i} Q(z')$ beziehungsweise $Q(q) \xrightarrow{i} Q(q')$ Kanten in τ sind.

Für die andere Richtung nehmen wir an, dass die Eigenschaften (1) und (2) gelten und zeigen, dass es für jedes $e \in E$ eine Interaktion $i \in \tau$ gibt, sodass $Q(z) \xrightarrow{i} Q(z') \in \tau$ für alle e-beschrifteten Kanten $z \xrightarrow{e} z'$ erfüllt ist. Sei also $e \in E$ beliebig aber fest. Wenn e genau einmal in A stattfindet, dann gilt die geforderte Bedingung, weil Eigenschaft (1) erfüllt ist.

Nehmen wir nun an, dass es wenigstens zwei e-beschriftete Kanten in A gibt. Analog zu dem Beweis von Lemma 4.2.1 erhalten wir, dass $|S_e^Q| \leq 2$ für das Signaturpotential von S_e^Q bezüglich Q: Wenn $|S_e^Q| \geq 3$, dann gibt es Kanten $z \xrightarrow{e} z'$ und $p \xrightarrow{e} p'$ sowie $q \xrightarrow{e} q'$ sodass $(Q(z), Q(z'))$ und $(Q(p), Q(p'))$ sowie $(Q(q), Q(q'))$ paarweise verschieden sind. Da $Q(z), Q(p), Q(q) \in \{0, 1\}$, gilt o.B.d. A, dass $Q(z) = Q(p)$ und $Q(z') \neq Q(p')$. Folglich gibt es kein $i \in \tau$, sodass $Q(z) \xrightarrow{i} Q(z')$ und $Q(q) \xrightarrow{i} Q(q')$ Kanten in τ sind. Dies widerspricht Eigenschaft (2).

1. Fall: Wenn $|S_e^Q| = 1$, dann gilt für je zwei Kanten $z \xrightarrow{e} z'$ und $q \xrightarrow{e} q'$, dass $(Q(z), Q(z')) = (Q(p), Q(p'))$. Daher gibt es $i \in \tau$, weil Eigenschaft (2) erfüllt ist.

2. Fall: Wenn $|S_e^Q| = 2$, dann gibt es zwei Kanten $z \xrightarrow{e} z'$ und $q \xrightarrow{e} q'$in A, sodass $(Q(z), Q(z')) \neq (Q(p), Q(p'))$. Weil Eigenschaft (2) erfüllt ist, gibt es eine Interaktion $i \in \tau$, sodass $Q(z) \xrightarrow{i} Q(z') \in \tau$ und $Q(q) \xrightarrow{i} Q(q') \in \tau$. Weil $|S_e^Q| = 2$, gilt somit für alle $(x, y) \in S_e^Q$, dass $x \xrightarrow{i} y \in \tau$. Insbesondere gilt $Q(p) \xrightarrow{i} Q(p') \in \tau$ für alle $p \xrightarrow{e} p' \in A$. Aus der Beliebigkeit von e folgt die Behauptung. □

An dieser Stelle sei auf folgenden algorithmischen Nutzen von Lemma 8.2 hingewiesen: Wenn (sup, sig) eine τ-Region von A ist und $e \in E$, dann gibt es eine Interaktion $i \in \tau$, nämlich $i = sig(e)$, sodass $sup(z) \xrightarrow{i} sup(z') \in \tau$ für alle $z \xrightarrow{e} z' \in \tau$. Lemma 8.2 besagt nun, dass wir die Interaktion i (für einen potentiellen Support Q) allein durch sukzessives paarweises Vergleichen von e-beschrifteten Kanten finden können, wenn es sie gibt.

Ausgehend von einer berechtigten Grundannahme Q bezüglich des Supports einer lösenden τ-Region $R = (sup, sig)$, verfolgen wir das Ziel, Q durch Hinzunahme weiterer Zustände zu einem passenden Support zu erweitern, wenn dies möglich ist. Dazu nutzen wir die rekursive Prozedur SUP, das heißt Algorithmus 4, die für ein gegebenes Trennungsproblem α einen τ-Support mit höchstens ϱ Elementen konstruiert, wenn es einen solchen gibt.

Die Eingabe (A, Q, ϱ, τ) für SUP besteht aus einem Transitionssystem $A = (Z, E, \delta, \iota)$, einer Menge $Q \subseteq Z$ von Zuständen, einer natürlichen Zahl ϱ und einem Netztyp τ. Dabei ist Q –gemäß der in Lemma 8.1 formulierten Grundannahmen– eine Teilmenge einer τ-Region $R = (sup, sig)$, die α löst, wenn es diese gibt. Wenn $\alpha = (z, z')$ zum Beispiel ein

Algorithmus 4 : SUP

Input : Transitionssystem $A = (Z, E, \delta, \iota)$, $Q \subseteq Z$, $\varrho \in \mathbb{N}$ und Netztyp τ.
Result : Menge $\mathfrak{S}$ mit τ-Supporten von A, sodass $Q \subseteq sup$ für alle $sup \in \mathfrak{S}$.

if es gibt $e \in E$ exakt einmal in A: $\big[(z \xrightarrow{e} z',\ Q(z) = 0,\ Q(z') = 1,\ \mathfrak{produce} \cap \tau = \emptyset)$ *oder* $(z' \xrightarrow{e} z,\ Q(z') = 1,\ Q(z) = 0,\ \mathfrak{consume} \cap \tau = \emptyset)\big]$ **then**
  if $|Q| \geq \varrho$ **then** // Q `erweitern ist unvereinbar mit` ϱ
    return $\emptyset$;
  else // $i \in \{\mathsf{nop}, \mathsf{used}\}$
    $Q_1 = Q \cup \{z\}$;
    return $\textsc{Sup}(A, Q_1, \varrho, \tau)$;

if *es gibt Kanten* $z \xrightarrow{e} z', q \xrightarrow{e} q' \in A$, *die in Konflikt mit* Q *sind* **then**
  if $|Q| \geq \varrho$ **then** // Q `erweitern ist unvereinbar mit` ϱ
    return $\emptyset$;
  if $Q(z) = Q(z') = Q(q) = 0,\ Q(q') = 1$ **then** // $i \in \{\mathsf{nop}, \mathsf{out}, \mathsf{set}, \mathsf{swap}, \mathsf{used}\}$
    $Q_1 = Q \cup \{z'\}$; $Q_2 = Q \cup \{q\}$; $Q_3 = Q \cup \{z\}$;
    return $\textsc{Sup}(A, Q_1, \varrho, \tau) \cup \textsc{Sup}(A, Q_2, \varrho, \tau) \cup \textsc{Sup}(A, Q_3, \varrho, \tau)$;
  if $Q(z) = 1, Q(z') = 0, Q(q) = 1, Q(q') = 1$ **then** // $i \in \{\mathsf{nop}, \mathsf{set}, \mathsf{used}\}$
    $Q_1 = Q \cup \{z'\}$;
    return $\textsc{Sup}(A, Q_1, \varrho, \tau)$;
  if $Q(z) = 1, Q(z') = 0, Q(q) = 0, Q(q') = 1$ **then** // $i \in \{\mathsf{nop}, \mathsf{set}, \mathsf{used}\}$
    $Q_1 = Q \cup \{z'\}$; $Q_2 = Q \cup \{q\}$;
    return $\textsc{Sup}(A, Q_1, \varrho, \tau) \cup \textsc{Sup}(A, Q_2, \varrho, \tau)$;
  if $Q(z) = 0, Q(z') = 1, Q(q) = 1, Q(q') = 1$ **then** // $i \in \{\mathsf{nop}, \mathsf{used}\}$
    $Q_1 = Q \cup \{z\}$;
    return $\textsc{Sup}(A, Q_1, \varrho, \tau)$;
  if $Q(z) = 0, Q(z') = 1, Q(q) = 0, Q(q') = 1$ **then** // $i \in \{\mathsf{nop}, \mathsf{used}\}$
    $Q_1 = Q \cup \{z\}$; $Q_2 = Q \cup \{q\}$;
    return $\textsc{Sup}(A, Q_1, \varrho, \tau) \cup \textsc{Sup}(A, Q_2, \varrho, \tau)$;
  if $Q(z) = 1, Q(z') = Q(q) = Q(q') = 0$ **then** // $i \in \{\mathsf{nop}, \mathsf{inp}, \mathsf{set}, \mathsf{swap}, \mathsf{used}\}$
    $Q_1 = Q \cup \{z'\}$; and $Q_2 = Q \cup \{q\}$; and $Q_3 = Q \cup \{q'\}$;
    return $\textsc{Sup}(A, Q_1, \varrho, \tau) \cup \textsc{Sup}(A, Q_2, \varrho, \tau) \cup \textsc{Sup}(A, Q_3, \varrho, \tau)$;
  if $Q(z) = 1, Q(z') = 0, Q(q) = 1, Q(q') = 0$ **then** // $i \in \{\mathsf{nop}, \mathsf{set}\ \mathsf{used}\}$
    $Q_1 = Q \cup \{z'\}$; and $Q_2 = Q \cup \{q'\}$;
    return $\textsc{Sup}(A, Q_1, \varrho, \tau) \cup \textsc{Sup}(A, Q_2, \varrho, \tau)$;
else
  return $\{Q\}$;

ZTP ist, welches durch R gelöst wird, dann gilt entweder $z \in sup$ oder $z' \in sup$. Um eine solche lösende Region für α zu finden, wird SUP jeweils einmal mit Eingabe $Q = \{z\}$ und $Q = \{z'\}$ gestartet. Der Algorithmus SUP prüft nun in jedem rekursiven Aufruf, ob die aktuelle Menge Q eine passende τ-Signatur erlaubt. Dazu wird für jedes Ereignis $e \in E$ von A untersucht, ob es eine (eindeutige) Kante $z \xrightarrow{e} z'$ (Zeile 1) beziehungsweise Kanten $z \xrightarrow{e} z'$ und $q \xrightarrow{e} q'$ (Zeile 7) in A gibt, die in Konflikt mit Q sind, das heißt, es kann kein konsistenter Signatur-Wert $sig(e)$ definiert werden. Wenn alle Kanten konfliktfrei sind, dann terminiert SUP mit der Ausgabe $\{Q\}$, wobei Q gemäß Lemma 8.2 der Support einer τ-Region ist. Andernfalls erlaubt Q keine τ-Signatur und der Konflikt muss durch hinzufügen weiterer Zustände gelöst werden. Zu jedem rekursiven Aufruf enthält Q jeweils nur die minimale Anzahl an notwendigen Zuständen – initial zum Beispiel gemäß der Grundannahme. Daraus folgt, dass das *Entfernen* von Zuständen von Q um Konflikte zu lösen, keine Option ist.

Wenn also eine eindeutige e-beschriftete Kante $z \xrightarrow{e} z'$ in Konflikt mit Q ist (Zeile 1), dann gilt einerseits entweder $Q(z) = 1$ und $Q(z') = 0$ oder $Q(z) = 0$ und $Q(z') = 1$, denn $Q(z) = Q(z')$ impliziert $Q(z) \xrightarrow{\mathsf{nop}} Q(z') \in \tau$. Daher gibt es genau einen Zustand $x \in \{z, z'\}$, um den Q erweitert werden kann, sodass der Konflikt gelöst wird. Für diese Erweiterung von Q startet der Algorithmus den rekursiven Aufruf $\textsc{Sup}(A, Q \cup \{x\}, \varrho, \tau)$ (Zeile 6) dessen Ergebnis der Ausgabe hinzugefügt wird.

Wenn zwei Kanten $z \xrightarrow{e} z'$ und $q \xrightarrow{e} q'$ in Konflikt sind, dann ist der einzige Weg Q potentiell zu einem passenden Support zu modifizieren, wenigstens einen der Zustände z, z', q, q' zu Q hinzuzufügen. Weil $z \xrightarrow{e} z'$ und $q \xrightarrow{e} q'$ in Konflikt sind, gilt darüber hinaus, dass wenigstens einer der Zustände z, z', q, q' zur Menge Q gehört. Andernfalls wären $Q(z) \xrightarrow{\mathsf{nop}} Q(z') \in \tau$ und $Q(q) \xrightarrow{\mathsf{nop}} Q(q') \in \tau$, denn nop ist Interaktion jedes Typs. Deshalb gibt es höchstens drei Kandidaten x, y, z aus $\{z, z', q, q'\}$, um die Q erweitert werden kann. Für jeden dieser Zustände, der tatsächlich nicht in Q ist, verzweigt der Algorithmus und startet SUP mit der entsprechenden Erweiterung von Q: SUP startet jeweils einen rekursiven Aufruf $\textsc{Sup}(A, Q \cup \{x\}, \varrho, \tau)$ und $\textsc{Sup}(A, Q \cup \{y\}, \varrho, \tau)$ sowie $\textsc{Sup}(A, Q \cup \{z\}, \varrho, \tau)$ und gibt die Vereinigung $\textsc{Sup}(A, Q \cup \{x\}, \varrho, \tau) \cup \textsc{Sup}(A, Q \cup \{y\}, \varrho, \tau) \cup \textsc{Sup}(A, Q \cup \{z\}, \varrho, \tau)$ der einzelnen Ausgaben zurück. Insbesondere hat die rekursive Prozedur SUP einen ternären Berechnungsbaum. Darüber hinaus geht jeder Erweiterung von Q eine Überprüfung voraus, ob diese Erweiterung mit der Schranke ϱ vereinbar ist (Zeile 2 beziehungsweise Zeile 8). Dies sichert unter anderem zu, dass SUP terminiert. Des Weiteren fügt SUP nur *notwendige* Zustände hinzu. Deshalb wird –im Falle ihrer Existenz– der Support einer minimalen τ-Region (sup, sig) mit ϱ-beschränkter Belegungszahl, die α löst, entlang wenigstens eines Pfades im Berechnungsbaum gefunden: $sup \in \textsc{Sup}$. Die Berechnungen von SUP liefern einen ternären Rekursionsbaum dessen Tiefe durch ϱ beschränkt ist. Deshalb findet SUP mit Zeitaufwand $\mathcal{O}^*(3^\varrho)$[1] eine passende Region $R = (sup, sig)$, die α löst, wenn

[1]Wir nutzen die $\mathcal{O}^*$-Notation um polynomielle Faktoren zu unterdrücken, das heißt, anstatt $\mathcal{O}(f(\varrho)|A|^c)$, wobei c eine Konstante ist, schreiben wir $\mathcal{O}^*(f(\varrho))$.

es eine solche gibt. Das folgende Lemma adressiert die korrekte Arbeitsweise von Sup.

Lemma 8.3 (Termination, Korrektheit, Vollständigkeit)**.** *Wenn $Q \subseteq Z$ eine Menge von Zuständen von A, ϱ eine natürliche Zahl und (A, Q, ϱ, τ) die Eingabe von* Sup *ist, dann*

1. *terminiert* Sup *und die Laufzeit ist $\mathcal{O}^*(3^\varrho)$ und für die Ausgabe $\mathfrak{S}$ von* Sup*;*
2. *ist jedes Element $sup \in \mathfrak{S}$ der Support einer τ-Region $R = (sup, sig)$ von A, die $Q \subseteq sup$ und $|sup| \leq \varrho$ erfüllt und*
3. *es gilt $sup \in \mathfrak{S}$ für jede Q-minimale ϱ-beschränkte τ-Region (sup, sig) von A.*

Beweis. (1): Es ist leicht zu sehen, dass Sup einen ternären Rekursionsbaum hat, denn zu jedem beliebigen aber festen Aufruf startet Sup höchstens drei neue Aufrufe und generiert damit höchstens drei Verzweigungen. Dabei ist jeder Knoten des Berechnungsbaumes mit der aktuellen Eingabe (A, Q, ϱ, τ) und der zugehörigen Ausgabe von Sup assoziiert. Wenn P ein beliebiger aber fester Pfad von der Wurzel zu einem Blatt in dem Berechnungsbaum ist, dann gilt darüber hinaus, dass P höchstens Länge ϱ hat: Wenn weder die If-Bedingung aus Zeile 1 noch jene aus Zeile 7 erfüllt ist, dann gibt Sup die Menge $\{Q\}$ zurück (Zeile 32) und terminiert. Insbesondere ist Q dann gemäß Lemma 8.2 ein τ-Support. Andernfalls ist eine der If-Bedingungen erfüllt und die Prozedur testet, ob $|Q| \geq \varrho$. Wenn dieser Test positiv ausfällt, dann gibt Sup die leere Menge $\emptyset$ zurück und terminiert, denn eine Erweiterung von Q wäre nicht ϱ-beschränkt. Wenn dieser Test negativ ausfällt, dann wird Q, je nachdem welche Konflikte vorliegen, zu Q_1 beziehungsweise Q_2 beziehungsweise Q_3 erweitert. Das heißt, für alle $i \in \{1, 2, 3\}$ gilt, wenn Q_i existiert, dann ist $|Q_i| > |Q|$ erfüllt und Sup wird rekursiv auf Eingabe (A, Q_i, ϱ, τ) gestartet. Da die Schranke ϱ nicht überschritten wird, kann die (initiale Menge) Q höchstens ϱ-mal entlang des Berechnungspfades P erweitert werden. Folglich gibt Sup nach höchstens $\varrho+1$ rekursiven Aufrufen (entlang P) entweder $\emptyset$ oder $\{Q'\}$ aus, wobei Q' von der Eingabe (A, Q', ϱ, τ) des aktuellen rekursiven Aufrufs herrührt. Insbesondere ist die Länge von P durch ϱ beschränkt.

Untersuchen wir nun die Laufzeit von Sup: Der rechnerisch aufwendigste Teil eines beliebigen aber festen Aufrufs ist der Test, ob es Kanten gibt, die in Konflikt sind. Im schlimmsten Fall muss für alle $e \in E$ und alle $z \xrightarrow{e} z', q \xrightarrow{e} q' \in A$ getestet werden, ob $z \xrightarrow{e} z'$ und $q \xrightarrow{e} q'$ in Konflikt sind. Weil A höchstens $|E| \cdot |Z|^2$ Kanten hat und $|Q| \leq |Z|$, ist dieser Aufwand jedoch polynomiell beschränkt durch die Größe von A. Darüber hinaus haben wir höchstens $3^{\varrho+1}$ verschiedene Aufrufe von Sup. Deshalb ist die Laufzeit von A in $\mathcal{O}^*(3^\varrho)$.

2. Wenn Q' ein Element der Ausgabe $\mathfrak{S}$ von $\textsc{Sup}(A, Q, \varrho, \tau)$ ist, dann gilt für alle $e \in E$, dass alle e-beschrifteten Kanten von A konfliktfrei sind. Deshalb ist Q' nach Lemma 8.2 der Support einer τ-Region von A. Darüber hinaus ist Q' per Konstruktion eine Erweiterung von Q, das heißt, $Q \subseteq Q'$. Wegen der Abbruchbedingungen der Zeilen 2 beziehungsweise 8 gilt außerdem $|Q'| \leq \varrho$.

3. Sei (sup, sig) eine beliebige aber feste Q-minimale τ-Region von A, sodass $|sup| \leq \varrho$. Weil sup Q-minimal ist, gilt $Q \subseteq sup$. Wir beweisen nun per Induktion über $n = |sup \setminus Q|$, dass sup ein Element der Ausgabe $\mathfrak{S}$ von Sup bei Eingabe (A, Q, ϱ, τ) ist. Wenn $n = 0$, das heißt $Q = sup$, dann gibt es keine Kanten (-paare) von A, die mit Q in Konflikt sind und $\{sup\}$ wird in Zeile 32 des Algorithmus ausgegeben. Dies beweist den Induktionsanfang. Nehmen wir nun an, die Behauptung gelte für $|sup \setminus Q| \leq n$, wobei $n \in \mathbb{N}$. Für den Induktionsschritt sei $|sup \setminus Q| = n + 1$. Der Support sup ist Q-minimal und offenbar gilt $|Q| < |sup|$. Deshalb gibt es ein Ereignis $e \in E$, sodass es eine (eindeutige) Kante $r \xrightarrow{e} r'$ beziehungsweise ein Paar von Kanten $r \xrightarrow{e} r'$ und $t \xrightarrow{e} t'$ in A gibt, die einerseits mit Q in Konflikt sind und andererseits in Zeile 1 beziehungsweise Zeile 7 von Sup gefunden werden. Weil sup ein Support ist und damit nicht Ursache eines Konflikts sein kann, gibt es einen Zustand $z \in \{r, r', t, t'\}$, sodass $sup(z) \neq Q(z)$. Darüber hinaus folgt aus $|Q| < |sup|$ und $|sup| \leq \varrho$ auch $|Q| < \varrho$. Deshalb gibt es im Berechnungsbaum des Laufs von Sup, gestartet auf der initialen Eingabe (A, Q, ϱ, τ), einen (Verzweigungs-) Knoten, an dem die die Menge $Q' = Q \cup \{z\}$ konstruiert wird, das heißt in einer der Zeilen 5, 11, 14, ..., 29. (Man beachte, dass wir ausdrücklich von den Kanten in Konflikt sprechen, die in Zeile 1 beziehungsweise Zeile 7 gefunden wurden.) Anschließend wird $\text{Sup}(A, Q', \varrho, \tau)$ gestartet. Weil $Q \subseteq sup$, impliziert dies darüber hinaus $z \in sup \setminus Q$ und somit $Q' \subseteq sup$. Wir zeigen nun, dass (sup, sig) auch Q'-minimal ist: Nehmen wir an, (sup, sig) sei nicht Q'-minimal. Dann gibt es (gemäß Bemerkung 8.1) eine Q'-minimale τ-Region (sup', sig') mit $|sup'| < |sup|$. Nun gilt aber $Q \subseteq Q' \subseteq sup'$. Wegen $|sup'| < |sup|$ widerspricht dies der Voraussetzung, dass (sup, sig) Q-minimal ist. Folglich ist (sup, sig) auch Q'-minimal. Wegen $|sup \setminus Q| = n + 1$, erhalten wir außerdem $|sup \setminus Q'| = n$. Weil sup Q'-minimal ist, gehört sup nach Induktionsvorausetzung zur Ausgabe von $\text{Sup}(A, Q', \varrho, \tau)$. Dies impliziert $sup \in \text{Sup}(A, Q, \varrho, \tau)$, denn $\text{Sup}(A, Q', \varrho, \tau) \subseteq \text{Sup}(A, Q, \varrho, \tau)$. □

Eine Berechnung von Sup liefert eine Menge von Supporten *aller* τ-Regionen von A, welche ϱ berücksichtigen und Q umfassen. Darunter können im Allgemeinen auch solche Supporte sein, die keiner lösenden Region entsprechen. Deshalb muss für jedes Element $Q \in \text{Sup}(A, Q, \varrho, \tau)$ noch überprüft werden, ob für die gemäß Lemma 8.1 zu Q gehörende negative Grundannahme F die Eigenschaft $Q \cap F = \emptyset$ erfüllt ist. Wenn $sup = Q$ ein gültiger Support ist, dann kann es potentiell exponentiell viele Signaturen sig geben, sodass $R = (sup, sig)$ eine τ-Region ist. Wir müssen also argumentieren, dass (eine lösende Region) R in Polynomialzeit bestimmt werden kann. Gemäß Lemma 4.2 ist die Menge $S_e^{sup} = \{(sup(z), sup(z')) \mid z \xrightarrow{e} z' \in A\}$ für jedes Ereignis $e \in E$ entweder ein- oder zweielementig. Im ersten Fall entspricht S_e^{sup} entweder $\{(0,0)\}$ oder $\{(0,1)\}$ oder $\{(1,1)\}$ oder $\{(1,0)\}$ und eine entsprechende Signatur $sig(e)$ kann dazu passend gewählt werden aus $\{\mathsf{nop}, \mathsf{res}, \mathsf{free}\}$ oder $\{\mathsf{out}, \mathsf{swap}, \mathsf{set}\}$ oder $\{\mathsf{nop}, \mathsf{set}, \mathsf{used}\}$ oder $\{\mathsf{inp}, \mathsf{swap}, \mathsf{res}\}$, je nachdem welche Interaktionen τ bereitstellt. Im zweiten Fall entspricht S_e^{sup} entweder $\{(0,0),(1,1)\}$ oder $\{(0,0),(1,0)\}$ oder $\{(0,1),(1,0)\}$ oder $\{(0,1),(1,1)\}$ und $sig(e)$ ist (der

Algorithmus 5 : Realisierung durch ein τ-Netz mit beschränkter Belegungszahl

Input : Netztyp τ, Transitionssystem $A = (Z, E, \delta, \iota)$, Menge $Q \subseteq Z$, Zahl $\varrho \in \mathbb{N}$.
Result : Menge $\mathcal{R}$ ϱ-beschränkter τ-Regionen von A, die ein Zeuge für die τ-ZTE und die τ-EZTE von A ist, wenn es überhaupt einen Zeugen gibt.

$\mathcal{R} = \emptyset$;
for *jedes ZTP* (z, z') *von* A **do**
  $R = \{Q \in \textsc{Sup}(A, \{z\}, \varrho, \tau) \mid Q \cap \{z'\} = \emptyset\} \cup$ // $sup(z) = 1, sup(z') = 0$
  $\{Q \in \textsc{Sup}(A, \{z'\}, \varrho, \tau) \mid Q \cap \{z\} = \emptyset\}$; // $sup(z) = 0, sup(z') = 1$
  if $R = \emptyset$ **then**
    return $\emptyset$;
  $\mathcal{R} = \mathcal{R} \cup R$;
for *jedes EZTP* (e, q) *von* A **do**
  $R = \{Q \in \textsc{Sup}(A, \{z \in Z \mid z \xrightarrow{e}\}, \varrho, \tau) \mid Q \cap \big(\{q\} \cup \{z \in Z \mid \xrightarrow{e} z\}\big) = \emptyset\} \cup$
    // $sup(q) = 0, sig(e) = \mathsf{inp}$
  $\{Q \in \textsc{Sup}(A, \{z, z' \in Z \mid z \xrightarrow{e} z'\}, \varrho, \tau) \mid Q \cap \{q\} = \emptyset\} \cup$
    // $sup(q) = 0, sig(e) = \mathsf{used}$
  $\{Q \in \textsc{Sup}(A, \{q\} \cup \{z \in Z \mid \xrightarrow{e} z\}, \varrho, \tau) \mid Q \cap \{z \in Z \mid z \xrightarrow{e}\} = \emptyset\} \cup$
    // $sup(q) = 1, sig(e) = \mathsf{out}$
  $\{Q \in \textsc{Sup}(A, \{q\}, \varrho, \tau) \mid Q \cap \{z, z' \in Z \mid z \xrightarrow{e} z'\} = \emptyset\}$;
    // $sup(q) = 1, sig(e) = \mathsf{free}$
  if $R = \emptyset$ **then**
    return $\emptyset$;
  $\mathcal{R} = \mathcal{R} \cup R$;
return $\mathcal{R}$

Reihenfolge entsprechend) in *eindeutiger* Weise durch S_e^{sup} bestimmt als nop, res, swap beziehungsweise set. Wenn e nicht dem augenblicklich zu lösenden Ereignis entspricht, dann ist unter allen passenden Signaturwerten die tatsächliche Wahl der Signatur von e nicht relevant. Wenn die Signaturwerte gemäß einer beliebigen aber festen Priorisierung der Interaktionen vergeben werden, kann daher eine exponentielle Explosion (in $|\tau|$) des Aufwands, eine passende Signatur zu bestimmen, verhindert werden. Wenn zum Beispiel $\tau = \{\mathsf{inp}, \mathsf{out}, \mathsf{free}, \mathsf{used\ nop}\}$ und e und *sup* derart, dass $S_e^{sup} = \{(0, 0)\}$, dann muss $sig(e)$ ein Element aus $\{\mathsf{nop}, \mathsf{res}, \mathsf{free}\}$ sein. Weil sowohl nop und free in τ sind, gibt es zunächst eine Wahl. Wenn aber *a priori* festgelegt ist, dass nop bezüglich free die höhere Priorität hat, dann ist die Lösung eindeutig. Die Priorisierung der Interaktionen verhindert das Finden einer Lösung nicht und erlaubt es uns, zu einem gültigen Support *sup* mit Zeitaufwand $\mathcal{O}(|Z|^2 \cdot |E|)$ eine passende Signatur *sig* zu bestimmen.

Algorithmus 5 beschreibt nun den vollständigen Prozess, der (zunächst nur) das Pro-

blem τ-RMBBZ entscheidet und im Falle einer positiven Entscheidung ein gesuchtes Netz konstruiert: Zunächst versucht Algorithmus 5 für jedes ZTP (z, z') eine τ-Region zu finden, die es löst und ϱ berücksichtigt. Dazu werden in Zeile 3 alle $\{z\}$-minimalen Regionen konstruiert und nur jene ausgewählt, die z und z' tatsächlich trennen. Analog werden in Zeile 4 die $\{z'\}$-minimalen Regionen bestimmt. Wenn es keine Region mit durch ϱ beschränkter Belegungszahl gibt, die (z, z') löst, dann erlaubt die aktuelle Eingabe für τ-RMBBZ keine positive Entscheidung und in Zeile 6 wird die leere Menge zurückgegeben. Andernfalls werden die gefundenen Regionen in Zeile 7 zur Menge $\mathcal{R}$ hinzugefügt, die potentiell zu einem gesuchten Zeugen für die τ-EZTE und die τ-ZTE von A ergänzt wird.

Wenn alle ZTP gelöst sind, wird ab Zeile 8 für jedes EZTP (e, q) von A eine τ-Region (sup, sig) gesucht, die es löst. Dabei werden alle vier Möglichkeiten, (e, q) zu lösen, untersucht: In Zeile 9 wird der Fall $sup(q) = 0$ und $sig(e) = \mathsf{inp}$ adressiert, in Zeile 10 der Fall $sup(q) = 0$ und $sig(e) = \mathsf{used}$, in den Zeile 11 der Fall $sup(q) = 1$ und $sig(e) = \mathsf{out}$ und in Zeile 11 der Fall $sup(q) = 1$ und $sig(e) = \mathsf{free}$. Wenn für keine dieser Möglichkeiten eine passende Region existiert, dann kann (e, q) nicht gelöst werden und es wird in Zeile 14 die leere Menge zurückgegeben und der Algorithmus terminiert. Andernfalls werden die gefundenen Regionen in Zeile 15 zur Menge $\mathcal{R}$ hinzugefügt. Wenn alle Trennungsprobleme erfolgreich gelöst werden konnten, wird ein entsprechender Zeuge in Zeile 16 zurückgegeben.

Weil höchstens $|Z|^2$ ZTP beziehungsweise $|E| \cdot |Z|$ EZTP von A auf ihre Lösbarkeit überprüft werden müssen und die Anzahl möglicher Grundannahmen (durch Q) für jedes Trennungsproblem konstant ist, nämlich 2 für ZTP und 4 für EZTP, erhalten wir zur bereits diskutierten Laufzeit $\mathcal{O}^*(3^\varrho)$ von Sup zusätzlich einen Faktor der polynomiell von der Größe von A abhängt. Deshalb ist die Gesamtlaufzeit $\mathcal{O}^*(3^\varrho)$. Darüber hinaus ist leicht zu sehen, dass Algorithmus 5 jeweils zu einer Prozedur von τ-EMBBZ beziehungsweise τ-SMBBZ modifiziert werden kann: Adressieren wir die τ-EMBBZ, dann sind nur die Zeilen 1 bis 7 des Algorithmus interessant und für die τ-SMBBZ interessieren nur die Zeilen 8 bis 15, jeweils natürlich um die Ausgabe in Zeile 16 erweitert. Insgesamt haben wir schließlich Satz 8.1 bewiesen.

8.2 Anwendung des Algorithmus an einem Beispiel

Im Rest dieses Kapitels werden wir nun ein ausführliches Beispiel für die Anwendung von Algorithmus 5 diskutieren.

Wir untersuchen das Transitionssystem A mit Zustandsmenge $Z = \{\iota, q, z\}$ und Ereignismenge $E = \{a, b, c\}$ aus Abbildung 8.1. Um eine τ-Realisierung für A zu finden, müssen die drei ZTP (ι, z), (ι, q), (q, z) und die fünf EZTP (c, ι), (a, q), (a, z), (b, z), (c, z) gelöst werden. Wir werden nun drei Instanzen (A, ϱ) von τ-RMBBZ mit A untersuchen, nämlich für $\varrho = 1$ und $\tau = \{\mathsf{nop}, \mathsf{set}, \mathsf{inp}\}$, für $\varrho = 2$ und $\tau = \{\mathsf{nop}, \mathsf{set}, \mathsf{inp}\}$ und für $\varrho = 2$ und $\tau = \{\mathsf{nop}, \mathsf{set}, \mathsf{inp}, \mathsf{out}\}$, und Algorithmus 5 (beziehungsweise Algorithmus 4) für jeden

dieser Fälle anwenden.

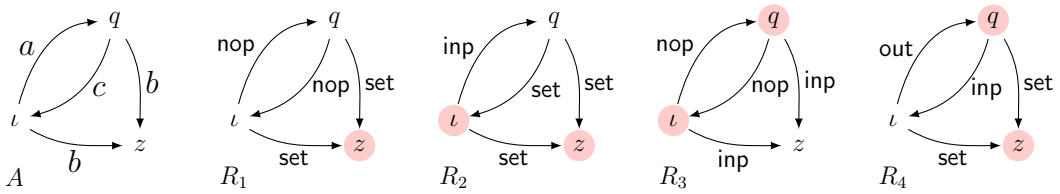


Abbildung 8.1: Das Transitionssystem A (links), das durch ein $\{\mathsf{nop}, \mathsf{set}, \mathsf{inp}, \mathsf{out}\}$-Netz N mit $\varrho_N \leq 2$ lösbar ist und einige seiner τ-Regionen mit 2-beschränkter Belegungszahl.

Fall ($\varrho = 1, \tau = \{\mathsf{nop}, \mathsf{set}, \mathsf{inp}\}$): Initial ist die Menge $\mathcal{R}$ gesuchter τ-Regionen leer. Wir starten mit dem ZTP (ι, z). Gemäß den Zeilen 3 und 4 von Algorithmus 5 starten wir SUP mit $Q = \{\iota\}$ und $Q = \{z\}$. Wir untersuchen zunächst den ersten Fall, das heißt $\text{SUP}(A, \{\iota\}, 1, \tau)$. Der Test in Zeile 7 von SUP fällt positiv aus, die Kanten $\iota \xrightarrow{b} z$ und $q \xrightarrow{b} z$ sind in Konflikt mit Q. Daraufhin wird in Zeile 8 festgestellt, dass $|S| \geq \varrho = 1$, sodass Q nicht erweitert werden kann und der Lauf $\text{SUP}(A, \{\iota\}, 1, \tau)$ die leere Menge zurückgibt. Für $Q = \{z\}$ starten wir $\text{SUP}(A, \{z\}, 1, \tau)$. Die Überprüfungen in Zeile 1 und Zeile 7 fallen negativ aus, das heißt, keine Kante ist in Konflikt mit Q. Eine gültige Signatur sig_1 ist definiert durch $sig_1(a) = sig_1(c) = \mathsf{nop}$ und $sig_1(b) = \mathsf{set}$. Folglich gibt $\text{SUP}(A, Q, \varrho, \tau)$ die Menge $Q = \{z\}$ als gültigen Support der 1-beschränkten τ-Region $R_1 = (\{z\}, sig_1)$ zurück, siehe Abbildung 8.1. Weil $\{z\} \cap \{\iota\} = \emptyset$, wird R_1 in Zeile 4 von Algorithmus 5 zu der Menge (von Regionen) R, die das ZTP (ι, z) lösen, hinzugefügt. Schließlich wird R_1 in Zeile 7 von Algorithmus 5 durch den Vereinigungsschritt auch zu $\mathcal{R}$ hinzugefügt. Insbesondere ist (ι, z) gelöst.

Das ZTP (q, z) kann ähnlich wie (ι, z) gelöst werden, wobei q und ι ihre Rollen tauschen.

Betrachten wir nun das ZTP (ι, q). Keine der a- beziehungsweise c-beschrifteten Kanten ist in Konflikt mit $\{\iota\}$. Jedoch sind die b-beschrifteten Kanten in Konflikt mit $\{\iota\}$, deshalb terminiert der Lauf von $\text{SUP}(A, \{\iota\}, 1, \tau)$ in Zeile 9 mit der Rückgabe der leeren Menge. Für den Lauf von $\text{SUP}(A, \{q\}, 1, \tau)$ ist die Bedingung in Zeile 7 erfüllt und der Konflikt von $\iota \xrightarrow{b} z$ und $q \xrightarrow{b} z$ mit $\{q\}$ wird festgestellt. Auch dieser Lauf terminiert in Zeile 9, denn $\{q\}$ kann nicht zu dem Support einer 1-beschränkten Region erweitert werden. Daher gibt es keine 1-beschränkte τ-Region, die (ι, q) löst und somit ist A nicht entsprechend implementierbar. Algorithmus 5 terminiert in Zeile 6 mit Rückgabe der leeren Menge.

Fall ($\varrho = 2, \tau = \{\mathsf{nop}, \mathsf{set}, \mathsf{inp}\}$): Weil ϱ hier größer ist als im vorherigen Fall, können wir die bereits diskutierten Regionen zur Lösung von (ι, z) und (q, z) verwenden. Betrachten wir nun (ι, q). In den Zeilen 3 und 4 von Algorithmus 5 wird SUP jeweils einmal für die Grundannahmen $\{\iota\}$ beziehungsweise $\{q\}$ aufgerufen. Während des Laufs von $\text{SUP}(A, \{\iota\}, 2, \tau)$ wird der Konflikt von $\iota \xrightarrow{b} z$ und $q \xrightarrow{b} z$ mit $Q = \{\iota\}$ entdeckt. Weil

$|S| < \varrho = 2$, darf Q erweitert werden und, gemäß des beobachteten Konflikts (Zeile 25), wird Sup in Zeile 26 jeweils auf $Q \cup \{z\}$ und $Q \cup \{q\}$ gestartet. (Es gibt keine dritte Erweiterung, weil z das Ziel beider b-beschrifteter Kanten ist.) Die Menge $\{\iota, q\}$ ist konfliktfrei und gehört deshalb als gültiger Support zur Ausgabe von $\textsc{Sup}(A, \{\iota, q\}, 2, \tau)$. Ebenso findet der Lauf von $\textsc{Sup}(A, \{\iota, z\}, 2, \tau)$ keinen Konflikt und gibt $\{\iota, z\}$ als gültigen Support einer 2-beschränkten Region zurück. In den Zeilen 3 und 4 von Algorithmus 5 wird aber nur der Support gewählt, der zu $\{q\}$ disjunkt ist, das heißt nur $\{\iota, z\}$. Ein passende Signatur sig_2 ist definiert durch $sig_2(a) = \mathsf{inp}$, $sig_2(b) = sig_2(c) = \mathsf{set}$ und wir erhalten die lösende Region $R_2 = (\{\iota, s\}, sig_2)$, siehe Abbildung 8.1. Darüber hinaus sei bemerkt, dass $\textsc{Sup}(A, \{q\}, 2, \tau)$ eine weitere gültig Region $R_2' = (\{q, z\}, sig_2')$ mit $sig_2'(a) = sig_2'(b) = \mathsf{set}$ und $sig_2'(c) = \mathsf{inp}$ liefert, die (ι, q) löst. Somit hat A einen Zeugen 2-beschränkter Regionen für die τ-ZTE und erlaubt damit insbesondere eine entsprechende τ-Einbettung.

Wir untersuchen nun die EZTP von A und starten dazu mit (a, q). Zum Lösen von EZTP ist unter allen Interaktionen von τ ist nur inp von Interesse. Entsprechend starten wir mit der Grundannahme $Q = \{t \in Z \mid t \xrightarrow{a} t' \in A\} = \{\iota\}$ und rufen $\textsc{Sup}(A, Q, 2, \tau)$ in Zeile 9 von Algorithmus 5 auf. Dieser Lauf findet in Zeile 7 von Sup den Konflikt von $\iota \xrightarrow{b} z$ und $q \xrightarrow{b} z$ mit $\{\iota\}$. Weil $|Q| < 2 = \varrho$ darf Q erweitert werden und gemäß des beobachteten Konflikts (Zeile 25) wird Sup in Zeile 27 auf $Q \cup \{q\}$ und $Q \cup \{z\}$ aufgerufen. Im ersten Fall wird $\{\iota, q\}$ als (konfliktfreier) Support zurückgegeben. Auch im zweiten Fall findet der rekursive Aufruf $\textsc{Sup}(A, \{\iota, z\}, 2, \tau)$ keinen Konflikt und gibt $\{\iota, z\}$ als gültigen Support zurück. Gemäß Zeile 9 von Algorithmus 5 werden jedoch nur solche Supports akzeptiert, die zu $\{q\}$ disjunkt sind, das heißt, nur $\{\iota, z\}$. Eine passende Signatur sig' ist definiert durch $sig'(b) = sig'(c) = \mathsf{set}$, $sig'(a) = \mathsf{inp}$ und wir erhalten die 2-beschränkte τ-Region $R' = (\{\iota, z\}, sig')$, die der oben definierten Region R_2 entspricht.

Das EZTP (c, ι) kann ähnlich wie (a, q) gelöst werden, wobei a und c beziehungsweise q und ι ihrer Rollen tauschen.

Für das EZTP (b, z) beginnen wir mit der Grundannahme $Q = \{t \in Z \mid t \xrightarrow{b}\} = \{\iota, q\}$. Der Lauf $\textsc{Sup}(A, Q, 2, \tau)$ stellt fest, das keine Kanten mit Q in Konflikt sind und $\{\iota, q\}$ wird als gültiger Support zurückgegeben. Wegen $\{\iota, q\} \cap \{z\} = \emptyset$, erlaubt dieser Support tatsächlich eine Region, die (b, z) löst. Eine passende Signatur sig_3 is definiert durch $sig_3(a) = sig_3(c) = \mathsf{nop}$, $sig_3(b) = \mathsf{inp}$ und wir erhalten die Region $R_3 = (\{\iota, q\}, sig_3)$, siehe Abbildung 8.1.

Für das EZTP (a, z) starten wir mit $Q = \{t \in Z \mid t \xrightarrow{a}\} = \{\iota\}$. Es findet der Lauf $\textsc{Sup}(A, Q, 2, \tau)$ heraus, dass $\iota \xrightarrow{b} z$ und $q \xrightarrow{b} z$ in Konflikt mit Q sind. Mit den Zeilen 25-27 von Algorithmus 4 folgen zwei rekursive Aufrufe $\textsc{Sup}(A, \{\iota, q\}, 2, \tau)$ und $\textsc{Sup}(A, \{\iota, z\}, 2, \tau)$. Beide Läufe finden keine Konflikte, aber in Zeile 9 von Algorithmus 5 werden beide resultierenden Supports ausgeschlossen. Folglich ist das EZTP (a, z) nicht durch eine 2-beschränkte τ-Region lösbar.

Analog verhält sich der Fall für das EZTP (c, z).

Fall ($\varrho = 2, \tau = \{\mathsf{nop}, \mathsf{set}, \mathsf{inp}, \mathsf{out}\}$): Weil der Parameter ϱ nicht kleiner ist als zuvor

und τ alle Interaktionen der bisher diskutierten Netztypen enthält, werden die bereits diskutierten lösenden Regionen ebenso gefunden. Folglich müssen wir nur noch die τ-Lösbarkeit der EZTP (a, z) und (c, z) untersuchen. Wir starten mit (a, z). Mit dem Ziel out als Signatur für a zu verwenden, gehen wir von der Grundannahme $Q = \{t \in Z \mid \xrightarrow{a} t\} = \{q\}$ aus (Zeile 11 von Algorithmus 5). Der Lauf $\textsc{Sup}(A, Q, 2, \tau)$ erkennt, dass $\iota \xrightarrow{b} z$ und $q \xrightarrow{b} z$ in Konflikt mit Q sind und gemäß des beobachteten Konflikts (Zeile 25) müssen die Erweiterungen $Q \cup \{\iota\}$ und $Q \cup \{z\}$ von Q untersucht werden. Für beide Erweiterungen findet Sup keinen Konflikt. Weil jedoch $\{\iota, q\}$ und $\{q\}$ nicht disjunkt sind, wird dieser Support nicht zu $\mathcal{R}$ hinzugefügt. Für $\{q, z\}$ finden wir die Signatur sig_4, die definiert ist durch $sig_4(a) = \mathsf{out}$, $sig_4(b) = \mathsf{set}$ sowie $sig_4(c) = \mathsf{inp}$ und erhalten die τ-Region $R_4 = (\{q, z\}, sig_4)$, die das EZTP (a, z) löst und $\varrho_{R_4} \leq 2$ erfüllt. Siehe erneut Abbildung 8.1. Das EZTP (c, z) wird analog zu (a, s) gelöst.

Insgesamt stellen wir damit Folgendes fest: Wenn $\tau = \{\mathsf{nop}, \mathsf{set}, \mathsf{inp}, \mathsf{out}\}$, dann gibt es einen Zeugen von τ-Regionen mit 2-beschränkter Belegungszahl für die τ-ZTE und die τ-EZTE für A.

8.3 Zusammenfassung und Ausblick

In diesem Kapitel wurde gezeigt, dass die Synthese boolescher Netze mit beschränkter Belegungszahl für alle Implementierungen fest-Parameter-berechenbar ist, wenn die Belegungszahl der Parameter ist und die Netze nop erlauben. Wegen der Zusammenhänge zwischen Regionen eines Transitionssystems $A = (Z, E, \delta, \iota)$ und Plätzen eines implementierenden Netzes N, konnten die entsprechenden Entscheidungsprobleme wieder bezüglich der Lösbarkeit von Trennungsproblemen von A durch Regionen mit beschränkter Belegungszahl formuliert werden. Insbesondere wurde ein parametrisierter Algorithmus vorgestellt, der für jedes Trennungsproblem eine lösende Region mit beschränkter Belegungszahl konstruiert, wenn es eine solche gibt. Dazu startet der Algorithmus mit einer (bezüglich gewisser Grundannahmen) minimalen Menge Q von Zuständen und erweitert diese rekursiv zu dem Support einer entsprechenden Region, wenn es eine solche gibt. In jedem rekursiven Aufruf wird dazu getestet, ob Q für alle Ereignisse eine passende Signatur erlaubt. Dabei genügt es, für jedes Ereignis e und alle e-beschrifteten Kanten *paarweise* zu überprüfen, ob sie mit Q in Konflikt sind. Wenn dies nicht der Fall ist, dann ist Q ein Support. Wenn es keine passende lösende Region gibt, dann wird das Trennungsproblem korrekt als nicht durch eine Region mit beschränkter Belegungszahl lösbar erkannt.

Priorisiert man die Interaktionen eines booleschen Netztyps τ, dann legt eine Menge $Q \subseteq Z$ von Zuständen in eindeutiger Weise eine τ-Region $R = (Q, sig)$ von A fest, wenn Q überhaupt eine solche erlaubt. Andererseits gilt für eine Abbildung $sig : E \to \tau$, dass entweder durch $sup(\iota) = 0$ und sig beziehungsweise durch $sup(\iota) = 1$ und sig eine τ-Region von A bestimmt ist oder dass es keine τ-Region von A gibt, die sig als Signatur hat. Darüber hinaus sind für sig wiederum nur jene Ereignisse interessant, deren Signatur

von nop verschieden sind, denn für die anderen Ereignisse können wir dann diese Signatur einfach ableiten. Diese Überlegungen zeigen, dass Regionen gewissermaßen (unter Beachtung der genannten Einschränkungen) bereits jeweils durch *sup* beziehungsweise durch *sig* (der Ereignisse, die keine nop-Signatur haben) beschrieben sind. Mit anderen Worten: *sup* und *sig* bedingen einander. Während die Beschränkung von *sup* jedoch einen Aspekt des *Verhaltens* des Netzes adressiert, zielt die Beschränkung von *sig* auf *strukturelle* Eigenschaften des Netzes ab. Die Ergebnisse dieses Kapitels und des Kapitels 6 zeigen nun: Aus parametrisierter Perspektive ist es leichter ein boolesches Netz zu synthetisieren, das bezügliche seines (Belegungs-) Verhaltens eingschränkt ist, als eines, dass bezüglich seiner (Platz-Umgebungs-) Struktur eingeschränkt ist.

Der parametrisierte Algorithmus dieses Kapitels funktioniert nur für boolesche Netztypen mit nop. Die parametrisierte Komplexität der Synthese boolescher Netze mit beschränkter Belegungszahl für die Netztypen ohne nop ist offen. In diesem Zusammenhang wäre zudem zunächst die (klassische) Komplexität des zu Grunde liegenden nichtparametrisierten Problems zu klären. Zum Beispiel haben wir in Abschnitt 4.2 gesehen, dass die klassischen Syntheseprobleme für alle Netztypen ohne nop und swap polynomiell sind. Gilt dies auch für die Synthese von Netzen mit beschränkter Belegungszahl? Analog gilt diese Frage auch für die Netztypen mit nop, deren klassische Syntheseprobleme polynomiell sind.

Zukünftige Forschung könnte auch die (parametrisierte) Komplexität der Synthese von Netzen mit beschränkter Belegungszahl für Netze untersuchen, deren Plätze mehr als eine Marke enthalten können. Dabei scheint es auf ersten Blick so, dass sich der Algorithmus dieses Kapitels auch für die Netztypen τ_{PT}^b und τ_{PPT}^b adaptieren ließe, denn auch hier ist der Supportwert eines Zustands Element einer endlichen Menge, nämlich $\{0, \ldots, b\}$. Spannender wäre es deshalb, die (parametrisierte) Komplexität des Problems für die Netztypen τ_{PT} und τ_{PPT} zu untersuchen, welche die mögliche Anzahl von Marken auf einem Platz nicht im Voraus einschränken.

Teil III

Verhaltensänderung zum Zwecke der Implementierbarkeit

Kapitel 9

Verhaltensmodifikation durch Neubeschriftung

Solange nicht ausdrücklich anders vereinbart, sei $\tau \in \{\tau_{PPT}, \tau_{PT}\}$ im Folgenden beliebig aber fest. Synthese umfasst die Aufgabe zu einem gegebenen Transitionssystem eine τ-Einbettung, τ-Sprachsimulation beziehungsweise τ-Realisierung zu finden, je nachdem nach welcher Implementierung im konkreten Fall gesucht wird. Unabhängig von der gesuchten Implementierung, gibt es nicht für jedes Transitionssystem A ein implementierendes τ-Netz. In diesem Fall bietet sich *Neubeschriftung* als Technik zur Modifikation von A zu einem implementierbaren Transitionssystem B an: B resultiert aus A, indem ursprünglich mit demselben Ereignis e beschriftete Kanten von A durch unterschiedliche Ereignisse $e_1, \ldots, e_k$ neu beschriftet werden. Die Ereignisse $e_1, \ldots, e_k$ werden als durch *Ereignisteilung* (engl. *label splitting* [CKLY98]) aus e entstanden interpretiert. Diese skizzierte Technik lässt sich gemäß der folgenden Definition 9.1 in natürlicher Weise auf eine beliebige Anzahl von Ereignissen von A fortsetzen.

Definition 9.1 (Ereignisteilung, Neubeschriftung)**.** Sei $A = (Z, E, \delta, \iota)$ ein Transitionssystem und $\mathfrak{E} = \{e_1, \ldots, e_n\} \subseteq E$ eine Teilmenge der Ereignisse von A. Die *Teilung der Ereignisse* $e_1, \ldots, e_n$ von $\mathfrak{E}$ in die paarweise voneinander (und von allen aus $E \setminus \mathfrak{E}$) verschiedenen Ereignisse $e_1^1, \ldots, e_1^{m_1}, \ldots, e_n^1, \ldots, e_n^{m_n}$, wobei $m_i \geq 2$ für alle $i \in \{1, \ldots, n\}$, ergibt die neue Ereignismenge $E' = (E \setminus \mathfrak{E}) \cup \bigcup_{i=1}^{n} \{e_i^1, \ldots, e_i^{m_i}\}$. Eine *$E'$-Neubeschriftung* von A ist ein Tripel (B, ϱ, φ), bestehend aus einem Transitionssystem $B = (Z, E', \delta', \iota)$ und einer Abbildung $\varrho : E' \to E$, sodass für alle $e \in E'$ gilt

$$\varrho(e) = \begin{cases} e_i, & \text{wenn es } i \in \{1, \ldots, n\} \text{ und } j \in \{1, \ldots, m_i\} \text{ mit } e = e_i^j \text{ gibt} \\ e, & \text{sonst} \end{cases}$$

und einer bijektiven Abbildung $\varphi : \Delta_B \to \Delta_A$ zwischen der (Kanten-) Menge $\Delta_B = \{(z, e, z') \mid z, z' \in Z, e \in E' : \delta'(z, e) = z'\}$ von B und $\Delta_A = \{(z, e, z') \mid z, z' \in Z, e \in E : \delta(z, e) = z'\}$ von A, sodass für alle $(z, e, z') \in \Delta_B$ gilt $\varphi((z, e, z')) = (z, \varrho(e), z')$. Wir nennen $\mathfrak{E}$ *die Menge der geteilten Ereignisse* (von A, die geteilt in B stattfinden).

Bemerkung 9.1. Die Forderung $m_i \geq 2$ für alle $i \in \{1, \ldots, m_i\}$ in Definition 9.1, sichert zu, dass ein in $\mathfrak{E}$ enthaltenes Ereignis tatsächlich geteilt wird und nicht nur eine einfache Umbenennung stattfindet.

Beispiel 9.1. Für das Transitionssystem A mit Ereignismenge $E = \{e\}$ aus Abbildung 9.1 gibt es kein τ-Netz, das es einbettet oder realisiert, denn das ZTP (z_0, z_1) ist nicht τ-lösbar: Wenn $R = (sup, sig)$ eine beliebige τ-Region ist, dann gilt $sup(z_0) = sup(z_1) - sig^-(e) + sig^+(e)$ und $sup(z_1) = sup(z_0) - sig^-(e) + sig^+(e)$. Dies impliziert $sup(z_0) = sup(z_1)$. Weil R beliebig war, ist (z_0, z_1) nicht τ-lösbar. Das Transitionssystem B mit Ereignismenge $E' = (E \setminus \{e\}) \cup \{e, \overline{e}\}$ aus Abbildung 9.1 ist jedoch einbettbar und realisierbar, wie das τ-Netz N aus Abbildung 9.1 zeigt. Insbesondere ist (B, ϱ, φ) eine E'-Neubeschriftung, wobei $\varrho(e) = \varrho(\overline{e}) = e$ und $\varphi((z_0, e, z_1)) = (z_0, e, z_1)$ und $\varphi((z_0, \overline{e}, z_1)) = (z_0, e, z_1)$.

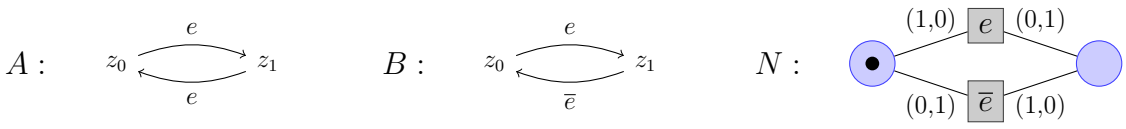


Abbildung 9.1: Transitionssysteme A und B sowie eine τ-Realisierung N von B.

Diese Technik der Neubeschriftung wurde urspünglich in [CKK+97a] für die Synthese von 1-gebundenen P/T-Netzen eingeführt und später von denselben Autoren für die Synthese von b-gebundenen P/T-Netzen erweitert [CCK10]. Die zugehörigen Heuristiken wurden in dem Synthese-Tool Petrify [CKK+97c] implementiert. Diese Heuristiken folgen einem *greedy*-Ansatz, der nicht-lösbare Trennungsproblem lösbar macht, indem sukzessive (lokal optimal) so wenig Ereignisse wie möglich geteilt werden, die sich einer passenden Signatur entziehen. In [Car12] wurden Beziehungen zwischen einzelnen Bestandteilen dieses Vorgehens und den NP-vollständigen Problemen Chromatic Number und Weighted Set Cover herausgearbeitet.

Die Erweiterung von *Ereignisteilung* auf eine beliebige Anzahl von Ereignissen von A macht *Neubeschriftung* zu einem mächtigen Werkzeug, das, unabhängig von der gesuchten Implementierung, stets ein implementierbares Transitionssystem liefern kann: spätestens wenn alle Kanten paarweise verschieden voneinander beschriftet sind, hat das resultierende Transitionssystem B sowohl die τ-ZTE als auch die τ-EZTE. (Man beachte, dass wir $\tau \in \{\tau_{PPT}, \tau_{PT}\}$ annehmen.) Mit zunehmender Anzahl der Ereignisse nimmt jedoch auch die Anzahl der Transitionen –und damit die Größe– des implementierenden Netzes zu. Dies nivelliert den entscheidenden Vorteil von Netzen, im Verhältnis zum Transitionssystem klein und damit überschaubarer zu sein. Darüber hinaus entfernt sich das resultierende Verhalten mit steigender Ereigniszahl zunehmend vom Eingabe-Verhalten. Wünschenswert ist es dagegen, das gegebene Transitionssystem so gering wie möglich zu modifizieren und dadurch dem eigentlichen Verhalten möglichst nah zu bleiben. Folglich möchten wir die Anzahl der zusätzlichen Ereignisse beziehungsweise die resultierende

Gesamtanzahl der durch Ereignisteilung entstandenen Ereignisse *minimieren*. Das aus diesen Überlegungen resultierende Optimierungsproblem führt in natürlicher Weise zu den folgenden drei Entscheidungsproblemen, deren Komplexität Gegenstand unserer Untersuchungen ist:

Neubeschriftung für τ-Einbettung	
Eingabe:	Ein endliches Transitionssystem $A = (Z, E, \delta, \iota)$ und $\kappa \in \mathbb{N}$.
Frage:	Gibt es eine E'-Neubeschriftung (B, ϱ, φ) von A, sodass $\lvert E' \rvert \leq \kappa$ erfüllt ist und B die τ-ZTE hat?

Neubeschriftung für τ-Sprachsimulation	
Eingabe:	Ein endliches Transitionssystem $A = (Z, E, \delta, \iota)$ und $\kappa \in \mathbb{N}$.
Frage:	Gibt es eine E'-Neubeschriftung (B, ϱ, φ) von A, sodass $\lvert E' \rvert \leq \kappa$ erfüllt ist und B die τ-EZTE hat?

Neubeschriftung für τ-Realisierung	
Eingabe:	Ein endliches Transitionssystem $A = (Z, E, \delta, \iota)$ und $\kappa \in \mathbb{N}$.
Frage:	Gibt es eine E'-Neubeschriftung (B, ϱ, φ) von A, sodass $\lvert E' \rvert \leq \kappa$ erfüllt ist und B die τ-ZTE und die τ-EZTE hat?

Schlachter und Wimmel [SW20] haben einerseits gezeigt, dass Neubeschriftung für τ-Einbettung NP-vollständig ist. Andererseits formulieren die Autoren die Komplexität der beiden anderen Entscheidungsprobleme als offene Frage. In Abschnitt 9.1 beantworten wir diese Frage und zeigen, dass Neubeschriftung für τ-Sprachsimulation und Neubeschriftung für τ-Realisierung NP-vollständig sind. Der Beweis der NP-Schwere basiert auf einer Reduktion des Problems Cubic Vertex Cover, die in einer Eingabe (A, κ) mit linearem Transitionssystem A resultiert. Insbesondere folgt daraus, dass beide Entscheidungsprobleme auch dann NP-vollständig sind, wenn das Transitionssystem der Eingabe linear ist.

In Abschnitt 9.2 werden wir auch die Komplexität des Problems Neubeschriftung für τ-Einbettung noch einmal untersuchen, wobei wir nun –im Gegensatz zu Schlachter und Wimmel [SW20]– den Grad des Eingabe-Transitionssystems berücksichtigen werden:

Definition 9.2 (Grad)**.** Sei $A = (Z, E, \delta, \iota)$ ein Transitionssystem. Der *Grad eines Zustands* $z \in Z$, notiert durch $deg(z)$, ist die maximale Anzahl von Kanten deren Ursprung beziehungsweise Ziel z ist, das heißt, $deg(z) = \max\{|\{e \in E \mid z \xrightarrow{e}\}|, |\{e \in E \mid \xrightarrow{e} z\}|\}$. Der *Grad von* A ist dann definiert als $\deg(A) = \max\{deg(z) \mid z \in Z\}$.

In [SW20] zeigen die Autoren die NP-Schwere des Problems der Neubeschriftung für τ-Einbettung durch eine Reduktion des Problems Subset Sum [GJ79]. Das Resultat dieser Reduktion ist ein Paar (A, κ), wobei das Transitionssystem A eine Verknüpfung von sechs Gadgets ist, welche die Autoren *strands* nennen. Jedes dieser Gadgets kann als Transitionssystem mit (höchstens) Grad drei interpretiert werden. Es ist relativ leicht zu

sehen, dass diese –ohne Verlust von Funktionalität– so „hintereinander“ arrangiert werden können, dass das resultierende Transitionssystem ebenso den Grad drei hat. Folglich ist NEUBESCHRIFTUNG FÜR τ-EINBETTUNG auch für Eingaben NP-vollständig, deren Transitionssystem höchstens den Grad drei hat. In Abschnitt 9.2 werden wir zeigen, dass NEUBESCHRIFTUNG FÜR τ-EINBETTUNG sogar dann NP-vollständig ist, wenn der Grad des Eingabe-Transitionssystems Zwei ist. Der Beweis der NP-Schwere basiert erneut auf einer Reduktion des Problems CUBIC VERTEX COVER, deren Resultat ein Paar (A, κ) ist, sodass $deg(A) = 2$. Darüber hinaus werden wir zeigen, dass diese Schranke scharf ist: Für Eingaben (A, κ) mit $deg(A) = 1$ ist NEUBESCHRIFTUNG FÜR τ-EINBETTUNG in Polynomialzeit entscheidbar.

Dies ist ein interessanter Kontrast zu den Ergebnissen für die anderen beiden Entscheidungsprobleme: NEUBESCHRIFTUNG FÜR τ-SPRACHSIMULATION und NEUBESCHRIFTUNG FÜR τ-REALISIERUNG sind auch für lineare Eingaben, das heißt insbesondere für Eingaben mit Grad eins, NP-vollständig.

Das folgende Lemma adressiert für alle eingeführten Entscheidungsprobleme, die Neubeschriftung betreffen, die Zugehörigkeit zu NP:

Lemma 9.1. NEUBESCHRIFTUNG FÜR τ-EINBETTUNG *und* NEUBESCHRIFTUNG FÜR τ-SPRACHSIMULATION *und* NEUBESCHRIFTUNG FÜR τ-REALISIERUNG *sind in* NP.

Beweis. Gemäß Definition 2.1 findet jedes Ereignis eines Transitionssystems an wenigstens einem Zustand statt. Daraus folgt für A und eine gesuchte Neubeschriftung (B, ϱ, φ), dass $|A_\Delta| = |B_\Delta| \leq |Z|^2 \cdot |E|$. Ebenso gilt $|E'| \leq |Z|^2 \cdot |E|$, denn andernfalls gäbe es wenigstens ein Ereignis in E', das nicht an einer Kante von B stattfindet. Es sind ϱ sowie φ durch $\{(e, e') \in E' \times E \mid \varrho(e) = e'\}$ sowie $\{(t, t') \in \Delta_B \times \Delta_A \mid \varphi(t) = t'\}$ vollständig bestimmt. Daher kann eine Lösung (B, ϱ, φ) nicht-deterministisch mit Zeitaufwand $\mathcal{O}(|Z|^2 \cdot |E|)$ geraten werden und es kann deterministisch in Zeit $\mathcal{O}(|Z|^4 \cdot |E|^2)$ verifiziert werden, dass (B, ϱ, φ) eine E'-Neubeschriftung von A ist, die $|E'| \leq \kappa$ erfüllt.

Schließlich hat B höchstens $|Z|^2$ ZTP und, wegen $|E'| \leq |Z|^2 \cdot |E|$, höchstens $|Z|^3 \cdot |E|^2$ EZTP, deren jeweilige τ-Lösbarkeit gemäß [BBD95, BD96a] mit polynomiellen Zeitaufwand in der Größe von B und damit (wegen $|\Delta_B| = |\Delta_A|$ und $|E'| \leq |Z|^2 \cdot |E|$) in der Größe von A entschieden werden kann. □

9.1 Neubeschriftung für Sprachsimulation und Realisierung

Der folgende Satz ist das Hauptresultat dieses Abschnitts:

Satz 9.1. *Sei* $\tau \in \{\tau_{PPT}, \tau_{PT}\}$. NEUBESCHRIFTUNG FÜR τ-SPRACHSIMULATION *und* NEUBESCHRIFTUNG FÜR τ-REALISIERUNG *sind auch dann* NP*-vollständig, wenn das Eingabe-Transitionssystem A linear ist.*

Der Beweis von Satz 9.1 basiert auf einer Reduktion des Problems Cubic Vertex Cover, dessen NP-Vollständigkeit in [FHJ98] bewiesen wurde:

Cubic Vertex Cover (CVC)	
Eingabe:	Ein Tripel $(\mathfrak{U}, M, \lambda)$ mit einer Menge $\mathfrak{U} = \{X_0, \ldots, X_{n-1}\}$ und einem Mengensystem $M = \{M_0, \ldots, M_{m-1}\}$, wobei $M_i = \{X_{i_0}, X_{i_1}\} \subseteq \mathfrak{U}$ für alle $i \in \{0, \ldots, m-1\}$ und $\lvert\{a \in M \mid X \in a\}\rvert = 3$ für alle $X \in \mathfrak{U}$, und einer natürlichen Zahl λ.
Frage:	Gibt es eine Knotenüberdeckung S für $(\mathfrak{U}, M)$, das heißt, $S \subseteq \mathfrak{U}$ und $\lvert S \cap M_i\rvert \neq \emptyset$ für alle $i \in \{0, \ldots, m-1\}$, die $\lvert S\rvert \leq \lambda$ erfüllt?

Beispiel 9.2 (CVC). Die Eingabe $(\mathfrak{U}, M, 3)$ mit Menge $\mathfrak{U} = \{X_0, X_1, X_2, X_3\}$ und Mengensystem $M = \{M_0, \ldots, M_5\}$, wobei $M_0 = \{X_0, X_1\}$, $M_1 = \{X_0, X_2\}$, $M_2 = \{X_0, X_3\}$, $M_3 = \{X_1, X_2\}$, $M_4 = \{X_1, X_3\}$ und $M_5 = \{X_2, X_3\}$, ermöglicht eine positive Entscheidung: $S = \{X_0, X_1, X_2\}$ ist eine gesuchte Knotenüberdeckung.

Solange nicht ausdrücklich anders vereinbart, sei $(\mathfrak{U}, M, \lambda)$ im weiteren Verlauf dieses Kapitels eine beliebige aber feste Eingabe von CVC, sodass $\mathfrak{U} = \{X_0, \ldots, X_{n-1}\}$ und $M = \{M_0, \ldots, M_{m-1}\}$ mit $M_i = \{X_{i_0}, X_{i_1}\}$ und $i_0 < i_1$ für alle $i \in \{0, \ldots, m-1\}$ sowie $\lambda \in \mathbb{N}$. Man beachte, dass $m = \frac{3n}{2}$, denn $(\mathfrak{U}, M)$ ist kubisch, das heißt, jedes Element von $\mathfrak{U}$ ist Element von genau drei Mengen aus M.

Für den Beweis von Satz 9.1 reduzieren wir $(\mathfrak{U}, M, \lambda)$ auf ein Paar (A, κ) mit linearem Transitionssystem A und natürlicher Zahl κ sodass gilt: Es gibt genau dann eine E'-Neubeschriftung (B, ϱ, φ) von A, sodass $\lvert E'\rvert \leq \kappa$ erfüllt ist und B die τ-EZTE hat, wenn es eine Knotenüberdeckung mit höchstens λ Elementen für $(\mathfrak{U}, M)$ gibt. Weil B linear ist, folgt mit Lemma 2.5, dass die τ-EZTE von B die τ-ZTE von B impliziert. Dies impliziert, dass $(\mathfrak{U}, M, \lambda)$ genau dann eine positive Entscheidung erlaubt, wenn (A, κ) eine positive Entscheidung erlaubt, unabhängig davon, nach welcher der beiden interessierenden Implementierungen gefragt wird.

Zunächst definieren wir $\kappa = n + 2(m-1) + \lambda$, wobei $n + 2(m-1)$ die Anzahl der Ereignisse des angekündigten Transitionssystems $A = (Z, E, \delta, \iota)$ ist. Da jedes geteilte Ereignis die Anzahl der Ereignisse (relativ zur ursprünglichen Ereignisanzahl) um wenigstens 1 erhöht, folgt sofort, dass höchstens λ Ereignisse von A geteilt werden können, das heißt $\lvert\mathfrak{E}\rvert \leq \lambda$.

Für jedes $i \in \{0, \ldots, m-1\}$ hat das Transitionssystem A das folgende Gadget T_i, auf dem die Elemente von $M_i = \{X_{i_0}, X_{i_1}\}$ als Ereignisse stattfinden:

$$T_i = \quad t_{i,0} \xrightarrow{X_{i_0}} t_{i,1} \xrightarrow{X_{i_1}} t_{i,2} \xrightarrow{X_{i_1}} t_{i,3} \xrightarrow{X_{i_0}} t_{i,4} \xrightarrow{X_{i_0}} t_{i,5}$$

Das Transitionssystem A mit Initialzustand $t_{0,0}$ ist dann gemäß Definition 5.9 die lineare Verbindung der Basis $(T_0, \ldots, T_{m-1})$. Man beachte, dass die Ereignismenge $E = \mathfrak{U} \cup \ominus \cup \oplus$ von A genau $\lvert E\rvert = n + 2 \cdot (m-1)$ Ereignisse hat. Abbildung 9.2 zeigt das Transitionssystem A, das gemäß der eingeführten Reduktion aus Beispiel 9.2 resultiert.

Für jedes $i \in \{0, \ldots, m-1\}$ korrespondiert T_i mit dem unlösbaren binären Wort $X_{i_0}X_{i_1}X_{i_1}X_{i_0}X_{i_0}$ [BESW16]. Der folgende Fakt impliziert für alle $i \in \{0, \ldots, m-1\}$ tatsächlich, dass T_i in keiner implementierbaren Neubeschriftung von A vorhanden sein kann:

Fakt 9.1. *Sei $i \in \{0, \ldots, m-1\}$ beliebig aber fest. Es gibt keine τ-Region von T_i, die das EZTP $(X_{i_0}, t_{i,2})$ löst.*

Beweis. Sei $i \in \{0, \ldots, m-1\}$ beliebig aber fest. Angenommen, $R = (sup, sig)$ ist eine τ-Region von T_i, die das EZTP $\alpha = (X_{i_0}, t_{i,2})$ löst. Nun gilt (1) $sig^-(X_{i_0}) \leq sup(t_{i,0})$, denn X_{i_0} findet an t_{i_0} statt; (2) $sup(t_{i,2}) < sig^-(X_{i_0})$, denn R löst α; (3) $sig^-(X_{i_0}) \leq sup(t_{i,4})$, denn X_{i_0} findet an $t_{i,4}$ statt. R ist eine τ-Region, daher ist $sup(z') = sup(z) - sig^-(e) + sig^+(e)$ für alle $z \xrightarrow{e} z' \in A$ erfüllt. Mit Ungleichungen (2) beziehungsweise (3) erhalten wir deshalb die folgenden Ungleichungen (4) beziehungsweise (5):

(4) $sup(t_{i,0}) - (sig^-(X_{i_0}) + sig^-(X_{i_1})) + (sig^+(X_{i_0}) + sig^+(X_{i_1})) < sig^-(X_{i_0})$
(5) $sig^-(X_{i_0}) \leq sup(t_{i,0}) - 2 \cdot (sig^-(X_{i_0}) + sig^-(X_{i_1})) + 2 \cdot (sig^+(X_{i_0}) + sig^+(X_{i_1}))$

Wenn wir in (4) auf beiden Seiten $sup(t_{i,0})$ subtrahieren und $sig^-(X_{i_0}) - sup(t_{i,0}) \leq 0$ aus (1) herleiten, dann erhalten wir $-(sig^-(X_{i_0}) + sig^-(X_{i_1})) + (sig^+(X_{i_0}) + sig^+(X_{i_1})) < 0$. Die Ungleichungen (4) und (5) ergeben aber $0 \leq -(sig^-(X_{i_0}) + sig^-(X_{i_1})) + (sig^+(X_{i_0}) + sig^+(X_{i_1}))$. Mithin war unsere Annahme falsch und es gibt keine τ-Region, die α löst. □

Lemma 9.2. *Wenn es eine E'-Neubeschriftung (B, ϱ, φ) von A gibt, sodass $|E'| \leq \kappa$ erfüllt ist und B die τ-EZTE hat, dann gibt es eine Knotenüberdeckung mit höchstens λ Elementen für $(\mathfrak{U}, M)$.*

Beweis. Sei (B, ϱ, φ) eine entsprechende E'-Neubeschriftung von A und sei $\mathfrak{E}$ die Menge der Ereignisse, die in B geteilt stattfinden. Weil $|E'| \leq n + 2(m-1) + \lambda$ und $|E| = n + 2(m-1)$, gilt $|\mathfrak{E}| \leq |E'| - |E| \leq \lambda$ nach Definition 9.1. Sei $i \in \{0, \ldots, m-1\}$ beliebig aber fest. Man überlegt sich, dass die Einschränkung R_i einer τ-Region R von B auf das Gadget T_i eine τ-Region von T_i definiert. Wenn R das EZTP $(X_{i_0}, t_{i,2})$ (in B) löst, dass heißt, $sig^-(X_{i_0}) > sup(t_{i,2})$ erfüllt, dann löst R_i dieses EZTP ebenso (in T_i). Gemäß Fakt 9.1, kann es eine solche Region R_i nicht geben. Weil B die τ-EZTE hat, folgt daraus, dass es T_i in B nicht geben kann. Weil B durch Neubeschriftung aus A resultiert, ist dies nur möglich, wenn eine Kante von T_i in B neu beschriftet ist. Deshalb gibt es ein Ereignis $e \in \{X_{i_0}, X_{i_1}\} \cap \mathfrak{E}$. Weil i beliebig war, gilt dies gleichzeitig für alle Pfade $T_0, \ldots, T_{m-1}$, das heißt mit $S = \mathfrak{U} \cap \mathfrak{E}$ gilt, dass $S \cap M_i \neq \emptyset$ für alle $i \in \{0, \ldots, m-1\}$. Insbesondere ist S eine gesuchte Knotenüberdeckung, denn es gilt $|S| = |\mathfrak{U} \cap \mathfrak{E}| \leq |\mathfrak{E}| \leq \lambda$. □

Es bleibt zu zeigen, dass es eine gesuchte E'-Neubeschriftung (B, ϱ, φ) von A gibt, wenn es eine gesuchte Knotenüberdeckung für $(\mathfrak{U}, M)$ gibt. Sei also $S = \{X_{j_0}, \ldots, X_{j_{\lambda-1}}\} \subseteq \mathfrak{U}$ eine solche Knotenüberdeckung. Für jedes $i \in \{0, \ldots, \lambda-1\}$ teilen wir das Ereignis X_{j_i} in die zwei Ereignisse X_{j_i} und $\overline{X_{j_i}}$ und erhalten die Ereignismenge $E' = (E \setminus S) \cup \bigcup_{i=0}^{\lambda-1} \{X_{j_i}, \overline{X_{j_i}}\}$.

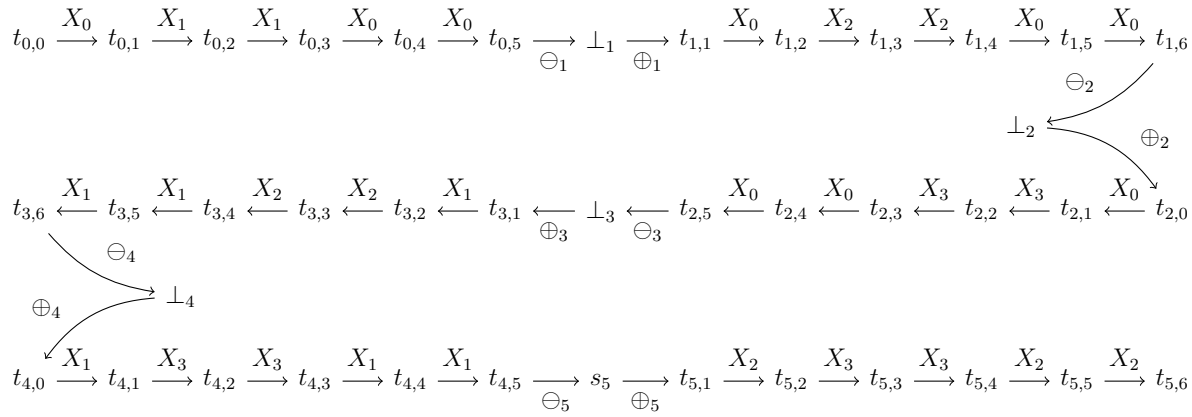


Abbildung 9.2: Das Transitionssystem A, das gemäß Reduktion aus Beispiel 9.2 resultiert.

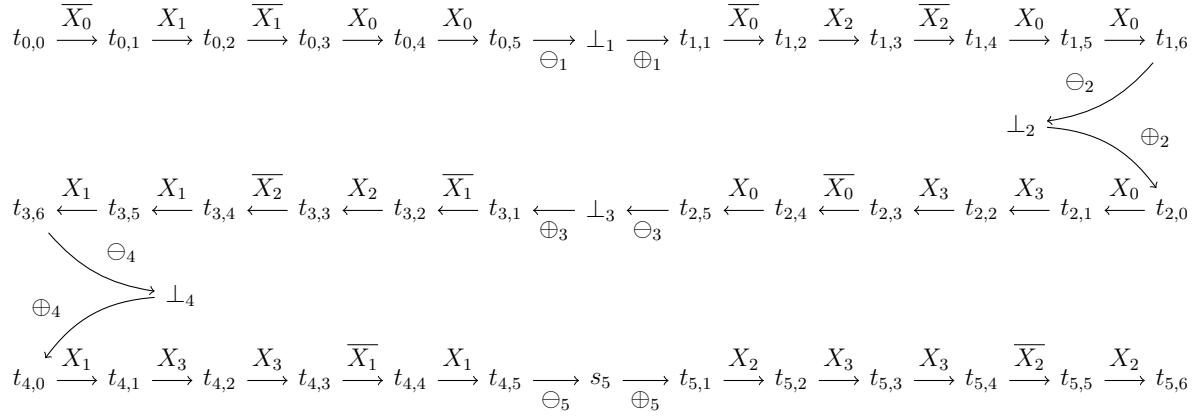


Abbildung 9.3: Das Transitionssystem B der E'-Neubeschriftung (B, ϱ, φ) von A (Abbildung 9.2) basierend auf der Knotenüberdeckung $\{X_0, X_1, X_2\}$ von $(\mathfrak{U}, M)$ aus Beispiel 9.2.

Wir definieren eine passende E'-Neubeschriftung (B, ϱ, φ) mit $\varrho(X_{j_i}) = \varrho(\overline{X_{j_i}}) = X_{j_i}$ für alle $i \in \{0, \ldots, \lambda - 1\}$ und $\varrho(e) = e$ für alle $e \in E \setminus S$ und Transitionssystem $B = (Z, E', \delta', t_{0,0})$ durch Neubeschriftung der Pfade $T_0, \ldots, T_{m-1}$; das heißt, für alle $e \in \ominus \cup \oplus$ stimmen A und B überein: $z \xrightarrow{e} z' \in A$ genau dann, wenn $z \xrightarrow{e} z' \in B$; für alle $i \in \{0, \ldots, m-1\}$ liefert die Definition von δ' auf den Zuständen von T_i gemäß der folgenden Regeln einen neuen Pfad $\overline{T_i}$:

- wenn $X_{i_0} \in S$ und $X_{i_1} \notin S$, dann $\overline{T_i} = t_{i,0} \xrightarrow{X_{i_0}} t_{i,1} \xrightarrow{X_{i_1}} t_{i,2}, \xrightarrow{X_{i_1}} t_{i,3} \xrightarrow{\overline{X_{i_0}}} t_{i,4} \xrightarrow{X_{i_0}} t_{i,5}$;
- wenn $X_{i_0}, X_{i_1} \in S$, dann $\overline{T_i} = t_{i,0} \xrightarrow{\overline{X_{i_0}}} t_{i,1} \xrightarrow{X_{i_1}} t_{i,2}, \xrightarrow{\overline{X_{i_1}}} t_{i,3} \xrightarrow{X_{i_0}} t_{i,4} \xrightarrow{X_{i_0}} t_{i,5}$;
- wenn $X_{i_0} \notin S$ und $X_{i_1} \in S$, dann $\overline{T_i} = t_{i,0} \xrightarrow{X_{i_0}} t_{i,1} \xrightarrow{X_{i_1}} t_{i,2}, \xrightarrow{\overline{X_{i_1}}} t_{i,3} \xrightarrow{X_{i_0}} t_{i,4} \xrightarrow{X_{i_0}} t_{i,5}$.

Das resultierende Transitionssystem ist B und die bijektive Funktion φ erhalten wir auf naheliegende Weise. Man beachte außerdem, dass B als die lineare Verbindung der Basis $C = (\overline{T_0}, \ldots, \overline{T_{m-1}})$ interpretiert werden kann. Abbildung 9.3 zeigt das Transitionssystem B, welches gemäß des beschriebenen Vorgehens aus dem Transitionssystem A aus Abbildung 9.2 und Beispiel 9.2 resultiert.

Wir werden im Folgenden die Lösbarkeit der Ereignisse von B diskutieren. Wenn möglich präsentieren wir Regionen $R = (sup, sig)$ wieder vereinfacht, durch den Wert $sup(t_{0,0})$ des Initialzustands $t_{0,0}$ sowie sig. Ereignisse gleicher Signatur (m, n) werden in einer Menge $\mathcal{E}^R_{m,n} = \{e \in E(A) \mid sig(e) = (m, n)\}$ zusammengefasst. Alle Ereignisse, die nicht explizit in einer Menge $\mathcal{E}^R_{m,n}$ mit $m \neq n$ aufgeführt sind, bekommen die Signatur $(0, 0)$, sodass wir die Menge $\mathcal{E}^R_{0,0}$ der Einfachheit halber jeweils nicht explizit definieren.

Jedes der Ereignisse aus $\oplus \cup \ominus$ findet gemäß der Definition 5.9 linearer Verbindungen genau einmal in A statt. Gemäß des folgenden Fakts sind alle diese Ereignisse τ-lösbar:

Fakt 9.2. *Sei $A = z_0 \xrightarrow{e_1} \ldots \xrightarrow{e_n} z_n$ ein lineares Transitionssystem. Für alle $i \in \{1, \ldots, n\}$ gilt: wenn e_i genau einmal in A stattfindet, dann ist e_i τ-lösbar.*

Beweis. Sei $i \in \{1, \ldots, n\}$ beliebig aber fest, sodass e_i genau einmal in A stattfindet, das heißt $z_{i-1} \xrightarrow{e_i}$ und $\neg z \xrightarrow{e_i}$ für alle $z \in \{z_0, \ldots, z_n\} \setminus \{z_{i-1}\}$. Die folgende τ-Region $R = (sup, sig)$ löst (e_i, z) für alle $z \in \{z_0, \ldots, z_{i-2}\}$: $sup(z_0) = 0$ und $\mathcal{E}^R_{0,1} = \{e_1, \ldots, e_{i-1}\}$ und $\mathcal{E}^R_{i-1,0} = \{e_i\}$.

Darüber hinaus ist klar, dass (e_i, z) für alle $z \in \{z_i, \ldots, z_n\}$ gelöst werden kann. □

Fakt 9.3. *Es gibt einen Zeugen $\mathcal{R}$ für die τ-ZTE und die τ-EZTE von B.*

Beweis. Weil B linear ist, genügt es zu argumentieren, dass es einen Zeugen für die τ-EZTE von B gibt. Wegen Fakt 9.2 sind alle Ereignisse aus $\oplus \cup \ominus$ τ-lösbar, denn diese finden jeweils genau einmal in B statt. Es genügt also, die Ereignisse aus $E' \setminus (\oplus \cup \ominus)$ auf ihre τ-Lösbarkeit hin zu untersuchen.

Sei also $x \in E' \setminus (\oplus \cup \ominus)$ beliebig aber fest. Weil G kubisch ist, gibt es genau drei Indizes $i < j < k \in \{0, \ldots, m-1\}$, sodass $x \in E(\overline{T_\ell})$ für alle $\ell \in \{i, j, k\}$. Sei für alle $\ell \in \{i, j, k\}$ durch $\text{anz}_\ell(x)$ die Anzahl der Vorkommen von x in T'_ℓ notiert.

Die folgende τ-Region $R = (sup, sig)$ löst (x, z) für alle $z \in Z(B) \setminus (Z(\overline{T_i}) \cup Z(\overline{T_j}) \cup Z(\overline{T_k}))$: Wenn $i = 0$, dann $sup(t_{0,0}) = \text{anz}_0(x)$, andernfalls $sup(t_{0,0}) = 0$; außerdem $\mathcal{E}^R_{0,\text{anz}_\ell(x)} = \{\ominus_i, \ominus_j, \ominus_k\}$ und $\mathcal{E}^R_{1,0} = \{x\}$.

Es bleibt zu zeigen, dass ein EZTP (x, z) auch dann τ-lösbar ist, wenn x und z zum gleichen Gadget $\overline{T_i}$ gehören, wobei $i \in \{0, \ldots, m-1\}$. Wir werden dazu wie folgt vorgehen: Sei $i \in \{0, \ldots, m-1\}$ beliebig aber fest. Wir argumentieren für alle möglichen Fälle $X_{i_0} \in S, X_{i_1} \notin S$ und $X_{i_0} \notin S, X_{i_1} \in S$ und $X_{i_0}, X_{i_1} \in S$, dass das EZTP (x, z) für alle Ereignisse $x \in \{X_{i_0}, X_{i_1}, \overline{X_{i_0}}, \overline{X_{i_1}}\}$ und Zustände $z \in Z(\overline{T_i})$ mit $\neg z \xrightarrow{x}$ τ-lösbar ist. Aus der Beliebigkeit von i folgt dann die τ-EZTE von B. Darüber hinaus werden wir uns darauf beschränken, entsprechende τ-Rümpfe für die Basis $C = (\overline{T_0}, \ldots, \overline{T_{m-1}})$ anzugeben; analog zu Lemma 5.3 zeigt man leicht, dass sich diese jeweils zu einer passenden τ-Region

von B fortsetzen lassen. Sei also $Z_\iota = \{t_{0,0}, \ldots, t_{m-1,0}\}$ die Menge der Initialzustände der Gadgets und $\underline{E} = E(B) \setminus (\oplus \cup \ominus)$ die Menge der Ereignisse dieser Basis sowie $\mathcal{E}^R_{m,n} = \{e \in \underline{E} \mid sig(e) = (m,n)\}$ für einen τ-Rumpf $R = (sup, sig)$ von C.

Wir diskutieren nun explizit den Fall $X_{i_0} \in S$ und $X_{i_1} \notin S$, das heißt:

$$\overline{T_i} = t_{i,0} \xrightarrow{X_{i_0}} t_{i,1} \xrightarrow{X_{i_1}} t_{i,2}, \xrightarrow{X_{i_1}} t_{i,3} \xrightarrow{\overline{X_{i_0}}} t_{i,4} \xrightarrow{X_{i_0}} t_{i,5}$$

(X_{i_0} und $\overline{X_{i_0}}$): Sei $j \neq k \in \{0, \ldots, m-1\} \setminus \{i\}$ sodass $X_{i_0} \in M_j \cap M_k$. Der folgende τ-Rumpf $R_1 = (sup_1, sig_1)$ löst (X_{i_0}, z) für alle $z \in \{t_{i,1}, t_{i,2}, t_{i,3}, t_{i,5}\}$: Für alle $z \in Z_\iota$, wenn $z = t_{i,0}$, dann $sup_1(z) = 1$; wenn $z \in \{t_{j,0}, t_{k,0}\}$, dann $sup_1(z) = 2$; andernfalls $sup_1(z) = 0$; außerdem $\mathcal{E}^{R_1}_{1,0} = \{X_{i_0}\}$ und $\mathcal{E}_{0,1} = \{\overline{X_{i_0}}\}$. (Man beachte, dass $\mathrm{anz}_j(e), \mathrm{anz}_\ell(e) \leq 2$, denn $X_{i_0} \in S$.)

Der folgende τ-Rumpf $R_2 = (sup_2, sig_2)$ löst $(\overline{X_{i_0}}, z)$ für alle $z \in \{t_{i,0}, t_{i,1}, t_{i,2}, t_{i,4}, t_{i,5}\}$: für alle $z \in Z_\iota$, wenn $z \in \{t_{j,0}, t_{k,0}\}$, dann $sup_2(z) = 2$; andernfalls $sup_2(z) = 0$; außerdem $\mathcal{E}^{R_2}_{2,0} = \{\overline{X_{i_0}}\}$ und $\mathcal{E}^{R_2}_{0,1} = \{X_{i_1}\}$. (Man beachte, dass $\mathrm{anz}_j(X_{i_0}), \mathrm{anz}_\ell(X_{i_0}) = 1$.)

(X_{i_1}): Sei $j \neq k \in \{0, \ldots, m-1\} \setminus \{i\}$, sodass $X_{i_1} \in M_j \cap M_k$. Der folgende τ-Rumpf $R_3 = (sup_3, sig_3)$ löst (X_{i_1}, z) für alle $z \in \{t_{i,0}, t_{i,3}, t_{i,4}\}$: für alle $z \in Z_\iota$, wenn $z \in \{t_{j,0}, t_{k,0}\}$, dann $sup_3(z) = 3$; andernfalls $sup_3(z) = 0$; außerdem $\mathcal{E}^{R_3}_{1,0} = \{X_{i_1}\}$ und $\mathcal{E}^{R_3}_{0,2} = \{X_{i_0}\}$. (Man beachte, dass $\mathrm{anz}_j(X_{i_1}), \mathrm{anz}_\ell(X_{i_1}) \leq 3$.)

Der folgende τ-Rumpf $R_4 = (sup_4, sig_4)$ löst $(X_{i_1}, t_{i,5})$: für alle $z \in Z_\iota$, wenn $z = t_{i,0}$, dann $sup_4(z) = 2$; wenn $z \in \{t_{j,0}, t_{k,0}\}$, dann $sup(z) = 3$; andernfalls $sup(z) = 0$; außerdem $\mathcal{E}^{R_4}_{1,0} = \{X_{i_1}\}$.

Analog findet man lösende τ-Regionen für alle EZTP von $\overline{T_i}$ für die verbleibenden Fälle $X_{i_0} \notin S, X_{i_1} \in S$ beziehungsweise $X_{i_0}, X_{i_1} \in S$. Weil i beliebig war, folgt das Lemma. □

Ausgehend von der Knotenüberdeckung S mit höchstens λ Elementen für $(\mathfrak{U}, M)$ haben wir die Neubeschriftung B für A konstruiert, die $|E'| \leq \kappa$ erfüllt. Mit Fakt 9.3 erhalten wir deshalb das folgende Lemma, das mit der Aussage von Lemma 9.2 sowie der Tatsache, dass die Reduktion offensichtlich polynomiell ist, den Beweis von Satz 9.1 abschließt:

Lemma 9.3. *Wenn es eine Knotenüberdeckung mit höchstens λ Elementen für $(\mathfrak{U}, M)$ gibt, dann gibt es eine E'-Neubeschriftung (B, ϱ, φ) für A, die $|E'| \leq \kappa$ erfüllt und die τ-ZTE und die τ-EZTE hat.*

9.2 Neubeschriftung für Einbettung

Der folgende Satz ist das Hauptresultat dieses Abschnitts:

Satz 9.2. Neubeschriftung für τ-Einbettung *ist auch dann NP-vollständig, wenn das Eingabe-Transitionssystem A höchstens Grad zwei hat. Wenn A Grad eins hat, dann ist* Neubeschriftung für τ-Einbettung *in Polynomialzeit entscheidbar.*

Für den Beweis von Satz 9.2 starten wir mit dem Polynomialzeit-Ergebnis. Zunächst ist leicht zu sehen, dass ein lineares Transitionssystem $A = z_0 \xrightarrow{e_1} \dots \xrightarrow{e_n} z_n$ stets die τ-ZTE hat: die τ-Region $R = (sup, sig)$, die durch $sup(z_i) = i$ für alle $i \in \{0, \dots, n\}$ und $sig(e) = (0, 1)$ für alle $e \in E$ definiert ist, bezeugt die τ-ZTE von A. Das folgende Lemma adressiert nicht-lineare 1-gebundene Transitionssysteme A, das heißt gerichtete beschriftete Kreise. Es besagt im Wesentlichen, dass A die τ-ZTE hat, wenn es ein Ereignis von A gibt, das genau einmal stattfindet:

Lemma 9.4. *Sei $A = z_0 \xrightarrow{e_1} \dots \xrightarrow{e_n} z_n$ ein nicht-lineares Transitionssystem mit Grad eins, das heißt, $z_0 = z_n$ und $z_i \neq z_j$ für alle $i \neq j \in \{0, \dots, n-1\}$. Wenn es ein $i \in \{0, \dots, n-1\}$ gibt, sodass $\neg z_j \xrightarrow{e_i}$ für alle $j \in \{0, \dots, n\} \setminus \{i-1\}$, das heißt, e_i findet genau einmal in A statt, dann hat A die τ-ZTE.*

Beweis. Sei (z, z') ein beliebiges aber festes ZTP von A mit $z_0 = z$ und $z_j = z'$ und $i \in \{j+1, \dots, n\}$. Die Situation lässt sich wie folgt skizzieren, wobei $i - 1 = j + k$ and $n = j + k + \ell + 1$:

$$A = z_0 \xrightarrow{e_1} \dots \xrightarrow{e_j} z_j \xrightarrow{e_{j+1}} \dots \xrightarrow{e_{j+k}} z_{i-1} \xrightarrow{e_i} z_i \xrightarrow{e_{i+1}} \dots \xrightarrow{e_{i+\ell}} z_n$$

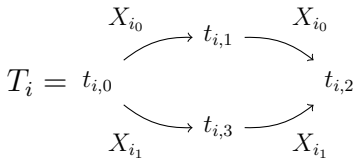
Diese Annahme ist keine wesentliche Einschränkung, weil A ein Kreis ist und durch einfache Umbenennung der Zustände, in die gewünschte Form gebracht werden kann.

Die folgende τ-Region $R = (sup, sig)$ löst (z_0, z_j): $sup(z_0) = \ell$ und $\mathcal{E}^R_{0,1} = E \setminus \{e_i\}$ und $\mathcal{E}^R_{j+k+\ell,0} = \{e_i\}$. Dies impliziert $sup(z_0) = \ell \neq \ell + j = sup(z_j)$, weil $j > 0$. Somit wird (z_0, z_j) von R gelöst und die Behauptung folgt. □

Wenn nun $A = (Z, E, \delta, \iota)$ ein nicht-lineares Transitionssystem mit Grad eins ist, welches nicht die τ-ZTE hat, dann muss, gemäß Lemma 9.4, jedes Ereignis von A wenigstens zweimal stattfinden. In diesem Fall erhalten wir eine E'-Neubeschriftung (B, ϱ, φ) von A, welche die τ-ZTE hat, durch Neubeschriftung einer beliebigen Kante von A: Sei $q \xrightarrow{x} q' \in A$ beliebig aber fest und $E' = (E \setminus \{x\}) \cup \{x, \overline{x}\}$. Die E'-Neubeschriftung (B, ϱ, φ) von A, welche in naheliegender Weise durch die Neubeschriftung $q \xrightarrow{\overline{x}} q'$ der Kante $q \xrightarrow{x} q'$ entsteht, erfüllt $|E'| = |E| + 1$ und hat gemäß Lemma 9.4 die τ-ZTE, denn $\overline{x}$ findet einmalig statt. Folglich erlaubt eine Eingabe (A, κ) mit nicht-linearem Transitionssystem A mit Grad eins genau dann eine positive Entscheidung, wenn $|E| + 1 \leq \kappa$.

Um den Beweis von Satz 9.2 zu vervollständigen, werden nun Eingaben (A, κ) betrachtet, sodass $deg(A) = 2$. Wir zeigen durch eine Reduktion von CVC, dass Neubeschriftung für τ-Einbettung eingeschränkt auf solche Eingaben NP-vollständig ist. Dazu wird die beliebige aber feste Instanz $(\mathfrak{U}, M, \lambda)$ des vorherigen Abschnitts auf eine Eingabe (A, κ) für Neubeschriftung für τ-Einbettung reduziert, wobei A Grad zwei hat. Insbesondere gibt es genau dann eine Knotenüberdeckung mit höchstens λ Elementen für $(\mathfrak{U}, M)$, wenn es für A eine gesuchte E'-Neubeschriftung (B, ϱ, φ) mit $|E'| \leq \kappa$ gibt.

Zunächst definieren wir $\kappa = n+m-1+\lambda$, wobei $n+m-1$ der Anzahl der Ereignisse von $A = (Z, E, \delta, \iota)$ entspricht, das heißt, λ beschränkt erneut die Anzahl der (in B) geteilt vorkommenden Ereignisse von A. Das Transitionssystem A hat für jedes $i \in \{0, \ldots, m-1\}$ das folgende Gadget T_i, auf dem die Elemente von $M_i = \{X_{i_0}, X_{i_1}\}$ als Ereignisse stattfinden:



Wir nutzen die Ereignisse $\oplus_1, \ldots, \oplus_{m-1}$ und für alle $i \in \{1, \ldots, m-1\}$ die Kante $t_{i-1,2} \xrightarrow{\oplus_i} t_{i,0}$, um $T_0, \ldots, T_{m-1}$ gemäß folgender Skizze zu A zu verbinden:

$$A = T_0 \xrightarrow{\oplus_1} T_1 \xrightarrow{\oplus_2} \cdots \xrightarrow{\oplus_{m-2}} T_{m-2} \xrightarrow{\oplus_{m-1}} T_{m-1}$$

Es ist $\iota = t_{0,0}$ der Initialzustand von A und mit $\oplus = \{\oplus_1, \ldots, \oplus_{m-1}\}$ gilt für die Menge $E(A) = \mathfrak{U} \cup \oplus$, dass $|E(A)| = n+m-1$. Es ist leicht zu sehen, dass A den Grad zwei hat.

Der folgende Fakt impliziert, dass die Gadgets $T_0, \ldots, T_{m-1}$ in keiner Neubeschriftung von A vorhanden sein können, die durch eine Einbettung implementierbar ist:

Lemma 9.5. *Sei $i \in \{0, \ldots, m-1\}$ beliebig aber fest. Es gibt keine τ-Region von T_i, die das ZTP $(t_{i,1}, t_{i,3})$ löst.*

Beweis. Sei $R = (sup, sig)$ eine beliebige aber feste τ-Region von T_i. Weil $z \xrightarrow{e} z' \in T_i$ stets $sup(z) \xrightarrow{sig(e)} sup(z') \in \tau$, das heißt $sup(z') = sup(z) - sig^-(e) + sig^+(e)$, impliziert, gelten die folgenden Gleichungen:

$$sup(t_{i,1}) = sup(t_{i,0}) - sig^-(X_{i_0}) + sig^+(X_{i_0}) \tag{9.1}$$
$$sup(t_{i,2}) = sup(t_{i,0}) - 2sig^-(X_{i_0}) + 2sig^+(X_{i_0}) \tag{9.2}$$
$$sup(t_{i,3}) = sup(t_{i,0}) - sig^-(X_{i_1}) + sig^+(X_{i_1}) \tag{9.3}$$
$$sup(t_{i,2}) = sup(t_{i,0}) - 2sig^-(X_{i_1}) + 2sig^+(X_{i_1}) \tag{9.4}$$

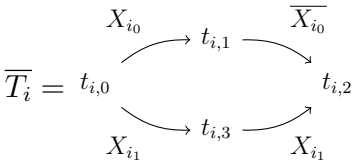
Subtrahieren der Gleichung 9.4 von der der Gleichung 9.2 liefert $0 = 2 \cdot (-sig^-(X_{i_0}) + sig^+(X_{i_0}) + sig^-(X_{i_1}) - sig^+(X_{i_1}))$ und somit $-sig^-(X_{i_1}) + sig^+(X_{i_1}) = -sig^-(X_{i_0}) + sig^+(X_{i_0})$. Mit den Gleichung 9.1 und 9.3 impliziert dies $sup(t_{i,1}) = sup(t_{i,3})$. Folglich löst R nicht $(t_{i,1}, t_{i,3})$ und weil R beliebig war, folgt die Behauptung. □

Lemma 9.6. *Wenn es eine E'-Neubeschriftung (B, ϱ, φ) von A gibt, die $|E'| \leq \kappa$ erfüllt und die τ-ZTE hat, dann gibt es eine Knotenüberdeckung mit höchstens λ Elementen für $(\mathfrak{U}, M)$.*

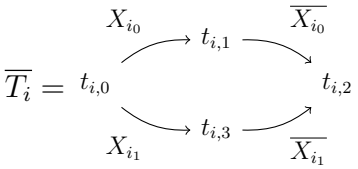
Beweis. Sei (B, ϱ, φ) eine entsprechende E'-Neubeschriftung von A und $\mathfrak{E}$ die Menge der Ereignisse von A, die geteilt in B vorkommen. Gemäß Definition 9.1 enthält E' für jedes geteilte Ereignis aus $\mathfrak{E}$ wenigstens zwei Ereignisse. Daher gilt $|E|-|\mathfrak{E}|+2\cdot|\mathfrak{E}| = |E|+|\mathfrak{E}| \leq |E'|$. Dies impliziert $|\mathfrak{E}| \leq |E'| - |E|$. Des Weiteren gelten $|E'| \leq n + m - 1 + \lambda$ und $|E| = n+m-1$. Somit erhalten wir $|\mathfrak{E}| \leq \lambda$. Weil B die τ-ZTE hat, ist das ZTP $(t_{i,1}, t_{i,3})$ für alle $i \in \{0, \ldots, m-1\}$ τ-lösbar. Für alle $i \in \{0, \ldots, m-1\}$ ist die Einschränkung R_i einer Region R von B auf das Gadget T_i eine Region. Gemäß Lemma 9.5 enthält B deshalb keines der Gadgets $T_0, \ldots, T_{m-1}$ mit der ursprünglichen Beschriftung. Weil B durch Neubeschriftung der Kanten von A entsteht, impliziert dies $\mathfrak{E} \cap E(T_i) \neq \emptyset$ für alle $i \in \{0, \ldots, m-1\}$. Insbesondere ist $S = \mathfrak{U} \cap \mathfrak{E}$ eine Knotenüberdeckung mit höchstens λ Elementen für $(\mathfrak{U}, M)$, denn es gilt $|\mathfrak{U} \cap \mathfrak{E}| \leq |\mathfrak{E}| \leq \lambda$. □

Um den Beweis von Satz 9.2 zu komplettieren, bleibt zu zeigen, dass es eine gesuchte E'-Neubeschriftung (B, ϱ, φ) von A gibt, wenn es eine Knotenüberdeckung mit höchstens λ Elementen für $(\mathfrak{U}, M)$ gibt. Sei also $S = \{X_{j_0}, \ldots, X_{j_{\lambda-1}}\} \subseteq \mathfrak{U}$ eine solche Knotenüberdeckung und sei $E' = (E \setminus S) \cup \{X_{j_0}, \overline{X_{j_0}}, \ldots, X_{j_{\lambda-1}}, \overline{X_{j_{\lambda-1}}}\}$. Sei $\varrho(X_{j_i}) = \varrho(\overline{X_{j_i}}) = X_{j_i}$ für alle $i \in \{0, \ldots, \lambda\}$ und $\varrho(e) = e$ für alle $E \setminus S$. Wir erhalten eine passende E'-Neubeschriftung (B, ϱ, φ) von A mit Transitionssystem $B = (Z, E', \delta', t_{0,0})$, wobei δ' auf den $\oplus$-beschrifteten Kanten mit δ übereinstimmt und für jedes $i \in \{0, \ldots, m-1\}$ bezüglich (der Zustände von) T_i wie folgt definiert ist, um $\overline{T_i}$ zu erhalten:

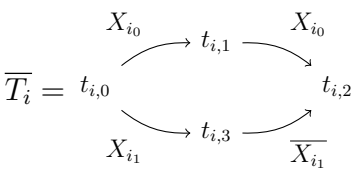
- wenn $X_{i_0} \in S$ und $X_{i_1} \notin S$, dann

$\overline{T_i} =$ 

- wenn $X_{i_0}, X_{i_1} \in S$, dann

$\overline{T_i} =$ 

- wenn $X_{i_0} \notin S$ und $X_{i_1} \in S$, dann

$\overline{T_i} =$ 

Wir zeigen nun, dass B die τ-ZTE hat:

Fakt 9.4. *Es gibt einen Zeugen $\mathcal{R}$ für die τ-ZTE von B.*

Beweis. Die folgende τ-Region $R_1 = (sup_1, sig_1)$ löst (z, z') für alle Zustände $z \neq z' \in Z$ für die gilt $\{z, z'\} \notin \{\{t_{i,1}, t_{i,3}\} \mid i \in \{0, \ldots, m-1\}\}$: $sup_1(t_{0,0}) = 0$ und $\mathcal{E}_{0,1}^{R_1} = E'$.

Seien $i \in \{0, \ldots, m-1\}$ und $a \in \{X_{i_0}, X_{i_1}\} \cap S$ beliebig aber fest. Dies impliziert $\overline{a} \in E'$. Die folgende τ-Region $R_2 = (sup_2, sig_2)$ löst $(t_{i,1}, t_{i,3})$: $sup_2(t_{0,0}) = 0$ und $\mathcal{E}_{0,1}^{R_2} = \{a\}$ und $\mathcal{E}_{1,0}^{R_2} = \{\overline{a}\}$ und $\mathcal{E}_{0,0}^{R_2} = E' \setminus (\{a, \overline{a}\})$. Aus der Beliebigkeit von i folgt, dass $(t_{i,1}, t_{i,3})$ für alle $i \in \{0, \ldots, m-1\}$ τ-lösbar ist. Dies beweist das Lemma. □

Mit Lemma 9.6 und Fakt 9.4 und der Polynomialität der Reduktion erhalten wir das folgende Lemma, dass den Beweis von Satz 9.2 vervollständigt:

Lemma 9.7. *Wenn es eine Knotenüberdeckung mit höchstens λ Elementen für $(\mathfrak{U}, M)$ gibt, dann gibt es eine E'-Neubeschriftung (B, ϱ, φ) von A, die $|E'| \leq \kappa$ erfüllt und die τ-ZTE hat.*

9.3 Zusammenfassung und Ausblick

Sei $\tau \in \{\tau_{PPT}, \tau_{PT}\}$. *Neubeschriftung* ist eine Technik, die darauf abzielt, ein nicht-implementierbares Transitionssystem A durch Neubeschriftung von Kanten so zu modifizieren, dass es durch ein τ-Netz N implementierbar ist. Genauer werden Kanten, die ursprünglich mit dem gleichen Ereignis e beschriftet waren, nun mit verschiedenen Ereignissen $e_1, \ldots, e_n$ neu beschriftet. Wir verstehen die Ereignisse $e_1, \ldots, e_n$ als durch Teilung von e hervorgegangen. Diese Methode kann auf natürliche Weise auf eine Vielzahl von Ereignissen von A fortgesetzt werden. *Neubeschriftung* ist ein mächtiges Werkzeug, das stets ein (aus Neubeschriftung resultierendes) implementierbares Transitionssystem B liefern kann: spätestens wenn alle Kanten paarweise verschieden beschriftet sind, ist das Resultat B implementierbar. Mit Zunahme der Ereignisse von B nimmt jedoch die Größe des implementierenden Netzes N zu und das neue Verhalten entfernt sich zunehmend vom ursprünglichen Verhalten. Deshalb soll die Anzahl der Ereignisse von B „klein“ bleiben. Diese Überlegungen führen auf natürliche Weise zu den Entscheidungsproblemen Neubeschriftung für τ-Einbettung und Neubeschriftung für τ-Sprachsimulation sowie Neubeschriftung für τ-Realisierung: für ein Transitionssystem A und eine Zahl κ wird gefragt, ob es ein Transitionssystem B gibt, dass aus A durch Neubeschriftung hervorgeht und höchstes κ Ereignisse hat sowie durch ein τ-Netz entsprechend implementierbar ist.

In diesem Kapitel wurde gezeigt, dass Neubeschriftung für τ-Sprachsimulation und Neubeschriftung für τ-Realisierung NP-vollständig sind und damit eine offene Frage aus [SW20] gelöst. Der Beweis der NP-Vollständigkeit basiert auf einer Reduktion von CVC, die in einem linearen TS resultiert, sodass diese Probleme auch für derart massiv eingeschränkte Eingaben NP-vollständig sind.

Die NP-Vollständigkeit von Neubeschriftung für τ-Einbettung wurde ursprünglich in [SW20] für allgemeine Transitionssysteme gezeigt; die Technik der Autoren impliziert, dass das Problem auch dann schwer ist, wenn das Eingabe-Transitionssystem den Grad drei hat, das heißt, jeder Zustand ist jeweils der Ursprung beziehungsweise das Ziel von höchstens drei Kanten. In diesem Kapitel zeigen wir ein stärkeres Resultat: Neubeschriftung für τ-Einbettung ist auch für Transitionssysteme mit Grad zwei NP-vollständig. Diese Schranke ist scharf: Wir zeigen, dass Neubeschriftung für τ-Einbettung für Transitionssysteme mit Grad eins polynomiell ist. Dies ist ein interessanter Kontrast gegenüber Neubeschriftung für τ-Sprachsimulation und Neubeschriftung für τ-Realisierung, die sogar für lineare Transitionssysteme NP-vollständig sind. Diese Ergebnisse zeigen außerdem, dass die Entscheidungsprobleme nicht fest-Paramter-berechenbar sind, wenn der Grad des Eingabe-Transitionssystems als Parameter gewählt wird.

In Rahmen zukünftiger Arbeit bleibt zum Beispiel die Komplexität des Problems der Neubeschriftung auch für weitere Netztypen zu untersuchen. Für viele der in dieser Arbeit behandelten Netztypen, deren zu Grunde liegende klassische Syntheseprobleme bereits NP-schwer sind, kann man durch einfache Reduktionen zeigen, dass die zugehörigen Probleme der Neubeschriftung ebenfalls NP-schwer sind. Für die b-beschränkten P/T-Netze reduziere man zum Beispiel die Eingabe A auf das Paar $(A', |E(A)| + 1)$, wobei A' aus A hervorgeht, indem eine bidirektionale Kante $\iota \xleftarrow{a} q$ mit neuem Zustand q und neuem Ereignis a hinzugefügt wird. Gemäß Beispiel 9.1 muss a geteilt werden, sodass eine resultierende E'-Neubeschriftung B einbettbar oder realisierbar ist. Wegen $\kappa = |E(A)| + 1$ kann jedoch kein weiteres Ereignis geteilt werden, sodass die verbleibenden Trennungsprobleme sowohl in A als auch in B sind und deren Lösbarkeit äquivalent ist. Interessant sind für Komplexitätsuntersuchungen deshalb vordergründig die Netztypen, deren klassische Syntheseprobleme polynomiell sind. Bezüglich der booleschen Netztypen trifft dies zum Beispiel für die Netztypen A4 und A10 aus Abbildung 4.3 in Abschnitt 4.1 zu. In [Tre20a] hat der Autor der vorliegenden Arbeit für die Netztypen A4 gezeigt, dass jedes der drei Entscheidungsprobleme der Neubeschriftung, je nach Implementierung, NP-vollständig ist. Für die Netztypen A10 ist deren Komplexität jedoch ungeklärt. Ebenso ungeklärt ist die Komplexität von τ-Neubeschriftung für Einbettung für $\tau \in \{\tau^b_{\mathbb{Z}PPT}, \tau^b_{\mathbb{Z}PT}\}$ mit $b \geq 2$, für die τ-Einbettung polynomiell ist.

Für zukünftige Arbeit wäre es ebenso interessant, die Entscheidungsprobleme Neubeschriftung für τ-Einbettung und Neubeschriftung für τ-Sprachsimulation sowie Neubeschriftung für τ-Realisierung aus Sicht der parametrisierten Komplexität zu untersuchen, wenn κ als (natürlicher) Parameter gewählt wird.

Kapitel 10

Verhaltensmodifikation durch Löschung

Unabhängig von Netztyp τ und gesuchter Implementierung ist nicht jedes Transitionssystem A durch ein τ-Netz implementierbar. Es ist deshalb von praktischer Relevanz Methoden zu entwickeln, die A möglichst geringfügig so modifizieren, dass ein implementierbares Verhalten resultiert. Eine der in der Literatur am häufigsten diskutierte Methoden, nämlich *Neubeschriftung*, haben wir im vorherigen Kapitel 9 kennengelernt. Wie wir gesehen haben, ist diese Methode für die Netztypen τ_{PT} und τ_{PPT} im Allgemeinen nicht effizient: die korrespondierenden Entscheidungsprobleme sind NP-schwer.

Es stellt sich natürlich die Frage, ob es auch andere vernünftige Möglichkeiten gibt, das Eingabe-Verhalten zum Zwecke der Implementierbarkeit zu modifizieren. Tatsächlich lassen sich in naheliegender Weise solche Modifikationen aus unserem Verständnis von Implementierungen ableiten: Die Zustände von A korrespondieren via einer Simulation mit den erreichbaren Markierungen eines implementierenden Netzes. Wenn A nicht implementierbar ist, dann können wir möglicherweise deshalb kein Netz finden, weil nicht jeder Zustand von A mit einer Markierung assoziiert werden kann. Deshalb könnte das *Löschen von* fraglichen *Zuständen* zu einem implementierbaren Verhalten führen. Des Weiteren korrespondiert das Stattfinden von Ereignissen an ihren Ursprüngen mit dem Schalten entsprechender Transitionen in Markierungen, die via der Simulation mit diesen Ursprüngen assoziiert werden. Wenn das Verhalten nicht implementierbar ist, dann könnte dies daran liegen, dass die Transitionen in den Markierungen nicht feuern können. Somit könnte das Entfernen der Ereignisse an den fraglichen Ursprüngen zu einem implementierbaren Verhalten führen. Wenn dies eine Option ist, dann gibt es wenigstens zwei Möglichkeiten, Ereignisse an gewissen Zuständen nicht mehr stattfinden zu lassen, je nachdem ob das Erhalten gewisser Ereignisse oder gewisser Kanten im Vordergrund steht: Legt der Modellierer (des zu Grunde liegenden Systems) den Fokus auf den Erhalt von Ereignissen (des ursprünglichen Verhaltens), dann wird er das *Löschen von* einigen *Kanten* bevorzugen, die möglicherweise mit verschiedenen Ereignisse beschriftet sind, und dabei gleichzeitig fordern, dass für jedes Ereignis wenigstens einige Kanten bleiben, sofern dies möglich ist. Andererseits könnte der Modellierer auch zu dem Schluss kommen, dass einige Ereignisse weniger interessant für das Verhalten des Systems sind, sodass von ihnen

abstrahiert werden kann. In diesem Fall würde er das vollständige *Löschen von Ereignissen* und aller zugehöriger Kanten bevorzugen, während die Kanten der verbleibenden Ereignisse vollständig erhalten bleiben.

Ebenso wie *Neubeschriftung* liefert das *Löschen von Kanten, Ereignissen* beziehungsweise *Zuständen* mächtige Werkzeuge, die stets zu einem implementierbaren Verhalten führen: Spätestens wenn das Eingabeverhalten zu einem einzigen Zustand degeneriert ist, haben wir ein implementierbares Transitionssystem. Sicherlich ist dies weder vernünftig noch wünschenswert. Stattdessen suchen wir nach einer Modifikation A' von A, sodass die Anzahl κ gelöschter Kanten, Ereignisse beziehungsweise Zustände so klein wie möglich ist. Wir erhalten das zu dem angedeuteten Optimierungsproblem gehörige Entscheidungsproblem, indem die Anzahl κ ein Teil der Eingabe wird: entscheide für ein Transitionssystem A und eine Zahl κ, ob A durch das Löschen von höchstens κ Kanten, Ereignissen oder Zuständen –je nachdem– zu einem Transitionssystem A' modifiziert werden kann, dass durch ein τ-Netz implementierbar ist. Es ist leicht zu sehen, dass die Komplexität dieses Entscheidungsproblems eine untere Schranke für die Komplexität der Optimierungsvariante ist. Darüber hinaus sieht man leicht, dass das Entscheidungsproblem NP-schwer ist, wenn das unterliegende Syntheseproblem NP-schwer ist: die triviale Reduktion von A auf (A, κ) mit $\kappa = 0$ beweist dann die NP-Schwere. Interessant sind deshalb insbesondere Netztypen, deren zu Grunde liegende Syntheseprobleme polynomiell sind wie zum Beispiel τ_{PPT} und τ_{PT}.

In diesem Kapitel adressieren wir die durch Kanten-, Ereignis- beziehungsweise Zustandslöschung implizierten Entscheidungsprobleme deshalb für die Netztypen τ_{PPT} beziehungsweise τ_{PT}, je nachdem, jeweils für alle Implementierungen, das heißt Einbettung, Sprachsimulation und Realisierung. Dieses Kapitel basiert auf der Publikation [Tre21b].

10.1 Kanten-, Ereignis- und Zustandslöschung: Definitionen und Abgrenzung

Solange nicht ausdrücklich anders vereinbart, sei $\tau \in \{\tau_{PPT}, \tau_{PT}\}$ in diesem Abschnitt beliebig aber fest. In diesem Kapitel adressieren wir die wie folgt definierten Methoden *Kanten-* und *Ereignis-* sowie *Zustandslöschung*, um Transitionssysteme zu modifizieren:

Definition 10.1 (Kantenlöschung)**.** Sei $A = (Z, E, \delta, \iota)$ ein Transitionssystem. Ein Transitionssystem $B = (Z', E', \delta', \iota)$ mit Zustandsmenge $Z' \subseteq Z$ und Ereignismenge $E' \subseteq E$ ist eine *Kantenlöschung* von A, wenn für alle Zustände $z, z' \in Z'$ und alle Ereignisse $e \in E'$ gilt: wenn $\delta'(z, e) = z'$, dann $\delta(z, e) = z'$. Wir definieren durch $\mathfrak{K} = \Delta_A \setminus \Delta_B$ *die Menge der* (aus A) *gelöschten Kanten.*

Definition 10.2 (Ereignislöschung)**.** Sei $A = (Z, E, \delta, \iota)$ ein Transitionssystem. Ein Transitionssystem $B = (Z', E', \delta', \iota)$ mit Zustandsmenge $Z' \subseteq Z$ und Ereignismenge $E' \subseteq E$ ist eine *Ereignislöschung* von A, wenn für alle Ereignisse $e \in E'$ gilt: $z \xrightarrow{e} z' \in B$ genau dann,

wenn $z \xrightarrow{e} z' \in A$. Wir definieren durch $\mathfrak{E} = E \setminus E'$ die *Menge der* (aus A) *gelöschten Ereignisse*.

Definition 10.3 (Zustandslöschung)**.** Sei $A = (Z, E, \delta, \iota)$ ein Transitionssystem. Ein Transitionssystem $B = (Z', E', \delta', \iota)$ mit Zustandsmenge $Z' \subseteq Z$ und Ereignismenge $E' \subseteq E$ ist eine *Zustandslöschung* von A, wenn die folgenden zwei Bedingungen gelten: (1) für alle $z, z' \in Z'$ und alle $e \in E'$ gilt genau dann $\delta'(z, e) = z'$, wenn $\delta(z, e) = z'$; (2) wenn es eine Kante $q \xrightarrow{e} q'$ in A gibt, die nicht in B ist, dann $q \notin Z'$ oder $q' \notin Z'$. Wir definieren durch $\mathfrak{Z} = Z \setminus Z'$ *die Menge der* (aus A) *gelöschten Zustände*.

Bemerkung 10.1. Gemäß unserer intuitiven Vorstellung von den diskutierten Methoden, sprechen wir von Kanten- und Ereignis- sowie Zustandslöschung häufig von Techniken, die ein Transitionssystem A modifizieren. Dies legt zunächst den Gedanken nahe, diese Modifikationen im Sinne eines Operators O zu definieren, dessen Anwendung $O(A)$ auf A ein entsprechendes Resultat B liefert. Stattdessen haben wir diese Modifikationen jedoch als relational, das heißt als Beziehungen zwischen A und (einem Transitionssystem) B, beschrieben. Der Grund dafür ist, dass es im Allgemeinen verschiedene Möglichkeiten geben kann, A unter Berücksichtigung von κ zu modifizieren, sodass sich eine funktionale Definition einfach nicht anbietet.

Folgende Zusammenhänge bestehen zwischen den eingeführten Modifikationen eines Transitionssystems A: Kantenlöschung ist die allgemeinste Modifikation, denn jede Ereignislöschung und jede Zustandslöschung ist eine Kantenlöschung. Jedoch ist nicht jede Kantenlöschung eine Ereignislöschung oder Zustandslöschung. Ebenso ist nicht jede Ereignislöschung eine Zustandslöschung und nicht jede Zustandslöschung ist eine Ereignislöschung. Insbesondere gibt es, ihrer Namensgebung entsprechend, substantielle Unterschiede zwischen diesen Methoden, deren Schwerpunkt auf unterschiedlichen Bestandteilen des Transitionssystems liegt: Wenn B zum Beispiel eine Kantenlöschung von A ist, dann kann es potentiell ein Ereignis $e \in E'$ geben, sodass es eine Kante $z \xrightarrow{e} z'$ in A gibt, die nicht in B ist. Wenn B jedoch eine Ereignislöschung ist, dann ist im Gegensatz dazu für jedes $e \in E'$ jede e-beschriftete Kante von A auch in B. Schließlich kann eine Kante $z \xrightarrow{e} z'$ von A nur dann nicht in einer Zustandslöschung B von A sein, wenn wenigstens einer der Zustände z oder z' nicht in Z' ist. Diese Bedingung muss für eine gelöschte Kante jedoch nicht erfüllt sein, wenn B eine Kanten- oder Ereignislöschung ist. Wir diskutieren diese Feinheiten im Folgenden an einigen Beispielen.

Das Transitionssystem B aus Abbildung 10.2 ist eine Zustandslöschung des Transitionssystems A aus Abbildung 10.1. Dabei entsteht B durch das Löschen des Zustands z_3, das heißt $\mathfrak{Z} = \{z_3\}$. B ist auch eine Kantenlöschung, wobei $\mathfrak{K} = \{z_2 \xrightarrow{x} z_3\}$. Jedoch ist das Transitionssystem B keine Ereignislöschung von A, denn x ist ein Ereignis in B, aber nicht alle x-beschrifteten Kanten von A sind auch in B.

Das Transitionssystem C aus Abbildung 10.3 ist eine Ereignislöschung von A, wobei $\mathfrak{E} = \{a\}$. C ist auch eine Kantenlöschung, wobei $\mathfrak{K} = \{t_0 \xrightarrow{a} t_1, q_0 \xrightarrow{a} q_1\}$, aber keine Zustandslöschung: Offensichtlich sind alle Zustände von A auch in C.

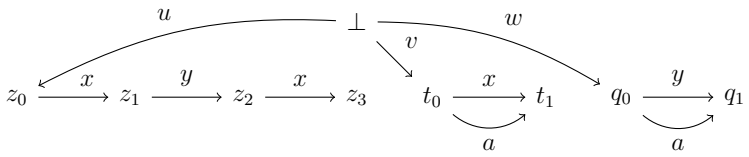


Abbildung 10.1: Transitionssystem A.

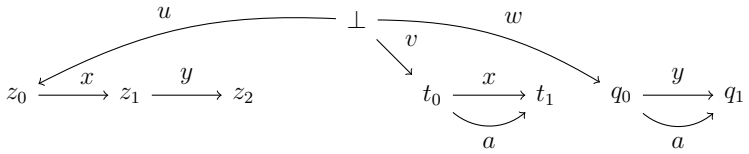


Abbildung 10.2: Zustandslöschung B des Transitionssystems A aus Abbildung 10.1, die durch das Löschen des Zustands z_3 entsteht.

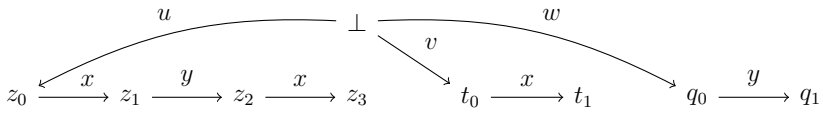


Abbildung 10.3: Ereignislöschung C von A, die durch das Löschen des Ereignisses a entsteht.

Das Transitionssystem A aus Abbildung 10.1 hat nicht die τ-EZTE, denn das EZTP $\alpha = (x, z_1)$ ist nicht τ-lösbar. Dies sieht man so: Angenommen, $R = (sup, sig)$ ist eine τ-Region, die α löst, das heißt $sig^-(x) > sup(z_1)$. Weil x an z_0 stattfindet, gilt $sig^-(x) \leq sup(z_0)$. Mit $sup(z_1) = sup(z_0) - sig^-(x) + sig^+(x)$ und $sig^-(x) > sup(z_1)$ impliziert dies $sig^-(x) > sig^+(x)$. Wegen der Kante $t_0 \xrightarrow{x} t_1$ impliziert dies auch $sup(t_0) > sup(t_1)$ und somit $sig^-(a) > sig^+(a)$, wegen $t_0 \xrightarrow{a} t_1$. Andererseits findet x an z_2 statt, sodass wir $sig^-(x) \leq sup(z_2)$ erhalten. Mit $sig^-(x) > sup(z_1)$ und $z_1 \xrightarrow{y} z_2$ ist dies nur möglich, wenn $sig^-(y) < sig^+(y)$. Dies impliziert jedoch $sup(q_0) < sup(q_1)$ und somit den Widerspruch $sig^-(a) < sig^+(a)$.

Für die Zustandslöschung B von A aus Abbildung 10.2 gibt es eine τ_{PT}-Region $R = (sup, sig)$ die α löst: $sup(\bot) = 2$ and $\mathcal{E}^R_{2,1} = \{x\}$ and $\mathcal{E}^R_{1,0} = \{a, y\}$ and $\mathcal{E}^R_{0,0} = E \setminus \{a, y, x\}$. Man überlegt sich, dass alle weiteren EZTP und alle ZTP von B τ_{PT}-lösbar sind: B ist durch ein τ_{PT}-Netz realisierbar. Wenn der Modellierer entscheidet, dass die Ereignisse x und y sowie alle zugehörigen Kanten für die Modellierung des Systemverhaltens essentiell sind, so führt auch (allein) das Entfernen des Ereignisses a zu einem durch (pure) P/T-Netze realisierbaren Verhalten, nämlich dem Transitionssystem C aus Abbildung 10.3:

Eine τ-Region $R' = (sup', sig')$, die zum Beispiel (x, z_1) in C löst, ist wie folgt definiert: $sup(\bot) = 1$ und $\mathcal{E}^{R'}_{1,0} = \{x\}$ und $\mathcal{E}^{R'}_{0,1} = \{y\}$ und $\mathcal{E}^{R'}_{0,0} = E \setminus \{x, y\}$.

10.2 Kantenlöschung für Sprachsimulation und Realisierung

Solange nicht ausdrücklich anders vereinbart, sei im Folgenden $\tau = \tau_{PT}$. In diesem Abschnitt werden wir die Komplexität der Aufgabe untersuchen, ein –bezüglich τ– nicht-implementierbares Transitionssystem A durch das Löschen einer möglichst geringen Anzahl von Kanten so zu modifizieren, dass ein implementierbares Verhalten resultiert. Diese Aufgabe gibt Anlass zu den folgenden Entscheidungsproblemen:

Kantenlöschung für τ-Einbettung	
Eingabe:	Ein endliches Transitionssystem $A = (Z, E, \delta, \iota)$ und $\kappa \in \mathbb{N}$.
Frage:	Gibt es eine Kantenlöschung B von A, sodass $\lvert\mathfrak{K}\rvert \leq \kappa$ erfüllt ist und B die τ-ZTE hat?

Kantenlöschung für τ-Sprachsimulation	
Eingabe:	Ein endliches Transitionssystem $A = (Z, E, \delta, \iota)$ und $\kappa \in \mathbb{N}$.
Frage:	Gibt es eine Kantenlöschung B von A, sodass $\lvert\mathfrak{K}\rvert \leq \kappa$ erfüllt ist und B die τ-EZTE hat?

Kantenlöschung für τ-Realisierung	
Eingabe:	Ein endliches Transitionssystem $A = (Z, E, \delta, \iota)$ und $\kappa \in \mathbb{N}$.
Frage:	Gibt es eine Kantenlöschung B von A, sodass $\lvert\mathfrak{K}\rvert \leq \kappa$ erfüllt ist und B die τ-ZTE und die τ-EZTE hat?

Die Arbeit von Schlachter und Wimmel [SW20] impliziert die folgende Charakterisierung der Komplexität von Kantenlöschung für τ-Einbettung:

Satz 10.1 ([SW20])**.** Kantenlöschung für τ-Einbettung *ist* NP*-vollständig.*

Die Autoren [SW20] bemerken explizit, dass dieses Ergebnis und die verwendete Technik, jedoch keine Rückschlüsse auf die Komplexität von Kantenlöschung für τ-Sprachsimulation oder Kantenlöschung für τ-Realisierung erlaubt. In diesem Kapitel schließen wir diese Lücke und zeigen, dass beide Probleme NP-vollständig sind. Wir beginnen mit der Zugehörigkeit zur Komplexitätsklasse NP:

Lemma 10.1. Kantenlöschung für τ-Sprachsimulation *und* Kantenlöschung für τ-Realisierung *sind in* NP*.*

Beweis. Sicherlich kann durch nicht-deterministisches Löschen von höchstens κ Kanten von A in Polynomialzeit ein Transitionssystem $B = (Z', E', \delta', \iota)$ bestimmt werden, das die

gefragte Eigenschaft hat, falls es ein solches gibt. Des Weiteren kann in deterministischer Polynomialzeit überprüft werden ob B tatsächlich eine Kantenlöschung von A ist, denn die Größe von B ist durch die Größe von A beschränkt und die klassischen Syntheseprobleme sind polynomiell für τ_{PT} [BBD15]. Schließlich hat B höchstens so, viele ZTP und EZTP wie A, sodass auch die τ-ZTE beziehungsweise τ-EZTE von B mit polynomiellem Zeitaufwand verifiziert werden kann. □

Der folgende Satz ist das Hauptresultat dieses Abschnitts:

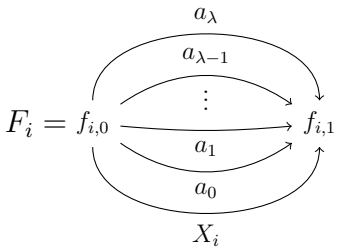
Satz 10.2. Kantenlöschung für τ-Sprachsimulation *und* Kantenlöschung für τ-Realisierung *sind* NP*-vollständig.*

Der Rest dieses Kapitels ist dem Beweis von Satz 10.2 gewidmet: Gemäß Lemma 10.1 sind beide Probleme in NP und es bleibt, die NP-Schwere zu zeigen. Dazu präsentieren wir eine geeignete Reduktion des Problems Cubic Vertex Cover (CVC), das in Kapitel 9 definiert wurde. Solange nicht ausdrücklich anders vereinbart, sei $(\mathfrak{U}, M, \lambda)$ im Folgenden eine beliebige aber feste Eingabe für CVC, sodass $\mathfrak{U} = \{X_0, \ldots, X_{n-1}\}$ und $M = \{M_0, \ldots, M_{m-1}\}$ sowie $M_i = \{X_{i_0}, X_{i_1}\}$ mit $i_0 < i_1$ für alle $i \in \{0, \ldots, m-1\}$. Wir reduzieren $(\mathfrak{U}, M, \lambda)$ auf ein Paar (A, κ) mit Transitionssystem A und natürlicher Zahl κ, sodass A genau dann eine Kantenlöschung hat, welche κ berücksichtigt und die τ-EZTE beziehungsweise die τ-EZTE und die τ-ZTE hat, wenn es eine Knotenüberdeckung mit höchstens λ Elementen für $(\mathfrak{U}, M)$ gibt. Wir beginnen mit der Konstruktion von (A, κ). Anschließend beweisen wir die angekündigte Funktionalität der Reduktion.

Zunächst definieren wir $\kappa = \lambda$. Für jedes $i \in \{0, \ldots, m-1\}$ hat das Transitionssystem A das folgende Gadget T_i, welches die Elemente X_{i_0} und X_{i_1} von $M_i = \{X_{i_0}, X_{i_1}\}$ als Ereignisse verwendet:

$$T_i = \quad t_{i,0} \xrightarrow{X_{i_0}} t_{i,1} \xrightarrow{X_{i_1}} t_{i,2} \xrightarrow{X_{i_0}} t_{i,3}$$

Man beachte, dass die Konstruktion von T_i wegen $i_0 < i_1$ für alle $i \in \{0, \ldots, m-1\}$ eindeutig ist. Darüber hinaus hat A für jedes $i \in \{0, \ldots, n-1\}$ das folgende Gadget F_i, welches das Element $X_i \in \mathfrak{U}$ als Ereignis verwendet:



Man beachte, dass jedes der Gadgets $F_0, \ldots, F_{n-1}$ für jedes $j \in \{0, \ldots, \lambda\}$ jeweils eine a_j-beschriftete Kante enthält, deren Richtung der X_i-beschrifteten Kante gleich ist.

Des Weiteren hat das Transitionssystem A den Initialzustand $\perp$, über den die eingeführten Gadgets mittels zusätzlicher Kanten miteinander verbunden werden. Genauer hat das Transitionssystem A für jedes $i \in \{0, \ldots, m-1\}$ die Kante $\perp \xrightarrow{w_i} t_{i,0}$ und für jedes $j \in \{0, \ldots, n-1\}$ die Kante $\perp \xrightarrow{w_j} f_{j,0}$. Das resultierende Transitionssystem ist A.

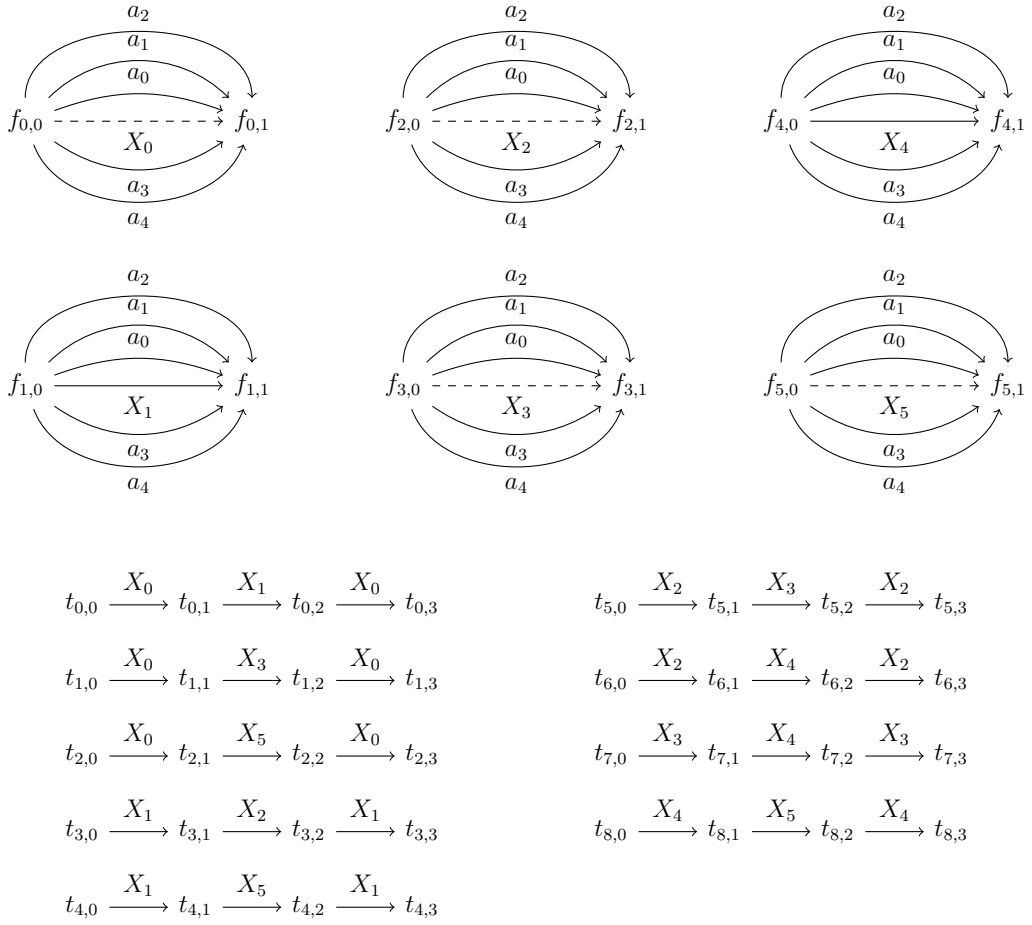
Abbildung 10.4 zeigt die Gadgets des Transitionssystems A, das gemäß der Reduktion aus der Eingabe des folgenden Beispiels 10.1 für CVC resultiert:

Beispiel 10.1. Die Eingabe $(\mathfrak{U}, M, 4)$ mit $\mathfrak{U} = \{X_0, \ldots, X_5\}$ und $M = \{M_0, \ldots, M_8\}$, wobei $M_0 = \{X_0, X_1\}$, $M_1 = \{X_0, X_3\}$, $M_2 = \{X_0, X_5\}$, $M_3 = \{X_1, X_2\}$, $M_4 = \{X_1, X_5\}$, $M_5 = \{X_2, X_3\}$, $M_6 = \{X_2, X_4\}$, $M_7 = \{X_3, X_4\}$ und $M_8 = \{X_4, X_5\}$, erlaubt eine positive Entscheidung, denn $S = \{X_0, X_2, X_3, X_5\}$ ist eine passende Kontenüberdeckung für $(\mathfrak{U}, M)$.

Für die eine Richtung zeigen wir nun, dass es eine gesuchte Knotenüberdeckung für $(\mathfrak{U}, M)$ gibt, wenn es eine gesuchte Kantenlöschung für A gibt, welche die τ-EZTE hat:

Lemma 10.2. *Wenn es eine Kantenlöschung $B = (Z', E', \delta', \iota)$ von A gibt, die $|\mathfrak{K}| \leq \kappa$ erfüllt und die τ-EZTE hat, dann gibt es eine Knotenüberdeckung mit höchstens λ Elementen für $(\mathfrak{U}, M)$.*

Beweis. Sei B eine entsprechende Kantenlöschung von A. Es ist unser Ziel, zu zeigen, dass die Menge $S = \{X \in \mathfrak{U} \mid (z, X, z') \in \mathfrak{K}\}$ der Ereignisse aus $\mathfrak{U}$, die eine gelöschte Kante beschriften, eine gesuchte Knotenüberdeckung definiert. Sei sei $i \in \{0, \ldots, m-1\}$ beliebig aber fest. Es gibt genau zwei Möglichkeiten: Entweder das Gadget T_i und alle seine Kanten sind in B vorhanden oder es gibt eine Kante von T_i, die nicht in B ist, das heißt, $\{(t_{i,0}, X_{i_0} t_{i,1}), (t_{i,1}, X_{i_1} t_{i,2}), (t_{i,2}, X_{i_0} t_{i,3})\} \cap \mathfrak{K} \neq \emptyset$. Im zweiten Fall gilt offenbar bereits $M_i \cap S \neq \emptyset$. Wir werden nun zeigen, dass dies auch im ersten Fall zutrifft: Weil T_i vollständig in B vorhanden ist, gibt es das EZTP $\alpha = (X_{i_0}, t_{i,1})$ in B. Weil B die τ-EZTE hat, ist α τ-lösbar. Wir wollen nun zeigen, dass dies impliziert, dass B wenigstens eine der Kanten $f_{i_0,0} \xrightarrow{X_{i_0}} f_{i_0,1}$ beziehungsweise $f_{i_1,0} \xrightarrow{X_{i_1}} f_{i_1,1}$ nicht enthält, das heißt, wenigstens eine dieser Kanten wurde gelöscht. Dazu zeigen wir wie folgt, dass die Annahme, beide der genannten Kanten seien in B, zum Widerspruch führt: Wir gehen von einer τ-Region $R = (sup, sig)$ aus, die α löst. Erst wird gezeigt, dass sowohl $sig^-(X_{i_0}) > sig^+(X_{i_0})$ als auch $sig^-(X_{i_1}) < sig^+(X_{i_1})$ gelten. Daraus folgen unverzüglich $sup(f_{i_0,0}) > sup(f_{i_0,1})$ und $sup(f_{i_1,0}) < sup(f_{i_1,1})$. Für jedes $j \in \{0, \ldots, \lambda\}$ gibt es eine a_j-beschriftete Kante $f_{i_0,0} \xrightarrow{a_j} f_{i_0,1}$ und $f_{i_1,0} \xrightarrow{a_j} f_{i_1,1}$ in A. Weil B durch die Löschung von höchstens λ Kanten von A entsteht, gibt es also ein $j \in \{0, \ldots, \lambda\}$, sodass $f_{i_0,0} \xrightarrow{a_j} f_{i_0,1}$ und $f_{i_1,0} \xrightarrow{a_j} f_{i_1,1}$ in B sind. (Durch das Löschen von λ Kanten, können höchstens λ verschiedene Indizes aus $\{0, \ldots, \lambda\}$ betroffen sein, das heißt, es muss einen Index j in $\{0, \ldots, \lambda\}$ geben, für den es keine gelöschte Kante gibt, sodass a_j in allen $F_0, \ldots, F_{n-1}$ stattfindet.) Mit $sup(f_{i_0,0}) > sup(f_{i_0,1})$ erhalten wir $sig^-(a_j) > sig^+(a_j)$ und mit $sup(f_{i_1,0}) < sup(f_{i_1,1})$ erhalten wir $sig^-(a_j) < sig^+(a_j)$ und somit einen Widerspruch. Folglich können nicht beide Kanten $f_{i_0,0} \xrightarrow{X_{i_0}} f_{i_0,1}$ und $f_{i_1,0} \xrightarrow{X_{i_1}} f_{i_1,1}$ in B sein. Es bleiben $sig^-(X_{i_0}) > sig^+(X_{i_0})$



$t_{0,0} \xrightarrow{X_0} t_{0,1} \xrightarrow{X_1} t_{0,2} \xrightarrow{X_0} t_{0,3}$

$t_{1,0} \xrightarrow{X_0} t_{1,1} \xrightarrow{X_3} t_{1,2} \xrightarrow{X_0} t_{1,3}$

$t_{2,0} \xrightarrow{X_0} t_{2,1} \xrightarrow{X_5} t_{2,2} \xrightarrow{X_0} t_{2,3}$

$t_{3,0} \xrightarrow{X_1} t_{3,1} \xrightarrow{X_2} t_{3,2} \xrightarrow{X_1} t_{3,3}$

$t_{4,0} \xrightarrow{X_1} t_{4,1} \xrightarrow{X_5} t_{4,2} \xrightarrow{X_1} t_{4,3}$

$t_{5,0} \xrightarrow{X_2} t_{5,1} \xrightarrow{X_3} t_{5,2} \xrightarrow{X_2} t_{5,3}$

$t_{6,0} \xrightarrow{X_2} t_{6,1} \xrightarrow{X_4} t_{6,2} \xrightarrow{X_2} t_{6,3}$

$t_{7,0} \xrightarrow{X_3} t_{7,1} \xrightarrow{X_4} t_{7,2} \xrightarrow{X_3} t_{7,3}$

$t_{8,0} \xrightarrow{X_4} t_{8,1} \xrightarrow{X_5} t_{8,2} \xrightarrow{X_4} t_{8,3}$

Abbildung 10.4: Gadgets für A resultierend aus $(G, 4)$ aus Beispiel 10.1. Oben: $F_0, \ldots, F_5$. Die gestrichtelten Kanten entsprechen denen, die gemäß der Konstruktion für Lemma 10.3 basierend auf dem 4-VC $\{X_0, X_2, X_3, X_5\}$ von $(\mathfrak{U}, M)$ für die Kantenlöschung B gelöscht werden. Unten: Die Gadgets $T_0, \ldots, T_8$.

und $sig^-(X_{i_1}) < sig^+(X_{i_0})$ zu zeigen: Weil R das EZTP α löst, gilt $sup(t_{i,1}) < sig^-(X_{i_0})$. Außerdem gilt $sig^-(X_{i_0}) \leq sup(t_{i,0})$, weil X_{i_0} an $t_{i,0}$ stattfindet, Mit dem Vorherigen erhalten wir $sup(t_{i,0}) > sup(t_{i,1})$ und folglich $sig^-(X_{i_0}) > sig^+(X_{i_0})$, wegen der Kante $t_{i,0} \xrightarrow{X_{i_0}} t_{i,1}$. Nun findet X_{i_0} aber auch an $t_{i,2}$ statt, sodass $sig^-(X_{i_0}) \leq sup(t_{i,2})$ gilt. Mit $sup(t_{i,1}) < sig^-(X_{i_0})$ impliziert dies $sup(t_{i,1}) < sup(t_{i,2})$ und somit $sig^-(X_{i_1}) < sig^+(X_{i_1})$.

Wir fassen zusammen: Weil i beliebig war, gilt für alle $i \in \{0, \ldots, m-1\}$, dass wenigstens eine der der Kanten $f_{i_0,0} \xrightarrow{X_{i_0}} f_{i_0,1}$ oder $f_{i_1,0} \xrightarrow{X_{i_1}} f_{i_1,1}$ nicht in B sind. Damit erfüllt die Menge $S = \{X \in \mathfrak{U} \mid (z, X, z') \in \mathfrak{K}\}$ einerseits $S \cap M_i \neq \emptyset$ für alle $i \in \{0, \ldots, m-1\}$ und andererseits gilt offenbar $|S| \leq |\mathfrak{K}| \leq \lambda$. Somit ist S eine Knotenüberdeckung mit höchstens λ Elementen für $(\mathfrak{U}, M)$. Dies beweist das Lemma. □

Wir zeigen für die Rückrichtung, dass (A, κ) eine positive Entscheidung erlaubt, wenn es eine gesuchte Knotenüberdeckung für $(\mathfrak{U}, M)$ gibt. Sei also $S = \{X_{i_0}, \ldots, X_{i_{\lambda-1}}\} \subseteq \mathfrak{U}$ eine Knotenüberdeckung mit λ Elementen für $(\mathfrak{U}, M)$. Wir erhalten eine passende Kantenlöschung $B = (Z(A), E(A), \delta', \bot)$, indem wir für jedes $\ell \in \{0, \ldots, \lambda - 1\}$ die Kante $f_{i_\ell,0} \xrightarrow{X_{i_\ell}} f_{i_\ell,1}$ aus A entfernen (und keine weitere Kante), das heißt, $\mathfrak{K} = \{f_{i_\ell,0} \xrightarrow{X_{i_\ell}} f_{i_\ell,1} \mid \ell \in \{0, \ldots, \lambda - 1\}\}$. Es ist klar, dass $|\mathfrak{K}| = |S| \leq \lambda$ erfüllt ist.

Wir zeigen nun, dass B die τ-ZTE und die τ-EZTE hat. Lösende Regionen $R = (sup, sig)$ präsentieren wir implizit über $sup(\bot)$ und sig. Wie gehabt fassen wir Ereignisse gleicher Signatur (m, n) in einer Menge $\mathcal{E}^R_{m,n} = \{e \in E(B) \mid sig(e) = (m, n)\}$ zusammen. Dabei gilt $e \in E^R_{0,0}$ für alle Ereignisse $e \in E(B)$, die nicht in einer Menge $\mathcal{E}^R_{m,n}$ mit $(m, n) \neq (0, 0)$ aufgelistet sind. Der Einfachheit halber werden wir deshalb die Menge $E^R_{0,0}$ jeweils nicht explizit definieren.

Der folgende Fakt adressiert die τ-ZTE von B:

Fakt 10.1. *Die Kantenlöschung B von A hat die τ-ZTE.*

Beweis. Sei $i \in \{0, \ldots, m - 1\}$ beliebig aber fest. Die folgende Region $R_1 = (sup_1, sig_1)$ löst (z, z') für alle $z \in Z(T_i)$ alle $z' \in Z(B) \setminus Z(T_i)$: $sup_1(\bot) = 0$ und $\mathcal{E}^{R_1}_{0,1} = \{w_i\}$. Analog definiert man für alle $j \in \{0, \ldots, n-1\}$ eine Region, die (z, z') für alle $z \in Z(F_j)$ und alle $z' \in Z(B) \setminus Z(F_j)$ löst.

Um den Beweis des Fakts abzuschließen, bleibt zu zeigen, dass jedes ZTP (z, z') auch dann τ-lösbar ist, wenn z und z' Zustände desselben Gadgets sind. Die folgende Region R_2 leistet das Gewünschte: $sup_2(\bot) = 0$ und $\mathcal{E}^{R_2}_{0,1} = E(B)$. □

Es bleibt zu zeigen, dass B die τ-EZTE hat:

Fakt 10.2. *Die Kantenlöschung B von A hat die τ-EZTE*

Beweis. Seien $W = \{w_0, \ldots, w_{m-1}\}$ und $Y = \{y_0, \ldots, y_{n-1}\}$.

Die folgende Region $R_1 = (sup_1, sig_1)$ löst (e, z) für alle $e \in W \cup Y$ und alle $z \in Z(B) \setminus \{\bot\}$: $sup_1(\bot) = 1$ und $\mathcal{E}^{R_1}_{1,0} = W \cup Y$.

Die folgende Region $R_2 = (sup_2, sig_2)$ löst $(e, \bot)$ für alle $e \in E \setminus (W \cup Y)$: $sup_2(\bot) = 0$ und $\mathcal{E}^{R_2}_{0,3} = W \cup Y$ und $\mathcal{E}^{R_2}_{1,0} = E \setminus (W \cup Y)$.

Wegen der Regionen R_1 und R_2 bleiben nur noch jene EZTP (a, q) von B zu betrachten, für die a und q jeweils aus einem der Gadgets sind. Wenn a und q jeweils aus verschiedenen Gadgets sind, dann definiert man (für a und q beliebig aber fest) eine Region $R_3 = (sup_3, sig_3)$, sodass $sup_3(\bot) = 0$ und für alle $e \in E(B)$: wenn $e = a$, dann $sig_3(e) = (1, 1)$; wenn $e \in W \cup Y$, sodass es eine e-beschriftete Kante in B zu einem Gadget von B gibt, dass in dem a stattfindet, dann $sig_3(e) = (0, 1)$; andernfalls $sig_3(e) = (0, 0)$. Um den Beweis des Fakts abzuschließen, müssen wir folglich nur noch jene EZTP (a, q) von B betrachten, für die a und q in demselben Gadget stattfinden.

Sei $i \in \{0, \ldots, m - 1\}$ beliebig aber fest. Wir zeigen nun, dass alle EZTP von B, die aus T_i herrühren, τ-lösbar sind. Dazu beginnen wir mit $(X_{i_0}, t_{i,1})$ und $(X_{i_0}, t_{i,3})$, wobei wir zwischen den Fällen $X_{i_0} \in S$ und $X_{i_0} \notin S$ unterscheiden:

1. $X_{i_0} \in S$. Die folgende Region $R_4 = (sup_4, sig_4)$ löst sowohl $(X_{i_0}, t_{i,1})$ als auch $(X_{i_0}, t_{i,3})$: $sup_4(\bot) = 0$ und $\mathcal{E}_{1,0}^{R_4} = \{X_{i_0}\}$ und $\mathcal{E}_{0,1}^{R_4} = \{X_{i_1}\} \cup \{w_i\} \cup (\mathfrak{U} \setminus S) \cup \{a_0, \ldots, a_\lambda\}$ und $\mathcal{E}_{0,3}^{R_4} = E(B) \setminus (\mathcal{E}_{1,0}^{R_4} \cup \mathcal{E}_{0,1}^{R_4})$.

2. $X_{i_0} \notin S$ (und folglich $X_{i_1} \in S$). Die folgende Region $R_5 = (sup_5, sig_5)$ löst $(X_{i_0}, t_{i,1})$ und $(X_{i_0}, t_{i,3})$: $sup_5(\bot) = 0$ und $\mathcal{E}_{1,0}^{R_5} = (\mathfrak{U} \setminus S) \cup \{a_0, \ldots, a_\lambda\}$ and $\mathcal{E}_{0,1}^{R_5} = \{X_{i_1}\} \cup \{w_i\}$ und $\mathcal{E}_{0,3}^{R_5} = E(B) \setminus (\mathcal{E}_{1,0}^{R_5} \cup \mathcal{E}_{0,1}^{R_5})$.

Man beachte, dass für beide Regionen $2sig^-(X_{j_0}) + sig^-(X_{j_1}) \leq 3$ für alle $j \in \{0, \ldots, m-1\}$ gilt, sodass diese tatsächlich wohldefiniert sind. Soweit sind wir mit X_{i_0} in T_i bereits fertig.

Wir fahren mit dem EZTP $(X_{i_1}, t_{i,0})$ fort und unterscheiden $X_{i_1} \in S$ und $X_{i_1} \notin S$:

1. $X_{i_1} \in S$. Die folgende Region $R_6 = (sup_6, sig_6)$ löst $(X_{i_1}, t_{i,0})$: $sup_6(\bot) = 1$ und $\mathcal{E}_{1,0}^{R_6} = \{X_{i_1}, w_i\}$ und $\mathcal{E}_{0,1}^{R_6} = \{X_{i_0}\} \cup (\mathfrak{U} \setminus S) \cup \{a_0, \ldots, a_\lambda\}$ und $\mathcal{E}_{0,3}^{R_6} = E \setminus (\mathcal{E}_{1,0}^{R_6} \cup \mathcal{E}_{0,1}^{R_6})$.

2. $X_{i_1} \notin S$ (und folglich $X_{i_0} \in S$). Die folgende Region $R_7 = (sup_7, sig_7)$ löst $(X_{i_1}, t_{i,0})$: $sup_7(\bot) = 1$ und $\mathcal{E}_{1,0}^{R_7} = \{w_i\} \cup (\mathfrak{U} \setminus S) \cup \{a_0, \ldots, a_\lambda\}$ und $\mathcal{E}_{0,1}^{R_7} = \{X_{i_0}\}$ und $\mathcal{E}_{0,3}^{R_7} = E(B) \setminus (\mathcal{E}_{1,0}^{R_7} \cup \mathcal{E}_{0,1}^{R_7})$.

Bezüglich T_i bleibt nur noch, die τ-Lösbarkeit von $(X_{i_1}, t_{i,2})$ und $(X_{i_1}, t_{i,3})$ zu zeigen. Dazu unterscheiden wir wieder:

1. $X_{i_1} \in S$. Die folgende Region $R_8 = (sup_8, sig_8)$ löst $(X_{i_1}, t_{i,2})$ und $(X_{i_1}, t_{i,2})$: $sup_8(\bot) = 1$ und $\mathcal{E}_{1,0}^{R_8} = \{X_{i_1}\}$ und $\mathcal{E}_{0,3}^{R_8} = (Y \cup W) \setminus \{w_i\}$.

2. $X_{i_1} \notin S$ (und folglich $X_{i_0} \in S$). Die folgende Region $R_9 = (sup_9, sig_9)$ löst $(X_{i_1}, t_{i,2})$ und $(X_{i_1}, t_{i,2})$: $sup_9(\bot) = 1$ und $\mathcal{E}_{1,0}^{R_9} = (\mathfrak{U} \setminus S) \cup \{a_0, \ldots, a_\lambda\}$ und $\mathcal{E}_{0,3}^{R_9} = (Y \cup W) \setminus \{w_i\}$.

Insgesamt haben wir gezeigt, dass alle EZTP von T_i τ-lösbar sind. Wenden wir uns nun den verbleibenden Gadgets zu, das heißt $\overline{F_0}, \ldots, \overline{F_{n-1}}$. Sei $i \in \{0, \ldots, n-1\}$ beliebig aber fest.

Die folgende Region $R_{10} = (sup_{10}, sig_{10})$ löst $(e, f_{i,1})$ für alle Ereignisse $e \in E(F_i)$: $sup_{10}(\bot) = 2$ und $\mathcal{E}_{1,0}^{R_{10}} = \{y_i\} \cup (\mathfrak{U} \setminus Z) \cup \{a_0, \ldots, a_\lambda\}$ und $\mathcal{E}_{0,3}^{R_{10}} = E(B) \setminus \mathcal{E}_{1,0}^{R_{10}}$.

Aus der Beliebigkeit von i folgt, dass alle EZTP (e, z) von B lösbar sind, die aus den Gadgets $F_0, \ldots, F_{n-1}$ herrühren. Insgesamt ist der Fakt bewiesen. □

Weil B eine Kantenlöschung von A ist, die $|\mathfrak{E}| \leq \kappa$ erfüllt, erhalten wir mit den Fakten 10.1 und 10.2 das folgende Lemma, das den Beweis von Satz 10.2 vervollständigt:

Lemma 10.3. *Wenn es eine Knotenüberdeckung mit höchstens λ Elementen für $(\mathfrak{U}, M)$ gibt, dann gibt es eine Kantenlöschung B von A, die $|\mathfrak{K}| \leq \kappa$ erfüllt und die τ-EZTE und die τ-ZTE hat.*

In diesem Abschnitt haben wir uns auf den Netztyp τ_{PT} beschränkt. Die folgende Bemerkung erklärt, warum die Konstruktion für τ_{PPT} tatsächlich nicht verwendbar ist:

Bemerkung 10.2. Die Konstruktion eignet sich nicht für $\tau = \tau_{PPT}$: Wenn es $i \neq j \in \{0, \ldots, m-1\}$ gibt, sodass $X_{i_0} \neq X_{j_0}$ und beide $X_{i_0} \notin S$ und $X_{j_0} \notin S$ erfüllen, dann ist $\alpha = (X_{i_0}, t_{j,0,0})$ nicht τ_{PPT}-lösbar. Das sieht man so: Ist $R = (sup, sig)$ eine beliebige aber feste τ_{PPT}-Region mit $sig(X_{i_0}) = (m, 0)$ und $m \neq 0$. Dann gibt es $\ell \in \{0, \ldots, \lambda\}$ mit $f_{i_0,0} \xrightarrow{a_j} f_{i_0,1}$ und $f_{i_1,0} \xrightarrow{a_j} f_{i_1,1}$. Mit $sig(X_{i_0}) = (m, 0)$ erhalten wir $sig(a_j) = (m, 0)$ und folglich $sig(X_{i_1}) = (m, 0)$. Wegen $t_{j,0} \xrightarrow{X_{j_0}}$ folgt daraus $m \leq sup(t_{j,0})$, das heißt, R löst α nicht. Aus der Beliebigkeit von R folgt die Behauptung.

10.3 Ereignislöschung

Solange nicht ausdrücklich anders vereinbart, sei im Folgenden $\tau \in \{\tau_{PPT}, \tau_{PT}\}$ beliebig aber fest. In diesem Abschnitt untersuchen wir die Komplexität der Aufgabe, ein Transitionssystem, das nicht durch ein τ-Netz implementierbar ist, durch das Löschen einer beschränkten Anzahl von Ereignissen implementierbar zu machen. Genauer gesagt sind die folgenden Entscheidungsprobleme Gegenstand unserer Untersuchungen:

Ereignislöschung für τ-Einbettung	
Eingabe:	Ein Transitionssystem $A = (Z, E, \delta, \iota)$ und $\kappa \in \mathbb{N}$.
Frage:	Gibt es eine Ereignislöschung B von A, sodass $\lvert\mathfrak{E}\rvert \leq \kappa$ erfüllt ist und B die τ-ZTE hat?

Ereignislöschung für τ-Sprachsimulation	
Eingabe:	Ein Transitionssystem $A = (Z, E, \delta, \iota)$ und $\kappa \in \mathbb{N}$.
Frage:	Gibt es eine Ereignislöschung B von A, sodass $\lvert\mathfrak{E}\rvert \leq \kappa$ erfüllt ist und B die τ-EZTE hat?

Ereignislöschung für τ-Realisierung	
Eingabe:	Ein Transitionssystem $A = (Z, E, \delta, \iota)$ und $\kappa \in \mathbb{N}$.
Frage:	Gibt es eine Ereignislöschung B von A, sodass $\lvert\mathfrak{E}\rvert \leq \kappa$ erfüllt ist und B die τ-ZTE und die τ-EZTE hat?

Alle adressierten Entscheidungsprobleme sind in NP:

Lemma 10.4. Ereignislöschung für τ-Realisierung *und* Ereignislöschung für τ-Sprachsimulation *und* Ereignislöschung für τ-Einbettung *in* NP.

Beweis. Im Falle ihrer Existenz kann eine passende Ereignislöschung B von A, die $\lvert\mathfrak{E}\rvert \leq \kappa$ erfüllt, sicherlich in nicht-deterministischer Polynomialzeit geraten werden. Anschließend kann in deterministischer Polynomialzeit verifiziert werden, dass B eine Ereignislöschung ist und die jeweilige Trennungs-Eigenschaften hat, denn die Größe von B ist durch

die Größe von A beschränkt und die klassischen Syntheseprobleme sind polynomiell für τ_{PT} [BBD15]. □

10.3.1 Ereignislöschung für Sprachsimulation und Realisierung

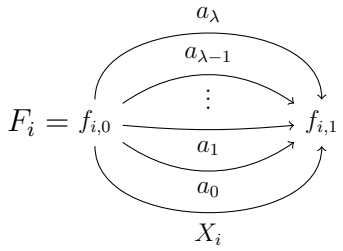
Solange nicht ausdrücklich anders vereinbart, sei im Folgenden $\tau = \tau_{PT}$. Der folgende Satz formuliert das Hauptresultat dieses Teilabschnitts:

Satz 10.3. Ereignislöschung für τ-Realisierung *und* Ereignislöschung für τ-Sprachsimulation *sind in* NP*-vollständig.*

Gemäß Lemma 10.4 sind die in Satz 10.3 adressierten Probleme in NP, sodass nur noch deren NP-Schwere zu zeigen ist.

Wir zeigen die NP-Schwere durch eine Reduktion von CVC, welche die Gadgets der Reduktion aus Abschnitt 10.2 und die beliebige aber feste Instanz $(\mathfrak{U}, M, \lambda)$ von CVC wiederverwendet: Genauer definieren wir zunächst $\kappa = \lambda$. Des Weiteren nutzen wir für alle $i \in \{0, \dots, m-1\}$ und alle $j \in \{0, \dots, n-1\}$ erneut die Gadgets T_i und F_j des vorherigen Abschnitts:

$$T_i = \quad t_{i,0} \xrightarrow{X_{i_0}} t_{i,1} \xrightarrow{X_{i_1}} t_{i,2} \xrightarrow{X_{i_0}} t_{i,3}$$



Das Transitionssystem A hat den Initialzustand $\perp$. Um die Gadgets über $\perp$ miteinander zu verbinden und um zu garantieren, dass wir für die Rückrichtung des Beweises eine Ereignislöschung von A definieren können, die ein initialisiertes Transitionssystem ist, fügen wir für jeden Zustand z der Gadgets eine Kante von $\perp$ zu z hinzu, die mit einem Ereignis beschriftet ist, dass genau einmal in A vorkommt: Für alle $i \in \{0, \dots, m-1\}$ und alle $j \in \{0, \dots, 3\}$ fügen wir die Kante $\perp \xrightarrow{w_i^j} t_{i,j}$ hinzu. Des Weiteren fügen wir für alle $i \in \{0, \dots, n-1\}$ und alle $j \in \{0, 1\}$ die Kante $\perp \xrightarrow{y_i^j} f_{i,j}$ hinzu. Das resultierende Transitionssystem ist A.

Lemma 10.5. *Wenn es eine Ereignislöschung $B = (Z', E', \delta', \perp)$ von A gibt, welche die τ-EZTE hat und $|\mathfrak{E}| \leq \kappa$ erfüllt, dann gibt es eine Knotenüberdeckung mit höchstens λ Elementen für $(\mathfrak{U}, M)$.*

Beweis. Sei $B = (Z', E', \delta', \perp)$ eine Ereignislöschung von A, die $|\mathfrak{E}| \leq \kappa$ erfüllt und die τ-EZTE hat. Weil $|\mathfrak{E}| \leq \kappa$, gibt es ein $j \in \{0, \dots, \lambda\}$, sodass das Ereignis a_j in B ist, das heißt $a_j \in E'$. Sei $i \in \{0, \dots, m-1\}$ beliebig aber fest. Wenn beide Ereignisse X_{i_0}

und X_{i_1} der Menge $M_i = \{X_{i_0}, X_{i_1}\}$ in B sind, das heißt $\{X_{i_0}, X_{i_1}\} \cap \mathfrak{E} = \emptyset$, dann ist T_i vollständig in B und die Kanten $f_{i_0,0} \xrightarrow{X_{i_0}} f_{i_0,1}$ sowie $f_{i_1,0} \xrightarrow{X_{i_1}} f_{i_1,1}$ sind ebenso vorhanden. Darüber hinaus ist $(X_{i_0}, t_{i,1})$ ein EZTP von B, weil T_i ganz vorhanden ist. Weil auch die Kanten $f_{i_0,0} \xrightarrow{a_j} f_{i_0,1}$ und $f_{i_1,0} \xrightarrow{a_j} f_{i_1,1}$ in B sind, erhalten wir analog zu der Argumentation im Beweis von Lemma 10.2 einen Widerspruch zur τ-EZTE von B. Daraus folgt $\{X_{i_0}, X_{i_1}\} \cap \mathfrak{E} \neq \emptyset$. Weil i beliebig war, definiert die Menge $S = \{X \in \mathfrak{U} \mid X \in \mathfrak{E}\}$ eine Knotenüberdeckung mit höchstens λ Elementen für $(\mathfrak{U}, M)$. □

Für die andere Richtung müssen wir zeigen, dass eine Knotenüberdeckung mit höchstens λ Elementen für $(\mathfrak{U}, M)$ eine gesuchte Ereignislöschung für A impliziert. Dies ist der Gegenstand des folgenden Lemmas:

Lemma 10.6. *Wenn es eine Knotenüberdeckung mit höchstens λ Elementen für $(\mathfrak{U}, M)$ gibt, dann gibt es eine Ereignislöschung B von A, welche die τ-EZTE und die τ-ZTE hat und $|\mathfrak{E}| \leq \kappa$ erfüllt.*

Beweis. Sei S Knotenüberdeckung mit höchstens λ Elementen für $(\mathfrak{U}, M)$. Wir erhalten $B = (Z, E', \delta', \bot)$ mit $Z = Z(A)$ in naheliegender Weise durch $E' = E(A) \setminus S$ und der Modifikation von δ', sodass für jedes Ereignis $e \in E'$ eine Kante $z \xrightarrow{e} z'$ genau dann in B ist, wenn $z \xrightarrow{e} z'$ eine Kante in A ist. Man beachte, dass B tatsächlich die Zustände von A erhält. Wegen der w_i^j beschrifteten Kanten von $\bot$ nach $t_{i,j}$ beziehungsweise wegen der y_i^j beschrifteten Kanten von $\bot$ nach $f_{i,j}$ ist B ein wohldefiniertes initialisiertes Transitionssystem.

Seien $W = \{w_0^0, \ldots, w_{m-1}^3\}$ und $Y = \{y_0^0, \ldots, y_{n-1}^1\}$. Die folgende Region $R_1 = (sup_1, sig_1)$ löst $(\bot, z)$ und (e, z) für alle $z \in Z \setminus \{\bot\}$ und alle $e \in W \cup Y$: Für alle $z \in Z$, wenn $z = \bot$, dann $sup_1(z) = 1$, sonst $sup_1(z) = 0$; für alle $e \in E'$, wenn $e \in W \cup Y$, dann $sig_1(e) = (1, 0)$, sonst $sig_1(e) = (0, 0)$.

Darüber hinaus löst die folgende Region $R_2 = (sup_2, sig_2)$ das EZTP $(e, \bot)$ für alle $e \in E' \setminus (W \cup Y)$: $sup_2(\bot) = 0$; für alle $i \in \{0, \ldots, m-1\}$ und alle $j \in \{0, \ldots, 3\}$, $sup_2(t_{i,j}) = 3 - j$ und $sig_2(w_i^j) = (0, 3 - j)$; für alle $i \in \{0, \ldots, n-1\}$ und alle $j \in \{0, 1\}$, $sup_2(f_{i,j}) = 1 - j$ und $sig_2(y_i^j) = (0, 1 - j)$; außerdem $sig_2(e) = (1, 0)$ für alle $e \in E' \setminus (W \cup Y)$.

Um den Beweis des Lemmas zu vervollständigen, bleibt die Lösbarkeit aller EZTP (e, z) beziehungsweise ZTP (z, z') von B mit $e \in E' \setminus (W \cup Y)$ und $z, z' \in Z \setminus \{\bot\}$ zu zeigen.

Dazu beachten wir, dass sich jede Region $R_i = (sup_i, sig_i)$ der Fakten 10.1 und 10.2 aus Abschnitt 10.2 auf naheliegende Weise zu einer Region $R_i' = (sup_i', sig_i')$ von B fortsetzen lässt. Die liegt daran, dass das Transitionssystem A der aktuellen Konstruktion als eine Erweiterung des Transitionssystems der Reduktion aus Abschnitt 10.2 interpretiert werden kann. Die Ereignisse $y_0^0, \ldots, y_0^{m-1}$ beziehungsweise $w_0^0, \ldots, w_0^{n-1}$ der vorliegenden Konstruktion korrespondieren dabei mit den Ereignissen $y_0, \ldots, y_{m-1}$ beziehungsweise $w_0, \ldots, w_{n-1}$ der Konstruktion aus Abschnitt 10.2. Genauer erhalten wir

$R'_i = (sup'_i, sig'_i)$ wie folgt: $sup'_i(z) = sup_i(z)$ für alle $z \in Z(B)$; außerdem definieren wir für alle $e \in E(B)$: wenn $e \in E(B) \setminus (W \cup Y)$, dann $sig'_i(e) = sig_i(e)$; für alle $i \in \{0, \ldots, m-1\}$, wenn $e = y_i^0$, dann $sig'_i(e) = sig_i(y_i)$; für alle $j \in \{0, \ldots, n-1\}$, wenn $e = w_j^0$, dann $sig'_i(e) = sig_i(w_j)$; wenn $e \in \{w_i^1, w_i^2, w_i^3 \mid i \in \{0, \ldots, m-1\}\}$ oder $e \in \{y_j^1 \mid j \in \{0, \ldots, n-1\}\}$, dann gibt es eine eindeutige e-beschriftete Kante $\bot \xrightarrow{e} z$ in B und wir definieren $sig'_i(e) = (sup_i(\bot) - sup_i(z), 0)$, falls $sup_i(\bot) > sup_i(z)$ und andernfalls $sig'_i(e) = (0, sup_i(z) - sup_i(\bot))$.

Somit erhalten wir die τ-Lösbarkeit der verbleibenden ZTP und EZTP aus den entsprechenden Regionen der Fakten 10.1 und 10.2 aus Abschnitt 10.2. Dies beweist das Lemma. □

Bemerkung 10.3. Ähnlich zu Bemerkung 10.2 argumentiert man, dass die vorliegende Konstruktion (dieses Abschnitts) für den Netztyp τ_{PPT} nicht funktioniert. Genauer gesagt, können wir nicht allein mit puren Regionen nachweisen, dass B die τ-EZTE hat. Deshalb haben wir uns auch für Ereignislöschung für τ-Realisierung und Ereignislöschung für τ-Sprachsimulation auf τ_{PT} beschränkt.

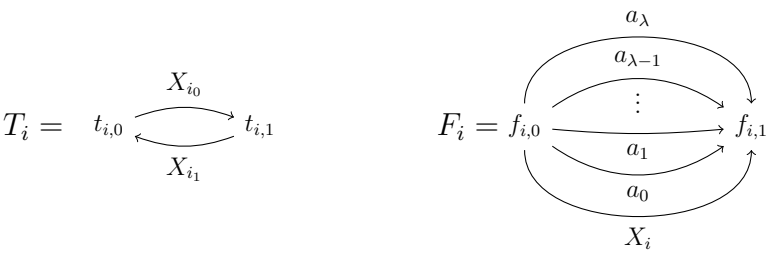
10.3.2 Ereignislöschung für Einbettung

Solange nicht ausdrücklich anders vereinbart, sei im Folgenden $\tau \in \{\tau_{PPT}, \tau_{PT}\}$ beliebig aber fest. Eine minimale Ereignislöschung zu finden, ist auch dann schwer, wenn wir eine Einbettung anstreben:

Satz 10.4. Ereignislöschung für τ-Einbettung *ist NP-vollständig.*

Gemäß Lemma 10.4 ist Ereignislöschung für τ-Einbettung in NP. Die NP-Schwere des Problems zeigen wir erneut mittels einer Reduktion von CVC:

Dabei reduzieren wir eine beliebige aber feste Instanz $(\mathfrak{U}, M, \lambda)$ von CVC wie folgt auf eine Paar (A, κ) mit Transitionssystem A und Zahl κ: Zunächst definieren wir $\kappa = \lambda$. Das Transitionssystem A hat für alle $i \in \{0, \ldots, m-1\}$ das folgende Gadget T_i, welches die Elemente der Menge $M_i = \{X_{i_0}, X_{i_1}\}$ als Ereignisse verwendet, und für alle $j \in \{0, \ldots, n-1\}$ das Gadget F_j, das wir bereits aus den vorherigen Reduktionen kennen:



Das Transitionssystem A hat den Initialzustand $\bot$. Um die eingeführten Gadgets über $\bot$ miteinander zu verbinden und um zu garantieren, dass wir für die Rückrichtung eine Ereignislöschung von A definieren können (wobei die Erreichbarkeit der Zustände über

einen initialisierten Pfad der entscheidende Punkt ist), fügen wir für jeden Zustand z der Gadgets eine Kante von $\bot$ zu z hinzu, die mit einem Ereignis beschriftet ist, dass genau einmal in A vorkommt: Für alle $i \in \{0, \ldots, m-1\}$ und alle $j \in \{0, 1\}$ fügen wir die Kante $\bot \xrightarrow{w_i^j} t_{i,j}$ hinzu. Des Weiteren fügen wir $i \in \{0, \ldots, n-1\}$ und alle $j \in \{0, 1\}$ die Kante $\bot \xrightarrow{y_i^j} f_{i,j}$ hinzu. Das resultierende Transitionssystem ist $A = (Z, E, \delta, \bot)$.

Lemma 10.7. *Wenn es eine Ereignislöschung B für A gibt, welche die τ-ZTE hat und $|\mathfrak{E}| \leq \kappa$ erfüllt, dann gibt es eine Knotenüberdeckung mit höchstens λ Elementen für $(\mathfrak{U}, M)$.*

Beweis. Sei $B = (Z', E', \delta', \bot)$ eine Ereignislöschung für A, welche die τ-ZTE hat und $|\mathfrak{E}| \leq \kappa$ erfüllt. Sei $i \in \{0, \ldots, m-1\}$ beliebig aber fest. Weil B die τ-ZTE hat, gibt es eine Region $R = (sup, sig)$, sodass $sup(t_{i,0}) > sup(t_{i,1})$ oder $sup(t_{i,0}) < sup(t_{i,1})$. Wenn sowohl X_{i_0} als auch X_{i_1} in B stattfinden, das heißt $\{X_{i_0}, X_{i_1}\} \cap \mathfrak{E} = \emptyset$, dann impliziert der erste Fall $sig^-(X_{i_0}) < sig^+(X_{i_0})$ und $sig^-(X_{i_1}) > sig^+(X_{i_1})$ und der zweite Fall impliziert $sig^-(X_{i_0}) > sig^+(X_{i_0})$ und $sig^-(X_{i_1}) < sig^+(X_{i_1})$. Weil $|\mathfrak{E}| \leq \kappa$ und $f_{i_0,0} \xrightarrow{X_{i_0}} f_{i_0,1}$ sowie $f_{i_1,0} \xrightarrow{X_{i_1}} f_{i_1,1}$ in B sind, gibt es ein $a \in \{a_0, \ldots, a_\lambda\} \cap E'$, sodass beide Fälle den Widerspruch $sig(a)^- < sig(a)^+$ und $sig(a)^- > sig(a)^+$ implizieren. Daher muss es ein $X \in \{X_{i_0}, X_{i_1}\}$ geben, das nicht in B stattfindet, das heißt $X \in \mathfrak{E}$. Weil i beliebig war, gilt $M_i \cap \mathfrak{E} \neq \emptyset$ für alle $i \in \{0, \ldots, m-1\}$ und die Menge $S = \{X \in \mathfrak{U} \mid X \in \mathfrak{E}$ definiert eine passende Knotenüberdeckung für $(\mathfrak{U}, M)$. □

Für die andere Richtung beweisen wir das folgenden Lemma, dass den Beweis von Satz 10.4 vervollständigt:

Lemma 10.8. *Wenn es eine Knotenüberdeckung mit höchstens λ Elementen für $(\mathfrak{U}, M)$ gibt, dann gibt es eine Ereignislöschung B für A, welche die τ-ZTE hat und $|\mathfrak{E}| \leq \kappa$ erfüllt.*

Beweis. Sei S eine passende Knotenüberdeckung von A. Wir erhalten $B = (Z, E', \delta', \bot)$ in naheliegender Weise durch $E' = E(A) \setminus S$ und das Löschen aller mit einem Ereignis aus S beschrifteter Kanten. Man bemerke, dass B tatsächlich die Zustände von A bewahrt. Die folgende Region $R = (sup, sig)$ löst alle ZTP von B: Zunächst definieren wir $sup(\bot) = 0$ sowie $sig(e) = (1, 0)$ für alle $e \in \mathfrak{U} \setminus S$ und alle $e \in \{a_0, \ldots, a_\lambda\}$; für alle $i \in \{0, \ldots, m-1\}$ definieren wir des Weiteren $sup(t_{i,0}) = 2i+2$ und $sup(t_{i,1}) = 2i+1$ und $sig(w_i^0) = (0, 2i+2)$ sowie $sig(w_i^1) = (0, 2i+1)$, falls $X_{i_0} \in E'$, andernfalls definieren wir $sup(t_{i,0}) = 2i+1$ und $sup(t_{i,1}) = 2i+2$ und $sig(w_i^0) = (0, 2i+1)$ sowie $sig(w_i^1) = (0, 2i+2)$; für alle $j \in \{0, \ldots, n-1\}$ definieren wir $sup(f_{j,0}) = 2m+2j+2$ und $sup(f_{j,0}) = 2m+2j+1$ und $sig(y_j^0) = (0, 2m+2j+2)$ sowie $sig(y_j^1) = (0, 2m+2j+1)$. Dies beweist das Lemma. □

10.4 Zustandslöschung

Das Problem der *Verhaltensmodifikation durch Zustandslöschung* besteht darin, für ein gegebenes Transitionssystem A und eine Zahl κ zu entscheiden, ob A durch das Löschen von höchstens κ Zuständen –und den jeweiligen inzidenten Kanten– zu einem Transitionsystem B modifiziert werden kann, das durch ein τ-Netz implementierbar ist. In diesem Abschnitt zeigen wir für die Netztypen τ_{PPT} beziehungsweise τ_{PT}, dass das Problem für jede der Implementierungen Einbettung oder Sprachsimulation beziehungsweise Realisierung NP-vollständig ist. Solange nicht ausdrücklich anders vereinbart, sei $\tau \in \{\tau_{PPT}, \tau_{PT}\}$ im Folgenden also beliebig aber fest.

Die in diesem Abschnitt adressierten Entscheidungsprobleme sind wie folgt definiert:

Zustandslöschung für τ-Einbettung	
Eingabe:	Ein endliches Transitionssystem $A = (Z, E, \delta, \iota)$ und $\kappa \in \mathbb{N}$.
Frage:	Gibt es eine Zustandslöschung B von A, sodass $\lvert\mathfrak{Z}\rvert \leq \kappa$ erfüllt ist und B die τ-ZTE hat?

Zustandslöschung für τ-Sprachsimulation	
Eingabe:	Ein endliches Transitionssystem $A = (Z, E, \delta, \iota)$ und $\kappa \in \mathbb{N}$.
Frage:	Gibt es eine Zustandslöschung B von A, sodass $\lvert\mathfrak{Z}\rvert \leq \kappa$ erfüllt ist und B die τ-EZTE hat?

Zustandslöschung für τ-Realisierung	
Eingabe:	Ein endliches Transitionssystem $A = (Z, E, \delta, \iota)$ und $\kappa \in \mathbb{N}$.
Frage:	Gibt es eine Zustandslöschung B von A, sodass $\lvert\mathfrak{Z}\rvert \leq \kappa$ erfüllt ist und B die τ-ZTE und die τ-EZTE hat?

Das folgenden Lemma konstatiert die Zugehörigkeit zur Komplexitätsklasse NP:

Lemma 10.9. Zustandslöschung für τ-Einbettung *und* Zustandslöschung für τ-Sprachsimulation *und* Zustandslöschung für τ-Realisierung *sind in* NP.

Beweis. Wenn es für ein Transitionssystem $A = (Z, E, \delta, \iota)$ und eine natürliche Zahl κ eine implementierbare Zustandslöschung $B = (Z', E', \delta', \iota)$ von A gibt, die $\lvert Z \setminus Z'\rvert \leq \kappa$ erfüllt, dann kann B sicherlich in nicht-deterministischer Polynomialzeit von einer Turing-Maschine geraten werden. Es ist die Größe von B durch die Größe von A beschränkt. Deshalb kann die Implementierbarkeit von B in (deterministischer) Polynomialzeit überprüft werden, indem ein Zeuge von τ-Regionen für die Trennungseigenschaft berechnet wird, die der gesuchten Implementierung entspricht [BBD15]. □

Die folgenden beiden Abschnitte adressieren die NP-Schwere der Entscheidungsprobleme. Diese beweisen wir mittels geeigneter Reduktionen von CVC. Solange nicht ausdrücklich anders vereinbart, sei $(\mathfrak{U}, M, \lambda)$ im Folgenden eine beliebige aber feste Eingabe von

CVC mit $\mathfrak{U} = \{X_0, \ldots, X_{n-1}\}$ und $M = \{M_0, \ldots, M_{m-1}\}$, sodass $M_i = \{X_{i_0}, X_{i_1}\} \subseteq \mathfrak{U}$ und $i_0 < i_1$ für alle $i \in \{0, \ldots, m-1\}$.

10.4.1 Zustandslöschung für Einbettung und Realisierung

Der folgende Satz umfasst das Hauptresultat dieses Abschnitts:

Satz 10.5. Zustandslöschung für τ-Einbettung *und* Zustandslöschung für τ-Realisierung *sind* NP*-vollständig.*

Der Rest dieses Abschnitts ist dem Beweis von Satz 10.5 gewidmet. Gemäß Lemma 10.9 sind die adressierten Probleme in NP.

Für die NP-Schwere reduzieren wie die Instanz $(\mathfrak{U}, M, \lambda)$ auf ein Paar (A, κ) mit Transitionssystem A und natürlicher Zahl κ, sodass A genau dann eine passende Zustandslöschung B erlaubt, welche die τ-ZTE beziehungsweise die τ-EZTE hat, wenn es eine Knotenüberdeckung mit höchstens λ Elementen für $(\mathfrak{U}, M)$ gibt: Zunächst definieren wir $\kappa = \lambda$. Wir erhalten das Transitionssystem $A = (Z, E, \delta, \bot)$ mit Initialzustand $\bot$ wie folgt:

- die Zustandsmenge ist definiert durch $Z = \mathfrak{U} \cup \{\bot\}$, das heißt, die Elemente aus $\mathfrak{U}$ werden Zustände in A;
- die Ereignismenge ist definiert durch $E = \{a_0, \ldots, a_{n-1}\} \cup \{M_0, \ldots, M_{m-1}\}$, wobei wir für alle $i \in \{0, \ldots, m-1\}$ das Ereignis M_i mit der Kante $M_i = \{X_{i_0}, X_{i_1}\}$ von $(\mathfrak{U}, M)$ assoziieren;
- für alle $i \in \{0, \ldots, x_{n-1}\}$ hat A die Kante $\bot \xrightarrow{a_i} X_i$;
- für alle $i \in \{0, \ldots, m-1\}$ hat A die beschrifteten Kanten $X_{i_0} \xrightarrow{M_i} X_{i_1}$ und $X_{i_1} \xrightarrow{M_i} X_{i_0}$, die mit der Kante $M_i = \{X_{i_0}, X_{i_1}\}$ von $(\mathfrak{U}, M)$ assoziiert wird.

Abbildung 10.5 zeigt das Transitionssystem A, welches gemäß der beschriebenen Reduktion aus der Eingabe für CVC resultiert, die in Beispiel 10.1 definiert wurde.

Wenn es eine Zustandslöschung B von A gibt, die $|\mathfrak{Z}| \leq \kappa$ erfüllt und eine τ-Einbettung oder eine τ-Realisierung erlaubt, dann hat B die τ-ZTE. So eine Zustandslöschung impliziert eine gesuchte Knotenüberdeckung für $(\mathfrak{U}, M)$:

Lemma 10.10. *Wenn es eine Zustandslöschung B von A gibt, die $|\mathfrak{Z}| \leq \kappa$ erfüllt und die τ-ZTE hat, dann gibt es eine Knotenüberdeckung mit höchstens λ Elementen für $(\mathfrak{U}, M)$.*

Beweis. Sei $B = (Z', E', \delta', \bot)$ eine Zustandslöschung von A, die $|\mathfrak{Z}| \leq \kappa$ erfüllt und die τ-ZTE hat. Wir zeigen zunächst, dass es keine bidirektionalen Kanten in B geben kann: Nehmen wir an, es gibt Zustände $z, z' \in Z'$ und ein Ereignis $e \in E'$, sodass beide Kanten $z \xrightarrow{e} z'$ und $z' \xrightarrow{e} z$ in B sind. Weil B die τ-ZTE hat, gibt es eine τ-Region, die

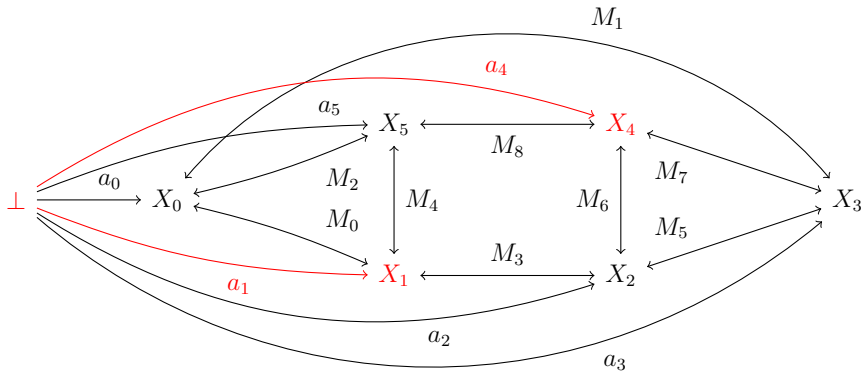


Abbildung 10.5: Das Transitionssystem A, welches aus Beispiel 10.1 resultiert. Gemäß des Beweises von Lemma 10.11 erhalten wir eine einbettbare Zustandslöschung B von A durch das Löschen der Elemente X_0, X_2, X_3, X_5 der Knotenüberdeckung aus Beispiel 10.1. Es ist B durch die rot markierten Zustände und Kanten skizziert.

das ZTP (z, z') löst. Sei $R = (sup, sig)$ so eine Region. Weil R eine Region ist, implizieren $z \xrightarrow{e} z'$ und $z' \xrightarrow{e} z$ jeweils, dass $sup(z) \xrightarrow{sig(e)} sup(z')$ und $sup(z') \xrightarrow{sig(e)} sup(z)$ Kanten in τ sind. Gemäß der Definition von τ folgt daraus $sup(z) = sup(z') - sig^-(e) + sig^+(e)$ und $sup(z') = sup(z) - sig^-(e) + sig^+(e)$. Dies impliziert $sup(z) = sup(z')$ und widerspricht der Annahme, dass R das ZTP (z, z') löst. Mitthin haben wir gezeigt, dass B keine bidirektionalen Kanten enthält.

Daraus folgt für jedes $i \in \{0, \ldots, m-1\}$, dass eine der M_i-beschrifteten Kanten $X_{i_0} \xrightarrow{M_i} X_{i_1}$ oder $X_{i_1} \xrightarrow{M_i} X_{i_0}$ nicht in B vorhanden ist. Weil B eine Knotenlöschung von A ist, impliziert dies $X_{i_0} \notin Z'$ oder $X_{i_1} \notin Z'$, das heißt $X_{i_0} \in \mathfrak{Z}$ oder $X_{i_1} \in \mathfrak{Z}$. Daraus folgt $\{X_{i_0}, X_{i_1}\} \cap \mathfrak{Z} \neq \emptyset$ für alle $i \in \{0, \ldots, m-1\}$. Mit $|\mathfrak{Z}| \leq \kappa = \lambda$ erhalten wir schließlich, dass die Menge $S = \mathfrak{U} \cap \mathfrak{Z}$ ein Knotenüberdeckung mit höchstens λ Elementen für $(\mathfrak{U}, M)$ ist. □

Mit Lemma 10.10 folgt, dass $(\mathfrak{U}, M, \lambda)$ eine positive Entscheidung erlaubt, wenn (A, κ) eine positive Entscheidung erlaubt. Das folgende Lemma beweist, das auch die umgekehrte Implikation wahr ist:

Lemma 10.11. *Wenn es eine Knotenüberdeckung mit höchstens λ Elementen für $(\mathfrak{U}, M)$ gibt, dann gibt es eine Zustandslöschung B von A, die $|\mathfrak{Z}| \leq \kappa$ erfüllt und die τ-ZTE und die τ-EZTE hat.*

Beweis. Sei $S \subseteq \mathfrak{U}$ eine Knotenüberdeckung mit höchstens λ Elementen für $(\mathfrak{U}, M)$. Wir erhalten $B = (Z', E', \delta', \bot)$ mit Zustandsmenge $Z' = Z(A) \setminus S$ in naheliegender Weise durch Löschung aller Zustände $X \in \{X_0, \ldots, X_{n-1}\}$ deren assoziiertes Ereignis $X \in \mathfrak{U}$ in

S ist. Zunächst überlegt man sich, dass B ein wohldefiniertes initialisiertes Transitionssystem ist, das heißt, jeder Zustand X_i, der in B präsent ist, kann durch eine a_i-beschriftete Kante von $\bot$ aus erreicht werden. (Siehe auch Abbildung 10.5.) Weil S eine Knotenüberdeckung von $(\mathfrak{U}, M)$ ist, ist für alle $i \in \{0, \dots, m-1\}$ wenigstens einer der Zustände X_{i_0}, X_{i_1} der Kanten $X_{i_0} \xrightarrow{M_i} X_{i_1}$ beziehungsweise $X_{i_1} \xrightarrow{M_i} X_{i_0}$ gelöscht. Dies impliziert, dass *keine* dieser Kanten in B vorhanden ist, sodass $\{M_0, \dots, M_{m-1}\} \cap E' = \emptyset$. Für die verbleibenden Ereignisse gilt jedoch, dass sie in B genau einmal stattfinden. Wie man leicht sieht, impliziert dies insbesondere, dass alle ZTP und alle EZTP von B τ-lösbar sind. Genauer erhalten wir für einen beliebigen aber festen Zustand $q \in Z'$ wie folgt eine τ-Region $R = (sup, sig)$: $sup(q) = 1$ und $sup(z) = 0$ für alle $z \in Z' \setminus \{q\}$; für alle $e \in E$, wenn $\xrightarrow{e} q$, dann $sig(e) = (0, 1)$; wenn $q \xrightarrow{e}$, dann $sig(e) = (1, 0)$, andernfalls $sig(e) = (0, 0)$. Offenbar werden q und alle von q ausgehenden Ereignisse e durch R gelöst. □

10.4.2 Zustandslöschung für Sprachsimulation

Auch wenn wir eine Sprachsimulation anstreben, kann eine passende Zustandslöschung im Allgemeinen nur schwer gefunden werden:

Satz 10.6. Zustandslöschung für τ-Sprachsimulation *ist NP-vollständig.*

Gemäß Lemma 10.9 ist Zustandslöschung für τ-Sprachsimulation in NP. Somit genügt es, die NP-Schwere zu zeigen, um den Beweis von Satz 10.6 komplettieren. Dazu präsentieren wir erneut eine Reduktion von CVC, welche der aus Abschnitt 10.4.1 ähnlich ist, diese jedoch erweitert: Es ist $\kappa = \lambda$ und ein Teil des hier konstruierten Transitionssystems A repräsentiert die Elemente aus $\mathfrak{U}$ beziehungsweise aus M der Eingabe $(\mathfrak{U}, M, \lambda)$ für CVC. Dazu hat A alle Zustände und alle beschrifteten Kanten der Konstruktion aus Abschnitt 10.4.1. Ziel ist es, erneut zu erzwingen, dass die τ-EZTE einer Knotenlöschung B von A, die $|\mathfrak{Z}| \leq \kappa = \lambda$ erfüllt, keine der bidirektionalen Kanten $X_{i_0} \xleftrightarrow{M_i} X_{i_1}$ mehr enthält, sodass die Menge der gelöschten Zustände ein passende Knotenüberdeckung für $(\mathfrak{U}, M)$ impliziert. Um dies zu erreichen werden wir die Konstruktion aus Abschnitt 10.4.1 wie folgt erweitern: Für alle $i \in \{0, \dots, m-1\}$ und alle $j \in \{0, \dots, \lambda\}$ wird der Pfad $\bot \xrightarrow{y_j^i} t_{i,j,0} \xrightarrow{M_i} t_{i,j,1}$ hinzugefügt. Das resultierende Transitionssystem ist A.

Lemma 10.12. *Wenn B eine Zustandslöschung von A ist, die $|\mathfrak{Z}| \leq \kappa$ erfüllt und die τ-EZTE hat, dann gibt es eine Knotenüberdeckung mit höchstens λ Elementen für $(\mathfrak{U}, M)$.*

Beweis. Sei B eine passende Zustandslöschung von A. Gemäß Konstruktion hat A für jedes $i \in \{0, \dots, m-1\}$ genau $\lambda+1$ M_i-beschriftete Kanten $t_{i,0,0} \xrightarrow{M_i} t_{i,0,1}, \dots, t_{i,\lambda,0} \xrightarrow{M_i} t_{i,\lambda,1}$. Weil B durch Löschung von höchstens $\kappa = \lambda$ Zuständen aus A resultiert, gibt es also für jedes $i \in \{0, \dots, m-1\}$ eine Kante $z \xrightarrow{M_i} z'$ mit $\neg z' \xrightarrow{M_i}$ in B. Weil B die τ-EZTE hat, ist

das entsprechende EZTP (M_i, z') τ-lösbar. Deshalb gibt es eine τ-Region $R = (sup, sig)$ von B, sodass $sup(z) \xrightarrow{sig(M_i)}$ und $\neg sup(z') \xrightarrow{sig(M_i)}$. Dies impliziert $sig^-(M_i) > sig^+(M_i)$. Daraus folgt, dass es keine bidirektionale Kante $X_{i_0} \overset{M_i}{\longleftrightarrow} X_{i_1}$ in B geben kann, denn solche Kante impliziert $sig^-(M_i) = sig^+(M_i)$ für alle τ-Regionen (sup, sig) von B. Wir schließen daraus, dass $\mathfrak{Z} \cap \{X_{i_0}, X_{i_1}\} \neq \emptyset$ für alle $i \in \{0, \ldots, m-1\}$. Weil $|\mathfrak{U} \cap \mathfrak{Z}| \leq |\mathfrak{Z}| \leq \kappa = \lambda$, ist die Menge $S = \mathfrak{U} \cap \mathfrak{Z}$ eine Knotenüberdeckung mit höchstens λ Elementen für $(\mathfrak{U}, M)$. □

Für die andere Richtung beweisen wir das folgende Lemma, welches den Beweis von Satz 10.6 abschließt.

Lemma 10.13. *Wenn es eine Knotenüberdeckung mit höchstens λ Elementen für $(\mathfrak{U}, M)$ gibt, dann gibt es eine Zustandslöschung B von A, die $|\mathfrak{Z}| \leq \kappa$ erfüllt und die τ-EZTE hat.*

Beweis. Sei S Knotenüberdeckung mit höchstens λ Elementen für $(\mathfrak{U}, M)$. Wir erhalten $B = (Z', E', \delta', x_0)$ in naheliegender Weise durch $Z' = Z(A) \setminus S$. Weil S eine Knotenüberdeckung ist, enthält B keine bidirektionalen Kanten $X_{i_0} \overset{M_i}{\longleftrightarrow} X_{i_1}$ für alle $i \in \{0, \ldots, m-1\}$. Sei nun $a \in E'$ beliebig aber fest und seien $q_0, \ldots, q_k$ genau die Ursprünge von a in B. Man überlegt sich, dass die folgende τ-Region $R = (sup, sig)$ wohldefiniert ist und das EZTP (a, p) für alle $p \in Z'$ mit $\neg p \xrightarrow{a}$ löst: für alle $z \in Z'$, wenn $z \in \{q_0, \ldots, q_k\}$, dann $sup(z) = 1$, sonst $sup(z) = 0$; für alle $e \in E'$, wenn $\xrightarrow{e} q$ für ein $q \in \{q_0, \ldots, q_k\}$, dann $sig(e) = (0, 1)$; wenn $q \xrightarrow{e}$ für ein $q \in \{q_0, \ldots, q_k\}$, dann $sig(e) = (1, 0)$; andernfalls $sig(e) = (0, 0)$. Weil a beliebig war, hat B die τ-EZTE. □

10.5 Zusammenfassung und Ausblick

In diesem Kapitel haben wir durch *Kantenlöschung*, *Ereignislöschung* und *Zustandslöschung* drei Möglichkeiten kennengelernt, ein Transitionssystem zum Zwecke der Implementierbarkeit zu modifizieren. Auf natürliche Weise können wir aus diesen Modifikationen Entscheidungsprobleme ableiten, die für ein gegebenes Transitionssystem A und eine Zahl κ fragen, ob sich A durch das Löschen von höchstens κ Kanten, Ereignissen oder Zuständen, je nachdem, zu einem implementierbaren Transitionssystem modifizieren lässt.

In diesem Kapitel haben wir für alle resultierenden Entscheidungsprobleme und jeweils jede der Implementierungen Einbettung, Sprachsimulation und Realisierung gezeigt, dass sie NP-vollständig sind, wenn das implementierende Netz ein P/T-Netz sein soll. Des Weiteren gelingt eine teilweise Charakterisierung, wenn auf pure P/T-Netze abgezielt wird: *Ereignislöschung* ist auch für pure P/T-Netze NP-vollständig ist, wenn eine Einbettung angestrebt wird. Darüber hinaus ist *Zustandslöschung* ebenso für pure P/T-Netze und jede der genannten Implementierungen NP-vollständig.

Für zukünftige Forschung bleibt zu klären, ob die durch *Kantenlöschung* und *Ereignislöschung* implizierten Entscheidungsprobleme für die puren P/T-Netze auch dann schwer bleiben, wenn eine Sprachsimulation oder Realisierung angestrebt werden.

Des Weiteren überlegt man sich, dass jedes der durch *Kantenlöschung*, *Ereignislöschung* und *Zustandslöschung* implizierten parametrisierten Probleme in XP ist (siehe Definition 2.30), wenn κ als Parameter gewählt wird und τ_{PT}- oder τ_{PPT}-Netze gesucht werden: Für ein gegebenes Paar $A = (Z, E, \delta, \iota)$ und κ muss für höchstens $\mathcal{O}(|\Delta_A|^\kappa)$ Mengen von Kanten, $\mathcal{O}(|E|^\kappa)$ Mengen von Ereignissen und $\mathcal{O}(|Z|^\kappa)$ Mengen von Zuständen überprüft werden, ob das Löschen der jeweiligen Kanten, Ereignisse oder Zustände, je nachdem, in einem implementierbaren Transitionssystem resultiert. Weil dieser Test für das (dann vorliegende) modifizierte Transitionssystem in Polynomialzeit durchführbar ist [BBD95, BD96a], erhalten wir jeweils einen Entscheidungsalgorithmus, dessen Laufzeit für eine (berechenbare) Funktion f durch $|A|^{f(\kappa)}$ beschränkt ist. Zukünftige Forschung könnte nun die Frage untersuchen, ob diese parametrisierten Probleme mit Parameter κ fest-Paramter-berechenbar sind.

Kapitel 11

Fazit

In der vorliegenden Arbeit haben wir unser Verständnis der Komplexität der Synthese von Petri-Netzen bezüglich verschiedener Syntheseprobleme und unterschiedlicher Netztypen erweitert. Dabei klassifizieren wir Netze gemäß zweier Dimensionen, nämlich einerseits bezüglich der erlaubten Anzahl von Marken, die ein Platz in jeder erreichbaren Markierung höchstens enthalten darf; dies führt zur Unterscheidung zwischen booleschen Netzen, b-beschränkten Netzen und beschränkten Netzen. Andererseits klassifizieren wir Netze gemäß der möglichen Fluss-Beziehungen zwischen Plätzen und Transitionen. Dies führt dazu, dass innerhalb der booleschen Netze genau 256 Klassen unterschieden werden, denn jede Teilmenge τ der booleschen Interaktionen nop, inp, out, res, set, swap, used und free definiert eine solche Klasse, deren Netze höchstens die Interaktionen aus τ verwenden. Innerhalb der b-beschränkten Netze unterscheiden wir die puren b-beschränkten P/T-Netze, b-beschränkten P/T-Netze und deren jeweilige Erweiterungen um die Gruppe $\mathbb{Z}_{b+1}$. Schließlich unterscheiden zwischen den (beschränkten) puren P/T-Netzen und den (beschränkten) P/T-Netzen.

Innerhalb dieser Netzklassen untersuchen wir, je nachdem, Probleme der Netzsynthese, welche die Frage nach der Existenz eines implementierenden Netzes überhaupt stellen: τ-Einbettung und τ-Sprachsimulation sowie τ-Realisierung. Für die Klasse der booleschen Netze gehen die ersten Arbeiten zur Komplexität dieser Syntheseprobleme zurück auf Badouel, Bernardinello und Darondeau [BBD97], die für die elementaren Netz-Systeme ($\{\mathsf{nop}, \mathsf{inp}, \mathsf{out}\}$) gezeigt haben, dass τ-Realisierung NP-vollständig ist, und auf Schmitt [Sch96], der für die Flip-Flop-Netze ($\{\mathsf{nop}, \mathsf{inp}, \mathsf{out}, \mathsf{swap}\}$) gezeigt hat, dass τ-Einbettung, τ-Sprachsimulation und τ-Realisierung in Polynomialzeit entscheidbar sind.

In der vorliegenden Arbeit setzen wir die Charakterisierung der Komplexität dieser Syntheseprobleme für boolesche Netzklassen fort: Eine vollständige Charakterisierung liefern wir für das Problem τ-Einbettung, das für 152 der 256 booleschen Netze NP-vollständig und für die verbleibenden 104 Netzklassen in Polynomialzeit entscheidbar ist. Bemerkenswert ist hier die Beobachtung, dass die Abwesenheit der Interaktion nop, die Unabhängigkeit zwischen Plätzen und Transitionen erlaubt, kein Garant für ein poly-

nomielles Entscheidungsverfahren ist: sogar die Synthese von Netzen, die nur Test auf Abwesenheit (free) und Invertierung (swap) erlauben, ist NP-vollständig.

Des Weiteren liefern wir eine Charakterisierung von τ-Sprachsimulation und τ-Realisierung für eine Majorität der booleschen Netze mit nop: Für 84 der 128 Netztypen mit nop sind beide Probleme NP-vollständig. Diese Charakterisierung umfasst (außer den Flip-Flop-Netzen) insbesondere alle weiteren booleschen Netzklassen, die bisher explizit in der Literatur diskutiert wurden: *contextual nets* (nop, inp, out, used, free) [MR95] und *event/condition nets* (nop, inp, out, used) [BBD15] als auch *inhibitor nets* (nop, inp, out, free) [Pie97] sowie *set nets* (nop, inp, out, set) [KKPR13] und *trace nets* (nop, inp, out, res, set, used, free) [BD95b]. Außerdem zeigen wir, dass τ-Sprachsimulation für 40 und τ-Realisierung für 36 dieser Netztypen in Polynomialzeit entscheidbar ist. Darüber hinaus beweisen wir die polynomielle Lösbarkeit von τ-Realisierung für die 64 booleschen Netzklassen, die weder nop noch swap enthalten. Dieses Ergebnis basiert wesentlich auf der Beobachtung, dass hier nur *stark zielorientierte* Transitionssysteme als Eingaben interessant sind, denn andernfalls kann die Eingabe sofort als nicht implementierbar abgelehnt werden.

Die Komplexität der Synthese von (puren) b-gebundenen P/T-Netzen formulieren Schlachter und Wimmel [SW17] als offene Frage. In dieser Dissertation beantworten wir diese Frage: Für alle $b \geq 1$ und alle Implementierungen Einbettung, Sprachsimulation und Realisierung ist das Problem, für ein Transitionssystem A zu entscheiden, ob es durch ein (pures) b-gebundenes P/T-Netzen implementiert werden kann, NP-vollständig.

Insbesondere ist die Synthese purer 1-gebundener P/T-Netze ist für jede der Implementierungen NP-vollständig. (Dies war durch die Ergebnisse zu den elementaren Netz-Systemen bereits bekannt [BBD15, TR18].) Im Kontrast dazu zeigt Schmitt [Sch96], dass die Erweiterung der 1-gebundenen P/T-Netze um die Gruppe $\mathbb{Z}_2$ eine Netzklasse liefert, deren Synthese für alle klassischen Implementierungen polynomiell ist.

In der vorliegenden Arbeit setzen wir Schmitts Ansatz, die (puren) b-gebundenen P/T-Netze um die Gruppe $\mathbb{Z}_{b+1}$ zu erweitern, auf $b \geq 2$ fort. Im Kontrast zu Schmitts Ergebnis für $b = 1$ stellt sich heraus, dass diese Erweiterung im Allgemeinen kein Polynomialzeit-Verfahren für die Synthese der resultierenden Netzklassen liefert: τ-Einbettung ist für die (puren) $\mathbb{Z}_{b+1}$-erweiterten b-gebundenen P/T-Netze für alle $b \geq 1$ in Polynomialzeit entscheidbar; τ-Sprachsimulation und τ-Realisierung sind für die (puren) $\mathbb{Z}_{b+1}$-erweiterten b-gebundenen P/T-Netze für alle $b \geq 2$ NP-vollständig. Nach bestem Wissen des Autors der vorliegenden Arbeit sind dies insbesondere die ersten Netzklassen überhaupt, für welche sich die Komplexität von τ-Einbettung nachweisbar unterscheidet von der Komplexität von τ-Sprachsimulation und τ-Realisierung.

Ein wesentlicher Beitrag der vorliegenden Arbeit ist im Zusammenhang der Charakterisierung der Komplexität von τ-Einbettung und τ-Sprachsimulation sowie τ-Realisierung für die genannten Klassen boolescher und b-beschränkter Netze, dass alle Beweise der NP-Vollständigkeit auf *einem* Reduktionsschema beruhen.

Ebenso untersuchen wir in dieser Dissertation die (parametrisierte) Komplexität der

Frage nach der Existenz eines implementierenden Netzes, das bezüglich seiner Struktur, genauer den Umgebungen seiner Plätze, oder seines Verhaltens, genauer der Anzahl der erreichbaren Markierungen in denen ein Platz markiert sein darf, gewissen Restriktionen genügen soll. Wir zeigen für eine Majorität der booleschen Netze mit nop (98 von 128) und für die P/T-Netze, dass die *Synthese von Netzen mit begrenzten Platzumgebungen* beziehungsweise *Platzbereichen*, je nachdem, NP-vollständig und nicht fest-Parameterberechenbar ist, genauer W[1]-schwer oder W[2]-schwer ist, wenn die begrenzende Anzahl als Parameter gewählt wird. Die Beweise der W[1]- und W[2]-Schwere basieren auf Reduktionstechniken, die als Erweiterung des im ersten Teils der Arbeit eingeführten Schemas interpretiert werden können:

- jedes der reduzierten Probleme fragt für ein Paar $(\mathfrak{U}, M)$ mit Menge $\mathfrak{U}$ und Mengensystem $M = \{M_0, \ldots, M_{m-1}\}$, ob es eine Teilmenge $S \subseteq \mathfrak{U}$ gibt, sodass S bezüglich $M_0, \ldots, M_{m-1}$ gewisse Bedingungen erfüllt (dies trifft auch auf des Problem CM 1 IN 3 3SAT zu); dabei soll die Kardinalität von S im Fall der parametrisierten Probleme noch durch den Parameter λ definierten Restriktionen genügen;
- alle Reduktionen liefern ein Transitionssystem A_τ, in dem die Mengen $M_0, \ldots, M_{m-1}$ und damit $(\mathfrak{U}, M)$ mittels beschrifteter Pfade repräsentiert werden, auf denen die Elemente der Mengen als Ereignisse stattfinden;
- alle Reduktionen binden die Existenz von S an die (passende) Lösbarkeit eines Trennungsproblems α;
- in jedem Fall impliziert die Lösbarkeit von α die Implementierbarkeit des Transitionssystem A_τ.

Das Fazit aus dieser Beobachtung ist deshalb: Ein wesentlicher Beitrag der ersten beiden Teile dieser Dissertation ist, neben den genannten Komplexitätsresultaten, ein in hohem Maße allgemeines Reduktionsverfahren, welches sich auf sehr verschiedene Fragen zur Netzsynthese und auf äußerst unterschiedliche Netztypen sowie auf verschiedene Arten der Komplexität anwenden lässt.

Schließlich untersuchen wir in dieser Arbeit mit *Neubeschriftung*, *Kantenlöschung*, *Ereignislöschung* und *Zustandslöschung* vier natürliche Techniken, ein Transitionssystem zum Zwecke seiner Implementierbarkeit zu modifizieren. Dabei ist in der Literatur insbesondere das Problem der *Neubeschriftung* (*label splitting* in der englischsprachigen Fachliteratur) adressiert und algorithmisch umgesetzt worden [CCK08a, CKLY95, CKLY98, SW20]. Die zugehörigen Entscheidungsprobleme fragen für ein Transitionssystem A und eine Zahl κ, ob sich A gemäß der entsprechenden Technik möglichst geringfügig, das heißt κ „berücksichtigend“, zu einem implementierbaren Transitionssystem B modifizieren lässt.

In [SW20] haben Schlachter und Wimmel gezeigt, das *Neubeschriftung*, die auf eine Einbettung abzielt, für P/T-Netze NP-vollständig ist; die Komplexität von *Neubeschriftung*, die auf eine Sprachsimulation oder Realisierung abzielt, formulieren die Autoren für diese Netzklasse als offene Frage.

In dieser Dissertation beantworten wir diese Frage: Das Problem der *Neubeschriftung* ist für P/T-Netze auch dann NP-vollständig, wenn eine Sprachsimulation oder eine Realisierung angestrebt wird. Die Charakterisierung des Problems für die puren P/T-Netze und die zugehörigen Implementierungen ist Gegenstand zukünftiger Forschung.

Des Weiteren zeigen wir, dass jedes der durch *Kantenlöschung*, *Ereignislöschung* und *Zustandslöschung* implizierten Entscheidungsprobleme für jede der Implementierungen Einbettung, Sprachsimulation und Realisierung NP-vollständig ist, wenn ein P/T-Netz gesucht wird. Darüber hinaus zeigen wir: für pure P/T-Netze ist *Ereignislöschung* NP-vollständig wenn eine Einbettung angestrebt wird und *Zustandslöschung* ist für diese Netze ebenso NP-vollständig, unabhängig davon, welche der klassischen Implementierungen angestrebt wird.

Die Komplexität von *Ereignislöschung*, die auf eine Sprachsimulation oder Realisierung durch ein pures P/T-Netz abzielt, ist Gegenstand zukünftiger Forschung. Alle Komplexitätsresultate zu den Modifikationen basieren auf Reduktionen des Problems Cubic Vertex Cover, deren Technik sich wesentlich von der Technik der ersten beiden Teile dieser Arbeit unterscheidet. Insbesondere verknüpfen diese Reduktionen die Existenz einer Knotenüberdeckung S nicht mit der Lösbarkeit *eines* Trennungsproblems, sondern mit der *Un*lösbarkeit *vieler* Trennungsprobleme, die dafür sorgen, dass die Existenz einer gesuchten Modifizierung eine gesuchte Knotenüberdeckung impliziert.

Für zukünftige Forschung wäre es interessant, die Probleme der Modifikation von Transitionssystemen zum Zwecke der Implementierbarkeit aus Sicht der parametrisierten Komplexität zu untersuchen. Ebenso bleibt für zukünftige Forschung die Komplexität weiterer Modifikationen zu untersuchen, die in der Literatur vorgeschlagen wurden, wie zum Beispiel die *Verfeinerung von Zuständen* [BBD15].

Literaturverzeichnis

[AB09] Sanjeev Arora and Boaz Barak. *Computational Complexity - A Modern Approach*. Cambridge University Press, 2009. URL: `http://www.cambridge.org/catalogue/catalogue.asp?isbn=9780521424264`.

[AP20] Aishah Ahmed and Marta Pietkiewicz-Koutny. Algorithms for the synthesis of elementary net systems with localities. In Michael Köhler-Bußmeier, Ekkart Kindler, and Heiko Rölke, editors, *Proceedings of the International Workshop on Petri Nets and Software Engineering co-located with 41st International Conference on Application and Theory of Petri Nets and Concurrency (PETRI NETS 2020), Paris, France, June 24, 2020 (due to COVID-19: virtual conference)*, volume 2651 of *CEUR Workshop Proceedings*, pages 86–107. CEUR-WS.org, 2020. URL: `http://ceur-ws.org/Vol-2651/paper6.pdf`.

[BBD95] Eric Badouel, Luca Bernardinello, and Philippe Darondeau. Polynomial algorithms for the synthesis of bounded nets. In *TAPSOFT*, volume 915 of *Lecture Notes in Computer Science*, pages 364–378. Springer, 1995. `doi:10.1007/3-540-59293-8\_207`.

[BBD97] Eric Badouel, Luca Bernardinello, and Philippe Darondeau. The synthesis problem for elementary net systems is NP-complete. *Theor. Comput. Sci.*, 186(1-2):107–134, 1997. `doi:10.1016/S0304-3975(96)00219-8`.

[BBD15] Eric Badouel, Luca Bernardinello, and Philippe Darondeau. *Petri Net Synthesis*. Texts in Theoretical Computer Science. An EATCS Series. Springer, 2015. `doi:10.1007/978-3-662-47967-4`.

[BCD02] Eric Badouel, Benoît Caillaud, and Philippe Darondeau. Distributing finite automata through Petri net synthesis. *Formal Asp. Comput.*, 13(6):447–470, 2002. `doi:10.1007/s001650200022`.

[BD95a] Éric Badouel and Philippe Darondeau. Dualities between nets and automata induced by schizophrenic objects. In David H. Pitt, David E. Rydeheard, and Peter T. Johnstone, editors, *Category Theory and Computer Science, 6th International Conference, CTCS '95, Cambridge, UK, August 7-11, 1995,*

Proceedings, volume 953 of *Lecture Notes in Computer Science*, pages 24–43. Springer, 1995. URL: https://doi.org/10.1007/3-540-60164-3_18, doi:10.1007/3-540-60164-3_18.

[BD95b] Eric Badouel and Philippe Darondeau. Trace nets and process automata. *Acta Inf.*, 32(7):647–679, 1995. doi:10.1007/BF01186645.

[BD96a] Eric Badouel and Philippe Darondeau. On the Synthesis of General Petri Nets. Research Report RR-3025, INRIA, 1996. URL: https://hal.inria.fr/inria-00073668.

[BD96b] Éric Badouel and Philippe Darondeau. Theory of regions. In Wolfgang Reisig and Grzegorz Rozenberg, editors, *Lectures on Petri Nets I: Basic Models, Advances in Petri Nets, the volumes are based on the Advanced Course on Petri Nets, held in Dagstuhl, September 1996*, volume 1491 of *Lecture Notes in Computer Science*, pages 529–586. Springer, 1996. URL: https://doi.org/10.1007/3-540-65306-6_22, doi:10.1007/3-540-65306-6_22.

[BD04] Éric Badouel and Philippe Darondeau. The synthesis of petri nets from path-automatic specifications. *Inf. Comput.*, 193(2):117–135, 2004. URL: https://doi.org/10.1016/j.ic.2004.04.004, doi:10.1016/j.ic.2004.04.004.

[BD09] Eike Best and Philippe Darondeau. A decomposition theorem for finite persistent transition systems. *Acta Informatica*, 46(3):237–254, 2009.

[BD15a] Eike Best and Raymond R. Devillers. State space axioms for t-systems. *Acta Informatica*, 52(2-3):133–152, 2015. URL: https://doi.org/10.1007/s00236-015-0219-0, doi:10.1007/s00236-015-0219-0.

[BD15b] Eike Best and Raymond R. Devillers. Synthesis of bounded choice-free petri nets. In Luca Aceto and David de Frutos-Escrig, editors, *26th International Conference on Concurrency Theory, CONCUR 2015, Madrid, Spain, September 1.4, 2015*, volume 42 of *LIPIcs*, pages 128–141. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2015. URL: https://doi.org/10.4230/LIPIcs.CONCUR.2015.128, doi:10.4230/LIPIcs.CONCUR.2015.128.

[BD15c] Eike Best and Raymond R. Devillers. Synthesis of live and bounded persistent systems. *Fundam. Informaticae*, 140(1):39–59, 2015.

[BD17] Eike Best and Raymond R. Devillers. Characterisation of the state spaces of marked graph petri nets. *Inf. Comput.*, 253:399–410, 2017. URL: https://doi.org/10.1016/j.ic.2016.06.006, doi:10.1016/j.ic.2016.06.006.

[BDS18] Eike Best, Raymond R. Devillers, and Uli Schlachter. Bounded choice-free petri net synthesis: algorithmic issues. *Acta Informatica*, 55(7):575–611, 2018. URL: https://doi.org/10.1007/s00236-017-0310-9, doi:10.1007/s00236-017-0310-9.

[Ber93] Luca Bernardinello. Synthesis of net systems. In Marco Ajmone Marsan, editor, *Application and Theory of Petri Nets 1993, 14th International Conference, Chicago, Illinois, USA, June 21-25, 1993, Proceedings*, volume 691 of *Lecture Notes in Computer Science*, pages 89–105. Springer, 1993. URL: https://doi.org/10.1007/3-540-56863-8_42, doi:10.1007/3-540-56863-8_42.

[Bes86] Eike Best. Structure theory of Petri nets: the free choice hiatus. In *Advances in Petri Nets*, volume 254 of *Lecture Notes in Computer Science*, pages 168–205. Springer, 1986. doi:10.1007/BFb0046840.

[BESW16] Eike Best, Evgeny Erofeev, Uli Schlachter, and Harro Wimmel. Characterising petri net solvable binary words. In Fabrice Kordon and Daniel Moldt, editors, *Application and Theory of Petri Nets and Concurrency - 37th International Conference, PETRI NETS 2016, Toruń, Poland, June 19-24, 2016. Proceedings*, volume 9698 of *Lecture Notes in Computer Science*, pages 39–58. Springer, 2016. URL: https://doi.org/10.1007/978-3-319-39086-4_4, doi:10.1007/978-3-319-39086-4_4.

[BHW18] Eike Best, Thomas Hujsa, and Harro Wimmel. Sufficient conditions for the marked graph realisability of labelled transition systems. *Theor. Comput. Sci.*, 750:101–116, 2018. URL: https://doi.org/10.1016/j.tcs.2017.10.006, doi:10.1016/j.tcs.2017.10.006.

[BP06] Nadia Busi and G. Michele Pinna. Characterizing workflow nets using regions. In Viorel Negru, Dana Petcu, Daniela Zaharie, Ajith Abraham, Bruno Buchberger, Alexandru Cicortas, Dorian Gorgan, and Joël Quinqueton, editors, *8th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC 2006), 26-29 September 2006, Timisoara, Romania*, pages 399–406. IEEE Computer Society, 2006. URL: https://doi.org/10.1109/SYNASC.2006.21, doi:10.1109/SYNASC.2006.21.

[BP09] Nadia Busi and G. Michele Pinna. Process discovery and petri nets. *Math. Struct. Comput. Sci.*, 19(6):1091–1124, 2009. URL: https://doi.org/10.1017/S0960129509990132, doi:10.1017/S0960129509990132.

[BS15] Eike Best and Uli Schlachter. Analysis of petri nets and transition systems. In Sophia Knight, Ivan Lanese, Alberto Lluch-Lafuente, and Hugo Torres Vieira, editors, *Proceedings 8th Interaction and Concurrency Ex-*

perience, ICE 2015, Grenoble, France, 4-5th June 2015, volume 189 of *EPTCS*, pages 53–67, 2015. URL: https://doi.org/10.4204/EPTCS.189.6, doi:10.4204/EPTCS.189.6.

[Cai99] Benoît Caillaud. Synet : A synthesizer of distributable bounded petri-nets from finite automata, 1999. URL: https://www.irisa.fr/s4/tools/synet/.

[Car12] Josep Carmona. The label splitting problem. *Trans. Petri Nets Other Model. Concurr.*, 6:1–23, 2012. URL: https://doi.org/10.1007/978-3-642-35179-2_1, doi:10.1007/978-3-642-35179-2_1.

[CCK08a] Josep Carmona, Jordi Cortadella, and Michael Kishinevsky. A region-based algorithm for discovering petri nets from event logs. In Marlon Dumas, Manfred Reichert, and Ming-Chien Shan, editors, *Business Process Management, 6th International Conference, BPM 2008, Milan, Italy, September 2-4, 2008. Proceedings*, volume 5240 of *Lecture Notes in Computer Science*, pages 358–373. Springer, 2008. URL: https://doi.org/10.1007/978-3-540-85758-7_26, doi:10.1007/978-3-540-85758-7_26.

[CCK+08b] Josep Carmona, Jordi Cortadella, Michael Kishinevsky, Alex Kondratyev, Luciano Lavagno, and Alexandre Yakovlev. A symbolic algorithm for the synthesis of bounded petri nets. In Kees M. van Hee and Rüdiger Valk, editors, *Applications and Theory of Petri Nets, 29th International Conference, PETRI NETS 2008, Xi'an, China, June 23-27, 2008. Proceedings*, volume 5062 of *Lecture Notes in Computer Science*, pages 92–111. Springer, 2008. URL: https://doi.org/10.1007/978-3-540-68746-7_10, doi:10.1007/978-3-540-68746-7_10.

[CCK09] Josep Carmona, Jordi Cortadella, and Michael Kishinevsky. Genet: A tool for the synthesis and mining of petri nets. In *Ninth International Conference on Application of Concurrency to System Design, ACSD 2009, Augsburg, Germany, 1-3 July 2009*, pages 181–185. IEEE Computer Society, 2009. URL: https://doi.org/10.1109/ACSD.2009.6, doi:10.1109/ACSD.2009.6.

[CCK10] Josep Carmona, Jordi Cortadella, and Michael Kishinevsky. New region-based algorithms for deriving bounded petri nets. *IEEE Trans. Computers*, 59(3):371–384, 2010. URL: https://doi.org/10.1109/TC.2009.131, doi:10.1109/TC.2009.131.

[CEP95] Allan Cheng, Javier Esparza, and Jens Palsberg. Complexity results for 1-safe nets. *Theor. Comput. Sci.*, 147(1&2):117–136, 1995. URL: https://doi.org/10.1016/0304-3975(94)00231-7, doi:10.1016/0304-3975(94)00231-7.

[CFK+15] Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015. doi:10.1007/978-3-319-21275-3.

[CHEP71] F. Commoner, Anatol W. Holt, Shimon Even, and Amir Pnueli. Marked directed graphs. *J. Comput. Syst. Sci.*, 5(5):511–523, 1971.

[CKK+96] Jordi Cortadella, Michael Kishinevsky, Alex Kondratyev, Luciano Lavagno, and Alex Yakovlev. Complete state encoding based on the theory of regions. In *2nd International Symposium on Advanced Research in Asynchronous Circuits and Systems (ASYNC '96), March 18-21, 1996, Aizu-Wakamatsu, Fukushima, Japan*, pages 36–47. IEEE Computer Society, 1996. URL: https://doi.org/10.1109/ASYNC.1996.494436, doi:10.1109/ASYNC.1996.494436.

[CKK+97a] J. Cortadella, M. Kishinevsky, A. Kondratyev, L. Lavagno, and A. Yakovlev. A region-based theory for state assignment in speed-independent circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 16(8):793–812, Aug 1997. doi:10.1109/43.644602.

[CKK+97b] Jordi Cortadella, Michael Kishinevsky, Alex Kondratyev, Luciano Lavagno, and Alex Yakovlev. Petrify: A tool for manipulating concurrent specifications and synthesis of asynchronous controllers. *IEICE Transactions on Information and Systems*, E80-D:315–325, 03 1997.

[CKK+97c] Jordi Cortadella, Michael Kishinevsky, Alex Kondratyev, Luciano Lavagno, and Alexandre Yakovlev. A region-based theory for state assignment in speed-independent circuits. *IEEE Trans. on CAD of Integrated Circuits and Systems*, 16(8):793–812, 1997. doi:10.1109/43.644602.

[CKK+02] Jordi Cortadella, Michael Kishinevsky, Alex Kondratyev, L. Lavagno, and Alex Yakovlev. *Logic Synthesis of Asynchronous Controllers and Interfaces*, volume 8. Springer, 01 2002. doi:10.1007/978-3-642-55989-1.

[CKLY95] Jordi Cortadella, Michael Kishinevsky, Luciano Lavagno, and Alexandre Yakovlev. Synthesizing petri nets from state-based models. In Richard L. Rudell, editor, *Proceedings of the 1995 IEEE/ACM International Conference on Computer-Aided Design, ICCAD 1995, San Jose, California, USA, November 5-9, 1995*, pages 164–171. IEEE Computer Society / ACM, 1995. URL: https://doi.org/10.1109/ICCAD.1995.480008, doi:10.1109/ICCAD.1995.480008.

[CKLY98] J. Cortadella, M. Kishinevsky, L. Lavagno, and A. Yakovlev. Deriving petri nets from finite transition systems. *IEEE Transactions on Computers*, 47(8):859–882, 1998.

[Dar98] Philippe Darondeau. Deriving unbounded petri nets from formal languages. In Davide Sangiorgi and Robert de Simone, editors, *CONCUR '98: Concurrency Theory, 9th International Conference, Nice, France, September 8-11, 1998, Proceedings*, volume 1466 of *Lecture Notes in Computer Science*, pages 533–548. Springer, 1998. URL: https://doi.org/10.1007/BFb0055646, doi:10.1007/BFb0055646.

[Dar01] Philippe Darondeau. On the petri net realization of context-free graphs. *Theor. Comput. Sci.*, 258(1-2):573–598, 2001. URL: https://doi.org/10.1016/S0304-3975(00)00162-6, doi:10.1016/S0304-3975(00)00162-6.

[Dar03] Philippe Darondeau. Unbounded petri net synthesis. In Jörg Desel, Wolfgang Reisig, and Grzegorz Rozenberg, editors, *Lectures on Concurrency and Petri Nets, Advances in Petri Nets [This tutorial volume originates from the 4th Advanced Course on Petri Nets, ACPN 2003, held in Eichstätt, Germany in September 2003. In addition to lectures given at ACPN 2003, additional chapters have been commissioned]*, volume 3098 of *Lecture Notes in Computer Science*, pages 413–438. Springer, 2003. URL: https://doi.org/10.1007/978-3-540-27755-2_11, doi:10.1007/978-3-540-27755-2_11.

[Dar05] P. Darondeau. Distributed implementations of ramadge-wonham supervisory control with petri nets. In *Proceedings of the 44th IEEE Conference on Decision and Control*, pages 2107–2112, 2005. doi:10.1109/CDC.2005.1582472.

[DE95] Jorg Desel and Javier Esparza. *Free Choice Petri Nets*. Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, 1995. doi:10.1017/CBO9780511526558.

[DEH19] Raymond R. Devillers, Evgeny Erofeev, and Thomas Hujsa. Synthesis of weighted marked graphs from constrained labelled transition systems: A geometric approach. *Trans. Petri Nets Other Model. Concurr.*, 14:172–191, 2019. URL: https://doi.org/10.1007/978-3-662-60651-3_7, doi:10.1007/978-3-662-60651-3_7.

[DEH21] Raymond R. Devillers, Evgeny Erofeev, and Thomas Hujsa. Efficient synthesis of weighted marked graphs with circular reachability graph, and beyond. *Trans. Petri Nets Other Model. Concurr.*, 15:75–100, 2021. URL: https://doi.org/10.1007/978-3-662-63079-2_4, doi:10.1007/978-3-662-63079-2_4.

[DF13] Rodney G. Downey and Michael R. Fellows. *Fundamentals of Parameterized Complexity*. Texts in Computer Science. Springer, 2013. doi:10.1007/978-1-4471-5559-1.

[dFKM18] David de Frutos-Escrig, Maciej Koutny, and Lukasz Mikulski. An efficient characterization of petri net solvable binary words. In Victor Khomenko and Olivier H. Roux, editors, *Application and Theory of Petri Nets and Concurrency - 39th International Conference, PETRI NETS 2018, Bratislava, Slovakia, June 24-29, 2018, Proceedings*, volume 10877 of *Lecture Notes in Computer Science*, pages 207–226. Springer, 2018. URL: `https://doi.org/10.1007/978-3-319-91268-4_11`, `doi:10.1007/978-3-319-91268-4\_11`.

[DH19] Raymond R. Devillers and Thomas Hujsa. Analysis and synthesis of weighted marked graph Petri nets: Exact and approximate methods. *Fundam. Inform.*, 169(1-2):1–30, 2019. URL: `https://doi.org/10.3233/FI-2019-1837`, `doi:10.3233/FI-2019-1837`.

[DR93] Jörg Desel and Wolfgang Reisig. The synthesis problem of petri nets. In Patrice Enjalbert, Alain Finkel, and Klaus W. Wagner, editors, *STACS 93, 10th Annual Symposium on Theoretical Aspects of Computer Science, Würzburg, Germany, February 25-27, 1993, Proceedings*, volume 665 of *Lecture Notes in Computer Science*, pages 120–129. Springer, 1993. URL: `https://doi.org/10.1007/3-540-56503-5_15`, `doi:10.1007/3-540-56503-5\_15`.

[DR96] Jörg Desel and Wolfgang Reisig. The synthesis problem of petri nets. *Acta Informatica*, 33(4):297–315, 1996. URL: `https://doi.org/10.1007/s002360050046`, `doi:10.1007/s002360050046`.

[DR12] Philippe Darondeau and Laurie Ricker. Distributed control of discrete-event systems: A first step. *Trans. Petri Nets Other Model. Concurr.*, 6:24–45, 2012. URL: `https://doi.org/10.1007/978-3-642-35179-2_2`, `doi:10.1007/978-3-642-35179-2\_2`.

[EN94] Javier Esparza and Mogens Nielsen. Decidability issues for Petri nets - a survey. *Bulletin of the EATCS*, 52:244–262, 1994.

[ER90a] Andrzej Ehrenfeucht and Grzegorz Rozenberg. Partial (set) 2-structures. part I: basic notions and the representation problem. *Acta Inf.*, 27(4):315–342, 1990. URL: `https://doi.org/10.1007/BF00264611`, `doi:10.1007/BF00264611`.

[ER90b] Andrzej Ehrenfeucht and Grzegorz Rozenberg. Partial (set) 2-structures. part II: state spaces of concurrent systems. *Acta Informatica*, 27(4):343–368, 1990. URL: `https://doi.org/10.1007/BF00264612`, `doi:10.1007/BF00264612`.

[Ero18] Evgeny Erofeev. *Characterisation of a Class of Petri Net Solvable Transition Systems*. PhD thesis, University of Oldenburg, Germany, 2018.

[Esp96] Javier Esparza. Decidability and complexity of petri net problems - an introduction. In Wolfgang Reisig and Grzegorz Rozenberg, editors, *Lectures on Petri Nets I: Basic Models, Advances in Petri Nets, the volumes are based on the Advanced Course on Petri Nets, held in Dagstuhl, September 1996*, volume 1491 of *Lecture Notes in Computer Science*, pages 374–428. Springer, 1996. URL: https://doi.org/10.1007/3-540-65306-6_20, doi:10.1007/3-540-65306-6_20.

[ET20] Evgeny Erofeev and Ronny Tredup. Occupancy number restricted boolean petri net synthesis: A fixed-parameter algorithm. In Violet Ka I Pun, Volker Stolz, and Adenilso Simão, editors, *Theoretical Aspects of Computing - ICTAC 2020 - 17th International Colloquium, Macau, China, November 30 - December 4, 2020, Proceedings*, volume 12545 of *Lecture Notes in Computer Science*, pages 143–160. Springer, 2020. URL: https://doi.org/10.1007/978-3-030-64276-1_8, doi:10.1007/978-3-030-64276-1_8.

[EW17] Evgeny Erofeev and Harro Wimmel. Reachability graphs of two-transition petri nets. In Wil M. P. van der Aalst, Robin Bergenthum, and Josep Carmona, editors, *Proceedings of the International Workshop on Algorithms & Theories for the Analysis of Event Data 2017 Satellite event of the conferences: 38th International Conference on Application and Theory of Petri Nets and Concurrency Petri Nets 2017 and 17th International Conference on Application of Concurrency to System Design ACSD 2017, Zaragoza, Spain, June 26-27, 2017*, volume 1847 of *CEUR Workshop Proceedings*, pages 39–54. CEUR-WS.org, 2017. URL: http://ceur-ws.org/Vol-1847/paper03.pdf.

[FG06] Jörg Flum and Martin Grohe. *Parameterized Complexity Theory*. Texts in Theoretical Computer Science. An EATCS Series. Springer, 2006. URL: https://doi.org/10.1007/3-540-29953-X, doi:10.1007/3-540-29953-X.

[FHJ98] Gerd Fricke, Stephen T. Hedetniemi, and David Pokrass Jacobs. Independence and irredundance in k-regular graphs. *Ars Comb.*, 49, 1998.

[GJ79] M. R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.

[Gor17] Roberto Gorrieri. *Process Algebras for Petri Nets - The Alphabetization of Distributed Systems*. Monographs in Theoretical Computer Science. An EATCS Series. Springer, 2017. URL: https://doi.org/10.1007/978-3-319-55559-1, doi:10.1007/978-3-319-55559-1.

[GR02] Mikael Goldmann and Alexander Russell. The complexity of solving equations over finite groups. *Inf. Comput.*, 178(1):253–262, 2002. doi:10.1006/inco.2002.3173.

[HDK14] Thomas Hujsa, Jean-Marc Delosme, and Alix Munier Kordon. On the reversibility of well-behaved weighted choice-free systems. In *Petri Nets*, volume 8489 of *Lecture Notes in Computer Science*, pages 334–353. Springer, 2014.

[Hir94] Kunihiko Hiraishi. Some complexity results on transition systems and elementary net systems. *Theor. Comput. Sci.*, 135(2):361–376, 1994. doi:10.1016/0304-3975(94)90112-0.

[HKG97] Lawrence E. Holloway, Bruce H. Krogh, and Alessandro Giua. A survey of Petri net methods for controlled discrete event systems. *Discrete Event Dynamic Systems*, 7(2):151–190, 1997. doi:10.1023/A:1008271916548.

[HKT96] P. W. Hoogers, H. C. M. Kleijn, and P. S. Thiagarajan. An event structure semantics for general petri nets. *Theor. Comput. Sci.*, 153(1&2):129–170, 1996. URL: https://doi.org/10.1016/0304-3975(95)00120-4, doi:10.1016/0304-3975(95)00120-4.

[Hol85] Anatol W. Holt. Coordination technology and petri nets. In *Applications and Theory in Petri Nets*, volume 222 of *Lecture Notes in Computer Science*, pages 278–296. Springer, 1985.

[KKPR13] Jetty Kleijn, Maciej Koutny, Marta Pietkiewicz-Koutny, and Grzegorz Rozenberg. Step semantics of boolean nets. *Acta Inf.*, 50(1):15–39, 2013. doi:10.1007/s00236-012-0170-2.

[KP06] Maciej Koutny and Marta Pietkiewicz-Koutny. Transition systems of elementary net systems with localities. In Christel Baier and Holger Hermanns, editors, *CONCUR 2006 - Concurrency Theory, 17th International Conference, CONCUR 2006, Bonn, Germany, August 27-30, 2006, Proceedings*, volume 4137 of *Lecture Notes in Computer Science*, pages 173–187. Springer, 2006. URL: https://doi.org/10.1007/11817949_12, doi:10.1007/11817949_12.

[KP07] Maciej Koutny and Marta Pietkiewicz-Koutny. Synthesis of elementary net systems with context arcs and localities. In Jetty Kleijn and Alexandre Yakovlev, editors, *Petri Nets and Other Models of Concurrency - ICATPN 2007, 28th International Conference on Applications and Theory of Petri Nets and Other Models of Concurrency, ICATPN 2007, Siedlce, Poland, June 25-29, 2007, Proceedings*, volume 4546 of *Lecture Notes in Computer Science*, pages 281–300. Springer, 2007. URL: https://doi.org/10.1007/978-3-540-73094-1_18, doi:10.1007/978-3-540-73094-1_18.

[LBDM07] Robert Lorenz, Robin Bergenthum, Jörg Desel, and Sebastian Mauser. Synthesis of petri nets from finite partial languages. In Twan Basten, Gabriel Juhás, and Sandeep K. Shukla, editors, *Seventh International Conference*

on Application of Concurrency to System Design (ACSD 2007), 10-13 July 2007, Bratislava, Slovak Republic, pages 157–166. IEEE Computer Society, 2007. URL: http://doi.ieeecomputersociety.org/10.1109/ACSD.2007.72, doi:10.1109/ACSD.2007.72.

[LMB07] Robert Lorenz, Sebastian Mauser, and Robin Bergenthum. Theory of regions for the synthesis of inhibitor nets from scenarios. In Jetty Kleijn and Alexandre Yakovlev, editors, *Petri Nets and Other Models of Concurrency - ICATPN 2007, 28th International Conference on Applications and Theory of Petri Nets and Other Models of Concurrency, ICATPN 2007, Siedlce, Poland, June 25-29, 2007, Proceedings*, volume 4546 of *Lecture Notes in Computer Science*, pages 342–361. Springer, 2007. URL: https://doi.org/10.1007/978-3-540-73094-1_21, doi:10.1007/978-3-540-73094-1_21.

[LMJ08] Robert Lorenz, Sebastian Mauser, and G. Juhas. How to synthesize nets from languages - a survey. In *Proceedings - Winter Simulation Conference*, pages 637–647. IEEE Computer Society, 2008. doi:10.1109/WSC.2007.4419657.

[mil00] Millenium-probleme, 2000. URL: https://de.wikipedia.org/wiki/Millennium-Probleme.

[MR95] Ugo Montanari and Francesca Rossi. Contextual nets. *Acta Inf.*, 32(6):545–596, 1995. doi:10.1007/BF01178907.

[MR01] Cristopher Moore and J. M. Robson. Hard tiling problems with simple tiles. *Discrete & Computational Geometry*, 26(4):573–590, 2001. doi:10.1007/s00454-001-0047-6.

[Muk92] Madhavan Mukund. Petri nets and step transition systems. *Int. J. Found. Comput. Sci.*, 3(4):443–478, 1992. URL: https://doi.org/10.1142/S0129054192000231, doi:10.1142/S0129054192000231.

[Nie06] Rolf Niedermeier. *Invitation to Fixed-Parameter Algorithms*. Oxford University Press, 2006. URL: https://doi.org/10.1093/ACPROF:OSO/9780198566076.001.0001, doi:10.1093/ACPROF:OSO/9780198566076.001.0001.

[NRT92] Mogens Nielsen, Grzegorz Rozenberg, and P. S. Thiagarajan. Elementary transition systems. *Theor. Comput. Sci.*, 96(1):3–33, 1992. URL: https://doi.org/10.1016/0304-3975(92)90180-N, doi:10.1016/0304-3975(92)90180-N.

[Pap94] Christos H. Papadimitriou. *Computational complexity*. Addison-Wesley, 1994.

[Pet73] C. A. Petri. Concepts of net theory. In *MFCS*, pages 137–146. Mathematical Institute of the Slovak Academy of Sciences, 1973.

[Pie97] Marta Pietkiewicz-Koutny. Transition systems of elementary net systems with inhibitor arcs. In *ICATPN*, volume 1248 of *Lecture Notes in Computer Science*, pages 310–327. Springer, 1997. doi:10.1007/3-540-63139-9_43.

[Pie99] Marta Pietkiewicz-Koutny. The synthesis problem for elementary net systems with inhibitor arcs. *Fundam. Informaticae*, 40(2-3):251–283, 1999. URL: https://doi.org/10.3233/FI-1999-402307, doi:10.3233/FI-1999-402307.

[Pie02] Marta Pietkiewicz-Koutny. Synthesising elementary net systems with inhibitor arcs from step transition systems. *Fundam. Informaticae*, 50(2):175–203, 2002. URL: http://content.iospress.com/articles/fundamenta-informaticae/fi50-2-05.

[Raj90] Arvind Rajan. Theory of linear and integer programming, by alexander schrijver, wiley, new york, 1986, 471 pp. price $71.95. *Networks*, 20(6):801, 1990. URL: https://doi.org/10.1002/net.3230200608, doi:10.1002/net.3230200608.

[Sch96] Vincent Schmitt. Flip-flop nets. In *STACS*, volume 1046 of *Lecture Notes in Computer Science*, pages 517–528. Springer, 1996. doi:10.1007/3-540-60922-9_42.

[Sch13] Uli Schlachter. Apt analysis of petri nets and labeled transition systems, 2013. URL: https://github.com/CvO-Theory/apt.

[Sch16] Uli Schlachter. Bounded Petri net synthesis from modal transition systems is undecidable. In Josée Desharnais and Radha Jagadeesan, editors, *27th International Conference on Concurrency Theory, CONCUR 2016, August 23-26, 2016, Québec City, Canada*, volume 59 of *LIPIcs*, pages 15:1–15:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2016. URL: https://doi.org/10.4230/LIPIcs.CONCUR.2016.15, doi:10.4230/LIPIcs.CONCUR.2016.15.

[Sch18] Uli Schlachter. *Petri net synthesis and modal specifications*. PhD thesis, University of Oldenburg, Germany, 2018.

[Sip97] Michael Sipser. *Introduction to the theory of computation*. PWS Publishing Company, 1997.

[Sta90] Peter H. Starke. *Analyse von Petri-Netz-Modellen*. Leitfäden und Monographien der Informatik. Teubner, 1990.

[SW17] Uli Schlachter and Harro Wimmel. k-bounded Petri net synthesis from modal transition systems. In *CONCUR*, volume 85 of *LIPIcs*, pages 6:1–6:15. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017. doi:10.4230/LIPIcs.CONCUR.2017.6.

[SW20] Uli Schlachter and Harro Wimmel. Optimal label splitting for embedding an LTS into an arbitrary Petri net reachability graph is NP-complete. *CoRR*, abs/2002.04841, 2020. URL: https://arxiv.org/abs/2002.04841, arXiv:2002.04841.

[Tar77] Robert Endre Tarjan. Finding optimum branchings. *Networks*, 7(1):25–35, 1977. doi:10.1002/net.3230070103.

[TCCS92] Enrique Teruel, Piotr Chrzastowski-Wachtel, José Manuel Colom, and Manuel Silva Suárez. On weighted t-systems. In *Application and Theory of Petri Nets*, volume 616 of *Lecture Notes in Computer Science*, pages 348–367. Springer, 1992.

[TCS97] Enrique Teruel, José Manuel Colom, and Manuel Silva Suárez. Choice-free petri nets: a model for deterministic concurrent systems with bulk services and arrivals. *IEEE Trans. Systems, Man, and Cybernetics, Part A*, 27(1):73–83, 1997. URL: https://doi.org/10.1109/3468.553226, doi:10.1109/3468.553226.

[TE20a] Ronny Tredup and Evgeny Erofeev. The complexity of boolean state separation. In Violet Ka I Pun, Volker Stolz, and Adenilso Simão, editors, *Theoretical Aspects of Computing - ICTAC 2020 - 17th International Colloquium, Macau, China, November 30 - December 4, 2020, Proceedings*, volume 12545 of *Lecture Notes in Computer Science*, pages 123–142. Springer, 2020. URL: https://doi.org/10.1007/978-3-030-64276-1_7, doi:10.1007/978-3-030-64276-1_7.

[TE20b] Ronny Tredup and Evgeny Erofeev. On the complexity of synthesis of nop-free boolean petri nets. In Wil M. P. van der Aalst, Robin Bergenthum, and Josep Carmona, editors, *Proceedings of the International Workshop on Algorithms & Theories for the Analysis of Event Data 2020 Satellite event of the 41st International Conference on Application and Theory of Petri Nets and Concurrency Petri Nets 2020, virtual workshop, June 24, 2020*, volume 2625 of *CEUR Workshop Proceedings*, pages 66–84. CEUR-WS.org, 2020. URL: http://ceur-ws.org/Vol-2625/paper-05.pdf.

[TE20c] Ronny Tredup and Evgeny Erofeev. On the parameterized complexity of d-restricted boolean net synthesis. In Jianer Chen, Qilong Feng, and Jinhui Xu,

editors, *Theory and Applications of Models of Computation, 16th International Conference, TAMC 2020, Changsha, China, October 18-20, 2020, Proceedings*, volume 12337 of *Lecture Notes in Computer Science*, pages 226–238. Springer, 2020. URL: https://doi.org/10.1007/978-3-030-59267-7_20, doi:10.1007/978-3-030-59267-7_20.

[TE20d] Ronny Tredup and Evgeny Erofeev. On the parameterized complexity of synthesizing boolean petri nets with restricted dependency. In Julien Lange, Anastasia Mavridou, Larisa Safina, and Alceste Scalas, editors, *Proceedings 13th Interaction and Concurrency Experience, ICE 2020, Online, 19 June 2020*, volume 324 of *EPTCS*, pages 78–95, 2020. URL: https://doi.org/10.4204/EPTCS.324.7, doi:10.4204/EPTCS.324.7.

[TE21] Ronny Tredup and Evgeny Erofeev. On the parameterized complexity of the synthesis of boolean nets with restricted place environments. *Theor. Comput. Sci.*, 890:36–69, 2021. URL: https://doi.org/10.1016/j.tcs.2021.08.014, doi:10.1016/j.tcs.2021.08.014.

[TR18] Ronny Tredup and Christian Rosenke. Narrowing down the hardness barrier of synthesizing elementary net systems. In Sven Schewe and Lijun Zhang, editors, *29th International Conference on Concurrency Theory, CONCUR 2018, September 4-7, 2018, Beijing, China*, volume 118 of *LIPIcs*, pages 16:1–16:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018. URL: https://doi.org/10.4230/LIPIcs.CONCUR.2018.16, doi:10.4230/LIPIcs.CONCUR.2018.16.

[TR19a] Ronny Tredup and Christian Rosenke. The complexity of synthesis for 43 boolean petri net types. In T. V. Gopal and Junzo Watada, editors, *Theory and Applications of Models of Computation - 15th Annual Conference, TAMC 2019, Kitakyushu, Japan, April 13-16, 2019, Proceedings*, volume 11436 of *Lecture Notes in Computer Science*, pages 615–634. Springer, 2019. URL: https://doi.org/10.1007/978-3-030-14812-6_38, doi:10.1007/978-3-030-14812-6_38.

[TR19b] Ronny Tredup and Christian Rosenke. On the hardness of synthesizing boolean nets. In Wil M. P. van der Aalst, Robin Bergenthum, and Josep Carmona, editors, *Proceedings of the International Workshop on Algorithms & Theories for the Analysis of Event Data 2019 Satellite event of the conferences: 40th International Conference on Application and Theory of Petri Nets and Concurrency Petri Nets 2019 and 19th International Conference on Application of Concurrency to System Design ACSD 2019, ATAED@Petri Nets/ACSD 2019, Aachen, Germany, June 25, 2019*, volume 2371 of *CEUR Workshop Proceedings*, pages 71–86. CEUR-WS.org, 2019. URL: http://ceur-ws.org/Vol-2371/ATAED2019-71-86.pdf.

[Tre19a] Ronny Tredup. Fixed parameter tractability and polynomial time results for the synthesis of b-bounded petri nets. In Susanna Donatelli and Stefan Haar, editors, *Application and Theory of Petri Nets and Concurrency - 40th International Conference, PETRI NETS 2019, Aachen, Germany, June 23-28, 2019, Proceedings*, volume 11522 of *Lecture Notes in Computer Science*, pages 148–168. Springer, 2019. URL: https://doi.org/10.1007/978-3-030-21571-2_10, doi:10.1007/978-3-030-21571-2_10.

[Tre19b] Ronny Tredup. Hardness results for the synthesis of b-bounded petri nets. In Susanna Donatelli and Stefan Haar, editors, *Application and Theory of Petri Nets and Concurrency - 40th International Conference, PETRI NETS 2019, Aachen, Germany, June 23-28, 2019, Proceedings*, volume 11522 of *Lecture Notes in Computer Science*, pages 127–147. Springer, 2019. URL: https://doi.org/10.1007/978-3-030-21571-2_9, doi:10.1007/978-3-030-21571-2_9.

[Tre19c] Ronny Tredup. Synthesis of structurally restricted b-bounded petri nets: Complexity results. In Emmanuel Filiot, Raphaël M. Jungers, and Igor Potapov, editors, *Reachability Problems - 13th International Conference, RP 2019, Brussels, Belgium, September 11-13, 2019, Proceedings*, volume 11674 of *Lecture Notes in Computer Science*, pages 202–217. Springer, 2019. URL: https://doi.org/10.1007/978-3-030-30806-3_16, doi:10.1007/978-3-030-30806-3_16.

[Tre19d] Ronny Tredup. Tracking down the bad guys: Reset and set make feasibility for flip-flop net derivatives np-complete. In Massimo Bartoletti, Ludovic Henrio, Anastasia Mavridou, and Alceste Scalas, editors, *Proceedings 12th Interaction and Concurrency Experience, ICE 2019, Copenhagen, Denmark, 20-21 June 2019*, volume 304 of *EPTCS*, pages 20–37, 2019. URL: https://doi.org/10.4204/EPTCS.304.2, doi:10.4204/EPTCS.304.2.

[Tre20a] Ronny Tredup. The complexity of the label-splitting-problem for flip-flop-nets. In Sylvain Schmitz and Igor Potapov, editors, *Reachability Problems - 14th International Conference, RP 2020, Paris, France, October 19-21, 2020, Proceedings*, volume 12448 of *Lecture Notes in Computer Science*, pages 148–163. Springer, 2020. URL: https://doi.org/10.1007/978-3-030-61739-4_10, doi:10.1007/978-3-030-61739-4_10.

[Tre20b] Ronny Tredup. Finding an optimal label-splitting to make a transition system petri net implementable: a complete complexity characterization. In Gennaro Cordasco, Luisa Gargano, and Adele A. Rescigno, editors, *Proceedings of the 21st Italian Conference on Theoretical Computer Science, Ischia, Italy, September 14-16, 2020*, volume 2756 of *CEUR Workshop Pro-*

ceedings, pages 131–144. CEUR-WS.org, 2020. URL: http://ceur-ws.org/Vol-2756/paper_18.pdf.

[Tre20c] Ronny Tredup. Parameterized complexity of synthesizing b-bounded (m, n)-t-systems. In Alexander Chatzigeorgiou, Riccardo Dondi, Herodotos Herodotou, Christos A. Kapoutsis, Yannis Manolopoulos, George A. Papadopoulos, and Florian Sikora, editors, *SOFSEM 2020: Theory and Practice of Computer Science - 46th International Conference on Current Trends in Theory and Practice of Informatics, SOFSEM 2020, Limassol, Cyprus, January 20-24, 2020, Proceedings*, volume 12011 of *Lecture Notes in Computer Science*, pages 223–235. Springer, 2020. URL: https://doi.org/10.1007/978-3-030-38919-2_19, doi:10.1007/978-3-030-38919-2_19.

[Tre21a] Ronny Tredup. The complexity of synthesizing nop-equipped boolean petri nets from g-bounded inputs. *Trans. Petri Nets Other Model. Concurr.*, 15:101–125, 2021. URL: https://doi.org/10.1007/978-3-662-63079-2_5, doi:10.1007/978-3-662-63079-2_5.

[Tre21b] Ronny Tredup. Edge, event and state removal: the complexity of some basic techniques that make transition systems petri net implementable. In Didier Buchs and Josep Carmona, editors, *42nd International Conference, PETRI NETS 2021, Virtual Event, June 23 - 25, 2021, Proceedings*, volume 12734 of *Lecture Notes in Computer Science*. Springer, 2021. doi:10.1007/978-3-030-76983-3.

[Tre21c] Ronny Tredup. Synthesis of petri nets with restricted place-environments: Classical and parameterized. In Didier Buchs and Josep Carmona, editors, *42nd International Conference, PETRI NETS 2021, Virtual Event, June 23 - 25, 2021, Proceedings*, volume 12734 of *Lecture Notes in Computer Science*. Springer, 2021. doi:10.1007/978-3-030-76983-3.

[TRW18] Ronny Tredup, Christian Rosenke, and Karsten Wolf. Elementary net synthesis remains np-complete even for extremely simple inputs. In Victor Khomenko and Olivier H. Roux, editors, *Application and Theory of Petri Nets and Concurrency - 39th International Conference, PETRI NETS 2018, Bratislava, Slovakia, June 24-29, 2018, Proceedings*, volume 10877 of *Lecture Notes in Computer Science*, pages 40–59. Springer, 2018. URL: https://doi.org/10.1007/978-3-319-91268-4_3, doi:10.1007/978-3-319-91268-4_3.

[vdA11] Wil M. P. van der Aalst. *Process Mining - Discovery, Conformance and Enhancement of Business Processes*. Springer, 2011. URL: https://doi.org/10.1007/978-3-642-19345-3, doi:10.1007/978-3-642-19345-3.

[vdAWM04] Wil M. P. van der Aalst, Ton Weijters, and Laura Maruster. Workflow mining: Discovering process models from event logs. *IEEE Trans. Knowl. Data Eng.*, 16(9):1128–1142, 2004. URL: https://doi.org/10.1109/TKDE.2004.47, doi:10.1109/TKDE.2004.47.

[Wim08] Harro Wimmel. *Entscheidbarkeit bei Petri Netzen*. eXamen.press. Springer, 2008. doi:10.1007/978-3-540-85471-5.

[Wol19] Karsten Wolf. Synthesis for various petri net classes with union/find. *Fundam. Informaticae*, 169(1-2):57–84, 2019. URL: https://doi.org/10.3233/FI-2019-1839, doi:10.3233/FI-2019-1839.

[WWZ11] ShouGuang Wang, Chengying Wang, and MengChu Zhou. A transformation algorithm for optimal admissible generalized mutual exclusion constraints on petri nets with uncontrollable transitions. In *IEEE International Conference on Robotics and Automation, ICRA 2011, Shanghai, China, 9-13 May 2011*, pages 3745–3750. IEEE, 2011. URL: https://doi.org/10.1109/ICRA.2011.5979727, doi:10.1109/ICRA.2011.5979727.

Glossar

$A = (Z, E, \delta)$ Transitionssystem 15

Z Menge von Zuständen 15

E Menge von Ereignissen 15

δ (partielle) (Zustands-) Übergangsfunktion 15

(z, e, z') eine e beschriftete Kante von z nach z' 15

Δ_A Kantenmenge von A 15

$z \xrightarrow{e}$ z ist Ursprung von e 15

$\xrightarrow{e} z$ z ist Ziel von e 15

$\neg z \xrightarrow{e}$ z ist kein Ursprung von e 15

$z \xrightarrow{e} z'$ Notation für (z, e, z') 15

$z \xleftrightarrow{e} z'$ bidirektionale Kante, zusammenfassend für $z \xrightarrow{e} z'$ und $z' \xrightarrow{e} z$ 15

$z \xrightarrow{\varepsilon} z$ leerer Pfad von z nach z 15

$A = (Z, E, \delta, \iota)$ initialisiertes Transitionssystem 16

ι Initialzustand 16

$Z(A)$ Zustandsmenge von A 16

$E(A)$ Ereignismenge von A 16

$\delta(A)$ partielle Zustandsübergangsfunktion von A 16

$\iota(A)$ Initialzustand von A 16

$L(A)$ Sprache von Transitionssystem A 16

τ Netztyp 17

N Netz, auch τ-Netz 17

P Menge von Plätzen 17

T Menge von Transitionen 17

f Funktion 17

$\mathfrak{m}_0$ Anfangsmarkierung 17

$\mathfrak{m}$ Markierung 17

$\mathfrak{m}[t\rangle$ Transition t ist aktiviert in Markierung $\mathfrak{m}$ 17

$\mathfrak{m}[t\rangle\mathfrak{m}'$ das Schalten von Transition t in Markierung $\mathfrak{m}$ führt zu $\mathfrak{m}'$ 17

$\mathfrak{M}$ Menge aller erreichbarer Markierungen 17

A_N Erreichbarkeitsgraph von N 17

φ Abbildung/Funktion 19, 27

$A \leq B$ B simuliert A 19

$A \hookrightarrow B$ B ist eine Einbettung von A 20

$A \triangleright B$ B ist eine Sprachsimulation von A 20

$A \cong B$ B ist eine Realisierung von A 21

R Region, auch τ-Region 23

sup Support 23

sig Signatur 23

P^R Bild des Pfades P unter Region R 23

$\mathcal{E}^R_{e_\tau}$ Menge der Ereignisse mit Signatur e_τ unter Region R 23

$\mathcal{R}$ Menge von (τ-) Regionen 23

α (Ereignis- oder Zustands-) Trennungsproblem 24

$\tau_0 \equiv \tau_1$ die Netztypen τ_0 und τ_1 sind kongruent 27

ψ Funktion 27

Σ (nicht-leeres, endliches) Alphabet 32

Σ^* Menge aller Wörter über Σ 32

L Sprache 32

χ_L charakteristische Funktion der Sprache L 32

ω ein Eingabewort 32

$\mathcal{A}$ Algorithmus 32

$f_{\mathcal{A}}$ Laufzeitfunktion von Algorithmus $\mathcal{A}$ 32

g Funktion 32, 35

c Konstante 32

P Komplexitätsklasse 32

NP Komplexitätsklasse 32

NPV Komplexitätsklasse 33

f_m^p polynomielle many-one Reduktion 33

$|A|$ Eingabelänge von Transitionssystem A 33

$|Z|$ Anzahl der Zustände 33

$|E|$ Anzahl der Ereignisse 33

κ natürliche Zahl, Parameter 33

FPT Komplexitätsklasse 34

XP Komplexitätsklasse 34

W[i] Komplexitätsklasse 34

h Funktion 35

f_m^{para} (Funktion einer) parametrisierte(n) *many-one* Reduktion 35

$\mathfrak{U}$ endliche Menge (zum Beispiel mit Variablen etc. 38

M Mengensystem 38

M_i Element eines Mengensystems 38

S erfragte Menge eines Entscheidungsproblems z.B. Eins-in-drei-Modell etc. 38

A_τ Transitionssystem aus Reduktion für Netztyp τ 39

$\perp_i$ typischer Zustand zur Verbindung von Gadgets 41

$\perp$ Menge mit (einigen) Zuständen $\perp_0, \perp_1, \ldots, \perp_i$, $i \in \mathbb{N}$ 41

$\oplus_i$ typisches Ereignis zur Verbindung von Gadgets 41

$\oplus$ Menge mit (einigen) Ereignissen $\oplus_0, \oplus_1, \ldots, \oplus_i$, $i \in \mathbb{N}$ 41

$\ominus_{i+1}$ typisches Ereignis zur Verbindung von Gadgets 41

$\ominus$ Menge mit (einigen) Ereignissen $\ominus_0, \ominus_1, \ldots, \ominus_i$, $i \in \mathbb{N}$ 41

nop boolesche Interaktion (Unabhängigkeit) 50

inp boolesche Interaktion (bedingtes Konsumieren) 50

out boolesche Interaktion (bedingtes Produzieren) 50

res boolesche Interaktion (unbedingtes Konsumieren) 50

set boolesche Interaktion (unbedingtes Produzieren) 50

swap boolesche Interaktion (unbedingtes Invertieren) 50

used boolesche Interaktion (Test ob belegt) 50

free boolesche Interaktion (Test ob frei) 50

Q Menge von Zuständen Q und deren charakteristische Funktion 51

S_e^Q Signaturpotential von Q bezüglich e 52

$\mathfrak{consume}$ die Menge $\{\mathsf{inp}, \mathsf{res}, \mathsf{swap}\}$ 53

$\mathfrak{produce}$ die Menge $\{\mathsf{out}, \mathsf{set}, \mathsf{swap}\}$ 53

$\mathfrak{save}_1$ die Menge $\{\mathsf{nop}, \mathsf{set}, \mathsf{used}\}$ 53

$\mathfrak{save}_0$ die Menge $\{\mathsf{nop}, \mathsf{res}, \mathsf{free}\}$ 53

$\mathfrak{save}$ die Menge $\mathfrak{save}_1 \cup \mathfrak{save}_0$ 53

$\mathfrak{inhibit}$ die Menge $\{\mathsf{inp}, \mathsf{out}, \mathsf{used}, \mathsf{free}\}$ 53

$\overline{E}$ die Menge $\{\overline{e} \mid e \in E\}$ der mit E assoziierten Ereignisse 98

τ_{PT}^b Netztyp der b-beschränkten P/T-Netze 116

τ_{PPT}^b Netztyp der puren b-beschränkten P/T-Netze 117

$\tau_{\mathbb{Z}PT}^b$ Netztyp der $\mathbb{Z}_{b+1}$ erweiterten b-beschränkten P/T-Netze 117

$\tau_{\mathbb{Z}PPT}^b$ Netztyp der puren $\mathbb{Z}_{b+1}$ erweiterten b-beschränkten P/T-Netze 117

sig^- 119

sig^+ 119

$|sig|$ 119

$(A_0, \ldots, A_n)$ Basis 121

$\mathbb{Z}_{b+1}^{k \times n}$ $k \times n$ Matrizen über $\mathbb{Z}_{b+1}$ 142

$\mathbb{Z}_{b+1}^k$ Menge der k-Tupel über $\mathbb{Z}_{b+1}$ 142

abs abstrakte Signatur 143

P_z eindeutiger Pfad von ι nach z 143

σ_z Parikh-Vektor des Pfades P_z 144

σ_t Fundamentalkreis der Sehne t 144

$\mathbf{x} \cdot \mathbf{y}$ Produktsumme 144

δ_{must} 151

δ_{may} 151

${}^\bullet p^\bullet$ (boolesche) Platzumgebung 154

${}^\bullet R^\bullet$ Umgebung von Region R 155

τ_{PT} Netztyp der P/T-Netze 201

τ_{PPT} Netztyp der puren P/T-Netze 201

$f^-(p,t)$ 202

${}^\bullet p$ Vorbereich von Platz p 202

$p^\bullet$ Nachbereich von Platz p 202

${}^\bullet p^\bullet$ Bereich von Platz p (bei P/T-Netzen) 202

ϱ_p Belegungszahl des Platzes p 218

ϱ_N Belegungszahl des Netzes N 218

ϱ_R Belegungszahl der Region R 219

Sup ein Algorithmus 224

(B, ϱ, φ) Neubeschriftung 235

$\mathfrak{E}$ Menge der geteilten/gelöschten Ereignisse 235, 251

$deg(z)$ Grad von Zustand z 237

$deg(A)$ Grad von Transitionssystem A 237

$\mathfrak{K}$ Menge der gelöschten Kanten 250

$\mathfrak{Z}$ Menge der gelöschten Zustände 251

Index

(ϱ, κ)-begrenzte τ-Region, 204
(ϱ, κ)-begrenzter Platzbereich, 202
Q-maximal, 86
τ-Einbettung, 21
τ-Implementierungen, 21
τ-Realisierung, 21
τ-Region, 23
τ-Sprachsimulation, 21
$\mathsf{W}[i]$-schwer, 35
b-beschränkte P/T-Netze, 116
b-beschränkte gruppenerweiterte P/T-Netze, 117
b-beschränkte gruppenerweiterte pure P/T-Netze, 117
b-beschränkte pure P/T-Netze, 117
d-begrenzt, 154
d-beschränkte Region, 155
k-konsistent, 124
slice-wise-polynomiell, 34
τ-EMBBZ, 220
τ-Einbettung mit beschränkter Belegungszahl, 220
τ-Einbettung, 38
τ-RMBBZ, 219
τ-RMBPB, 204
τ-Realisierung mit begrenzten Platz-Bereichen, 204
τ-Realisierung mit beschränkter Belegungszahl, 219
τ-Realisierung, 38
τ-SMBBZ, 219
τ-SMBPB, 204
τ-Sprachsimulation mit begrenzten Platz-Bereichen, 204
τ-Sprachsimulation mit beschränkter Belegungszahl, 219
τ-Sprachsimulation, 38
Bτ-RMBP, 156
Bτ-SMBP, 156
Boolesche τ-Realisierung mit begrenzter Platzumgebung, 156
Boolesche τ-Sprachsimulation mit begrenzter Platzumgebung, 156
CM 1 in 3 3Sat, 38
CVC, 239
Cubic Monotone 1 in 3 3Sat, 38
Cubic Vertex Cover, 239
Ereignislöschung für τ-Einbettung, 259
Ereignislöschung für τ-Realisierung, 259
Ereignislöschung für τ-Sprachsimulation, 259
HS, 186
Hitting Set, 186
Kantenlöschung für τ-Einbettung, 253
Kantenlöschung für τ-Realisierung, 253
Kantenlöschung für τ-Sprachsimulation, 253
Neubeschriftung für τ-Einbettung, 237

Neubeschriftung für τ-Realisierung, 237
Neubeschriftung für τ-Sprachsimulation, 237
OS, 172
Odd Set, 172
RIS, 159
Regular Independent Set, 159
Zustandslöschung für τ-Einbettung, 264
Zustandslöschung für τ-Realisierung, 264
Zustandslöschung für τ-Sprachsimulation, 264
Modale Synthese, 151
Sprachäquivalenz, 150

abstrakte Region, 143
Aktivierung (von Transitionen), 17
Anfangsmarkierung, 17
assoziierte(s) Ereignis(se), 98

Basis, 121
Belegungszahl einer Region, 219
Belegungszahl eines Netzes, 218
Belegungszahl eines Platzes, 218
Bereich, 202
bezeugt, 25
Bild eines Pfades, 23
boolesche Interaktion, 50
boolescher Netztyp, 49
boolesches Netz, 49

dünn verteilt, 124

Effekt, 202
Einbettung, 20
eindeutiges Ereignis, 90
eindeutiges Transitionssystem, 90
eindeutiges Ziel, 81
Eins-in-drei-Modell, 38
einseitige Schleifen-Erweiterung, 98
Endzustand eines Pfades, 15
Entscheidungsproblem, 32
Ereignis, 15
Ereignis findet statt, 15
Ereignis-Zustands-Trennbarkeits-Eigenschaft, 25
Ereignis-Zustands-Trennungs-Problem, 25
Ereignisbasis, 121
Ereignislöschung, 250
Ereignisteilung, 235
Erreichbarkeitsgraph, 17
EZTE, 25
EZTP, 25

fest-Parameter-berechenbar, 34
Fluss, 17
Fundamentalkreis, 144

Grad eines Transitionssystems, 237
Grad eines Zustands, 237
Grundannahme, 221

implizit definierte τ-Region, 23
induzierte Region, 146
initialisierter Pfad, 16
initialisiertes Transitionssystem, 16
Initialzustand, 16

Kante, 15
Kantenlöschung, 250
Kantenmenge, 15
Kettenverbindung, 136
klassische Syntheseprobleme, 37
Komplexitätsklasse $\mathsf{W}[i]$, 34
Komplexitätsklasse FPT, 34
Komplexitätsklasse XP, 34
Komplexitätsklasse NP, 32
Komplexitätsklasse NPV, 33
Komplexitätsklasse P, 32
Konflikt (von Kanten), 222
konfliktfrei, 222
kongruent (Netztypen), 27
Kongruenz (zwischen Netztypen), 27

konsistent, 124

Laufzeit, 32
leerer Pfad, 15
lineare Verbindung, 122
lineares Transitionssystem, 31

Millenium-Problem, 32
minimale Region, 221
modales Transitionssystem, 151

Nachbereich, 202
Netz (vom Typ τ), 17
Netztyp, 17
Neubeschriftung, 235
normalisierte Region, 93

P/T-Netze, 201
Parameter, 33
parametrisierte (many one) Reduktion, 35
parametrisiertes Problem, 33
Parikh-Vektor, 144
Pfad, 15
Platz, 17
Platz-Umgebung, 154
Platz-Umgebungszahl, 154
polynomielle (many-one) Reduktion, 33
Produktsumme, 144
pure P/T-Netze, 201

Rückwärts-Erweiterung, 98
Realisierung, 21
relevanter Pfad, 177
Rumpf, 121

Schalten (von Transitionen), 17
Schleife, 15
Schleifen-Erweiterung, 99
schleifenfrei, 15
Sehne, 143
Signatur, 23
Signaturpotential, 52
Simulation, 19
Spannbaum, 143
Sprache (eines Transitionssystems), 16
Sprachsimulation, 20
stark zielorientiert, 81
Startzustand eines Pfades, 15
Support, 23
synthetisiertes Netz, 23

Terminalzustand, 31
Transition, 17
Transitionssystem, 15

Übergangsfunktion, 15
Umgebung einer Region, 155
Unabhängigkeit (zwischen Platz und Transition), 17
Ursprung, 15

Vorbereich, 202

Zeuge, 25
Ziel, 15
ZTE, 24
ZTP, 24
ZTP und EZTP von Basen, 121
Zustand, 15
Zustands-Trennbarkeits-Eigenschaft, 24
Zustands-Trennungs-Problem, 24
Zustandsbasis, 121
Zustandslöschung, 251

Selbstständigkeitserklärung

Hiermit versichere ich, dass ich die vorliegende Arbeit selbstständig und ohne fremde Hilfe verfasst habe, andere als die von mir angegebenen Quellen und Hilfsmittel nicht verwendet habe und die den benutzten Werken wörtlich oder inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Ronny Tredup — Rostock, den 8. November 2021

Lebenslauf

Ronny Tredup
geboren am 24. April 1983 in Wolfen

08/2006 – 07/2009 Abendgymnasium Rostock, Abschluss: Abitur (Note: 1.4)

10/2009 – 07/2011 Universität Rostock, Studiengang Lehramt Gymnasium, Hauptfächer Mathematik und Deutsch

10/2011 – 12/2015 Universität Rostock, Studiengang Lehramt Gymnasium, Hauptfächer Mathematik und Informatik, Nebenfach Deutsch, Abschluss: 1. Staatsexamen (Note: 1.2)

01/2016 – 10/2021 Universität Rostock, IEF, Institut für Informatik, wissenschaftlicher Mitarbeiter am Lehrstuhl für Theoretische Informatik, Promotion zum Thema „Zur Komplexität der Synthese von Petri-Netzen“ (Note: summa cum laude)