



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Bachelorarbeit

Florian Reiter

Modellierung und Analyse von Szenarien des Living
Place mit rekonfigurierbaren Petrinetzen

Florian Reiter

Modellierung und Analyse von Szenarien des Living Place mit rekonfigurierbaren Petrinetzen

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung

im Studiengang Angewandte Informatik
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuende Prüferin : Prof. Dr. Julia Padberg
Zweitgutachter : Prof. Dr. rer. nat. Kai von Luck

Abgegeben am 11.07.2012

Florian Reiter

Thema der Bachelorarbeit

Modellierung und Analyse von Szenarien des Living Place mit rekonfigurierbaren Petrinetzen

Stichworte

Rekonfigurierbare Petrinetze, dekorierte S/T-Netze, intelligente Wohnung, Living Place Hamburg, Modellierung, Szenarien

Kurzzusammenfassung

Diese Arbeit befasst sich mit der Modellierung von kontextabhängigen Szenarien in einer intelligenten Wohnung, dem Living Place Hamburg. Mit rekonfigurierbaren Petrinetzen, einer Erweiterung der klassischen Petrinetze, werden verschiedene morgendliche Abläufe eines Bewohners modelliert. Dabei steht die Interaktion zwischen dem Bewohner und der intelligenten Wohnung im Vordergrund. Anhand dieser Netzmodelle werden einige Ansätze zu einer möglichen Analyse ausgearbeitet.

Florian Reiter

Title of the paper

Modeling and analysis of scenarios of the living place using reconfigurable petri nets

Keywords

Reconfigurable Petri Nets, Decorated P/T Nets, Smart Home, Living Place Hamburg, Modeling, Scenarios

Abstract

This thesis concerns the modeling of context-dependent scenarios in a smart home, i.e. the Living Place Hamburg. Reconfigurable petri nets – an extension of classic petri nets – are used to model different matutinal routines of a resident. Priority is given to the interaction of the resident and the smart home. On the basis of these net models suggestions for a possible analysis are presented.

Inhaltsverzeichnis

Abbildungsverzeichnis	6
1 Einleitung	7
1.1 Motivation	7
1.2 Gliederung	8
2 Petrinetze	9
2.1 Einordnung	9
2.2 Graphische Einführung	10
2.3 Klassische Darstellung	13
2.4 Petrinetz Eigenschaften	14
3 Rekonfigurierbare Petrinetze	20
3.1 Einführung	20
3.2 Universell-algebraische Darstellung	21
3.3 Dekorierte S/T-Netze	22
3.4 Netztransformationen	25
3.5 Fazit	32
4 Modellierung der Szenarien	33
4.1 Living Place	33
4.1.1 Living Place Hamburg	33
4.1.2 Beschreibung der Szenarien	36
4.2 Modellierung	38

4.2.1	Prinzipien der Modellierung.....	38
4.3	Modellierung der Szenarien.....	45
4.3.1	Grundlegende Modellierungsentscheidungen	45
4.3.2	Szenario A: "kurz, schnell (Arbeitstag)"	48
4.3.3	Modellierung von Szenario A	48
4.3.4	Szenario B: "ausgiebig (Arbeitstag)"	54
4.3.5	Modellierung von Szenario B	55
4.3.6	Szenario C: "ausgiebig (Wochenende)"	60
4.3.7	Modellierung von Szenario C	61
4.4	Bewertung der Modellierung.....	65
5	Analysemöglichkeiten.....	67
5.1	Einführung.....	67
5.2	Konfluenz.....	68
5.3	Erreichbarkeitsanalyse	72
5.4	Statischer Gesamtüberblick	76
6	Schluss	78
6.1	Zusammenfassung	78
6.2	Ausblick	79
	Literaturverzeichnis	80
	Anhang.....	84
A	Regeln	85
A.1	Übersicht aller Regeln	85
A.2	Satz aller Regeln (exkl. Löschregeln) aus Szenario A.....	86
B	Netze.....	96
B.1	Alle Netze aus Szenario A.....	96
C	Inhalt der CD	108

Abbildungsverzeichnis

2.1	Brücke über Fluss.....	10
2.2	Petrinetz "Eine Brücke"	10
2.3	Vor- und Nachbereich einer Transition	11
2.4	Petrinetz "Eine Brücke - SAV"	12
2.5	Petrinetz "Eine Brücke" (vgl. Abbildung 2.2) - farblich ergänzt.....	14
2.6	Ausschnitt vom Erreichbarkeitsgraphen des Netzes "Eine Brücke"	15
2.7	Petrinetz "Zwei Brücken".....	16
2.8	Petrinetz "Zwei Brücken - Zwei Autos".....	17
2.9	Petrinetz "Zwei Brücken - Ticketvergabe"	18
3.1	Petrinetz "Eine Brücke - Gefärbte Labels"	23
3.2	Veränderung der Transitionsfarbe	23
3.3	Strukturbewahrende, dekorierte S/T-Netz Morphismen.....	26
3.4	Transformationsprozess "Brückenveränderung"	27
3.5	Die Idee der Netzersetzungsregel aus [Padberg (2011)].....	28
3.6	Regel zur Veränderung der Brücke.....	29
3.7	Zusammenhang der Regelkomponenten aus [Padberg (2011)].....	29
3.8	Zusammenhang der Regelkomponenten inklusive NAC	30
3.9	Negative Anwendungsbedingung bei Veränderung der Brücke	31
4.1	Living Place Hamburg aus [Ellenberg u.a. (2011)]	34
4.2	Architektur des Living Place aus [Ellenberg u.a. (2011)]	35
4.3	Aktionsraum im Living Place.....	37
4.4	Teilnetzvergleich.....	42
4.5	Teil des statischen Gesamtüberblicks.....	44
4.6	Basisnetz NO aller Szenarien	46
4.7	Regel R3	49
4.8	Netz NA5.....	50
4.9	Regel R20	51
4.10	Netz NA22.....	53
4.11	Regel R4	55
4.12	Löschregel LöRe39.....	56
4.13	Ausschnitt aus Netz NB7.....	57
4.14	Ausschnitt aus Netz NB15.....	58
4.15	Ausschnitt aus Netz NB26.....	59
4.16	Löschregel LöRe36.....	61
4.17	Vergleich der Netze NC5 und Netz NC13.....	62

4.18	Netzausschnitte von Netz NC40 - NC44	64
4.19	Netz NC51	65
5.1	Lokale Konfluenz.....	70
5.2	Konfluenz bei mehreren Schalt- und Regelschritten.....	71
5.3	Globale Konfluenz (Skizze).....	71
5.4	Iterative Veränderung der Netzstruktur (Auszug aus Szenario A)	73
5.5	Hypothetischer Ansatz eines Erreichbarkeitsgraphen	73
5.6	Abstrakter Erreichbarkeitsgraph aus [Padberg (2012)].....	74
5.7	Ein möglicher Pfad (=Szenario).....	75
5.8	Statischer Gesamtüberblick aller anwendbaren Regeln	77

1 Einleitung

1.1 Motivation

In den 50 Jahren ihres Bestehens haben sich Petrinetze sowohl in der Theorie als auch in der Praxis bewährt [vgl. [Priese und Wimmel \(2008\)](#)]. Bis heute entwickelte sich diese Netztheorie zu einem bedeutsamen Entwurfs- und Darstellungsinstrument für verteilte Systeme und nebenläufige Prozesse. Die zunehmende Bedeutung solcher Systeme in Anwendungen aller Art hat aufgrund der Komplexität ihres Verhaltens die Entwicklung der Petrinetze beeinflusst. Dies ist dadurch begründet, dass Petrinetze besonders zur Analyse und Verifikation verteilter und nebenläufiger Systeme geeignet sind.

Rekonfigurierbare Petrinetze stellen eine Erweiterung der Petrinetze dar. Sie ergänzen die Petrinetze um einen Satz von Regeln, mit denen das Netz verändert werden kann. Dies erfolgt durch den Austausch von Teilnetzen. Charakteristisches Merkmal sind dynamische Veränderungen, die sich während des sogenannten Tokenspiels ergeben. Aufgrund dieser dynamischen Anpassungen dienen rekonfigurierbare Petrinetze vorzugsweise der Modellierung nebenläufiger Systeme, die sich selbst dynamisch verändern.

Das „Living Place Hamburg“ [vgl. [von Luck u.a. \(2010\)](#)] ist ein Projekt der HAW Hamburg, in dem erforscht werden soll, inwieweit das Leben eines Menschen innerhalb einer Wohnung durch Computer unterstützt werden kann. Im Rahmen dieses Projektes gibt es viele Forschungsansätze, die auf der Basis von intelligenten, kontextabhängigen Szenarien entwickelt werden. Die Interaktion zwischen dem Bewohner und der Wohnung ist bei der Entwicklung dieser Szenarien häufig von Bedeutung. Auf welche Art und Weise können solche komplexen Szenarien beschrieben werden?

Rekonfigurierbare Petrinetze bieten sich zur Beschreibung dieser Szenarien an. Sie ermöglichen mit ihrer Dynamik eine permanente Anpassung der Netzstruktur an den vorliegenden Kontext. So kann eine Unterscheidung von verschiedenen Szenarien gemacht werden. Im Vergleich dazu wäre eine Modellierung mit klassischen Petrinetzen zu statisch, da alle Elemente in einem Netz vereint werden müssten. Die explizite Unterscheidung von Szenarien würde demnach nicht erfolgen.

In dieser Arbeit wird der Versuch unternommen, eine Modellierung von ausgewählten Szenarien aus dem Living Place Hamburg mit rekonfigurierbaren Petrinetzen zu entwickeln. Dabei soll anhand bestimmter Modellierungsprinzipien die Umsetzung erklärt und die Szenarien schrittweise mittels graphischer Darstellungen beschrieben werden. Weiterhin werden Möglichkeiten zur Analyse dieser Netzmodelle aufgeführt.

1.2 Gliederung

Zunächst beschäftigt sich diese Arbeit mit den klassischen *Petrinetzen*. Es werden die zum Verständnis dieser Arbeit notwendigen Grundlagen dieser Netztheorie anhand graphischer Beispiele anschaulich beschrieben. Nach einer formalen Beschreibung, werden einige Begriffe und Eigenschaften der Petrinetze erläutert.

Im anschließenden Kapitel *Rekonfigurierbare Petrinetze* wird anfangs eine weitere Darstellungsart der S/T-Netze eingeführt, die universell-algebraische Darstellung. Es folgen dekorierte S/T-Netze, die eine Erweiterung der klassischen S/T-Netze sind. Basierend auf diesen Netzen werden im nächsten Abschnitt Netztransformationen vorgestellt, welche regelbasiert angewendet werden und die spezifische Dynamik der RKPn ausmachen. Eine abschließende Definition der RKPn komplettiert dieses Kapitel und schließt die Grundlagenbeschreibung ab.

In der *Modellierung der RKPn aus den Szenarien* wird zunächst das Forschungsprojekt „Living Place Hamburg“ der HAW Hamburg vorgestellt. Es folgt eine Beschreibung der Szenarien, auf denen die Modellierung basiert. Bevor drei Szenarien vorgestellt und mit den rekonfigurierbaren Petrinetzen konkret modelliert werden, sollen einige Prinzipien dieser Modellierung beschrieben werden. Das Kapitel endet mit einer kurzen Bewertung der Modellierung.

In Kapitel 5 werden *Analysemöglichkeiten* aufgeführt, die für eine Untersuchung der rekonfigurierbaren Petrinetze anwendbar sind oder potenziell möglich wären. In der ersten Hälfte wird auf Konfluenz eingegangen, während sich die zweite Hälfte mit dem umfassenden Bereich der Erreichbarkeitsanalyse beschäftigt.

Zum *Schluss* werden alle Erkenntnisse dieser Arbeit zusammengefasst. Letztlich erfolgt ein Ausblick auf mögliche weitere Forschungs- und Anwendungsrichtungen der rekonfigurierbaren Petrinetze.

2 Petrinetze

Die ausgewählten Szenarien des „Living Place Hamburg“ sollen mit rekonfigurierbaren Petrinetzen modelliert werden. Vorab werden jedoch zunächst die klassischen Petrinetze erläutert, denn sie bilden die Basis der rekonfigurierbaren Petrinetze.

Nach einer kurzen Einordnung der verschiedenen Petrinetzarten folgt eine graphische Beschreibung der Netze. Hier werden die einzelnen Komponenten eingeführt und anschließend formal beschrieben. Abschließen wird dieses Kapitel eine Erläuterung spezifischer Eigenschaften der Petrinetze.

2.1 Einordnung

Petrinetze wurden 1962 von C.A. Petri in seiner Dissertation „*Kommunikation mit Automaten*“ [vgl. [Petri \(1962\)](#)] eingeführt. Im Laufe der weiteren Jahre gab es viele Arbeiten zur weiteren Entwicklung und Anwendung dieser Netztheorie. Grundlegend kann zwischen zwei Arten von Petrinetzen unterschieden werden. Es gibt Low-Level-Netze, wie z.B. Bedingungs-Ereignis-Netze und Stellen-Transitions-Netze, die vor allem in den ersten Jahren konstruiert wurden. Die sogenannten High-Level-Netze, z.B. Gefärbte Petrinetze und Petrinetze mit individuellen Token, sind zwar formal aufwendiger, können aber modellierte Systeme genauer beschreiben.

Diese Arbeit befasst sich ausschließlich mit den Stellen-Transitions-Netzen (im Folgenden *S/T-Netze*). Sie ähneln der Konzeption von Automaten, unterscheiden sich aber in einem elementaren Punkt. Während bei Automaten nur Zustände als Knoten abgebildet werden, können *S/T-Netze* zusätzlich auch die Übergänge zwischen den Zuständen als explizite Knoten darstellen. Diese Erweiterung gibt den Petrinetzen ihre wohl wichtigste Eigenschaft. Sie können nebenläufige und verteilte Systeme modellieren und mathematisch spezifizieren. Daneben können sie auch graphisch repräsentiert werden. Ihre Modellierungsmächtigkeit macht Petrinetze geeignet zum Einsatz auf den verschiedensten Gebieten [vgl. [Baumgarten \(1990\)](#)].

2.2 Graphische Einführung

Anhand der klassischen, graphischen Darstellung sollen nun die Petrinetzkomponenten eingeführt werden. Dabei soll ein kleines Beispiel helfen, die Begriffe einfach und leicht verständlich zu erklären.

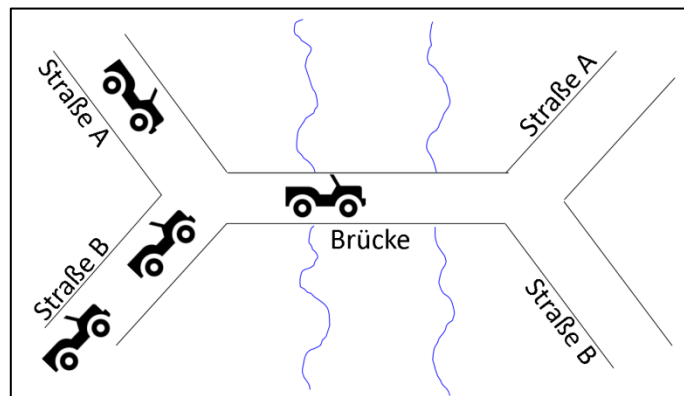


Abbildung 2.1: Brücke über Fluss

Die Skizze in Abbildung 2.1 zeigt eine Brücke, die über einen Fluss führt. Zwei Straßen führen zu dieser Brücke. Auf jeder Straße kommen jeweils zwei Autos zur Brücke gefahren. Die Brücke darf nur von einem Auto überquert werden. Erst wenn das Auto auf der anderen Seite angekommen ist und seine Fahrt auf einer der Straßen fortführt, darf das nächste Auto auf die Brücke fahren. Diese Situation soll nun mit einem Petrinetz dargestellt werden. Das Netz „Eine Brücke“ in Abbildung 2.2 zeigt eine mögliche Variante.

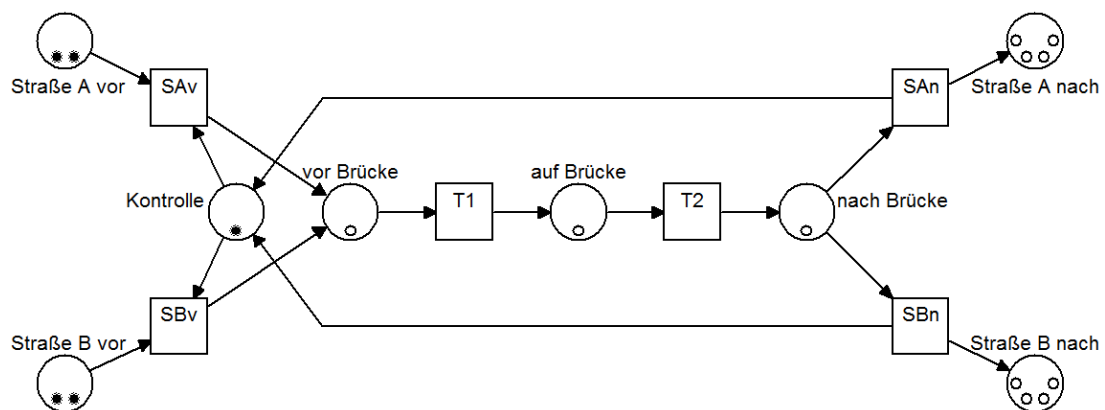


Abbildung 2.2: Petrinetz "Eine Brücke"¹

¹ Alle Netze dieser Arbeit sind mit dem Petrinetz-Tool *Netlab* konstruiert worden. URL: <http://www.irt.rwth-aachen.de/index.php?id=101> (zuletzt zugegriffen: 10.07.2012)

Ein Petrinetz ist ein gerichteter Graph. Wie bereits in der Einführung erwähnt wurde, können Petrinetze Zustände und Zustandsübergänge separat abbilden. Zustände heißen *Stellen* und werden durch Kreise dargestellt. Die Rechtecke repräsentieren die Aktionen und werden als *Transitionen* bezeichnet. Stellen und Transitionen werden durch gerichtete *Kanten* verbunden. Dabei ist zu beachten, dass entweder eine Stelle mit einer Transition oder eine Transition mit einer Stelle verbunden werden darf. Knoten vom gleichen Typ dürfen nicht durch eine Kante verknüpft werden. Außerdem darf es nur eine Kante zwischen zwei Knoten unterschiedlichen Typs geben. Die Kanten können zusätzlich noch eine Gewichtung bekommen. Zu diesem Zweck wird die Kante mit einer natürlichen Zahl gekennzeichnet. Hat eine Kante ein Gewicht gleich eins, dann kann man nach allgemein üblicher Konvention die Angabe der Gewichtung weglassen.

Stellen können durch Marken, den sogenannten *Token*, belegt werden. Die Anzahl an Token, die eine Stelle aufnehmen kann, wird durch die *Kapazität* der Stelle begrenzt. Die Kapazität kann einen Wert zwischen eins und unendlich haben. Eine Verteilung der Token auf alle Stellen wird *Markierung* des S/T-Netzes genannt. Im Beispiel „Eine Brücke“ haben die Stellen „Straße A vor“ und „Straße B vor“ jeweils zwei Token und die Stelle „Kontrolle“ ein Token. Dies ist die Markierung des Netzes und weil sie die Anfangssituation wiedergibt, wird sie auch als *Anfangsmarkierung* bezeichnet.

Transitionen sind die aktiven Komponenten der S/T-Netze. Durch sie lassen sich die Netze verändern. Eine Transition hat einen *Vorbereich* und einen *Nachbereich*. Als Vorbereich werden alle Stellen bezeichnet, die eine Kante zur Transition hin haben. Entsprechend werden alle Stellen als Nachbereich bezeichnet, die von der Transition weg führen. Die Transition „SAv“ in Abbildung 2.3 hat zwei Stellen im Vorbereich (rot markiert) und eine Stelle im Nachbereich (blau markiert).

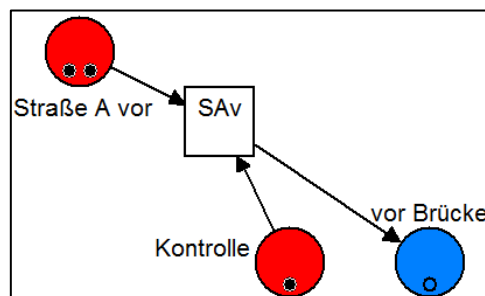


Abbildung 2.3: Vor- und Nachbereich einer Transition

Nachdem die Syntax der S/T-Netze erläutert wurde, soll nun das Schaltverhalten beschrieben werden. Wie bereits erwähnt wurde, sind Transitionen die aktiven Komponenten in S/T-Netzen. Sie können die Markierung des Netzes verändern und die Token auf den Stellen „wandern“ lassen. Auch hier soll das Brückenbeispiel genutzt werden, um diesen Sachverhalt zu verdeutlichen.

Eine Transition kann *schalten*, wenn auf allen Stellen im Vorbereich mindestens so viele Token liegen, wie die Gewichtung der jeweiligen Kante angibt. Zusätzlich müssen alle Stellen im Nachbereich so viele freie Kapazitäten haben, wie die ausgehenden Kanten der Transition angeben. Die Transition wird als *aktiviert* bezeichnet, wenn diese Konstellation zutrifft. Im Ausschnitt des Beispiels „Eine Brücke“ (siehe Abbildung 2.3) hat die Transition „SAv“ zwei Stellen im Vorbereich und beide Kanten die zur Transition führen haben ein Gewicht von eins. Da auf der Stelle „Straße A vor“ zwei Token liegen und auf der Stelle „Kontrolle“ ein Token liegt, sind die Vorbereichsbedingungen erfüllt. Im Nachbereich von „SAv“ gibt es die Stelle „vor Brücke“, zu der auch eine eins-gewichtete Kante führt und die eine freie Kapazität hat. Auch der Nachbereich entspricht den Schaltbedingungen. Die Transition „SAv“ ist somit *aktiviert* und kann *schalten*.

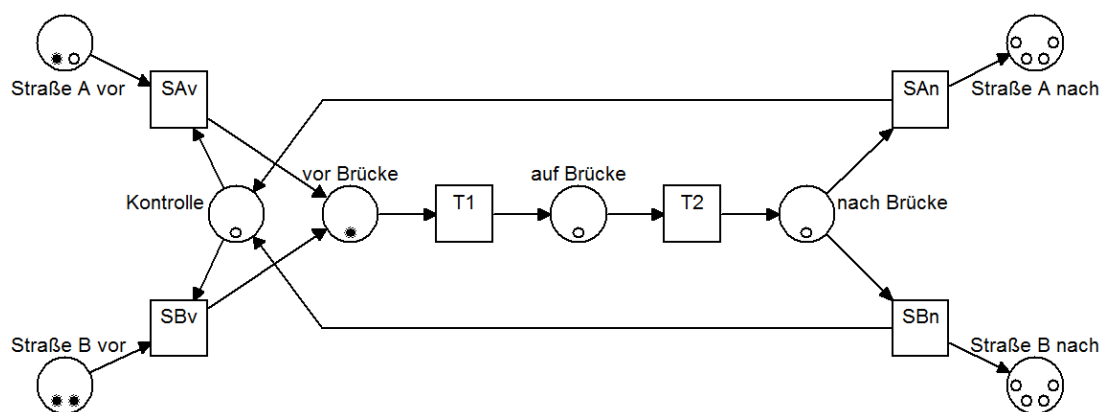


Abbildung 2.4: Petrinetz "Eine Brücke - SAV"

Das Schalten einer Transition hat folgende Auswirkung. Die Token aller Stellen im Vorbereich werden entfernt und bei allen Stellen im Nachbereich werden Token hinzugefügt. Die Anzahl der zu entfernenden Token wird durch die Gewichtung der jeweiligen Kante vorgegeben. Entsprechendes gilt beim Hinzufügen von Token, dort spielt die Kantengewichtung dieselbe Rolle. Das mehrmalige Schalten von Transitionen und das daraus resultierende „Wandern“ der Token werden auch als *Tokenspiel* bezeichnet. In Abbildung 2.4 soll das Resultat eines Schaltvorgangs gezeigt werden. Die Stelle „SAv“ war aktiviert und konnte schalten. Von beiden Stellen im Vorbereich wurde jeweils ein Token, übereinstimmend mit der Kantengewichtung, entfernt und der Stelle „vor Brücke“ wurde ein Token hinzugefügt. Diese Markierung des Netzes simuliert den Sachverhalt, dass ein Auto auf die Brücke fährt. Aus der *Anfangsmarkierung* hat sich also eine neue Markierung ergeben. Diese Markierung wird als *Folgemarkierung* bezeichnet. Die Namen der Stellen und Transitionen haben auf das Schaltverhalten keinen Einfluss.

Aus dieser Folgemarkierung im Netz „Eine Brücke – SAV“ lässt sich nun auch der Zweck von der Stelle „Kontrolle“ erkennen. Sie ist dafür verantwortlich, dass immer nur ein Auto auf die Brücke fahren kann. Während in Abbildung 2.2 die Transitionen „SAv“ und „SBv“

aktiviert und schaltfähig sind, was den Zugang der Autos von beiden Straßen auf die Brücke darstellen soll, zeigt das Netz in Abbildung 2.4, dass nach Schalten der Transition „SAv“ der Zugang zur Brücke verhindert wird. Denn die Transitionen „SAv“ und „SBv“ sind nicht mehr aktiviert. Das Token auf der Stelle „Kontrolle“ wurde entfernt. Einzig die Transition „T1“ ist aktiviert und kann schalten. Erst wenn das Auto von der Brücke runter gefahren ist, kann ein weiteres Auto die Brücke überqueren. Die Transitionen „SAn“ und „SBn“ geben nach dem Schalten die Brücke und auch die „Kontrolle“ wieder frei.

2.3 Klassische Darstellung

Die Stärke der Petrinetze ist ihre mathematische Fundierung. In vielen Fällen (z.B. in Texten, zur formalen Beschreibung oder als Datenstruktur) sind formaltextuelle Darstellungen nützlicher, als die graphischen Darstellungen [vgl. [Valk und Moldt \(2006\)](#)]. Um diese Variante vorzustellen, sollen die bisher eingeführten Begriffe und Komponenten der Petrinetze formal beschrieben werden. Dabei wird das Petrinetz als Menge dargestellt. Im Hinblick auf rekonfigurierbare Petrinetze ist diese Art der Beschreibung erforderlich, weil die Analyse von rekonfigurierbaren Petrinetzen auf die formal, mathematische Darstellung basiert.

Ein S/T-Netz kann auch in einer Mengendarstellung formal beschrieben werden. Ein S/T-Netz N kann als 6-Tupel in dieser Weise formuliert werden,

$N = (S, T, F, W, K, m_0)$, wobei folgendes gilt:

- S ist die Menge aller Stellen von N
- T ist die Menge aller Transitionen von N
- außerdem gilt: $S \cap T = \emptyset$
- F ist die Flussrelation von N , für die gilt: $F \subseteq (S \times T) \cup (T \times S)$
- W ist das Kantengewicht jeder Kante, mit $F \rightarrow \mathbb{N}^+$
- K ist die Kapazität jeder Stelle, mit $S \rightarrow \mathbb{N}_\omega^+$
- m_0 ist die Anfangsmarkierung von N

\mathbb{N}^+ steht für die natürlichen Zahlen ohne die Null. \mathbb{N}_ω^+ ist die Menge der natürlichen Zahlen ohne die Null einschließlich ω , was in diesem Fall für unendlich steht. Die Beschriftungen der Stellen und Transitionen sind in diesen Beispielen zum besseren Verständnis ergänzt worden. Sie sind keinesfalls zwingend notwendig und werden dementsprechend auch nicht in der formalen Darstellung mit aufgeführt. Die Beschriftung von Stellen und Transitionen wird bei den dekorierten S/T-Netzen in Abschnitt 3.3 eingeführt.

In der Literatur wird anstatt der Bezeichnung *S/T-Netz* auch häufig der Begriff *P/T-Netz* bzw. *P/T-Net²* genutzt. Da diese Arbeit in deutscher Schrift verfasst ist, soll dieser Stil auch bei der Netzbezeichnung beibehalten werden.

2.4 Petrinetz Eigenschaften

Nachdem der Aufbau der *S/T-Netze*, ihre einzelnen Komponenten und das Schaltverhalten beschrieben wurden, sollen nun weitere Eigenschaften der *S/T-Netze* erläutert werden. Einige dieser Begriffe beschreiben Situationen, die sich lokal im Laufe der Schaltdynamik der *S/T-Netze* ergeben. Andere Begriffe kennzeichnen Eigenschaften vom gesamten, globalen Netz, und gehen der Frage nach, welche generellen Eigenschaften man aus der Struktur der Netze ableiten kann [vgl. Kleuker (2009)]. Diese Konzepte werden vor allem zu Analyse Zwecken genutzt.

Erreichbarkeit

In Abschnitt 2.2 wurden bereits die Begriffe Anfangsmarkierung und Folgemarkierung erläutert. Während die Anfangsmarkierung den ersten Markierungszustand des Petrinetzes beschreibt, ergibt sich durch einmaliges oder mehrmaliges Schalten von Transitionen eine Folgemarkierung. Wird eine Folgemarkierung m_1 , beginnend von einer Anfangsmarkierung m_0 , durch das Schalten von einer Transition/mehreren Transitionen erreicht, dann ist m_1 von m_0 aus erreichbar. Laut [Priese und Wimmel (2008)] ist dies eine der Kernfragen zu Petrinetzen.

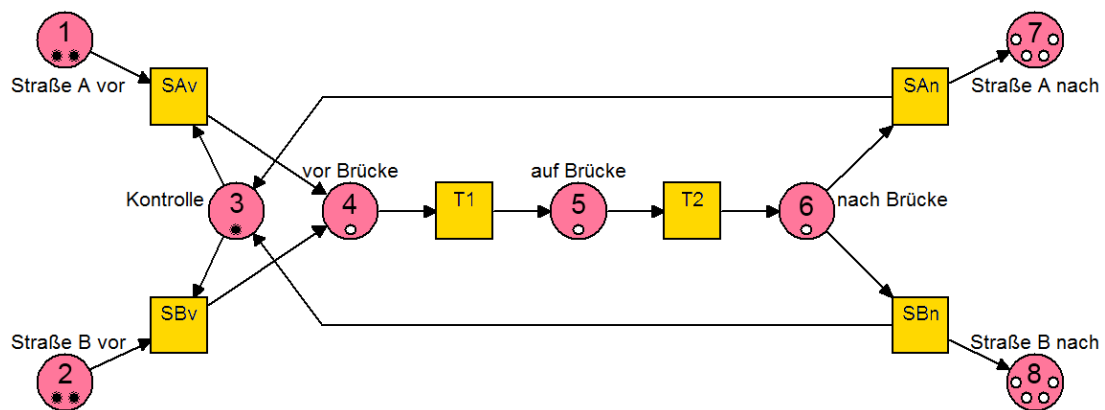


Abbildung 2.5: Petrinetz "Eine Brücke" (vgl. Abbildung 2.2) - farblich ergänzt

² engl. *Place/Transition Net*

Ein Graph, der die Erreichbarkeit aller möglichen Zustände (bzw. Markierungen des Netzes) erfasst, ist der sogenannte *Erreichbarkeitsgraph*. In diesem Graphen werden nicht nur alle erreichbaren Markierungen angezeigt, sondern es werden auch alle Zustandswechsel dargestellt, die durch das Schalten von Transitionen herbeigeführt werden.

Nachfolgend soll ein Ausschnitt vom Erreichbarkeitsgraphen des Petrinetzes „Eine Brücke“ gezeigt werden. Eine abgewandelte Darstellung des Petrinetzes soll die Beschreibung erleichtern. In Abbildung 2.5 sind alle Stellen und Transitionen gefärbt und die Stellen zusätzlich durchnummeriert worden. Mit Hilfe dieser Darstellung soll der Aufbau des Erreichbarkeitsgraphen erläutert werden.

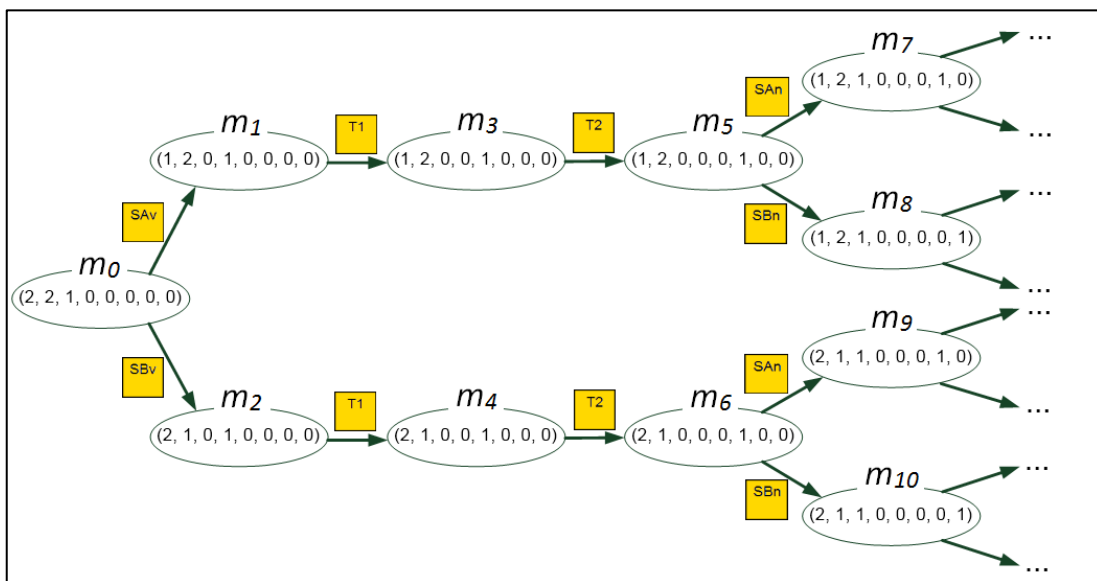


Abbildung 2.6: Ausschnitt vom Erreichbarkeitsgraphen des Netzes "Eine Brücke"

Der Ausschnitt des Erreichbarkeitsgraphen (siehe Abbildung 2.6) zeigt links die Anfangsmarkierung $m_0 = (2, 2, 1, 0, 0, 0, 0, 0)$. Diese Darstellung gibt die Anzahl der Token von jeder Stelle wieder. Alle Stellen haben ihre Position entsprechend ihrer Nummer. Man kann die Durchnummerierung auch als Index auffassen, der beliebig über alle Stellen gesetzt werden kann. In der Anfangsmarkierung m_0 besitzt die Stelle „Straße A vor“ zwei Token und steht wegen ihres Index 1 an erster Position. Analog gilt dies auch für alle weiteren Stellen. Ausgehend von m_0 kann entweder die Transition „SAv“ oder „SBv“ geschaltet werden.

Die Kanten des Graphen stellen also die geschalteten Transitionen dar, während die Knoten alle von m_0 aus erreichbaren Markierungen repräsentieren. Der Erreichbarkeitsgraph dieses Petrinetzes hat 81 mögliche erreichbare Markierungen. Aus diesem Grund ist hier nur der

Anfangsausschnitt illustriert und die Fortsetzung des Graphen auf der rechten Seite angedeutet.

Nicht immer lässt sich der Erreichbarkeitsgraph darstellen. Hat eine Stelle beispielsweise eine unendlich große Kapazität und nimmt unendlich viele Token auf, würde auch der Erreichbarkeitsgraph unendlich groß werden. In diesem Zusammenhang wird der Begriff *Beschränktheit* benutzt. Ein Petrinetz wird als *beschränkt* bezeichnet, wenn alle Stellen eine endlich große Kapazität haben. Daraus kann man folgern, dass ein Petrinetz genau dann beschränkt ist, wenn sein Erreichbarkeitsgraph endlich ist. Ist ein Petrinetz *unbeschränkt* und sein Erreichbarkeitsgraph unendlich groß, verschafft der *Überdeckungsgraph* Abhilfe. Siehe hierzu auch [Priese und Wimmel (2008)].

Konflikt

Während des Schaltvorgangs eines S/T-Netzes kann es zu Konfliktsituationen kommen. Wenn zwei Transitionen gleichzeitig aktiviert sind und um Token einer gemeinsamen Stelle im Vorbereitungsbereich „konkurrieren“, dann stehen sie im *Konflikt*.

Um diesen Sachverhalt anschaulich darzustellen, wird das Brückenbeispiel weiterentwickelt. Angenommen es gibt nicht nur eine Brücke, sondern zwei Brücken. Wobei immer noch die Bedingung gilt, dass nur ein Auto zurzeit über beide Brücken fahren darf. Das S/T-Netz in Abbildung 2.7 soll diese Sachlage anschaulich wiedergeben.

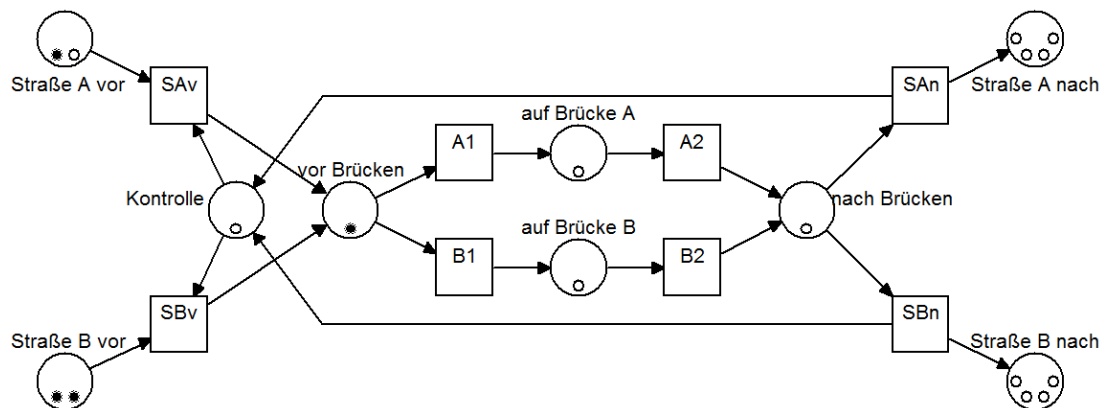


Abbildung 2.7: Petrinetz "Zwei Brücken"

Sobald ein Auto auf die Brücke fährt, hat es die Auswahl, über welche Brücke der Fluss überquert werden soll. In Abbildung 2.7 sind die Transitionen „A1“ und „B1“ aktiviert und können schalten. Sobald jedoch eine Transition schaltet, kann die andere Transition nicht mehr schalten, da das entsprechende Token von der Stelle „vor Brücken“ entfernt wurde. In diesem Fall ist die Vorbereitungsbedingung der anderen Transition verletzt. Es entsteht also

ein Konflikt. Anders formuliert kann man auch sagen, ein Konflikt entsteht bei nicht-*nebenläufiger*, gleichzeitiger Aktiviertheit von Transitionen [vgl. [Baumgarten \(1990\)](#)].

Nebenläufigkeit

Im Gegensatz zum Konflikt gibt es auch Situationen, in denen Transitionen nicht voneinander abhängig sind bzw. sich nicht beeinflussen. Wenn zwei Transitionen unabhängig voneinander schalten können, dann sind sie *nebenläufig*.

Das Brückenbeispiel soll erneut erweitert werden. Von nun an dürfen zwei Autos gleichzeitig über die beiden Brücken fahren, je ein Auto auf einer Brücke. Dazu wird die Kontrolle um ein weiteres Token ergänzt, sowie ebenfalls die Stellen „vor Brücken“ und „nach Brücken“. Das S/T-Netz in Abbildung 2.8 zeigt diese Erweiterung.

Jeweils ein Auto ist auf einer der Brücken. Die Stellen „auf Brücke A“ und „auf Brücke B“ sind mit einem Token belegt. Die Transitionen „A2“ und „B2“ sind beide aktiviert. Im Unterschied zur Konfliktsituation ist es diesmal egal, welche Transition wann schaltet. Beide sind unabhängig voneinander und beeinflussen sich nicht. Es ist unerheblich, in welcher Reihenfolge die beiden Transitionen schalten.

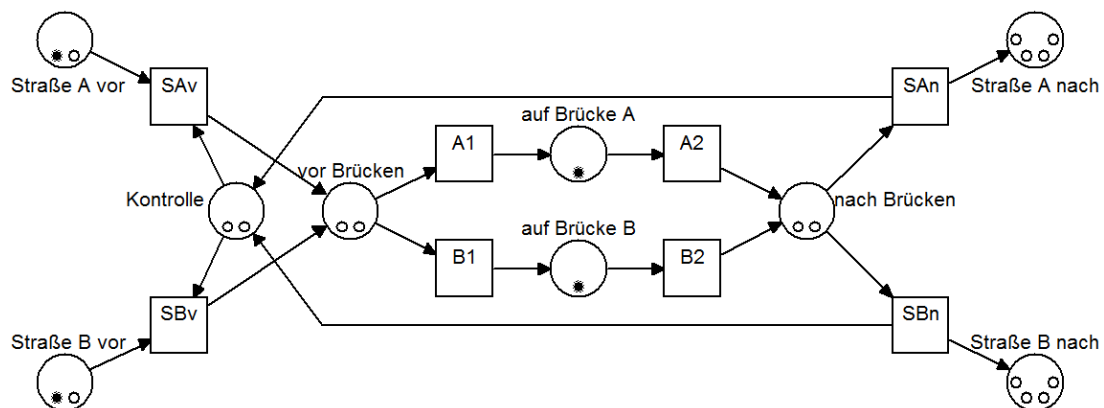


Abbildung 2.8: Petrinetz "Zwei Brücken - Zwei Autos"

Die Aufspaltung von Zustand und Zustandsübergang in unterschiedliche Knoten ist maßgeblich für die Existenz von Nebenläufigkeit in Petrinetzen verantwortlich. Dieses Prinzip macht Petrinetze im Vergleich zu einfachen Automaten so mächtig.

Verklemmung und Lebendigkeit

Die Nebenläufigkeit verleiht den S/T-Netzen zwar eine große Modellierungsstärke, jedoch ergeben sich dadurch auch andere Eigenschaften, die das Verhalten des gesamten Netzes

beeinflussen. In einem S/T-Netz kann es zu *Verklemmungen* kommen, so dass ein S/T-Netz nicht mehr in der Lage ist weiter zu schalten.

Das Brückenbeispiel soll ein weiteres Mal ergänzt werden. Diesmal wird zusätzlich ein Checkpoint eingerichtet. Das erweiterte Brückenbeispiel in Abbildung 2.9 soll den neuen Umstand demonstrieren.

Aus Sicherheitsgründen darf nur noch ein Auto zurzeit die beiden Brücken überqueren. Der Checkpoint verlangt pro Brücke zwei „Tickets“. Erst wenn beim Checkpoint zwei Tickets eingereicht wurden, kann die Brücke wieder verlassen werden. Die beiden Brücken im S/T-Netz wurden durch eine weitere Stelle und Transition jeweils „gestreckt“. Außerdem wurden die ausgehenden Kanten von der Stelle „Checkpoint“ mit einem Kantengewicht von zwei belegt. Dies soll den Umstand darstellen, dass ein Auto mit zwei eingereichten Tickets die Brücke auch wieder verlassen darf, indem der Checkpoint die zwei eingereichten Tickets wieder „freigibt“.

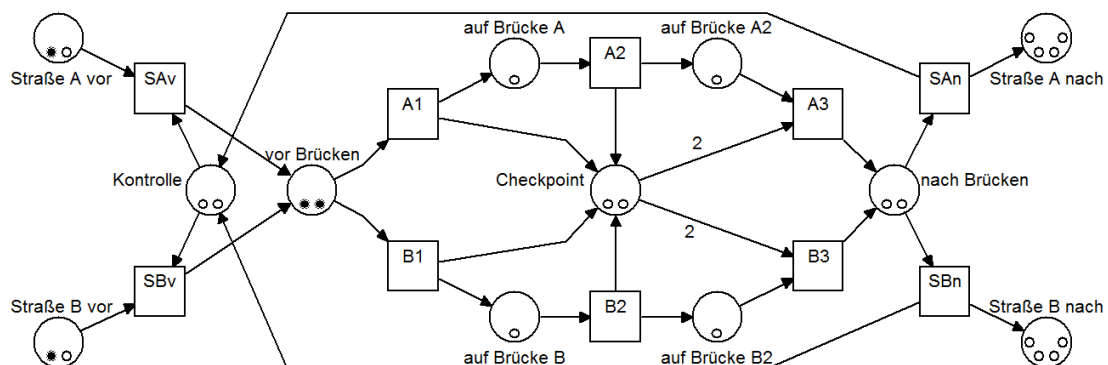


Abbildung 2.9: Petrinetz "Zwei Brücken - Ticketvergabe"

Das S/T-Netz in Abbildung 2.9 zeigt die Situation, dass zwei Autos vor der Brücke stehen. Wegen dem eingerichteten Checkpoint sollte nun zuerst das eine Auto die Brücke überqueren und dann das andere. Doch laut Netzstruktur ist es auch möglich, dass beide Autos auf die zwei Brücken fahren. In diesem Fall können die ersten beiden Transitionen der zwei Brücken, „A1“ und „B1“, noch schalten, denn sie sind beide aktiviert. Danach ist das System allerdings blockiert. Eine *Verklemmung*³ ist entstanden. Denn keines der beiden Autos kann nun weiterfahren. Die Stelle „Checkpoint“ hat die Kapazität zwei und deshalb können die Transitionen „A2“ und „B2“ nicht schalten, weil die Nachbereichsbedingungen verletzt sind.

Es gilt zu beachten, dass auch an anderen Stellen Verklemmungen entstehen können, die eine andere Relevanz haben. Wenn alle vier Autos die Brücken überquert haben und sich

³ engl. *Deadlock*

wieder auf den Straßen A und B befinden, dann existieren keine weiteren Autos (in diesem Brückenbeispiel), die eine der Brücken überqueren möchten. Vier Token sind demnach verteilt auf die Stellen „Straße A nach“ und „Straße B nach“. Auch hier befindet sich das Petrinetz in einer Verklemmung; es kann nicht mehr Schalten. Jedoch ist dieser Kontext anders zu bewerten. In diesem Fall ist der erreichte Gesamtzustand des Petrinetzes ein *Endzustand*. Gelegentlich kann eine solche Situation wünschenswert sein. Vorwiegend wird eine Verklemmung jedoch interessant und relevant, wenn er nicht einen Endzustand repräsentiert. Mit dem Erreichbarkeitsgraphen kann ein Netz auf Verklemmungen untersucht werden.

In diesem Zusammenhang wird auch häufig der Begriff der *Lebendigkeit* verwendet. Wenn eine Transition unter einer erreichbaren Markierung aktivierbar ist, dann heißt sie *lebendig*. Korrespondierend dazu heißt ein S/T-Netz lebendig, wenn alle seine Transitionen lebendig sind. In der Literatur gibt es viele weitere und auch unterschiedliche Bezeichnungen von Lebendigkeit. In [Priese und Wimmel (2008)] werden die Lebendigkeitsdefinitionen in *1-lebendig* bis *4-lebendig* angegeben. Andere Autoren, wie beispielsweise [Baumgarten (1990)], sprechen bei dieser Thematik u.a. von *schwach-lebendig* und *stark-lebendig*.

Es gibt eine Vielzahl von weiteren Eigenschaften und Konzepten, die bei der Analyse und Berechnung von S/T-Netzen und allgemein von Petrinetzen zu finden sind. Da die Angabe aller bzw. vieler dieser Eigenschaften den Rahmen dieser Arbeit übersteigen würde und nur ein grundsätzliches Verständnis für Petrinetze vermittelt werden soll, wird auf eine ausgiebige Beschreibung verzichtet. Auch in Bezug auf die Modellierung (Kapitel 4) und den Analysemöglichkeiten (Kapitel 5) sollen nur die Begriffe erläutert werden, die in dieser Arbeit von Bedeutung sind bzw. angewendet werden. Umfangreiche und detaillierte Literatur zu den Eigenschaften von Petrinetzen sind in [Girault und Valk (2003)] und [Priese und Wimmel (2008)] zu finden.

3 Rekonfigurierbare Petrinetze

Nachdem die klassischen Petrinetze vorgestellt wurden, soll nun eine Erweiterung dieser Netze beschrieben werden. Die rekonfigurierbaren Petrinetze bilden die Grundlage dieser Arbeit und sollen zum Modellieren der Morgen-Szenarien des Living Place genutzt werden. Dazu wird nach einer kurzen Einführung die universell-algebraische Darstellung vorgestellt. Im Anschluss werden dekorierte S/T-Netze beschrieben und formal definiert. Es folgt ein Abschnitt über Netztransformationen, in dem zum einen die Netzersetzungsregeln und zum anderen negative Anwendungsbedingungen erläutert werden.

3.1 Einführung

Rekonfigurierbare Petrinetze (auch *RKPN* oder *RCPN*⁴) sind Petrinetze mit zusätzlichen Eigenschaften. Die Grundidee hierbei ist die sukzessive Veränderung der Netze durch Netztransformationen. Die Veränderung der Netzstruktur soll dabei dynamisch während des Schaltverhaltens der Netze erfolgen. Mit Hilfe von Regeln sollen die Veränderungen der Netze realisiert werden. Dieser Ansatz der regelbasierten Netztransformation erhöht die Ausdruckskraft von Petrinetzen und ermöglicht neben den bekannten aktiven Schaltvorgängen der Netze, dem sogenannten Tokenspiel, eine formale Beschreibung der strukturellen Veränderungen [vgl. [Biermann u.a. \(2008\)](#)]. Ein typisches Anwendungsgebiet von RKPN ist die Modellierung von dynamischen Strukturen [vgl. [Padberg \(2012\)](#)].

Ebenso wie bei den klassischen Petrinetzen gibt es unterschiedliche Varianten von rekonfigurierbaren Petrinetzen. Dabei können sowohl Low-Level-Varianten, wie z.B. normale S/T-Netze, als auch High-Level-Varianten verwendet werden. Wie im vorherigen Kapitel bleibt der Fokus auf S/T-Netze gerichtet. Bevor das Hauptmerkmal der RKPN erläutert wird, sollen *dekorierte S/T-Netze* vorgestellt werden, eine weitere Variation der S/T-Netze.

⁴ engl. *Reconfigurable Petri Nets*

3.2 Universell-algebraische Darstellung

Wie zu Beginn des Kapitels erwähnt wurde, soll eine weitere Darstellungsart von Petrinetzen vorgestellt werden. Die universell-algebraische Darstellung (oder auch Monoid-Darstellung) von S/T-Netzen dient hier vor allem als Darstellungsgrundlage für die rekonfigurierbaren Petrinetze, da diese Darstellung im Hinblick auf Netzmorphismen angemessen ist. Bevor die universell-algebraische Darstellung erläutert wird, müssen zwei weitere Begriffe eingeführt werden.

Ein *freies, kommutatives Monoid* über M ist eine algebraische Struktur (bzw. ein algebraisches Objekt), das aus einer Menge M besteht. Es hat eine zweistellige Verknüpfung „ \oplus “ und ein neutrales Element „ ε “.

Das freie, kommutative Monoid M^\oplus wird als Tripel (M, \oplus, ε) notiert, bestehend aus der Menge M , der zweistelligen Verknüpfung „ \oplus “, mit

- \oplus : für $a, b \in M^\oplus$, gilt $a \oplus b \in M^\oplus$

und dem neutralen Element „ ε “, mit

- ε : für $a \in M^\oplus$, gilt $a \oplus \varepsilon = \varepsilon \oplus a = a$

Des Weiteren gilt für M^\oplus die Assoziativität und die Kommutativität, mit

- *Assoziativität*: für $a, b, c \in M^\oplus$, gilt $a \oplus (b \oplus c) = (a \oplus b) \oplus c$
- *Kommutativität*: für $a, b \in M^\oplus$, gilt $a \oplus b = b \oplus a$

Mit dem freien, kommutativen Monoid und seinen Eigenschaften lassen sich *lineare Summen* beschreiben. Ist z.B. die Menge $M = (a, b, c)$ gegeben, dann könnte „baabca“ ein Wort in M^* sein. Da Wörter nicht kommutativ sind, wäre aber folgende Beschreibung als lineare Summe möglich: „ $3a \oplus 2b \oplus c$ “. Wegen der Kommutativität ist die Reihenfolge der Elemente nicht von Bedeutung und sie können somit aufsummiert werden.

Ein S/T-Netz in universell-algebraischer Darstellung wird folgendermaßen formuliert,

$N = (S, T, pre, post, m)$, wobei folgendes gilt:

- S ist die Menge aller Stellen von N
- T ist die Menge aller Transitionen von N
- pre ist eine Abbildung, die den Vorbereich der Transitionen beschreibt, mit $T \rightarrow S^\oplus$

- $post$ ist eine Abbildung, die den Nachbereich der Transitionen beschreibt, mit $T \rightarrow S^\oplus$
- m ist die Markierung von N und wird durch eine lineare Summe beschrieben, mit $m \in S^\oplus$

Im Vergleich zur klassischen Darstellung lässt sich erkennen, dass die Flussrelation F und die Kantengewichtung W nicht mehr aufgeführt sind. Die Flussrelation und die Kantengewichtung repräsentieren die Netzstruktur (entsprechen den graphischen Kanten des S/T-Netzes), die bei der universell-algebraischen Darstellung durch pre und $post$ beschrieben werden.

Mit den Abbildungen pre und $post$ wird der Vorbereich (pre) und der Nachbereich ($post$) für jede einzelne Transition als lineare Summe gekennzeichnet. Dabei wird die Summe aus den Stellen der Transition gebildet, die im Vor- bzw. im Nachbereich liegen. Das Gewicht der Kanten wird dabei als Koeffizient dargestellt. Diese Abbildungen geben keine Menge wieder, sondern das freie, kommutative Monoid S^\oplus über der Menge S aller Stellen. Auch die Markierung m des Netzes N ist eine lineare Summe bzw. das freie, kommutative Monoid S^\oplus über der Menge S aller Stellen.

Der Vorteil der universell-algebraischen Darstellung von S/T-Netzen ist das gleichzeitige Berechnen vom Schalten einzelner Transitionen und vom Schalten nebenläufiger, also mehrerer Transitionen. Hierzu lässt sich ein Transitionsvektor definieren, der das Schalten von mehreren, unterschiedlichen Transitionen beschreibt. Außerdem kann bei Berechnungen die Häufigkeit des Schaltens mit einbezogen werden.

Mithilfe der Monoid-Darstellung der S/T-Netze und der Transitionsvektoren kann die Aktivierung von Transitionen, das Schalten mehrerer Transitionen (sowohl nebenläufig als auch sequentiell nacheinander) und die entsprechenden Folgemarkierungen notiert, analysiert und berechnet werden [vgl. [Padberg \(2011\)](#)].

3.3 Dekorierte S/T-Netze

Wie bereits in der Einführung erwähnt, bleibt das Augenmerk auf S/T-Netze gerichtet. Im vorherigen Kapitel wurde die universell-algebraische Darstellung der S/T-Netze eingeführt. Für die weiteren Ausführungen soll ausschließlich diese Darstellung der Netze verwendet werden, da die Monoid-Darstellung für Netzmorphismen (siehe Abschnitt 3.4) angemessener ist. Sie erhöht die Anschaulichkeit und erleichtert die Handhabung von Abbildungen.

Die rekonfigurierbaren Petrinetze in dieser Arbeit basieren auf den sogenannten *dekorierten S/T-Netzen*. Auch in diesem Kapitel soll das Thema anhand eines Beispiels

erläutert werden. Dazu soll erneut das Brückenbeispiel aus Kapitel 2 dienen. In Abbildung 3.1 wird die bereits bekannte Situation illustriert.

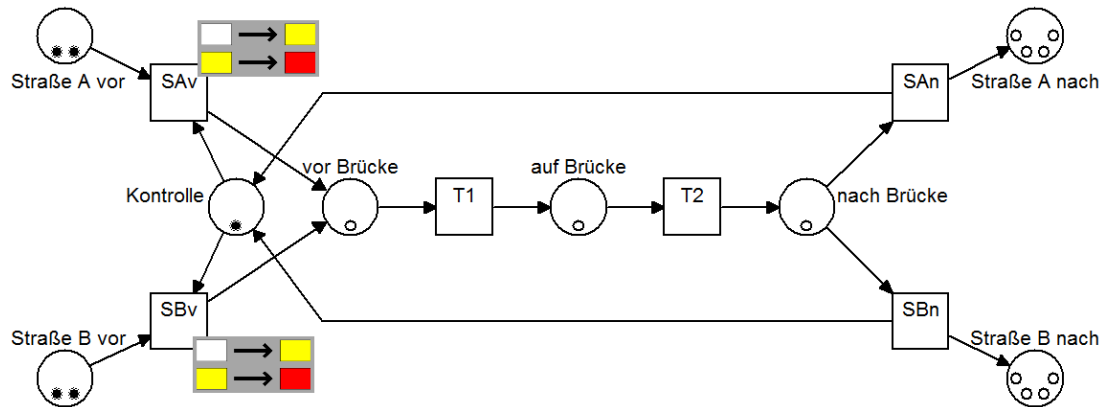


Abbildung 3.1: Petrinetz "Eine Brücke - Gefärbte Labels"

Doch in diesem Fall soll zusätzlich das Passieren von Autos an zwei Transitionen gekennzeichnet werden. Hierzu werden den Transitionen „SAv“ und „SBv“ jeweils eine kleine Erneuerungsfunktion (oder auch Tabelle) hinzugefügt, die zu einer Veränderung der Farbe führen. Schaltet eine Transition, verändert sich entsprechend der Abbildung ihre eigene Farbe. Die Transition „SAv“ hat zuerst die Farbe *Weiß*. Laut Funktion verändert sich also die Farbe beim Schalten von *Weiß* in *Gelb*. Schaltet die Transition danach erneut wird sie *Rot* (siehe Abbildung 3.2).

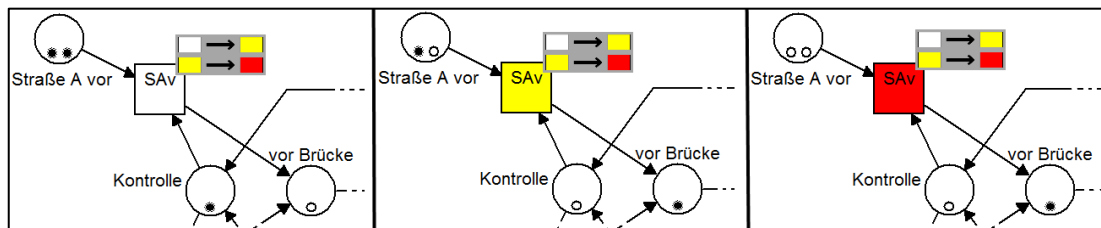


Abbildung 3.2: Veränderung der Transitionsfarbe

Transitionen bekommen also eine weitere Eigenschaft hinzugefügt. Diese Erweiterung der Transitionen wird als *Label* bezeichnet, das sich während der Schaltvorgänge verändern kann. In der Definition von dekorierten S/T-Netzen wird es als *tlb* ausgezeichnet. Dabei ist die Art der Labels nicht auf Farben beschränkt. Auch andere Formen sind denkbar, wie etwa natürliche Zahlen. Das Verändern der Labels wird auch *Label-Erneuerung* genannt. Das Erneuern der Labels erfolgt durch die Funktion *rnw*. Die *rnw*-Funktion ist eine Abbildung, in der die Label-Erneuerung definiert ist. In dem Beispiel aus Abbildung 3.1 ist *rnw* die

Abbildung einer Farbe auf eine Farbe. Bezogen auf natürliche Zahlen als Label, wären auch Funktionsterme vorstellbar, die das simple Zählen von Transitionsschaltungen ermöglichen.

Neben den Transitions-Labels gibt es eine weitere Ergänzung. Stellen und Transitionen bekommen eine Beschriftung. Diese Beschriftung dient, anders als in Kapitel 2, nicht nur dem besseren Verständnis der Beispiele, sondern ist eine notwendige Bedingung für die dekorierten S/T-Netze. Die Beschriftung von Stellen wird als *sname* und die der Transitionen als *tname* definiert. Die Bedeutung dieser Beschriftungen wird im nächsten Abschnitt deutlich, wenn die Netztransformationen vorgestellt und erläutert werden [vgl. Padberg (2011)].

Formale Definition

Nachfolgend soll eine formale Beschreibung der dekorierten S/T-Netze in der universell- algebraischen Mengendarstellung erfolgen.

Ein dekoriertes S/T-Netz *dekN* kann als 10-Tupel in dieser Weise formuliert werden,

$dekN = (S, T, pre, post, m, kap, sname, tname, tlb, rnw)$, wobei folgendes gilt:

- S ist die Menge aller Stellen von N
- T ist die Menge aller Transitionen von N
- pre ist eine Abbildung, die den Vorbereich der Transitionen beschreibt, mit $T \rightarrow S^\oplus$
- $post$ ist eine Abbildung, die den Nachbereich der Transitionen beschreibt, mit $T \rightarrow S^\oplus$
- m ist die Markierung von N und wird durch eine lineare Summe beschrieben, mit $m \in S^\oplus$
- kap ist die Kapazität jeder Stelle, mit $S \rightarrow \mathbb{N}_\omega^+$
- $sname$ ist die Beschriftung der Stellen, mit $S \rightarrow A_S$
- $tname$ ist die Beschriftung der Transitionen, mit $T \rightarrow A_T$
- tlb ist das Transitions-Label, mit $T \rightarrow W_T$
- rnw ist die Erneuerungsfunktion der Labels, mit $T \rightarrow END(W_T)$

A_S und A_T sind Namensräume. W_T sind wechselnde Labels. $END(W_T)$ ist ein Endomorphismus auf W_T und beschreibt die Menge aller Erneuerungsfunktionen. $END(W_T)$ ist die Menge aller Abbildungen von W_T auf sich selbst. Ein dekoriertes S/T-Netz

ist somit ein S/T-Netz inklusive der Abbildungen $sname$, $tname$, tlb und rnw [vgl. [Padberg \(2012\)](#)].

3.4 Netztransformationen

Netztransformationen sind das Hauptmerkmal der RKPN. Sie sind für die Veränderung der Netze verantwortlich, die sich zusätzlich zum Tokenspiel ergeben, und verleihen den Netzen ihre Dynamik. Dabei können die Netze u.a. vergrößert, verfeinert, gefaltet oder auch in andere Netze eingebettet werden [vgl. [Baumgarten \(1990\)](#)]. Die Transformationen der Netze basieren auf Netzmorphismen.

„Wie in algebraischen Theorien allgemein üblich, werden strukturerhaltende Abbildungen Homomorphismen oder kürzer Morphismen genannt. Um strukturverträgliche Transformationen zu definieren, wurde der Begriff auch für Netze eingeführt“ [vgl. [Valk und Moldt \(2006\)](#)].

Ein *Netzmorphismus* besteht aus einer Abbildung, welche die Struktur und die Markierung zwischen zwei Netzen erhält. Auf dekorierte S/T-Netze bezogen bedeutet dies, dass Stellen auf Stellen und Transitionen auf Transitionen abgebildet werden. Der Vor- und Nachbereich der Transitionen und die Markierung der Netze werden bewahrt. Weiterhin bleiben Labels und Beschriftungen gleichermaßen erhalten.

Zu Beginn dieses Kapitels wurde die universell-algebraische Darstellung vorgestellt, welche die Beschreibung des freien, kommutativen Monoids beinhaltet. Analog zum Netzmorphismus ist ein *frei erzeugter Monoidhomomorphismus* definiert als eine Abbildung zwischen zwei freien Monoiden.

f_S^\oplus ist ein *frei erzeugter Monoidhomomorphismus* über $f_S : S_1 \rightarrow S_2$. Jede Stelle aus einem Netz N_1 wird über f_S^\oplus auf eine Stelle aus einem Netz N_2 abgebildet. Für s_1 und $s_2 \in S_1$ von N_1 gilt dabei:

- $f_S^\oplus(0) = 0$
- $f_S^\oplus(k_1 \cdot s_1 + k_2 \cdot s_2) = k_1 \cdot f_S(s_1) + k_2 \cdot f_S(s_2)$

Der Begriff des Netzmorphismus soll nun, bezogen auf dekorierte S/T-Netze, formal beschrieben werden. Seien $dekN_1$ und $dekN_2$ dekorierte S/T-Netze über A_S , A_T und W_T mit $dekN_i = (S_i, T_i, pre_i, post_i, m_i, kap_i, sname_i, tname_i, tlb_i, rnw_i)$ für $i = 1, 2$.

Ein Netzmorphismus $f : N_1 \rightarrow N_2$ von N_1 nach N_2 ist ein Paar von Abbildungen $f = (f_S : S_1 \rightarrow S_2, f_T : T_1 \rightarrow T_2)$, so dass gilt:

- Bewahrung Vor- und Nachbereich: $f_S^\oplus \circ pre_1 = pre_2 \circ f_T$ und

$$f_S^\oplus \circ post_1 = post_2 \circ f_T$$

- Bewahrung der Markierung: $m_1(s) = m_2(f_S(s))$ für alle $s \in S_1$

- Bewahrung der Kapazitäten: $kap_1 = kap_2 \circ f_S$

- Bewahrung der Labels: $sname_1 = sname_2 \circ f_S$

$$tname_1 = tname_2 \circ f_T$$

$$tlb_1 = tlb_2 \circ f_T$$

$$rnw_1 = rnw_2 \circ f_T$$

Der Netzmorphismus f wird dabei als *strikt* bezeichnet, wenn f_S und f_T beide injektiv sind [vgl. [Padberg \(2012\)](#)].

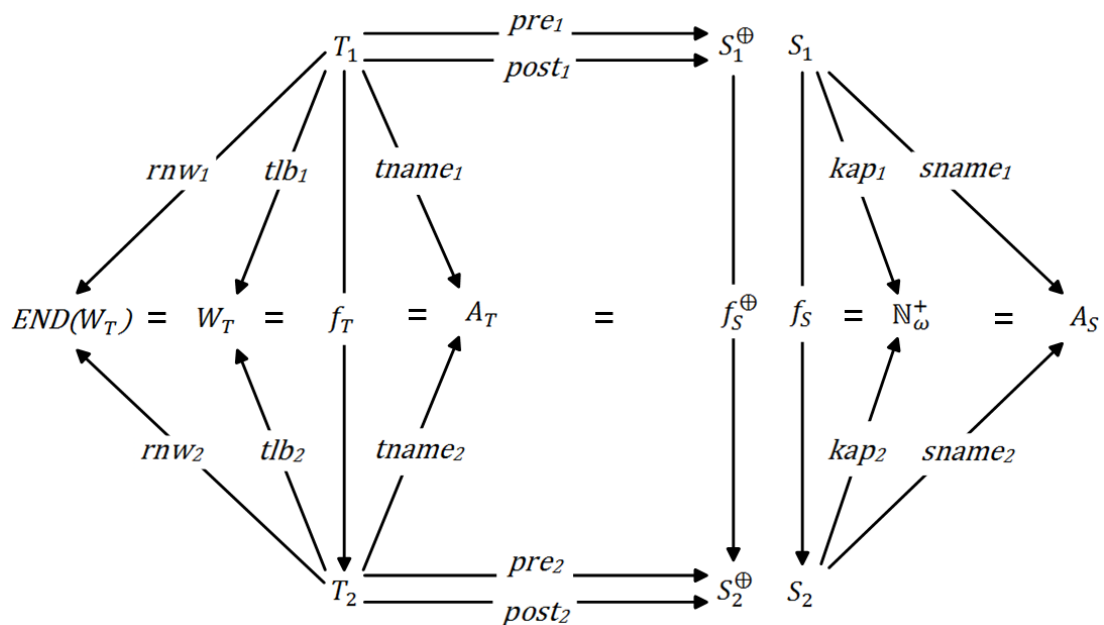


Abbildung 3.3: Strukturbewahrende, dekorierte S/T-Netz Morphismen

Die formal beschriebenen Abbildungen und Bedingungen eines dekorierten S/T-Netz Morphismus sollen veranschaulicht werden. In Abbildung 3.3 wird der Sachverhalt verdeutlicht. Die Gleichheitszeichen in der Skizze sollen den Aspekt der Struktur- und Kapazitätsbewahrung hervorheben, die den Netzmorphismus auszeichnen. Die Bewahrung der Kapazitäten $kap_1 = kap_2 \circ f_S$ fordert Gleichheit zum einen zwischen den Kapazitäten von Netz $dekn_1$, sprich kap_1 , und zum anderen von den Kapazitäten von Netz $dekn_2$, also kap_2 , nachdem die

Abbildung f_S von $S_1 \rightarrow S_2$ erfolgte. Analog dazu verhält es sich mit den anderen Komponenten und Abbildungen. Jetzt wird auch deutlich, weshalb die Beschriftungen der Stellen (*sname*) und der Transitionen (*tname*) wichtig sind. Sie sind bei Netzmorphismen ebenfalls notwendig für die Bewahrung der Struktur.

Netzersetzungsregel

Die Veränderung der Netze soll anhand von Regeln umgesetzt werden. Die soeben vorgestellten Netzmorphismen bilden die Grundlage dafür. Die Intuition der regelbasierten Netztransformation ist die Vorgabe einer Richtlinie, nach der die Netze modifiziert werden. Diese Richtlinie sollte nicht nur graphisch, sondern auch formal beschrieben werden, um eine Semantik für die Netztransformationen zu definieren. Eine erste formale Definition ist in [Padberg (2012)] zu finden.

Abermals soll das Brückenbeispiel zur Veranschaulichung verwendet werden. Das dekorierte S/T-Netz aus Abbildung 3.1 soll mittels regelbasierter Netztransformation verändert werden. Die bisherige Brücke soll durch eine neue, stärkere⁵ Brücke ersetzt werden. Dieser Transformationsprozess wird in Abbildung 3.4 dargestellt.

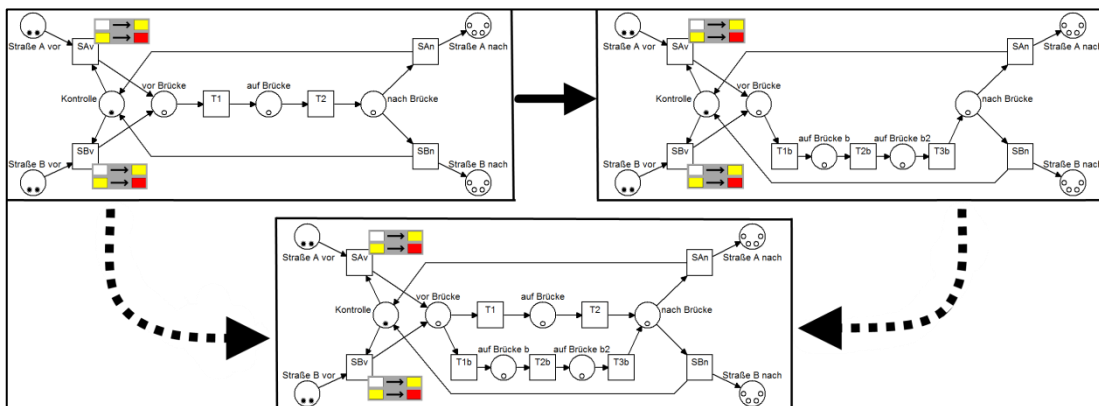


Abbildung 3.4: Transformationsprozess "Brückenveränderung"

Links oben ist das Ausgangsnetz abgebildet, welches bereits in Abbildung 3.1 gezeigt wurde. Rechts oben wird das veränderte Netz so dargestellt, wie es nach der Transformation aussehen soll. Das untere Netz zeigt eine Art Kombination beider Netze, in dem alle Netzkomponenten enthalten sind, sowohl die alte Brücke, die neue Brücke als auch alle übrigen Elemente.

Der Grundgedanke der Netzersetzungsregel (im Folgenden auch *Regel*) ist die Veränderung eines Netzes N_1 zu einem Netz N_2 . Dabei soll die Veränderung durch die Angabe einer Regel

⁵ Hier äußert sich die Stärke der Brücke durch Hinzufügen einer zusätzlichen Stelle und Transition.

erfolgen, die zum Einen darstellt, was aus dem gegebenen Netz N_1 gelöscht wird, und zum Anderen darstellt, was zum resultierenden Netz N_2 hinzugefügt wird [vgl. Rein u.a. (2008)]. Um dies darzustellen, gibt es eine *linke Seite* (L) und eine *rechte Seite* (R). Es soll also ein Netz L aus N_1 ausgewählt werden, dass anschließend in ein Netz R verändert wird (siehe Abbildung 3.5).

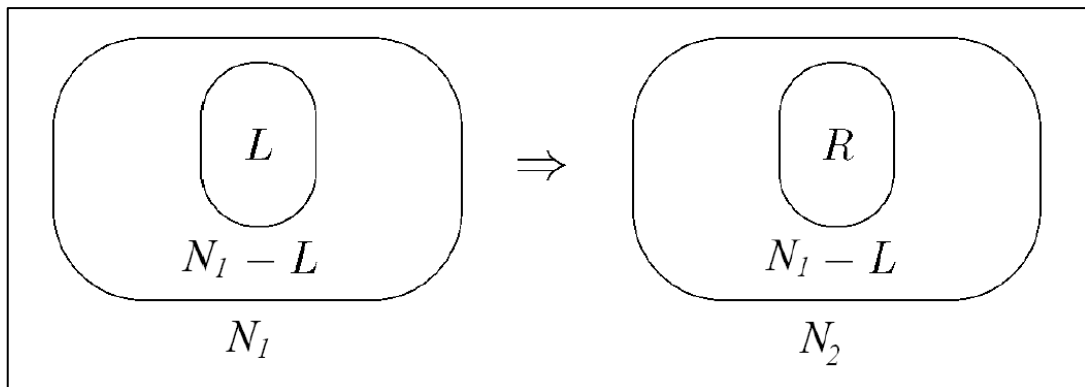


Abbildung 3.5: Die Idee der Netzersetzungsregel aus [Padberg (2011)]

Verglichen mit der Brückenveränderung stellt die alte Brücke den linken Teil der Regel (L) dar und die neue Brücke den rechten Teil (R). Das Netz links oben in Abbildung 3.4 würde N_1 entsprechen, während das Netz rechts oben N_2 entspricht. Das untere Netz, die „Kombination“ der Netze, beinhaltet einerseits N_1 als Teilnetz und andererseits N_2 als Teilnetz. Diese Kombination wird als *Klebenetz* (K) bezeichnet. Die Beziehung zwischen der *linken Seite* L , der *rechten Seite* R und dem *Klebenetz* K einer Netzersetzungsregel (r) über A_S , A_T und W_T wird folgendermaßen notiert:

- $r = \langle L \subseteq K \supseteq R \rangle$ wobei L , R und K Netze über A_S , A_T und W_T sind.

Wie bereits erwähnt wurde, soll ein Netz L aus N_1 ausgewählt werden. Hier kommt nun der oben eingeführte Netzmorphismus zur Anwendung. Die Auswahl eines Netzes, der linken Seite L , wird durch einen Netzmorphismus $\nu : L \rightarrow N_1$ beschrieben. Durch die Kontaktbedingung und durch die Identifikationsbedingung wird garantiert, dass das resultierende Netz N_2 inklusive der rechten Seite R auch wieder ein Netz ist. Die Kontaktbedingung sorgt beispielsweise dafür, dass keine knotenlosen⁶ Kanten entstehen.

⁶ Eine Kante ist knotenlos, wenn sie an einem Ende oder an beiden Enden keinen Kontakt zu einer Stelle oder Transition hat.

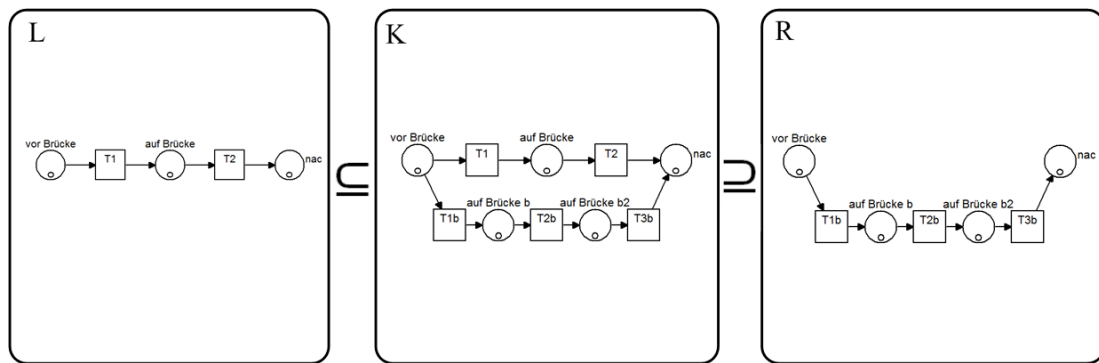


Abbildung 3.6: Regel zur Veränderung der Brücke

In Abbildung 3.6 wird die Regel zur Veränderung der Brücke illustriert. Darin ist zu erkennen, dass nicht nur die zu löschenden Elemente in der linken Seite abgebildet werden, sondern auch diejenigen, die aufgrund der Kontaktbedingung nicht gelöscht werden dürfen. Die Stelle „vor Brücke“ z.B. ist sowohl in L als auch in R enthalten (dementsprechend auch in K). Es werden also auch Komponenten aufgeführt, die im Ausgangsnetz und im resultierenden Netz enthalten sind und auch erhalten bleiben. Der Zusammenhang aller Regelkomponenten wird in Abbildung 3.7 veranschaulicht. Netz D ist ein „Klebenetz“ und beinhaltet jeweils N_1 und N_2 als Teilnetz. v , d und h sind Netzmorphismen.

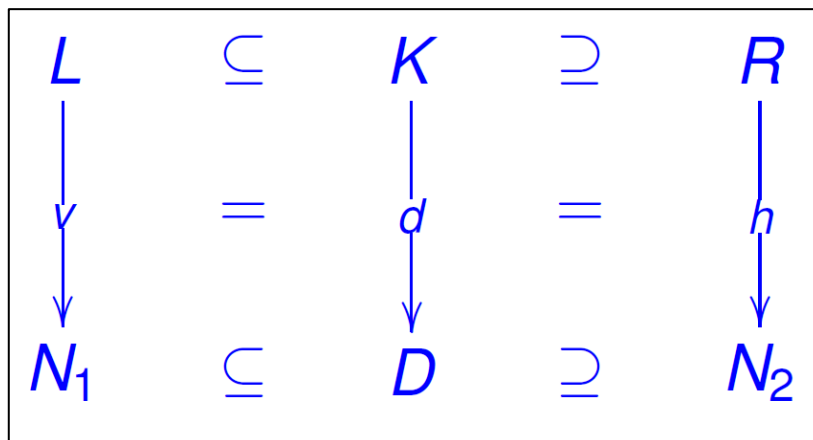


Abbildung 3.7: Zusammenhang der Regelkomponenten aus [Padberg (2011)]

Der Transformationsansatz $\langle L \subseteq K \supseteq R \rangle$ ist eine Variante der klassischen, algebraischen Transformationen und wird als *co-span* Ansatz bezeichnet. Anders als beim klassischen DPO^7 Ansatz werden hier zuerst neue Netzelemente hinzugefügt und

⁷ engl. *Double Push Out*

anschließend einige alte Elemente gelöscht. Dieser Ansatz ist vorteilhafter, weil er intuitiv besser zu verstehen ist [vgl. Ede u.a. (2012)].

Die Illustration der Regeln ist insbesondere bei großen Teilnetzen nicht immer anschaulich, weil drei Netze (L , R und K) abgebildet werden müssen. Deshalb ist eine optimale Darstellung der Regeln bzgl. der Lesbarkeit schwer zu realisieren.

In dieser Arbeit wird eine abgewandelte Form der Regeldarstellung angewendet. Die Regeln werden nur durch die Angabe der linken und rechten Seite notiert: $L \Rightarrow R$. Lediglich durch die optionale Angabe von negativen Anwendungsbedingungen (siehe nächster Abschnitt) werden die Regeln erweitert. Auf die Veranschaulichung des Klebenetzes wird verzichtet. Diese Darstellungsart ist ausreichend genug, um den semantischen Inhalt einer Regel wiederzugeben. Auf die Abbildung der Netze N_1 , D und N_2 wird ebenfalls verzichtet, wobei die Netze N_1 und N_2 bei der Modellierung der Szenarien (siehe Abschnitt 4.3) vor bzw. nach dem Anwenden einer Regel teilweise abgebildet werden. Der komplette Satz aller Netze und Regeln ist auf der beiliegenden CD zu finden.

Negative Anwendungsbedingung

Es können sich Situationen ergeben, in denen eine Regelanwendung möglich ist, ihre Anwendung aber dennoch verhindert werden soll. *Negative Anwendungsbedingungen* (im Folgenden NAC^8), bekannt aus Graphersetzungssystemen [vgl. Rein u.a. (2008)], bieten solch eine Kontrollstruktur. NAC s können optional zu den Netzersetzungsgesetzen hinzugefügt werden. Sie können die Ausführung einer Regel unterbinden und somit bestimmte Kontexte verbieten.

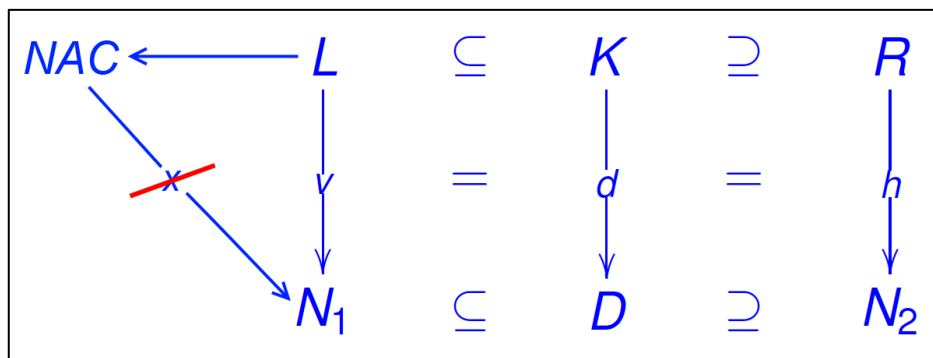


Abbildung 3.8: Zusammenhang der Regelkomponenten inklusive NAC

⁸ engl. *Negative Application Condition*

Die Anwendung einer Regel wird dann verhindert, wenn die negative Anwendungsbedingung nicht erfüllt wird. Anders formuliert bedeutet dies, eine Regel kann nur dann angewendet werden, wenn die *NAC* erfüllt wird. Ebenso wie die linke Seite *L* und die rechte Seite *R*, ist auch *NAC* ein Netz über A_S , A_T und W_T . Eine *NAC* gilt als erfüllt, wenn kein Netzmorphismus $x : NAC \rightarrow N_1$ existiert (siehe Abbildung 3.8). Das Netz der *NAC* darf also nicht im Ausgangsnetz N_1 enthalten sein.

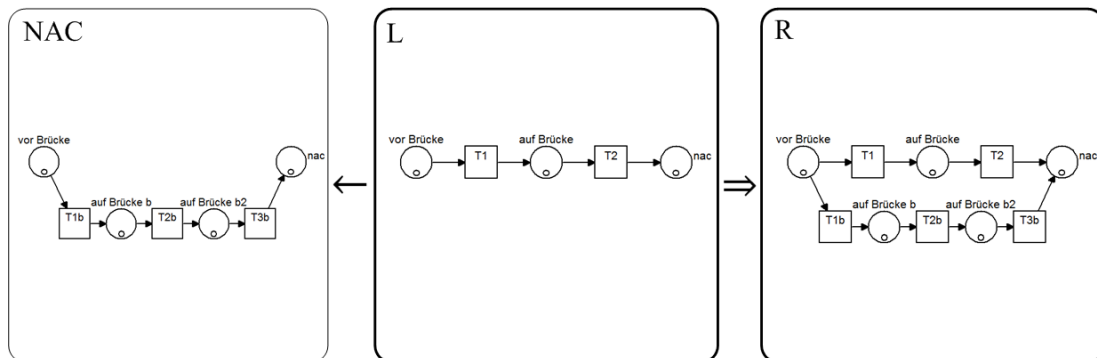


Abbildung 3.9: Negative Anwendungsbedingung bei Veränderung der Brücke

Das Beispiel der Brückenveränderung soll nun durch eine negative Anwendungsbedingung erweitert werden. Dazu wird das Beispiel ein wenig verändert. Zu der alten Brücke soll zusätzlich eine neue, starke Brücke hinzugefügt werden, so dass zwei Brücken nebeneinander stehen. Die beiden Netze *L* und *R* in Abbildung 3.9 geben diesen Kontext wieder. Bereits an dieser Stelle wird die abgewandelte Regeldarstellung $L \Rightarrow R$ angewendet. Die *NAC* soll verhindern, dass mehrere neue Brücken hinzugefügt werden. Ist das Netz der *NAC* bereits im Ausgangsnetz N_1 enthalten (bzw. wurde bereits eine neue Brücke hinzugefügt), existiert ein Netzmorphismus und die Regelanwendung wird verhindert.

Es gibt die Möglichkeit, mehrere *NACs* anzuwenden. In diesem Fall gelten die gleichen Bedingungen. Eine Regel kann nur dann angewendet werden, wenn alle negativen Anwendungsbedingungen erfüllt sind. Sobald ein Netzmorphismus zwischen einem der *NACs* und dem Ausgangsnetz existiert, wird die Anwendung der Regel verhindert.

Formale Definition

Die dekorierten *S/T*-Netze, der darauf bezogene Netzmorphismus und die regelbasierten Netzersetzungsregeln (im *co-span* Ansatz), die nun alle beschrieben und erläutert wurden, ergeben zusammen die rekonfigurierbaren Petrinetze. Folglich kann eine Definition der RKPN formuliert werden [vgl. [Padberg \(2012\)](#)].

Ein rekonfigurierbares, dekoriertes *S/T*-Netz $RKdekN = (dekN, \mathcal{R})$ ist gegeben durch ein dekoriertes *S/T*-Netz $dekN = (S, T, pre, post, m, kap, sname, tname, tlb, rnw)$ und einem Satz von Regeln \mathcal{R} , wobei \mathcal{R} durch $NAC \leftarrow L \rightarrow R$ beschrieben wird.

Im weiteren Verlauf der Arbeit soll der Begriff der rekonfigurierbaren Petrinetze (und auch die Abkürzung RKPN) synonym für die soeben formulierte Definition von rekonfigurierbaren, dekorierten S/T-Netzen verwendet werden.

3.5 Fazit

In diesem Kapitel wurden rekonfigurierbare Petrinetze vorgestellt, die in dieser Arbeit die Grundlage bilden und bei der Modellierung der Szenarien Anwendung finden. Aufbauend auf den aus Kapitel 2 eingeführten S/T-Netzen, wurden dekorierte S/T-Netze in der universell-algebraischen Darstellung beschrieben, die ein S/T-Netz durch Beschriftungen und Erneuerung der Transitions-Labels erweitern. Es folgte eine Erläuterung von Netzmorphismen im Hinblick auf dekorierte S/T-Netze, die die Basis für Netztransformationen bilden. Anschließend wurden Netzersetzungsregeln sowie negative Anwendungsbedingungen vorgestellt, mit denen die Rekonfigurierbarkeit der Petrinetze geregelt werden können. Auf diese Weise konnten schrittweise alle nötigen Konzepte, Begriffe und Definitionen eingeführt werden, um rekonfigurierbare Petrinetze zu beschreiben.

Mit dem Abschluss dieses Kapitels ist zugleich auch die Einführung der Grundlagen abgeschlossen. Es wurde versucht eine anschauliche und formale Einführung von rekonfigurierbaren Petrinetzen wiederzugeben, wobei der Schwerpunkt in erster Linie auf ersterem liegt. Im nächsten Kapitel soll die Anwendung von rekonfigurierbaren Petrinetzen zur Modellierung von Szenarien erfolgen.

4 Modellierung der Szenarien

Im folgenden Kapitel wird das „Living Place Hamburg“ (Abschnitt 4.1) vorgestellt, das als Inspirationsquelle und als Grundlage der Szenarien fungierte. Weiterhin werden verschiedene Prinzipien der Modellierung (Abschnitt 4.2) erläutert. Im nachfolgenden Abschnitt werden drei verschiedene Szenarien vorgestellt und jeweils anschließend ihre Modellierung präsentiert. Abschließend endet das Kapitel mit einer kurzen Bewertung der Modellierungen.

4.1 Living Place

4.1.1 Living Place Hamburg

Das „Living Place Hamburg“⁹ (im Folgenden auch Living Place) ist ein Appartement auf dem Campus der HAW Hamburg, in dem das Leben in einer computergestützten, intelligenten Wohnung erforscht und analysiert werden soll. Um dies zu erreichen, ist die 140 m² große Wohnung in einen Wohnbereich und einen Entwicklungsbereich eingeteilt (siehe Abbildung 4.1).

Der Begriff „Bewohner“ stellt im Folgenden eine fiktive Person dar, die in dieser Wohnung lebt. Auch im weiteren Verlauf dieser Arbeit, insbesondere bei den Szenarien, soll der Begriff mit dieser Bedeutung verknüpft sein. Damit die Szenarien und Beschreibungen nicht zu unübersichtlich werden und die Lesbarkeit der Texte erhalten bleiben, wird auf eine geschlechtsneutrale Formulierung verzichtet.

⁹ URL: <http://livingplace.informatik.haw-hamburg.de/blog/> (zuletzt zugegriffen: 10.07.2012)

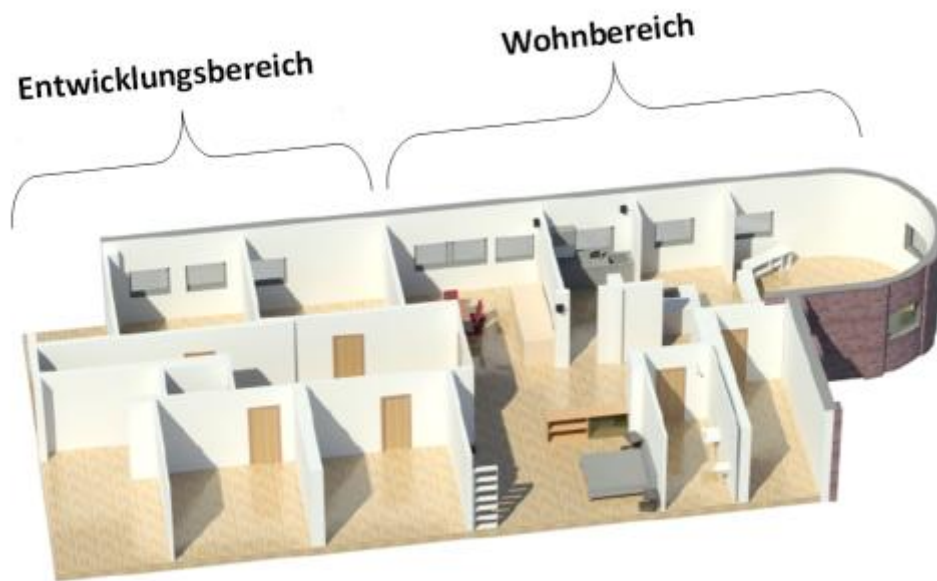


Abbildung 4.1: Living Place Hamburg aus [Ellenberg u.a. (2011)]

Die Wohnung ist nahezu komplett eingerichtet, um ein Leben unter realen Bedingungen zu ermöglichen. Zudem ist es mit zahlreichen Sensoren, Mikrofonen, Funknetzen und Kameras ausgestattet [vgl. von Luck u.a. (2010)]. Diese können den Anwendungen (Agenten) im Living Place Informationen liefern, damit sie den Bewohner im alltäglichen Leben unterstützen. Als Anwendung werden Dienste oder Programme der Wohnung bezeichnet, die eine bestimmte Aufgabe bei der Unterstützung haben. Sie können auch als Agenten bezeichnet werden.

Das Living Place kann auch als Multi-Agenten-System [vgl. Tennstedt (2011)] bezeichnet werden, in dem die verschiedenen Agenten über ein Nachrichtennetzwerkssystem untereinander verbunden sind und Daten austauschen können. Um die Interaktion und die Kommunikation von Sensoren und Agenten zu gewährleisten, wurde eine Blackboard-Architektur im Living Place eingesetzt. Sie fungiert als Schnittstelle zwischen den Anwendungen untereinander sowie mit allen Sensoren der Wohnung.

In Abbildung 4.2 ist die Architektur des Living Place abgebildet. Die verschiedenen Interpretationslevels repräsentieren Ebenen, in denen Daten verwertet und für andere Komponenten bereitgestellt werden. Auf der unteren Ebene (Low Level Interpretation) werden Sensordaten analysiert. Die beiden anderen Ebenen verarbeiten diese Informationen und die Sensordaten anhand von Modellen zu neuen Informationen. Das Blackboard ist die zentrale Einheit dieser Architektur und bildet die Schnittstelle der Kommunikation [vgl. Ellenberg u.a. (2011)].

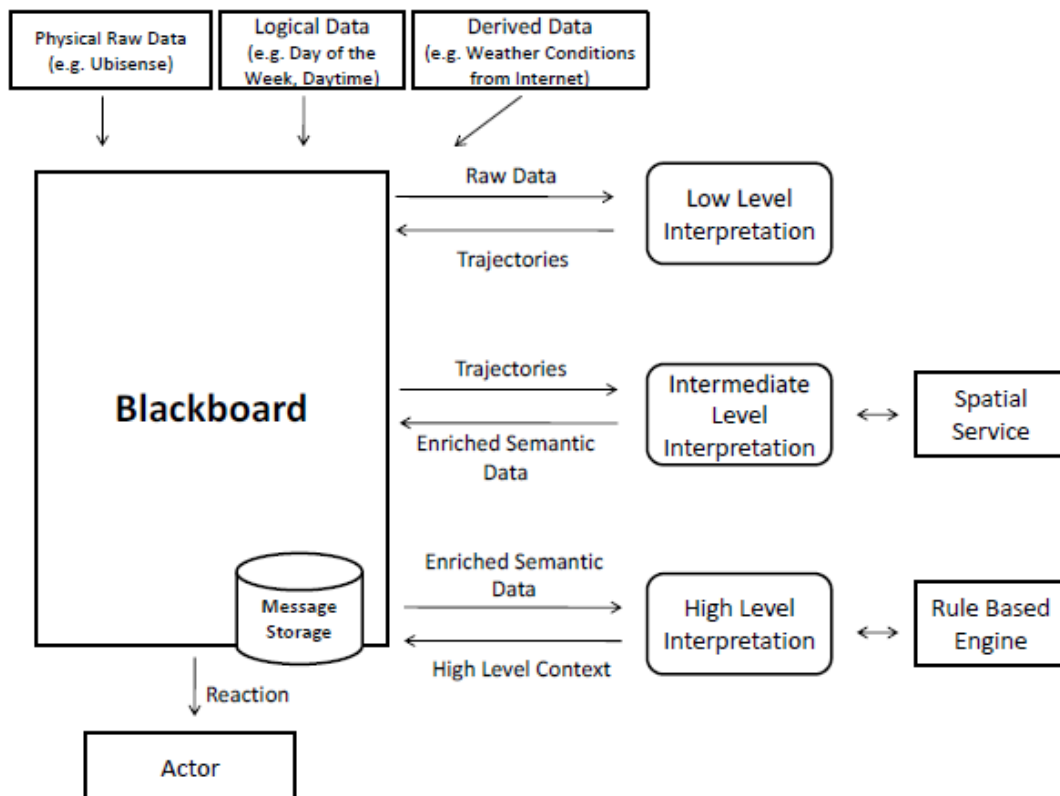


Abbildung 4.2: Architektur des Living Place aus [Ellenberg u.a. (2011)]

Dem Blackboard liegt das Prinzip des Publish-Subscribe¹⁰ zugrunde. Es erlaubt den Komponenten der Wohnung Informationen zu publizieren. Andere Komponenten (bspw. Agenten), die an solchen Informationen interessiert sind und sie verwenden möchten, können den Empfang dieser Daten abonnieren. In [Otto und Voskuhl (2010)] wurden einige Publish-Subscribe Systeme explizit für das Living Place untersucht und evaluiert.

Die nahtlose Interaktion des Bewohners mit der Wohnung ist ein wesentlicher Bestandteil der Forschung und Entwicklung im Living Place [vgl. Rahimi und Vogt (2011)]. Diesem Grundsatz folgen bereits zahlreiche Projekte, wie z.B. *Das intelligente Bett* [vgl. Hardenack (2011)], der *Wecker 2.0* [vgl. Ellenberg (2010)] und ein Sensornetzwerk [vgl. Schneider (2010)].

¹⁰ Ein Entwurfsmuster; deutsch „veröffentlichen-abonnieren“

4.1.2 Beschreibung der Szenarien

Bei einigen Entwicklungsprojekten und Forschungsarbeiten im Rahmen des Living Place Hamburg werden Szenarien eingesetzt [vgl. [von Luck u.a. \(2010\)](#)]. Dies sind Darstellungen für verschiedene, mögliche Abfolgen von Ereignissen und können je nach Intention und Ziel des Projektes bzw. der Arbeit angepasst werden.

Auch in der vorliegenden Arbeit werden Szenarien als Grundlage genutzt, um eine Modellierung und deren Analyse zu realisieren. Die Eigenschaft der freien, individuellen Gestaltung und die Auswahl völlig unterschiedlicher Abläufe in den Szenarien sollen dazu führen, verschiedene Ansätze einer Analyse der rekonfigurierbaren Petrinetze zu finden.

Die ausgewählten Szenarien beschreiben die morgendlichen Abläufe eines fiktiven Bewohners im Living Place, vom Aufwachen bis zum Verlassen der Wohnung. Im Vordergrund stehen die Handlungen und Aktivitäten des Bewohners. Die drei Szenarien, die in Abschnitt 4.3 modelliert und detailliert vorgestellt werden, stellen jeweils eine mögliche Abfolge der Aktionen und Geschehnisse des Bewohners dar. Es ist ein möglicher Ablauf von vielen. Falls der Bewohner von dem gegebenen Szenario abweicht, sollte die Wohnung die Abweichung erkennen. Folglich befindet sich der Bewohner in einem vielleicht ähnlichen, aber anderen Szenario. Im Idealfall kennt die Wohnung alle denkbaren Szenarien.

Bei der Auswahl wurde darauf geachtet, dass die Szenarien zwar unterschiedliche Abläufe widerspiegeln, dennoch beinhalten sie z.T. gleiche Teilaspekte. Die Teilaspekte, wie z.B. die Phase des Aufwachens, weichen dennoch in allen drei Versionen voneinander ab. So ergeben sich unterschiedliche Ablaufmuster. Die wesentliche Vorlage dieser Szenarien bilden die Ausarbeitungen von Jens Ellenberg, Bastian Karstaedt und Sören Voskuhl.

Wie schon im vorherigen Abschnitt beschrieben, nutzt die intelligente Wohnung viele verschiedene Sensoren, Kameras und auch Agenten, wie z.B. eine Kontextinterpretation [vgl. [Voskuhl \(2012\)](#)] oder eine Aktivitätserkennung [vgl. [Ellenberg \(2011\)](#)], um den Bewohner zu unterstützen. Die Szenarien in dieser Arbeit sollen jedoch nicht die technischen Komponenten beinhalten. Welcher Sensor oder welche Anwendung zu einer bestimmten Zeit aktiv ist und ein Event auslöst oder einen anderen Agenten anstößt, ist nicht entscheidend. Wenn die Wohnung sich entscheidet, ausgehend von den ermittelten Sensor- und Kontextinformationen, dass sie den Bewohner mit einer Aktion unterstützt, dann soll nur diese Aktion in den Szenarien wiedergespiegelt werden. Die Ermittlung der Daten und die Verarbeitung der Daten werden in den Szenarien nicht mit aufgeführt. Lediglich die resultierende Entscheidung der Wohnung, eine unterstützende Handlung zu vollziehen, die für die Interaktion des Bewohners mit dem Living Place von Bedeutung ist, spielt bei diesen Szenarien eine Rolle.

Außerdem fließen noch Gegebenheiten aus dem Umfeld der Wohnung mit ein. Diese können, wie z.B. das Wetter oder die Lichtverhältnisse innerhalb oder außerhalb des Living Place, ebenfalls von einem Sensor in der Wohnung gemessen werden (intern) oder von

externen Informationsquellen [vgl. [Ellenberg \(2011\)](#)] und [vgl. [Voskuhl \(2012\)](#)] stammen. Auch hier sind die Herkunft und die Entstehung der Informationen unerheblich. Nur die letztendlich relevante Information und die dadurch resultierende Entscheidung über den Ablauf des Szenarios bzw. über die Veränderung des rekonfigurierbaren Petrinetzes sind in dieser Arbeit ausschlaggebend.

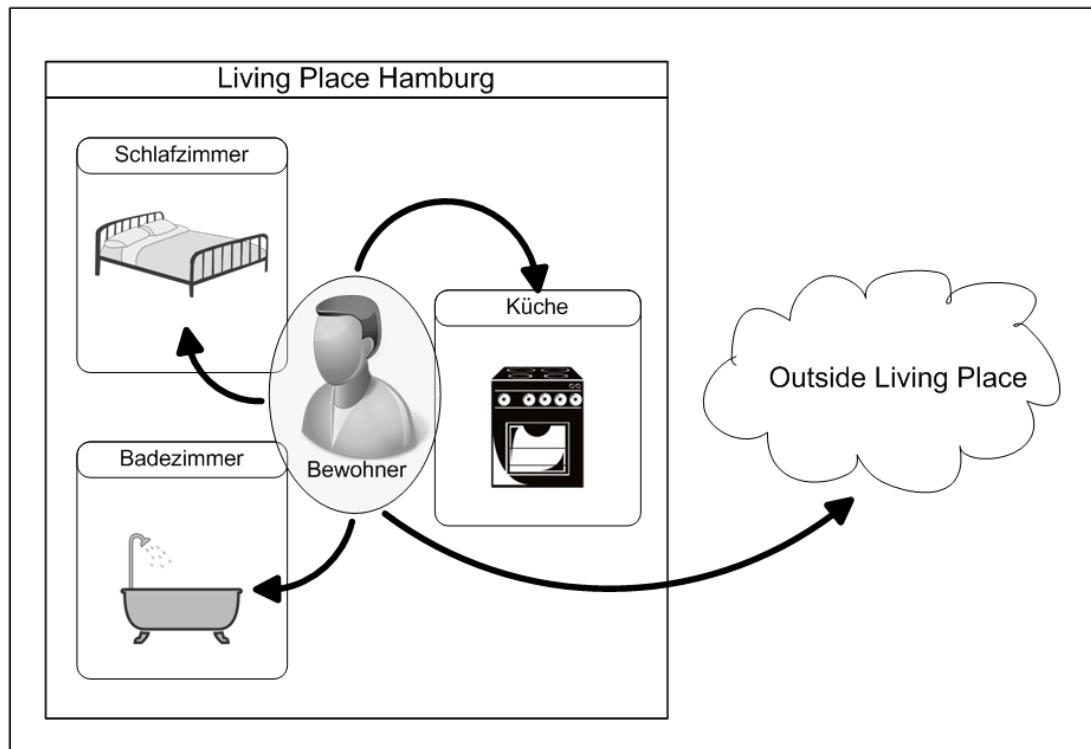


Abbildung 4.3: Aktionsraum im Living Place

Die Abbildung 4.3 verdeutlicht den Aktionsraum des Bewohners, wie er in den Szenarien in dieser Arbeit vorkommt. Innerhalb des Living Place hat der Bewohner die Möglichkeit sich im Schlafzimmer, im Badezimmer oder in der Küche aufzuhalten. Alle drei Szenarien enden damit, dass der Bewohner die Wohnung verlässt und er sich somit außerhalb des Living Place befindet.

Bei zahlreichen anderen Forschungsarbeiten steht die Kontexterkennung im Mittelpunkt. Die Wohnung soll die Aktivitäten des Bewohners erkennen und diese Informationen anderen Komponenten bereitstellen. Dazu muss man die Aktivitäten differenziert betrachten.

Läuft das Wasser in der Dusche und der Bewohner steht in der Dusche, dann ist der Bewohner am Duschen. Dies ist mit wenigen Sensoren einfach und klar zu erkennen und

könnte auf der unteren Ebene des Blackboards interpretiert werden. Doch einige Aktivitäten lassen sich nicht so simpel und eindeutig feststellen. Komplexere Aktivitäten, wie z.B. das Frühstücken, können aus mehreren Bestandteilen bestehen, von denen einige notwendig und einige hinreichend sein können. Wenn man davon ausgeht, dass „sich in der Küche befinden“ eine notwendige Bedingung ist, bleibt die Frage offen, wo genau man sich befinden muss. Muss der Bewohner am Tisch sitzen oder kann er auch im Stehen frühstücken? Die Aktivität Frühstücken zu erkennen ist für eine intelligente Wohnung aus diesem Grund komplizierter, da viele verschiedene Faktoren berücksichtigt werden müssen. Dies ist Aufgabe der High Level Interpretation, die solche Kontexte erkennen soll.

Diese Kontexterkenkung oder Aktivitätserkenkung der Wohnung wird bei den Szenarien in dieser Arbeit vorausgesetzt und ist somit auch kein Aspekt in meiner Analyse. Dieser Themenkomplex wird in [Ellenberg (2011)], [Voskuhl (2012)], [Tennstedt (2011)] u.a. konkreter und intensiver beleuchtet.

4.2 Modellierung

Der grundlegende Aufbau von Petrinetzen wurde in Kapitel 2 vorgestellt. Neben den formalen Darstellungen wurde darin auch die graphische Darstellung erläutert. Im anschließenden Kapitel 3 wurden die spezifischen Eigenschaften der rekonfigurierbaren Petrinetze eingeführt, bestehend aus den Label-Erneuerungen und Netztransformationen.

Im Folgenden soll nun erklärt werden, in welcher Art und Weise diese Eigenschaften bei der Modellierung eingesetzt und umgesetzt wurden. Es sollen die Prinzipien der Modellierung beleuchtet werden. Dabei soll in erster Linie die Bedeutung der einzelnen Petrinetzkomponenten hinsichtlich der Szenarien bzw. der intelligenten Wohnung im Vordergrund stehen.

4.2.1 Prinzipien der Modellierung

Im Hinblick auf die Beschreibung von den Prinzipien der Modellierung lassen sich diese in zwei Bereiche gliedern. Auf der einen Seite gibt es die Modellierungsgrundsätze der S/T-Netze, bezüglich der Stellen, Transitionen und den weiteren Komponenten. Auf der anderen Seite sollen auch die besonderen Merkmale der rekonfigurierbaren Petrinetze, bestehend aus den Label-Erneuerungen und den Netztransformationen, verdeutlicht werden.

Bezüglich dieser beiden Bereiche sind folgende Abbildungen¹¹ prinzipiell beobachtbar. Die S/T-Netze bzw. das Netz- oder Schaltverhalten der S/T-Netze entsprechen dem Handeln des Bewohners. Die Regeln und deren Netztransformationen stehen hingegen für das Verhalten und die unterstützenden Maßnahmen der intelligenten Wohnung. Dieses Schema trifft nicht auf alle Bereiche der Modellierung zu, kann aber dennoch als richtungsweisend gesehen werden.

S/T-Netze

Die klassische Syntax der Petrinetze, wurde in Kapitel 2 gezeigt und gibt an, dass Zustände durch Stellen und Aktionen durch Transitionen gekennzeichnet werden. Dieses Paradigma ist größtenteils auch in der vorliegenden Arbeit so umgesetzt worden. In den meisten Fällen beschreiben die Stellen den Zustand des Bewohners bzw. den Zustand der Wohnung, seltener einen weiteren Zustand von einer Gegebenheit, die sich innerhalb der Wohnung ergibt. Wenn der Bewohner z.B. im Bad steht, dann liegt ein Token auf der zugehörigen Stelle. Ebenso verhält es sich mit den Transitionen. Auch hier erfolgte die Umsetzung überwiegend klassisch, d.h. eine Aktivität des Bewohners oder eine Aktion/Entscheidung der Wohnung wurde durch eine Transition gekennzeichnet.

Diese Einteilung von Zuständen und Transitionen ist ein grundlegender und sehr wichtiger Schritt im Modellierungsprozess. Je genauer und schlüssiger die Einteilung sowie die Benennung der einzelnen Knoten sind, desto einfacher ist es für den Betrachter das Konzept bzw. das System zu verstehen. Dieser Einteilungsprozess lässt sich jedoch nicht immer problemlos und einfach realisieren. Auch bei der vorliegenden Modellierung dieser Arbeit gibt es manchmal leichte Abweichungen, die zwar durch die Gestaltungsfreiheit erlaubt sind, jedoch zu unterschiedlichen Auffassungen des Betrachters führen können. Aus diesem Grund sollen im weiteren Verlauf einige Prinzipien aufgezeigt werden, die bei dieser Modellierung von Bedeutung sind.

Ein Bewohner in der Wohnung führt viele unterschiedliche Aktivitäten aus. Meistens sind es zahlreiche Aktionen nacheinander. Bei der Gestaltung von Petrinetzen dürfen jedoch nicht mehrere Transitionen hintereinander geschaltet werden. Dies würde die Syntax der Petrinetze verletzen. Deshalb ist es erforderlich, geeignete Einteilungen zwischen Stellen und Transitionen zu finden. Wie bereits erwähnt, ist in dieser Arbeit versucht worden, eine solche Einteilung zu finden, um ein möglichst verständliches Bild der Szenarien zu vermitteln. Dennoch ergeben sich Situationen, bei denen die Knoten nicht 1:1 einen Zustand bzw. eine Aktion darstellen. Beispielsweise wird das „Frühstücken“ in dieser Modellierung als Stelle repräsentiert. Es wäre durchaus möglich, dieses als Aktion und somit auch als Transition darzustellen, was der Bedeutung des Wortes eventuell näher kommt. Da das Frühstücken jedoch ein signifikanter Bestandteil des Morgenprozesses ist

¹¹ Gemeint sind keine Illustrationen, sondern Abbildungen im Sinne einer Zuordnung

und in einem zeitlichen Kontext einen höheren Stellenwert hat als andere, zeitlich kürzere Aktionen, wurde das Frühstück als Stelle dargestellt. Es ist ein Beispiel dafür, dass die Gestaltungsfreiheit von Petrinetzen sehr groß ist, wobei die Lesbarkeit und die Verständlichkeit des Systems dabei im primären Fokus stehen sollten.

In dieser Arbeit ist bei der Einteilung von Stellen und Transitionen das Augenmerk vordergründig auf die Auswahl und Benennung der Stellen gerichtet worden. Dabei wurde versucht, mit Blick auf das Living Place und seinem Wohnbereich die wichtigen Bestandteile vom morgendlichen Handlungsablauf des Bewohners herauszufiltern. Hier ist vor allem der räumliche Aspekt zu nennen. Wie unter Abschnitt 4.1.2 bereits aufgezeigt wurde, hat der Bewohner in der Wohnung einen Aktionsraum (bestehend aus Schlafzimmer, Badezimmer und Küche). Diese Raumkomponenten sind zunächst einmal als Stellen gekennzeichnet worden. Andere Stellen wiederum spiegeln Zustände innerhalb der Räume wieder. Anhand ihrer Lokalität im Netz lassen sich diese Stellen den jeweiligen Räumen zuordnen.

Neben den Stellen und den Transitionen sind als weitere syntaktische Komponenten der Petrinetze die Token zu erwähnen. Beim Betrachten der Szenarien, also des Bewohners innerhalb der intelligenten Wohnung, könnte man annehmen, dass es sinnvoll wäre, ein Token zu nutzen, das den Bewohner repräsentiert. Diese Annahme wäre nicht nur intuitiv verständlich, sondern auf den ersten Blick sogar sinnvoll. Würde man jedoch immer nur ein Token „herumwandern“ lassen, könnte man nur den Bewohner repräsentieren und nicht die unterstützenden Aktivitäten der Wohnung darstellen. Es würde sich ein sequentieller Ablauf ergeben, der auch mit Automaten dargestellt werden könnte und eine der wichtigsten Eigenschaften der Petrinetze, die Nebenläufigkeit, ungenutzt bleiben würde. Dennoch lässt sich eine Abbildung des Bewohners bei einigen Token wiederfinden. Auch in diesem Fall kann man gerade in Bezug auf den räumlichen Kontext einen Zusammenhang zwischen dem Bewohner und den Token herstellen.

Die Abbildung¹² von Bewohner und Token ist somit nicht immer korrekt. In dieser Modellierung soll der morgendliche Ablauf in einer intelligenten Wohnung, die den Bewohner unterstützen soll, dargestellt werden. Demzufolge müssen auch die von der Wohnung durchgeführten Maßnahmen zur Unterstützung berücksichtigt werden. Wie man beispielsweise in Regel R9 erkennt, kann die Wohnung beim Eintritt in die Küche das Licht anschalten, wenn die Sensoren wenig Tageslicht messen. Zum einen stellen die Token also die Handlungen und Aktivitäten des Bewohners dar und zum anderen drücken sie Maßnahmen der Wohnung aus, resultierend aus Sensorwerten oder Interpretationsergebnissen.

Diese unterschiedliche Bedeutung der einzelnen Token ist nicht direkt in den Petrinetzen gekennzeichnet, da bei dieser Modellierung mit dem S/T-Netz ein Elementarnetz gewählt wurde, deren Token ausschließlich schwarze Marken sind. Petrinetze, in denen es auch unterschiedliche Marken (Token) gibt und deren Stellen sogenannte Multimengen speichern können, gehören zur Familie der High-Level-Netze. Diese werden in [Reisig

¹² hier: Abbildung im Sinne einer Zuordnung

(2010)] genauer beschrieben. Dennoch lässt sich die Bedeutung der Token aus der lokalen Umgebung im Petrinetz und teilweise aus der Bezeichnung der Knoten erkennen.

Eine weitere Eigenschaft der Stellen in S/T-Netzen ist die Kapazität von Token. Bei dieser Modellierung haben alle Stellen die gleiche Kapazität. Jede Stelle kann maximal ein Token aufnehmen, demzufolge ist die Kapazität gleich eins. Die Kapazitäten bei dieser Modellierung werden mit aufgeführt. Nur wenn die Petrinetze *1-sicher* wären [vgl. Baumgarten (1990)], was bedeutet, dass aus Netzstrukturgründen die Anzahl von Tokens auf jeder einzelnen Stelle nie größer als eins werden kann, könnte man die Kapazitäten vernachlässigen.

Auch im Sinne des intuitiven „Bewohner-Token“ (siehe oben) wurde diese Limitierung gesetzt. Wie bereits erwähnt wurde, soll trotzdem die Nebenläufigkeit ein erkennbarer Faktor in den Petrinetzen sein. So ergeben sich teilweise Markierungen, bei denen mehrere Stellen ein Token besitzen. Damit einhergehend ist in dieser Arbeit auf eine unterschiedliche Gewichtung der Kanten verzichtet worden. Alle Kanten haben ein Kantengewicht gleich 1.

Rekonfigurierbare Petrinetze

Regelbasierte Netztransformation ist das Hauptmerkmal rekonfigurierbarer Petrinetze. Wie im Kapitel 3 bereits gezeigt wurde, können durch Regeln die Petrinetze dynamisch verändert werden.

Beim Anwenden von Regeln werden Teilnetze oder Netzkomponenten hinzugefügt oder gelöscht. Demzufolge müssen Teilnetze für die Regeln gefunden bzw. erstellt werden. Dazu wurde überprüft, welche notwendigen Verbindungen und Abhängigkeiten im morgendlichen Ablauf des Bewohners vorzufinden sind. Basierend auf diesen Überlegungen konnten Teilnetze lokalisiert und definiert werden. Bei der Auswahl der Teilnetze wurde versucht, einen Auszug aus dem gesamten Morgenprozess des Bewohners zu extrahieren. Dieser Auszug sollte dem Kontext entsprechend inhaltlich stimmig sein und im besten Fall eine Aktivität des Bewohners oder der Wohnung repräsentieren. Nach diesem Prinzip wurden die Regeln schrittweise abgeleitet und ausformuliert.

Die Granularität der Regeln und die daraus implizit resultierenden Teilnetze ist ein wesentlicher Schritt bei der Modellierung der rekonfigurierbaren Petrinetze. Ebenso wie bei der Erzeugung von Stellen und Transitionen kann auch bei der Auswahl von Netzmorphismen, also dem Aufbau von Regeln, frei entschieden werden. Natürlich müssen die Klebebedingungen der Netztransformationen eingehalten werden.

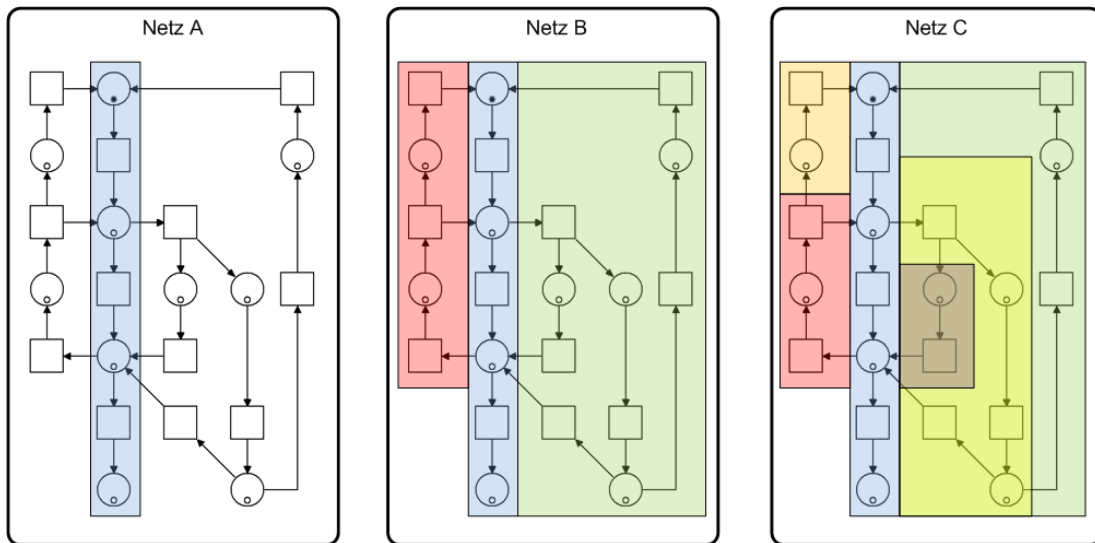


Abbildung 4.4: Teilnetzvergleich

Grundlegend ist zu beachten, dass die Größe der Netze innerhalb der Regeln auch die Anzahl der Regelanwendungen von der gesamten Modellierung bestimmt. Anhand von Abbildung 4.4 soll dies veranschaulicht werden. Alle drei Netze beinhalten das gleiche Petrinetz. Netz A zeigt das Basisnetz (blau). Ausgehend vom Basisnetz werden in Netz B und in Netz C unterschiedliche Teilnetze markiert. Während in Netz B nur zwei Teilnetze gekennzeichnet sind, zeigt Netz C fünf unterschiedliche Teilnetze. Wenn das Petrinetz nun alle Teilnetze zum Schalten benötigt und man davon ausgeht, dass nur das Basisnetz existiert und alle benötigten Teilnetze jeweils durch eine Regel hinzugefügt werden müssen, dann gäbe es zwangsläufig zwei (bei Netz B) bzw. fünf (bei Netz C) Regelanwendungen.

Bei dieser Modellierung wurde versucht, die Anzahl der Regeln und die Feinheitstufe der Gliederung der Teilnetze auf ein geeignetes Mittelmaß zu bringen. Eine zu feingranulare Einteilung von Teilnetzen und Regeln beeinträchtigt im Zusammenhang mit den Szenarien die Anschaulichkeit des gesamten Modells, weil es eine große Menge an Regeln gäbe und das Schalten, die eigentliche Dynamik der Petrinetze, in den Hintergrund geraten würde.

Dieser Aspekt lässt erkennen, dass eine ansprechende, interessante Dynamik der RKP von zwei Faktoren abhängt. Zum einen hängt die Dynamik vom Schalten der Transitionen ab und zum anderen auch maßgeblich vom Anwenden regelbasierter Netztransformationen. In den Szenarien soll sich, ausgehend von einem Basisnetz, eine Veränderung dieses Netzes durch diese beiden Faktoren ergeben. Durch diese Modifizierung kann das Netz größer bzw. kleiner werden und es wird eine dynamische Veränderung der Netzstruktur erzeugt. In diesem Fall ist dies gleichbedeutend mit den Aktivitäten und Handlungen des Bewohners bzw. der intelligenten Wohnung. In dieser Arbeit wurde versucht, eine angemessene

Balance zwischen diesen beiden Faktoren zu etablieren. Eine wechselnde Abfolge von Transitionsschaltungen und Regelanwendungen soll die Anschaulichkeit der Modellierungen erhöhen und einen ansprechenden, dynamischen Ablauf erzeugen.

Bei der Anwendung von Regeln können RKPn verändert werden. Komponenten werden hinzugefügt oder gelöscht. Ein grundlegendes Prinzip bei der Erstellung der Morgen-Szenarien in dieser Arbeit ist die Unterscheidung der Regeln hinsichtlich dieser beiden Möglichkeiten. Die eine Gruppe der Regeln fügt Teilnetze hinzu, während die andere Gruppe diese Netze wieder löscht. Dieser Aspekt verdeutlicht die Einteilung und den Gebrauch von Teilnetzen, da jedes Teilnetz mit einer Regel hinzugefügt und mit einer korrespondierenden „Löschregel“ (auch im weiteren Verlauf der Arbeit wird dieser Begriff verwendet) entfernt werden kann. Aufgrund der Bedingungen (Klebe- und negative Anwendungsbedingungen) von RKPn ist eine 1:1 Beziehung der veränderbaren Elemente zwischen Regel und Löschregel nicht vorzufinden. Es kann also vorkommen, dass eine Stelle oder Transition, die von einer Regel R_{xx} hinzugefügt wurde, bei der dazu passenden Löschregel $LÖR_{exx}$ nicht wieder entfernt wird. Im weiteren Verlauf der Arbeit wird der Begriff „Löschregel“ benutzt, um diejenigen Regeln zu benennen, die dem Löschen von Elementen zugeschrieben werden.

Die Löschregeln sollen bei dieser Modellierung hauptsächlich dazu verwendet werden, um Teilnetze wieder zu entfernen, die in bestimmten Momenten nicht benötigt werden und keine Beziehung zum aktuellen Kontext haben. Ersichtlich wird dies u.a. auch dadurch, dass der Großteil aller Löschregeln keine Token in der linken und rechten Seite der Regel enthält. Die negativen Anwendungsbedingungen sind davon ausgeschlossen. Dies ermöglicht den RKPn seine eigene Netzstruktur der aktuellen Umgebung anzupassen und per Regel benötigte Teilnetze hinzuzufügen oder zu entfernen. So ist das Netz im Stande, seine Größe dynamisch zum Kontext zu vergrößern oder zu verkleinern.

Die Veränderung des Petrinetzes durch Regelanwendungen soll einen weiteren Aspekt konstatieren. In Abschnitt 2.4 wurde bereits der Begriff des Konfliktes erläutert. Dieser ergibt sich, wenn zwei aktivierte Transitionen um ein Token einer Stelle konkurrieren. Es bestehen demnach mehrere Verwendungsmöglichkeiten für das Token (bzw. Ressource/Objekt). Zugleich kennzeichnen solche Konflikte auch einen lokalen Nichtdeterminismus. Dieser stellt in gewisser Weise eine Abstraktion der vom Bewohner vorliegenden Entscheidungsgründe bzw. -mechanismen dar. Beim Nichtdeterminismus wird durch das Petrinetz nicht vorgegeben, welche Transition als nächstes schaltet.

Wie später im Abschnitt 4.3 zu erkennen ist, lässt sich dieser Sachverhalt in der Modellierung erkennen. In vielen Situationen können verschiedene Regeln angewendet oder unterschiedliche Transitionen geschaltet werden. Analog zum Verhalten des Bewohners ergibt sich in der Modellierung also auch ein nichtdeterministischer Kontext aus Regelanwendungen und Schalten von Transitionen, bei dem vom Petrinetz nicht vorgegeben wird was passiert.

Bereits bei den Themen der Kontextinterpretation in intelligenten Wohnungen [vgl. Voskuhl (2012)] und [vgl. Tennstedt (2011)] sind zeitliche Abhängigkeiten zu finden. Dort wird die in [Allen (1983)] vorgestellte „*interval-based temporal logic*“ eingeführt. Sie dient der Einordnung voneinander abhängiger Aktionen. Die Aktionen werden in zeitlicher Relation zueinander gesetzt, wodurch die Beziehung und die Abhängigkeit der Aktionen untereinander dargestellt werden. Besonders beim Erkennen von komplexen Aktionen ist eine solche Einteilung in Relationen nützlich. Solche Relationen können der intelligenten Wohnung helfen die Aktionen zu identifizieren, zeitlich einzuordnen und dann, falls gefordert, entsprechende Maßnahmen auszuführen.

Es gibt noch eine weitere Beziehung zwischen Aktionen, die in dieser Arbeit eine tragende Rolle einnimmt. Hier ist neben der zeitlichen auch die kausale Abhängigkeit zu nennen. In manchen Situationen bestimmt die Kausalität den Ablauf von Ereignissen. Sie bezeichnet das Zusammenspiel von Ursache und Wirkung.

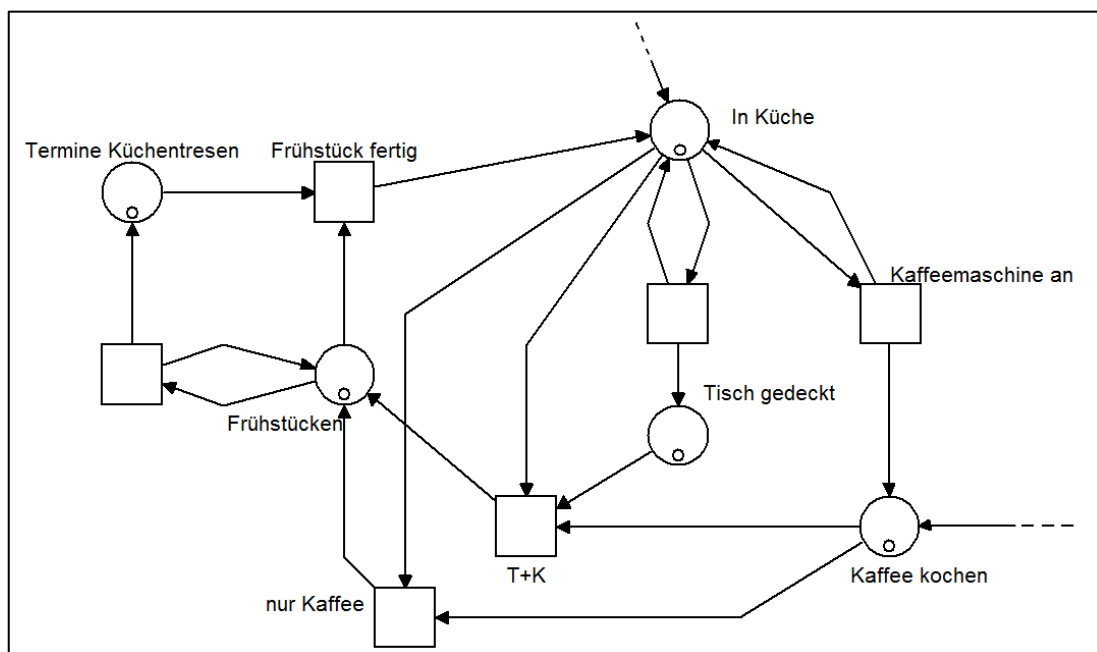


Abbildung 4.5: Teil des statischen Gesamtüberblicks

Anhand von einem Teil des statischen Gesamtüberblicks in Abbildung 4.5 lässt sich der Kontext der Abhängigkeiten gut verdeutlichen. Wenn der Bewohner frühstücken möchte, muss er zuerst einmal in der Küche sein. Um zu frühstücken kann der Bewohner Kaffee kochen und/oder den Tisch decken. Danach hat er die Möglichkeit zu frühstücken. Die Ursache in dieser kausalen Beziehung liegt hierbei im Kaffee kochen und/oder Tisch decken. Es muss also nicht zwingend eine Ursache sein, sondern es kann auch mehrere Ursachen geben. Das Frühstücken ist die resultierende Wirkung. Gleichzeitig beinhaltet die kausale

Beziehung auch eine zeitliche Reihenfolge. Denn erst wenn der Bewohner in der Küche ist und erst wenn er Kaffee gekocht hat und/oder den Tisch gedeckt hat, kann er frühstücken. Bei der Kausalität treten Ursache und Wirkung in einer chronologischen Reihenfolge auf, weil die Wirkung erst durch die Ursache ausgelöst wird.

Der wesentliche Inhalt der Szenarien sind die Handlungen und Aktivitäten des Bewohners. Die Wohnung versucht diese Aktivitäten zu erkennen. Zeitangaben haben dabei zwar schon einen Einfluss, sind in dieser Arbeit aber abstrahiert worden. Möchte man Zeiten mit in die Petrinetze integrieren, könnte man auf sogenannte Zeitbehaftete Petrinetze [vgl. [Wang \(1998\)](#)] zurückgreifen. Obwohl Zeitpunkte nicht beachtet werden, spielen dennoch zeitliche Abhängigkeiten, Nichtdeterminismus und vor allem kausale Abhängigkeiten eine bedeutende Rolle.

4.3 Modellierung der Szenarien

Nachdem das Living Place Hamburg und die Szenarien in Abschnitt 4.1 und einige Prinzipien der Modellierung in Abschnitt 4.2 vorgestellt wurden, soll nun in diesem Abschnitt die Modellierung der Szenarien erläutert werden.

Nach einigen Modellierungsentscheidungen werden drei Szenarien nacheinander vorgestellt und nach jedem Szenario die sich daraus ergebende Modellierung beschrieben. Dabei werden nicht alle Netze und Regeln abgebildet. Die Modellierung von Szenario A wird dabei detailliert und ausführlich beschrieben. Der komplette Regelsatz und alle Netze von Szenario A sind im Anhang zu finden. Bei Szenario B und C werden nur wichtige und interessante Aspekte beleuchtet. Um auch die komplette Modellierung von Szenario B und Szenario C zu begutachten, sind alle Netze und Regeln (inklusive aller Löschregeln) auf der beiliegenden CD enthalten.

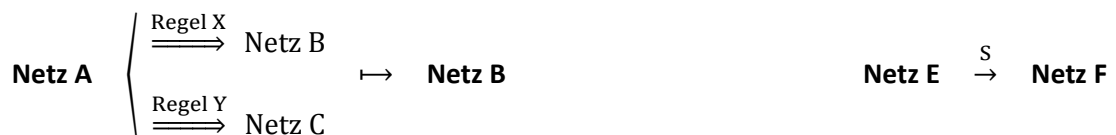
4.3.1 Grundlegende Modellierungsentscheidungen

Jedes Szenario vom morgendlichen Ablauf beginnt mit einem Basisnetz NO (siehe Abbildung 4.6), welches das Schlafen des Bewohners darstellt. Es beinhaltet die Stellen *Im Bett* und *Outside Living Place*, wobei die Stelle *Im Bett* entsprechend ein Token besitzt. Das Basisnetz besteht aus den beiden Stellen, weil es den Anfang und das Ende repräsentiert. Das Morgen Szenario beginnt immer damit, dass der Bewohner im Bett liegt und schläft. Es endet mit dem Verlassen der Wohnung.



Abbildung 4.6: Basisnetz N0 aller Szenarien

Wie bereits im vorherigen Abschnitt 4.2 erläutert wurde, wird durch den Einsatz der rekonfigurierbaren Petrinetze und deren Regeln, eine Veränderung des Netzes bewirkt und eine Art Netzfluss geschaffen. Dieser Netzfluss soll neben der graphischen Petrinetz Darstellung auch formal abgebildet werden. Hierzu sollen die folgenden Beispiele dienen.



Das Netz A hat die Möglichkeit Regel X oder Regel Y anzuwenden. Je nachdem, welche der Regeln angewendet wird, sind Netz B und Netz C die resultierenden Folgenetze. In diesem Fall sind Regel X und das Folgenetz B grau markiert. Es wurde also Regel X angewendet und entsprechend Netz B hinter dem Pfeil als Auswahl zum Folgenetz aufgeführt.

Mit dieser formalen Darstellungsweise soll vor allem der in Abschnitt 4.2 bereits erläuterte nichtdeterministische Kontext aus Regelanwendungen und Schalten von Transitionen zur Geltung gebracht werden.

Im rechten Beispiel wird das Schalten im Petrinetz abgebildet. Das ‚S‘ steht fürs Schalten und symbolisiert eine Netzveränderung durch Schalten einer Transition, bei dem von Netz E in Netz F übergegangen wird. Welche Transition geschaltet wird ist hier nicht dargestellt. Dies kann den Petrinetzen entnommen werden. Es ist auch möglich, dass ein Netz zwischen mehreren Regeln und zwischen mehreren Schaltvorgängen auswählen kann.

Bei dieser formalen Abbildung der Regelauswahl werden nur Regeln angegeben, die auch anwendbar sind. Alle Regeln, die aufgrund der negativen Anwendungsbedingung nicht ausführbar sind, werden nicht mit aufgeführt. Ansonsten würde durch die größere Anzahl an aufgeführten Auswahlmöglichkeiten die Übersichtlichkeit beeinträchtigt werden.

Wie im Abschnitt 4.2.1 erläutert wurde, gibt es zwei unterschiedliche „Arten“ von Regeln. Es gibt Regeln, die Teilnetze hinzufügen, und es gibt Löschrregeln, die Teilnetze wieder löschen. Die Anwendung der Löschrregeln wird in den ersten beiden Modellierungen der Szenarien nicht beachtet. Das Anwenden der Löschrregeln soll vor allem Bestandteil in der Modellierung von Szenario C sein und ausführlich beschrieben werden. Auch der komplette Abbau des Netzes wird dort erläutert, der auch am Ende der ersten beiden Modellierungen möglich wäre. Hier wird das Augenmerk jedoch auf andere Aspekte gerichtet, so dass auf eine Beschreibung vom Abbau der Netze in den Modellierungen von Szenario A und B verzichtet wurde.

Die zwei verschiedenen „Arten“ von Regeln stehen in folgender Beziehung zueinander. Während einerseits Teilnetze von den Regeln R1 – R35 hinzugefügt werden, bauen die Löschrregeln LöRe36 – LöRe70 die Teilnetze wieder ab. Für jede Regel, die ein Teilnetz hinzufügt, gibt es also eine korrespondierende Löschrregel, die das entsprechende Teilnetz wieder entfernt. Mit Ausnahme von Löschrregel LöRe71 sind demnach alle anderen Netze jeweils als Paar für dieselben Teilnetze verantwortlich. Eine Übersicht aller Regeln und Löschrregeln ist im Anhang (siehe Abschnitt A.1) zu finden.

Am Ende des Abschnitts 3.4 wurde eine Definition für rekonfigurierbare Petrinetze angegeben, so wie sie in dieser Arbeit verwendet wird. Demzufolge besteht ein RKPN aus einem dekorierten S/T-Netz $RKdekN$ und einem Satz von Regeln \mathcal{R} . Jedem Szenario wird also ein Satz von Regeln zugeordnet. Da es sich beim ersten Szenario um einen kurzen, schnellen Arbeitstag handelt, ist nicht der komplette Satz der Regeln, sondern nur eine kleinere Auswahl enthalten. Alle grau unterlegten Regeln (siehe Übersicht im Anhang A.1) sind nicht im Szenario A enthalten. Bei Szenario B und C besteht \mathcal{R} hingegen aus allen Regeln, also R1 – R35 und LöRe36 – LöRe71. Bezogen auf die Regelmenge trifft folgende Notation zu:

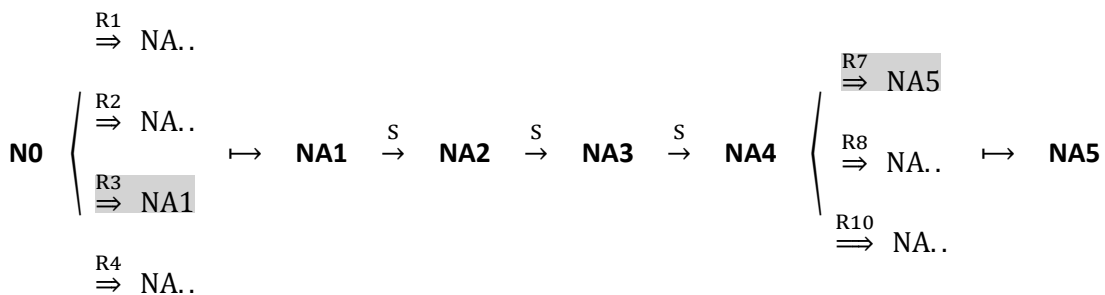
$$\text{Szenario A} \subseteq \text{Szenario B} = \text{Szenario C}$$

4.3.2 Szenario A: "kurz, schnell (Arbeitstag)"

1. Der Bewohner liegt in seinem Bett und schläft.
2. Er möchte mit dem Wecker geweckt werden.
3. Es ist bereits hell durch das Tageslicht.
4. Der Wecker klingelt.
5. Der Bewohner ist aufgestanden und der Wecker ist deaktiviert.
6. Der Bewohner zieht sich seine Kleidung an.
7. Der Bewohner geht in die Küche.
8. Sobald der Bewohner die Küche betritt, wird die Kaffeemaschine gestartet.
9. Der Kaffee ist fertig und der Bewohner trinkt den Kaffee am Küchentresen.
10. Während er trinkt, werden auf dem Küchentresen alle Termine angezeigt.
11. Der Bewohner hat den Kaffee ausgetrunken.
12. Der Bewohner geht ins Bad.
13. Er geht ans Waschbecken, putzt sich die Zähne und macht sich frisch.
14. Der Bewohner verlässt die Wohnung.

4.3.3 Modellierung von Szenario A

Bei dem ersten Szenario handelt es sich um einen Arbeitstag, an dem der Bewohner nur einen kurzen und schnellen Morgen verbringt. Dies könnte beispielsweise daran liegen, dass die Wohnung durch Verkehrsmeldungen einen Stau registriert. Da der Bewohner mit dem Auto zum ersten Termin fahren möchte, wird sich die Fahrtzeit voraussichtlich verlängern. Also verkürzt sich die Zeit für den morgendlichen Ablauf dementsprechend. Der Bewohner liegt in seinem Bett und schläft. Er hat am Abend vorher angegeben, dass er gerne mit dem Wecker geweckt werden möchte. Es ist bereits Tageslicht vorhanden und deshalb hell im Schlafzimmer. Das Licht muss demnach nicht eingeschaltet werden.



Ausgehend vom Basisnetz N0 gibt es die Möglichkeit, vier verschiedene Regeln anzuwenden, bestehend aus Regel R1 – R4. Regel R1 und R2 betreffen nur das Aufstehen ohne Wecker. In Regel R3 und R4 wird der Wecker genutzt, aber zusätzlich zwischen *Licht an* oder *Licht aus* unterschieden. Die Wohnung lässt das Licht aus und entscheidet sich für die Nutzung des Weckers. In diesem Fall wird also Regel R3 (Abbildung 4.7) genutzt, um ins nachfolgende Netz NA1 überzugehen. Die NAC verhindert ein mehrmaliges Anwenden der Regel.

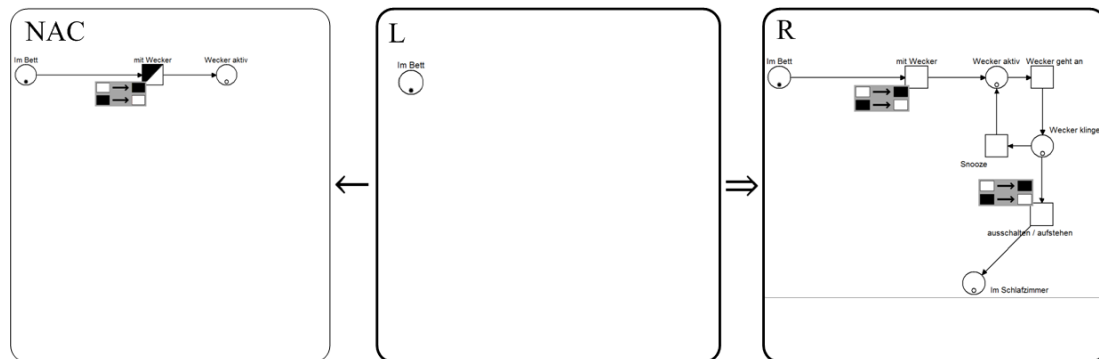


Abbildung 4.7: Regel R3

Wie in der rechten Seite der Regel R3 zu erkennen ist, besitzen die Transitionen „mit Wecker“ und „ausschalten / aufstehen“ eine *rnw*-Erneuerungsfunktion für ihre Labels (tlb). Diese Funktion ändert das entsprechende Label beim Schalten der jeweiligen Transition von weiß auf schwarz und von schwarz auf weiß. Bei der Modellierung von Szenario B (siehe Abschnitt 4.3.5) wird dieser Sachverhalt erneut aufgegriffen und seine Bedeutung hinsichtlich der Anwendung der Löschrregeln erläutert.

Es folgt ein dreimaliges Schalten von Transitionen, die das Klingeln des Weckers und das Aufstehen des Bewohners simulieren. Zu welchem Zeitpunkt der Wecker klingelt, kann die Wohnung aufgrund von verschiedenen Informationen feststellen. Beispielsweise lassen sich die Termine des Bewohners von einem Kalenderagenten bestimmen oder externe Informationen wie Wetter- oder Verkehrslage ermitteln [vgl. Voskuhl (2012)].

Der Bewohner ist aufgestanden und steht nun im Schlafzimmer, dargestellt durch das Netz NA4. Hier hat der Bewohner erneut verschiedene Auswahlmöglichkeiten, was er als nächstes tun möchte. Er geht zum Kleiderschrank, öffnet die Tür und zieht sich neue Kleidung an. Um diesen Vorgang zu beschreiben, benötigt das Petrinetz die Regel R7. Das Netz NA5 (siehe Abbildung 4.8) zeigt den Zustand nach der Anwendung dieser Regel. Die Labels der Transitionen „mit Wecker“ und „ausschalten / aufstehen“ sind schwarz, da sie jeweils einmal geschaltet haben und ihr Label durch die Erneuerungsfunktion geändert wurde.

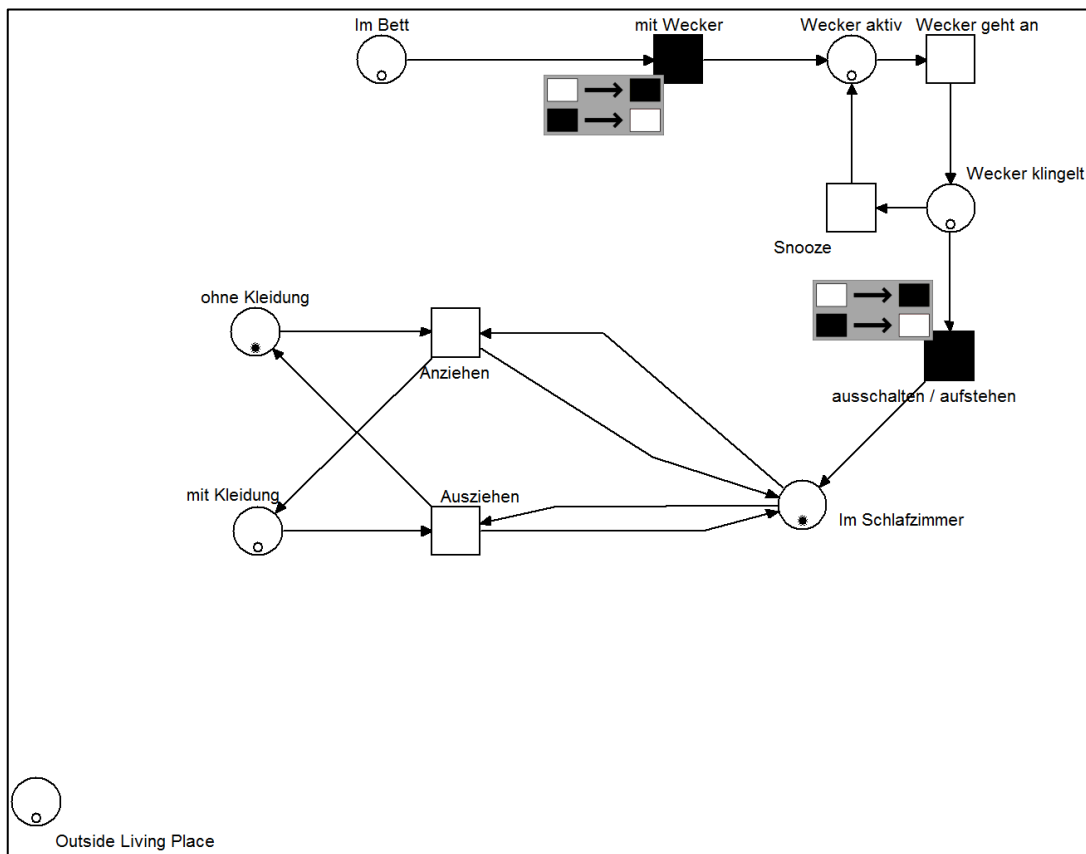
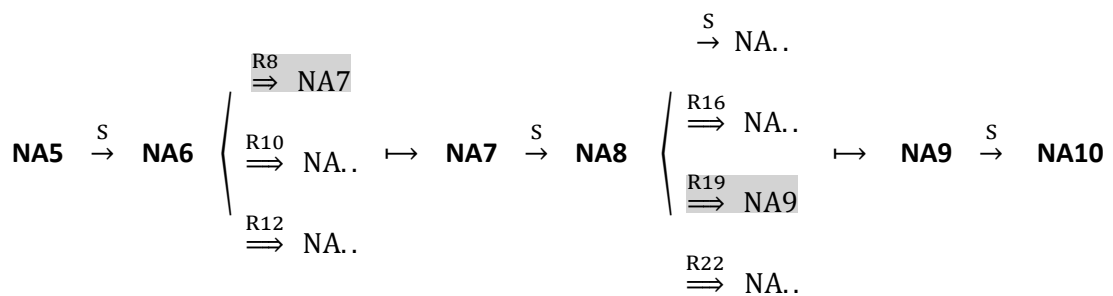


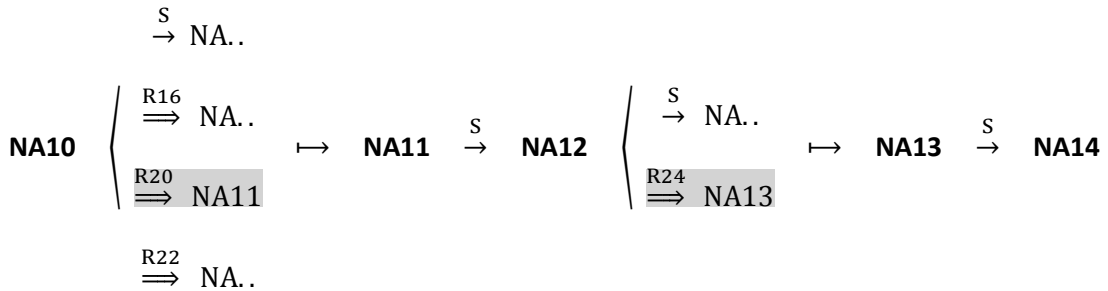
Abbildung 4.8: Netz NA5

Nachdem der Bewohner sich angezogen und die passende Transition geschaltet hat, wird nun Regel R8 angewendet. Während der Bewohner die Küche betritt, wird dies von der Wohnung erkannt und automatisch die Kaffeemaschine gestartet. Anhand von Regel R19 wird das Netz verändert. Das Starten der Kaffeemaschine wird durch entsprechendes Schalten der Transition aktiviert. Formal dargestellt mit dem Schaltübergang von Netz NA9 zu Netz NA10.



Solange der Bewohner noch in der Küche steht und der Kaffee kocht, besitzt er die Freiheit die Küche auch wieder zu verlassen. Diese Entscheidungsfreiheit wird im nächsten Schritt

festgelegt. Der Bewohner entscheidet sich dafür seinen Kaffee zu trinken, folglich wird Regel R20 (Abbildung 4.9) angewendet. Hier wird das Teilnetz „Frühstücken“ hinzugefügt. Der Name der Stelle kann sowohl für das klassische Frühstück¹³ stehen, als auch für das einfache Kaffee trinken. In diesem Fall wird davon ausgegangen, dass der Bewohner zumindest einen Kaffee trinkt, weshalb auf die Option „nur Tisch decken“ verzichtet wurde.



Regel R20 hat zwei negative Anwendungsbedingungen. NAC 1 soll einen mehrmaligen Gebrauch dieser Regel vermeiden. Demgegenüber verhindert NAC 2 eine falsche Auswahl der Frühstücks-Regeln. Das Teilnetz „Frühstücken“ kann durch zwei Regeln hinzugefügt werden. Zum einen wenn nur Kaffee getrunken wird (wie in diesem Szenario) und zum anderen beim klassischen Frühstück. NAC 2 verhindert also den Gebrauch der Regel, die für das klassische Frühstück benötigt wird.

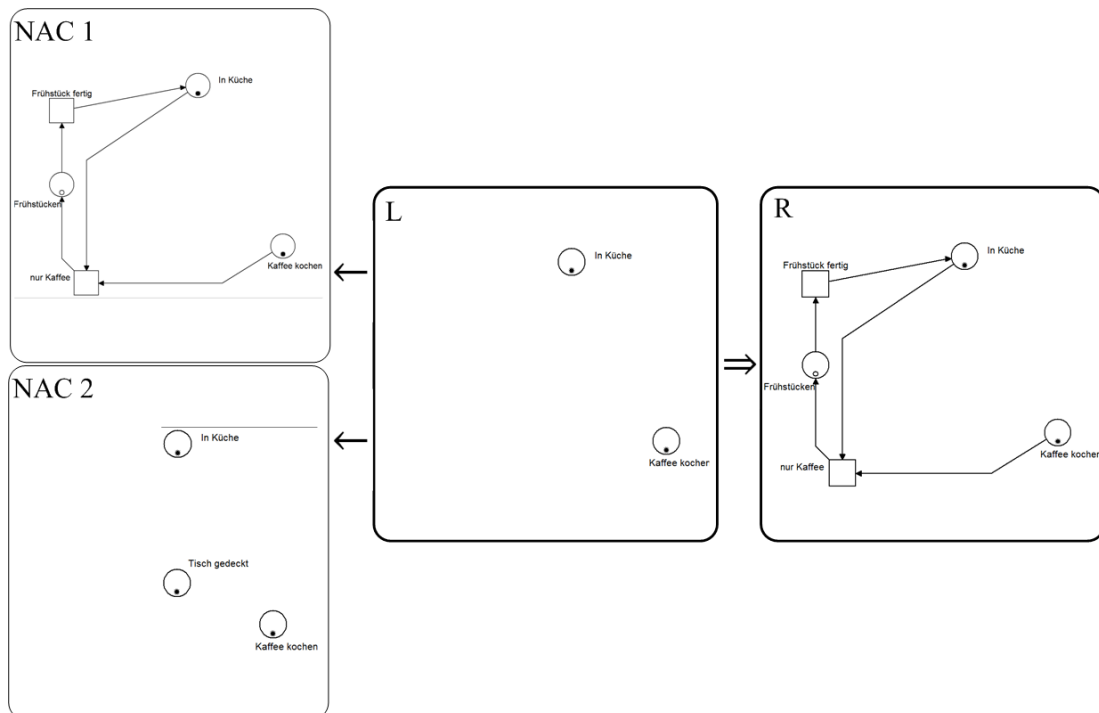
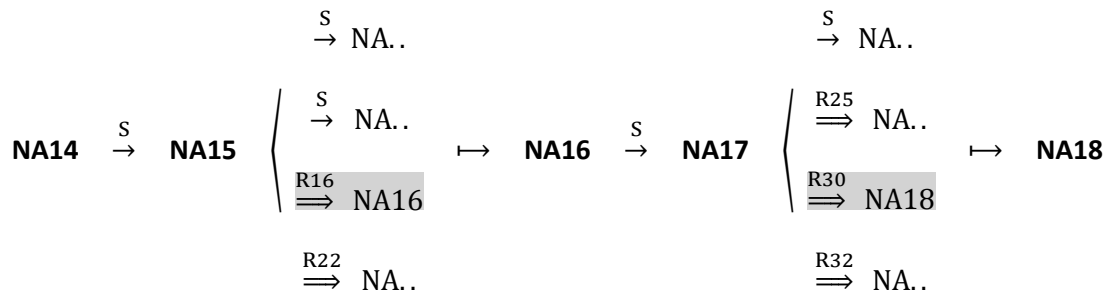


Abbildung 4.9: Regel R20

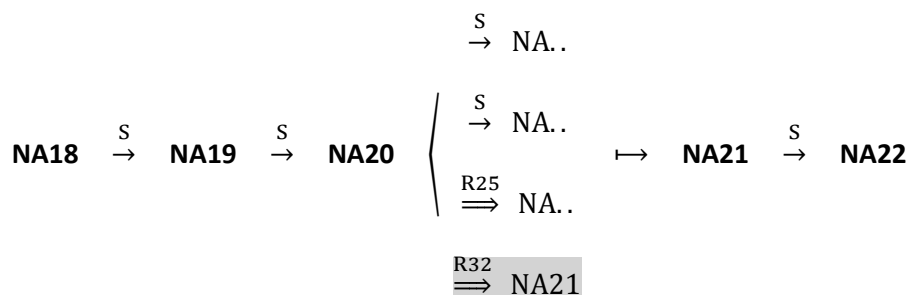
¹³ Das klassische Frühstück besteht hier aus einem Getränk (Kaffee) und Speisen (auf dem Tisch).

Während der Bewohner seinen Kaffee am Küchentresen trinkt, werden auf diesem Küchentresen seine tagesaktuellen Termine angezeigt. Regel R24 fügt diesen Kontext zum Netz hinzu. Der Küchentresen könnte in diesem Fall eine Multitouch-Oberfläche haben, wie sie in [Rahimi und Vogt (2011)] vorgestellt wird. Auf dieser Oberfläche könnten nicht nur die Termine, sondern auch andere Inhalte oder Dokumente angezeigt werden, mit denen der Bewohner diese per Berührung interaktiv organisieren oder sogar bearbeiten kann.

Hat der Bewohner den Kaffee ausgetrunken, wird der Zustand des Frühstücks verlassen. Erneut steht der Bewohner in der Küche und hat die freie Entscheidung darüber, was er als nächstes machen möchte. In diesem Szenario geht er in das Badezimmer, realisiert durch das Anwenden der Regel R16. Im Badezimmer angekommen möchte er sich die Zähne putzen und sich am Waschbecken frisch machen. Das Netz NA17 wird durch die Regel R30 in das Netz NA18 überführt.



Die Schaltreihenfolge $\text{NA18} \xrightarrow{S} \text{NA19} \xrightarrow{S} \text{NA20}$ beschreibt den Vorgang, der sich durchs Waschen und Zähne putzen ergibt. Nach dieser Aktion möchte der Bewohner die Wohnung verlassen. Regel R32 verändert das Netz NA20 hinsichtlich dieser Entscheidung.



Zum Abschluss wird in Abbildung 4.10 das Netz NA22 präsentiert. Bestehend aus allen Petrinetzkomponenten, die dynamisch während des gesamten Szenarios erstellt wurden, wird in diesem Netz erkennbar, dass sich nur noch ein Token im Netz befindet. Dieses Token liegt auf der Stelle „Outside Living Place“ und symbolisiert das Ende dieses ersten Szenarios.

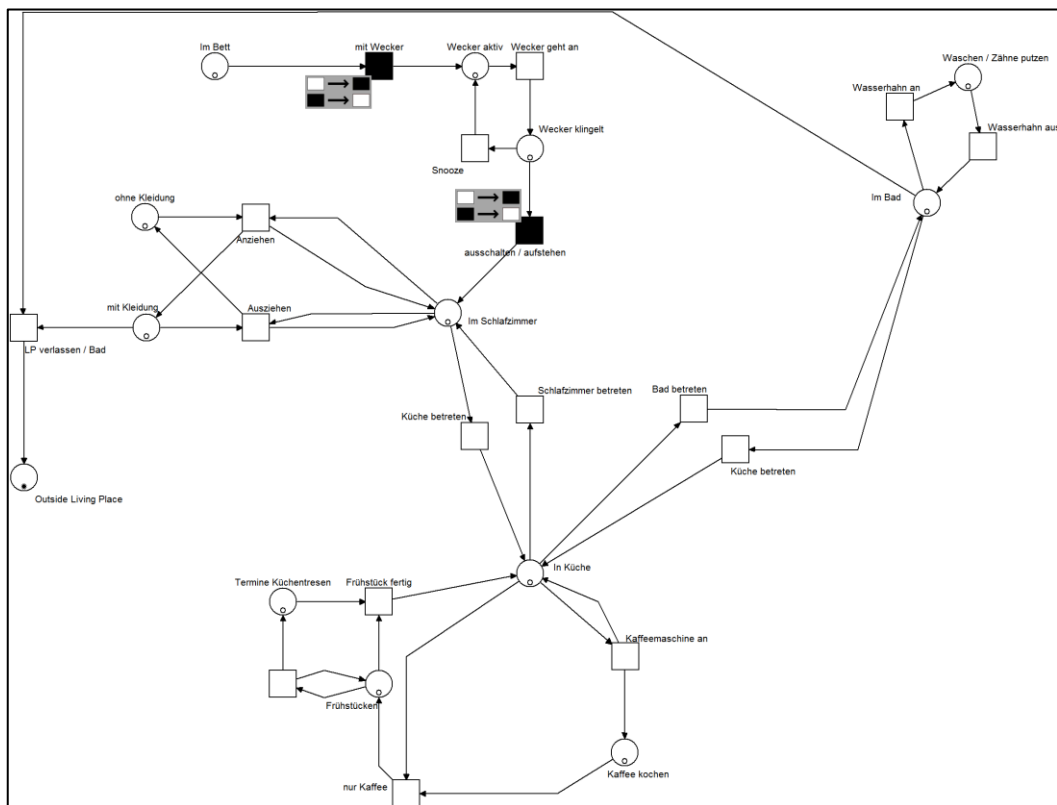


Abbildung 4.10: Netz NA22

Wie schon in Abschnitt 4.3.1 hingewiesen wurde, werden in diesem Szenario die Löschrregeln nicht beachtet. Das Netz NA22 ist nun zwar nicht mehr schaltfähig (es befindet sich in einem Endzustand¹⁴), kann aber mit den Löschrregeln abgebaut werden. Anhand des Netzes NA22 wird deutlich, dass die Transition zum Verlassen der Wohnung nur aktiv ist, wenn der Bewohner sich angezogen hat. Alle Regeln, die das Verlassen der Wohnung betreffen, wurden mit dieser Bedingung verknüpft. Ob diese Bedingung in der Realität sinnvoll ist, soll nicht Bestandteil dieser Arbeit sein.

¹⁴ Gleichzeitig auch in einer (gewollten) Verklemmung

4.3.4 Szenario B: "ausgiebig (Arbeitstag)"

1. Der Bewohner liegt in seinem Bett und schläft.
2. Er möchte mit dem Wecker geweckt werden.
3. Es ist noch kein Tageslicht vorhanden.
4. Der Wecker klingelt und das Licht geht an.
5. Der Bewohner drückt die „Snooze“-Taste und schläft kurz weiter.
6. Das Licht geht wieder aus und der Wecker klingelt nicht mehr.
7. Der Wecker klingelt nach ein paar Minuten erneut und das Licht geht wieder an.
8. Der Bewohner ist aufgestanden und der Wecker ist deaktiviert.
9. Der Bewohner geht ins Bad.
10. Er geht in die Dusche.
11. Während er in der Dusche ist, wird die Kaffeemaschine in der Küche gestartet und
12. die Handtücher im Badschrank werden erwärmt.
13. Der Bewohner ist fertig mit duschen.
14. Er trocknet sich mit den aufgewärmten Handtüchern ab.
15. Er geht zurück ins Schlafzimmer.
16. Der Bewohner zieht sich seine Kleidung an.
17. Anschließend geht der Bewohner in die Küche.
18. Er deckt den Tisch auf.
19. Der Kaffee ist fertig und der Bewohner frühstückt.
20. Während der Bewohner frühstückt, werden auf dem Tisch alle Termine angezeigt.
21. Er beendet das Frühstück.
22. Der Bewohner geht noch einmal ins Bad.
23. Er geht ans Waschbecken und frisiert sich die Haare.
24. Der Bewohner verlässt die Wohnung.
25. Das Licht wird ausgeschaltet.

4.3.5 Modellierung von Szenario B

Das zweite Szenario behandelt einen Arbeitstag, in dem der Bewohner einen ausgiebigen Morgen hat. Sein erster Termin könnte beispielsweise erst am späten Vormittag sein, weshalb er sich zwar vom Wecker wecken lässt, aber trotzdem genügend Zeit für ein ausgedehntes Morgenprogramm hat. Wie bereits zu Beginn dieses Abschnittes erläutert, wird dieses Szenario nicht so ausführlich beschrieben wie Szenario A. Lediglich einige interessante Auszüge sollen hier vorgestellt werden.

Im Unterschied zu Szenario A ist bei diesem Szenario noch kein Tageslicht vorhanden. Deshalb veranlasst die Wohnung beim Klingeln des Weckers zusätzlich noch das Anschalten des Lichtes. Die rechte Seite von Regel R4 (Abbildung 4.11) zeigt, dass die Stelle „Licht an / SZ“ mit im Teilnetz enthalten ist.

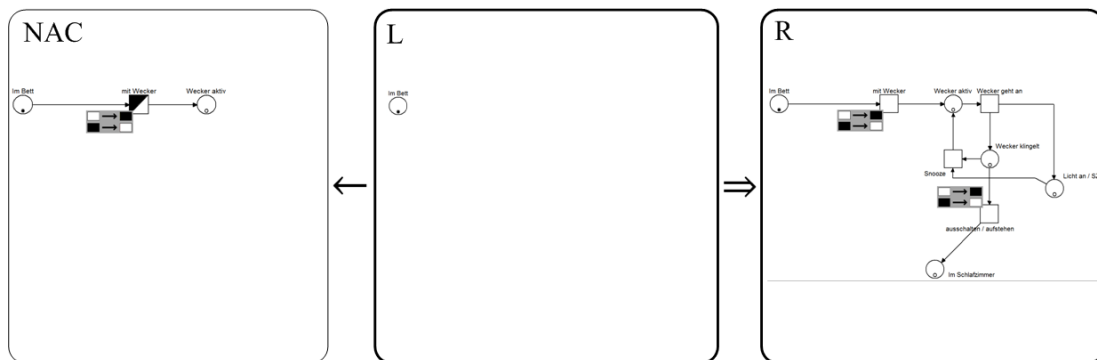


Abbildung 4.11: Regel R4

Nachfolgend soll nun der Zweck der Transitions-Labels erläutert werden. Bei der Modellierung von Szenario A wurde bereits die *rnw*-Funktion der beiden Transitionen „mit Wecker“ und „ausschalten / aufstehen“ beschrieben. Wenn der Bewohner aufgestanden ist und im Schlafzimmer steht, dann sind die Labels beider Transitionen schwarz, weil sie beide geschaltet haben. Egal wie oft dieses Teilnetz „durchlaufen“ werden würde, sobald der Bewohner wieder im Schlafzimmer steht, sind die Farben der beiden Labels identisch. Umgekehrt bedeutet dies, dass bei ungleicher Farbe der Labels dieses Teilnetz gerade „aktiv“ ist. Diese Eigenschaft kann man sich zu Nutze machen. In diesem Fall können die beiden Labels bei der Anwendung der entsprechenden Löschrregel nützlich sein. Die korrespondierende Löschrregel zu Regel R4 ist LöRe39. In Abbildung 4.12 wird die Löschrregel LöRe39 dargestellt.

Die negativen Anwendungsbedingungen NAC 2 und NAC 3 zeigen, dass dieses Teilnetz nicht entfernt werden darf, wenn die Farbe der beiden Transitions-Labels „mit Wecker“ und „ausschalten / aufstehen“ ungleich sind.

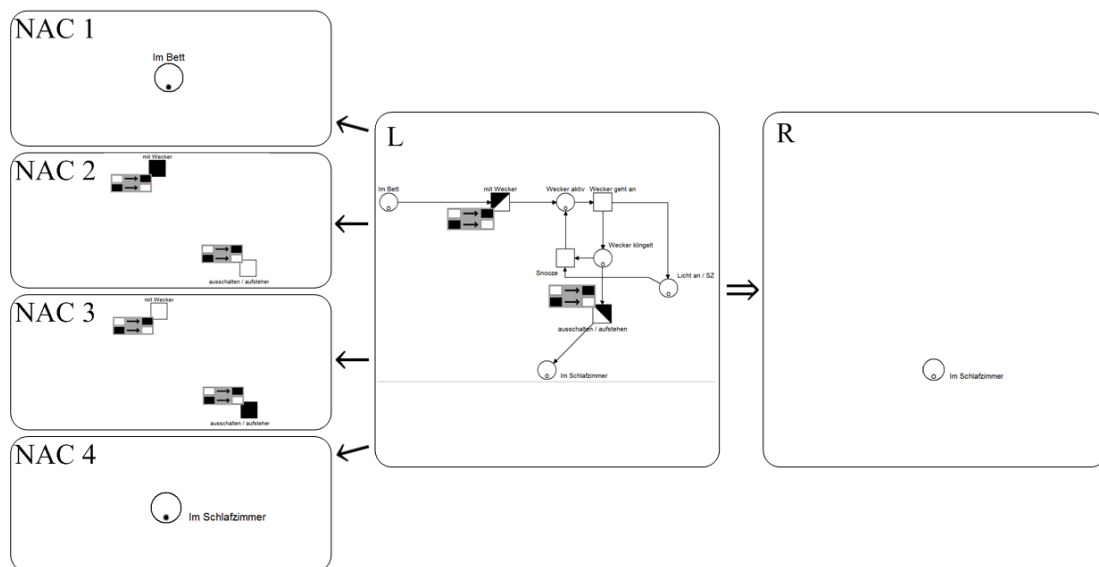


Abbildung 4.12: Löschregel LöRe39

Gesetz den Fall die Netzstruktur zwischen diesen beiden Transitionen wäre um ein vielfaches größer und ein Token würde durch verschiedene Stellen „wandern“, wäre der Vorteil dieser Variante noch deutlicher erkennbar. Denn wenn alle Stellen mit einer separaten NAC überprüft werden müssten, ob sich ein Token auf ihnen befindet, würde die Anzahl der benötigten NACs äquivalent zur Anzahl der Stellen steigen. Bei dieser Variante ist nur die „Abfrage“ der Labels nötig. Die Farben der beiden Transitions-Labels in der linken Seite der Löschregel sind irrelevant, weshalb die Labels zur Hälfte schwarz und zur Hälfte weiß sind.

Auch im Falle eines Raumwechsels wird bei der Regelanwendung das Licht beachtet. Beispielsweise können die Regeln R9 und R11 nur angewendet werden, wenn das Licht in der Wohnung an ist. Beim Wechsel von einem Raum zum anderen wird das Licht beim Verlassen ausgemacht und beim Eintreten in den neuen Raum angemacht. Mittels Lokationstechnologie, die u.a. in [Karstaedt (2012)] beschrieben wird, könnte die Wohnung immer den Aufenthaltsort des Bewohners erfassen und bei einem Raumwechsel verbunden mit dem An- und Ausschalten des Lichtes Anwendung finden. In Abbildung 4.13 wird im Ausschnitt des Netzes NB7 Punkt 9 des Szenarios abgebildet. Wenn der Bewohner vom Schlafzimmer ins Badezimmer geht, dann wird das Licht im Schlafzimmer ausgeschaltet und im Badezimmer das Licht angeschaltet.

Bei den Szenarien in dieser Arbeit wird davon ausgegangen, dass sich die Tageslicht- und entsprechend auch die Wohnungslichtverhältnisse während des Morgenablaufs nicht verändern. Wird das Licht zu Beginn aktiviert, bleibt es auch im restlichen Szenario immer an. Eine Integration von wechselnden Lichtverhältnissen in den Netzen wäre auch umsetzbar. Lediglich die Regeln, die einen Raumwechsel betreffen, müssten geändert bzw. ergänzt werden.

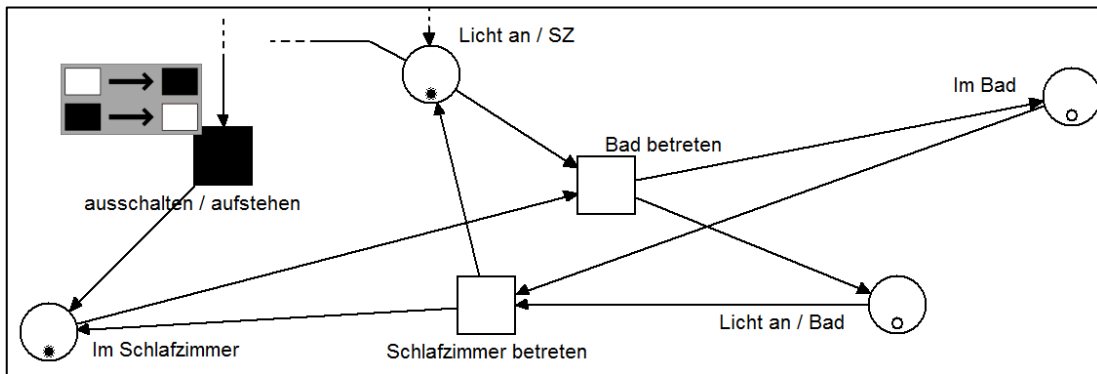
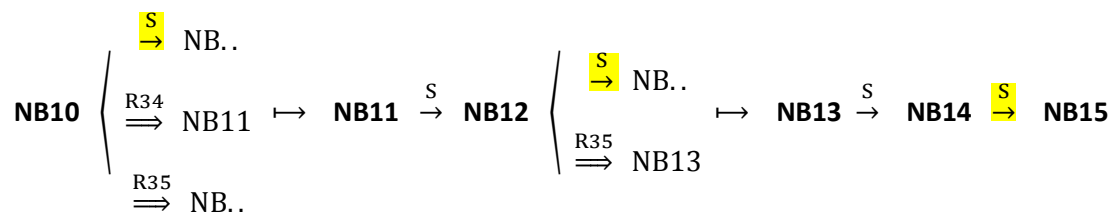


Abbildung 4.13:Ausschnitt aus Netz NB7

Die folgende formale Netzflussdarstellung zeigt die Entwicklung des Petrinetzes von Netz NB10 bis zu Netz NB15. Im Netz NB10 ist der Bewohner gerade im Badezimmer und duscht. Wie im Szenario beschrieben, wird während der Duschphase die Kaffeemaschine in der Küche gestartet (Punkt 11), umgesetzt mit Regel R34, und die Handtücher im Badschrank erwärmt (Punkt 12), dargestellt durch Regel R35. Netz NB15 zeigt letztendlich den Zustand, in dem der Bewohner mit dem Duschen fertig ist und im Anschluss wieder im Badezimmer steht.



Durch diese Schalt- und Regelanwendungsphase lässt sich nicht nur der Nichtdeterminismus identifizieren, sondern auch eine resultierende Nebenläufigkeit. In NB10 hat das Netz die Möglichkeit zu schalten oder 2 Regeln anzuwenden. Hier entsteht also der schon in Abschnitt 4.2 beschriebene lokale Nichtdeterminismus. Zunächst ist hier unbestimmt, welche Auswahl getroffen wird. Es ist allerdings beobachtbar, durch das Szenario, welcher Weg eingeschlagen wird. Im Netz NB12 besteht ebenfalls eine Auswahlmöglichkeit, bei der auch wieder geschaltet oder erneut Regel R35 angewendet werden kann.

Nachdem die beiden Regeln angewendet wurden, symbolisiert der Netzübergang von Netz NB14 zu Netz NB15 das Ende vom Duschen des Bewohners. Die gelb unterlegte Schaltdarstellung stellt diesen Vorgang dar. Wie man sehen kann, ist das Schalten dieser Transition auch bei den vorherigen Auswahlmöglichkeiten von Netz NB10 und Netz NB12 eine Alternative.

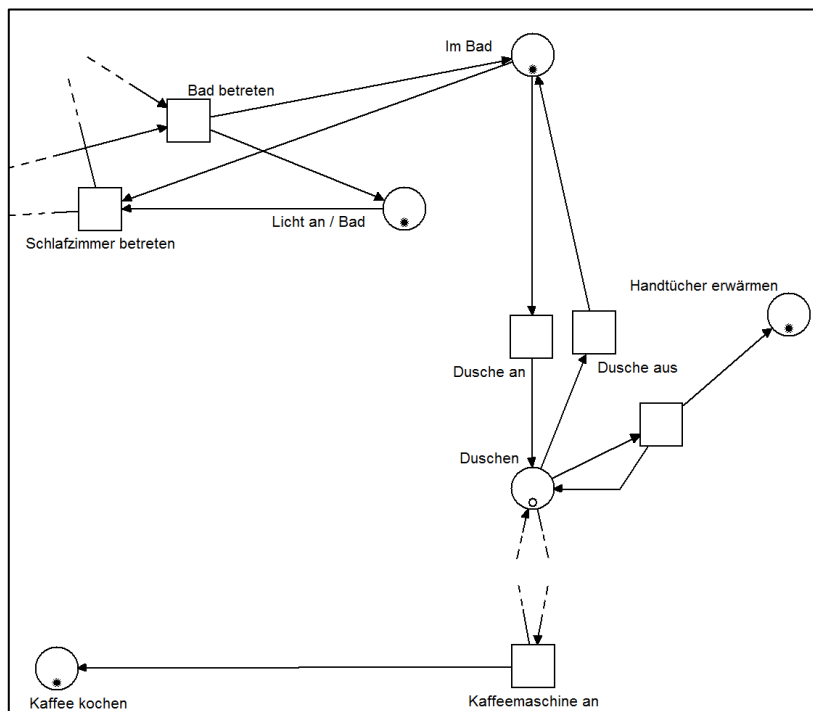


Abbildung 4.14:Ausschnitt aus Netz NB15

Aus diesem Nichtdeterminismus bzw. aus dieser Konfliktsituation entwickelt sich ein nebenläufiger Zustand. In Abschnitt 4.2 wurde bereits erwähnt, dass auch eine Markierung möglich ist, bei der mehrere Stellen ein Token besitzen. In Abbildung 4.14 lässt sich dieser Umstand erkennen. Der Ausschnitt von Netz NB15 zeigt vier mit Token belegte Stellen, resultierend aus den vorgestellten Netzflussübergängen von Netz NB10 zu NB15. Zwar kann das Tokenspiel der Stellen *Kaffee kochen* und *Handtücher erwärmen* nicht unabhängig weitergeführt werden, was eine tatsächliche nebenläufige Situation ergeben würde, dennoch ist es eine „quasi-nebenläufige“ Konstellation. Wenn man sich vorstellt, dass die beiden Stellen einen weiteren eigenen Ablauf konstruieren würden, wären nebenläufige Schritte denkbar. Das Kaffee kochen könnte einen weiteren Prozess innerhalb der Wohnung anstoßen. Wenn der Bewohner beispielsweise während des Kaffeetrinkens gerne die digitale Zeitung des Tages liest, dann könnte die Wohnung bereits veranlassen, dass die Zeitung aus dem Internet heruntergeladen und auf dem Küchentresen zur Verfügung gestellt wird.

Im Gegensatz zu Nebenläufigkeit gibt es auch Konstrukte, die mehrere Teilabläufe wieder vereinen. Hierbei ist zu erkennen, dass sich die Rollen von Transitionen und Stellen vertauschen. Mehrere Stellen mit Token liegen im Vorbereich einer Transition, so dass diese aktiviert und schaltfähig ist. Der Ausschnitt vom Netz NB26 (Abbildung 4.15) veranschaulicht diesen Zusammenhang.

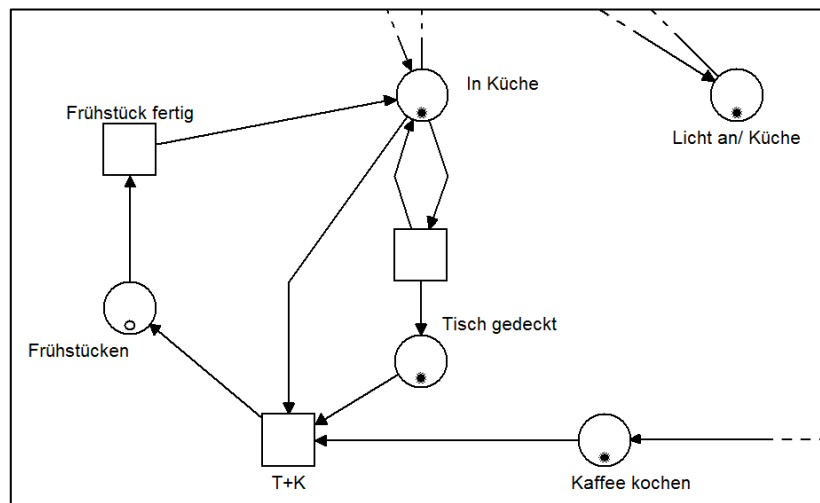


Abbildung 4.15:Ausschnitt aus Netz NB26

Die Transition $T+K$ ¹⁵ ist erst aktiviert, wenn der Tisch gedeckt ist, der Kaffee kocht und der Bewohner sich in der Küche befindet. Nur wenn die Vorbereitsbedingungen erfüllt sind, kann die Transition schalten. Solche Gebilde bieten also die Möglichkeit die Anzahl der Token im Petrinetz wieder zu reduzieren.

¹⁵ $T+K = \underline{T}$ isch ist gedeckt und Kaffee ist gekocht (entspricht dem klassischen Frühstück)

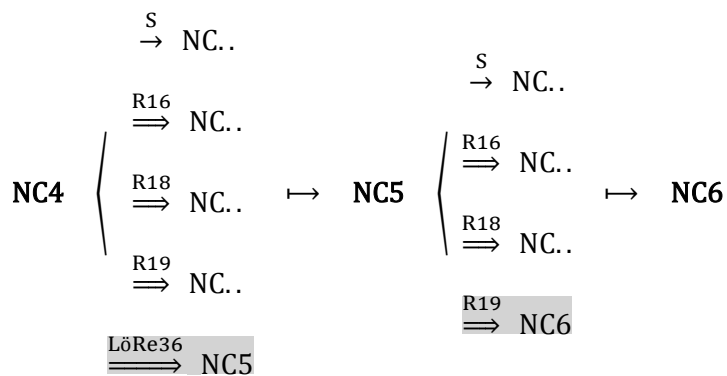
4.3.6 Szenario C: "ausgiebig (Wochenende)"

1. Der Bewohner liegt in seinem Bett und schläft.
2. Er möchte nicht mit dem Wecker geweckt werden, sondern vom Tageslicht.
3. Es ist Tageslicht vorhanden.
4. Der Bewohner ist aufgestanden.
5. Er geht in die Küche.
6. Sobald der Bewohner die Küche betritt, wird die Kaffeemaschine gestartet.
7. Der Kaffee ist fertig und der Bewohner trinkt den Kaffee am Küchentresen.
8. Weil Wochenende ist, werden auf dem Küchentresen keine Termine angezeigt.
9. Der Bewohner hat den Kaffee ausgetrunken.
10. Er geht zurück ins Schlafzimmer.
11. Der Bewohner zieht sich sein Kleidung an.
12. Er geht ins Bad.
13. Er geht ans Waschbecken und putzt sich die Zähne.
14. Der Bewohner geht wieder zurück ins Schlafzimmer.
15. Er zieht sich seine Kleidung aus.
16. Er legt sich erneut ins Bett und liest ein Buch.
17. Nach einer gewissen Zeit steht der Bewohner wieder auf.
18. Der Bewohner geht ins Bad.
19. Er geht in die Dusche.
20. Während er in der Dusche ist, werden die Handtücher im Badschrank erwärmt.
21. Der Bewohner ist fertig mit duschen.
22. Er trocknet sich mit den aufgewärmten Handtüchern ab.
23. Dann geht er ans Waschbecken und macht sich frisch.
24. Er geht zurück ins Schlafzimmer.
25. Der Bewohner zieht sich erneut seine Kleidung an.
26. Der Bewohner verlässt die Wohnung.

4.3.7 Modellierung von Szenario C

Im Gegensatz zu den beiden vorherigen Szenarien soll das Augenmerk in diesem Szenario auf das Anwenden der Löschrregeln gerichtet werden. Dabei sollen, ähnlich wie beim Szenario B, nur interessante Auszüge aus dem Szenario beschrieben werden.

Beim Szenario C handelt es sich um einen Tag am Wochenende, an dem der Bewohner nicht zur Arbeit muss und auch keine Termine hat. Demnach möchte er nicht vom Wecker geweckt werden. Zum Zeitpunkt des Aufstehens ist bereits Tageslicht vorhanden, weshalb kein Licht benötigt wird. Der Bewohner geht in die Küche.



Hier gibt es nun verschiedene Möglichkeiten zur Fortsetzung. Im Szenario wird die Kaffeemaschine gestartet, damit der Bewohner einen Kaffee trinken kann. Das Netz hat jedoch auch die Möglichkeit seine Struktur wieder zu verändern, indem es eine Löschrregel anwendet. Durch das Anwenden von Löschrregel LöRe36 kann das „Aufstehen“ Teilnetz wieder gelöscht werden (siehe Abbildung 4.16).

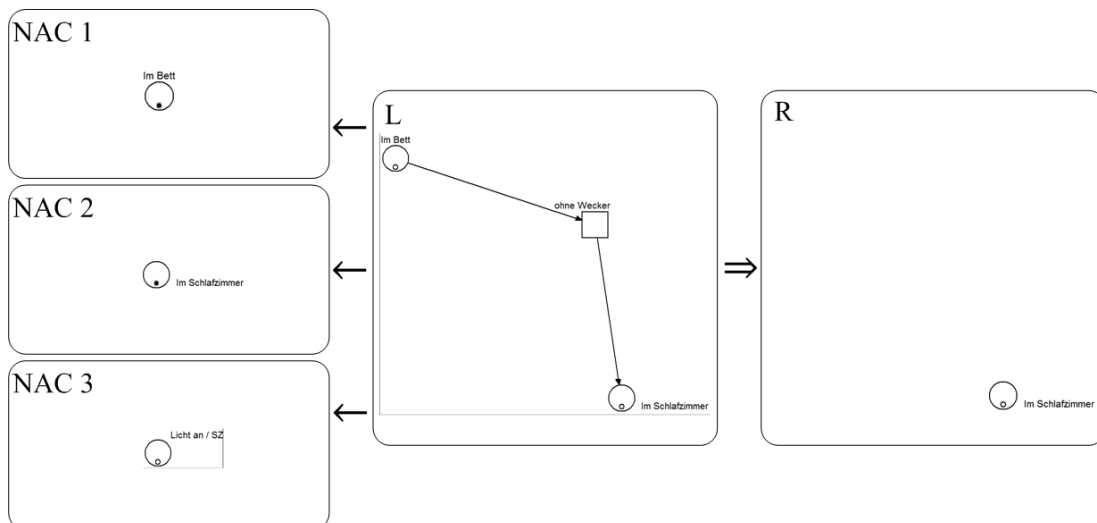


Abbildung 4.16: Löschrregel LöRe36

Beim Übergang von Netz NC4 zu Netz NC5 wird die Anwendung der Löschregel LöRe36 veranschaulicht. Die drei verschiedenen negativen Anwendungsbedingungen NAC 1, 2 und 3 gewährleisten, dass diese Regel nicht angewendet werden darf, wenn sich der Bewohner noch im Bett (NAC 1) oder im Schlafzimmer (NAC 2) befindet. Außerdem verhindert NAC 3 eine Anwendung, falls das Licht in der Wohnung nach dem Aufstehen angeschaltet wurde. Bei angeschaltetem Licht würde Löschregel LöRe37 in Betracht kommen.

Die Löschregeln sollen zur Verkleinerung der Netze verwendet werden (siehe Abschnitt 4.2.1) und haben in einigen Situationen keinen Einfluss auf den nächsten Schritt. Dieser Umstand wird besonders beim Übergang von Netz NC4 zu Netz NC6 deutlich. Beim ersten Übergang von Netz NC4 zu Netz NC5 gibt es fünf Auswahlmöglichkeiten. Das Netz kann eine Transition schalten, es kann drei Regeln und eine Löschregel anwenden. Hier wird also das Netz verkleinert durch die Anwendung von Löschregel LöRe36 (grau markiert). Beim zweiten Übergang (Netz NC5 zu Netz NC6) gibt es immer noch die gleichen Möglichkeiten wie zuvor, ausschließlich der Löschregel. Das Anwenden der Löschregel LöRe36 hat also keinerlei Einfluss auf den nächsten Schritt.

Nachdem der Bewohner mit dem Kaffee trinken fertig ist, geht er wieder zurück ins Schlafzimmer (Punkt 10). Das Teilnetz „Frühstücken“ wird durch zwei Löschregeln wieder entfernt. Löschregel LöRe54 und LöRe55 realisieren diese Veränderung. Der Vorteil der dynamischen Veränderung soll anhand der Abbildung in 4.17 gezeigt werden.

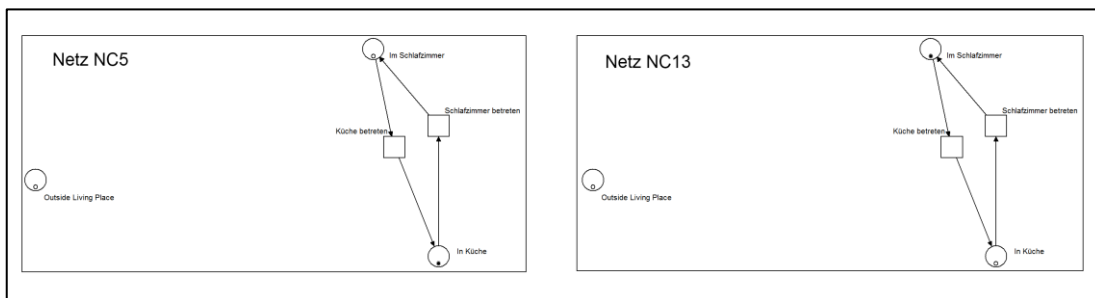
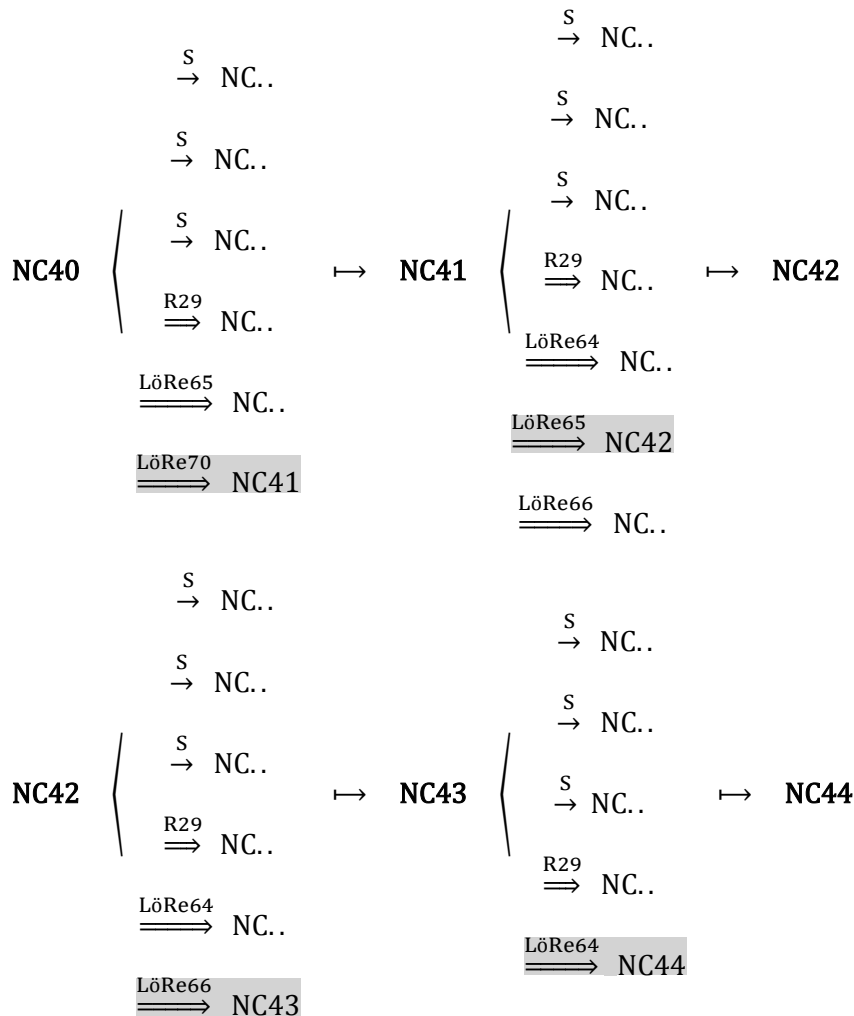


Abbildung 4.17: Vergleich der Netze NC5 und Netz NC13

Auf der linken Seite der Abbildung wird das Netz NC5 dargestellt und auf der rechten Seite wird das Netz NC13 abgebildet. Verglichen mit dem Szenario entspricht Netz NC5 Punkt 5 und Netz NC13 entspricht Punkt 10. In beiden Fällen wechselt der Bewohner den Raum. Obwohl beide Netze vom Verlauf gesehen unterschiedliche Zeitpunkte widerspiegeln, ist die Größe beider Netze identisch. Hier wird die Eigenschaft der Netztransformation dahingehend genutzt, dass die Größe der RKPn durch Anwenden von Regeln (in diesem Fall durch das Anwenden der Löschregeln) maßgeblich beeinflusst werden kann. Ohne das Anwenden der Löschregeln würden sich beide Netze deutlich in ihrer Größe unterscheiden. Das Entfernen von Kontexten (bzw. Teilnetzen), die im aktuellen Ablauf nicht benötigt werden, erhöht die Anschaulichkeit und Lesbarkeit der RKPn und verbessert i. Allg. das Verständnis beim Betrachter.

Der Bewohner hat sich zwischenzeitlich erneut ins Bett gelegt, um ein Buch zu lesen. Im Anschluss daran geht er ins Bad, duscht sich, trocknet sich ab, macht sich frisch und geht wieder ins Schlafzimmer. An dieser Stelle wendet das Netz eine Reihe von Löschrregeln an, die den Badezimmer Kontext entfernt.



Die formale Darstellung der Netzübergänge von Netz NC40 bis zu Netz NC44 illustriert diesen Verlauf. Schrittweise werden vier Löschrregeln angewendet (grau markiert), die nacheinander alle Komponenten entfernen. Dabei stehen die Löschrregeln teilweise in einer abhängigen Beziehung zueinander. Wie beim Übergang von Netz NC40 zu Netz NC41 zu erkennen ist, hat das Netz die Auswahl zwischen drei Transitionsschaltungen, der Regel R29 und zwei Löschrregeln. Nach dem Anwenden von Löschrregel LöRe70 ergibt sich ein anderes Bild. Die Auswahlmöglichkeiten beim nächsten Übergang sind größer geworden. Neben den Transitionsschaltungen und der Regel, hat das Netz zusätzlich die Auswahl zwischen drei Löschrregeln, LöRe64, LöRe65 und LöRe66. Diese Mehrzahl an Auswahlmöglichkeiten resultiert durch das Vorhandensein negativer Anwendungsbedingungen in LöRe64 (NAC 2)

und LöRe66 (NAC 4). Diese beiden Löschregeln können nämlich erst dann angewendet werden, wenn LöRe70 bereits ausgeführt wurde (vorausgesetzt dieses Teilnetz wurde zuvor von Regel R35 hinzugefügt). Danach können die Teilnetze sukzessive abgebaut werden. Dieser Prozess wird in Abbildung 4.18 abgebildet. Darin werden Ausschnitte von den Netzen NC40 bis NC44 gezeigt und der schrittweise Abbau der Teilnetze demonstriert.

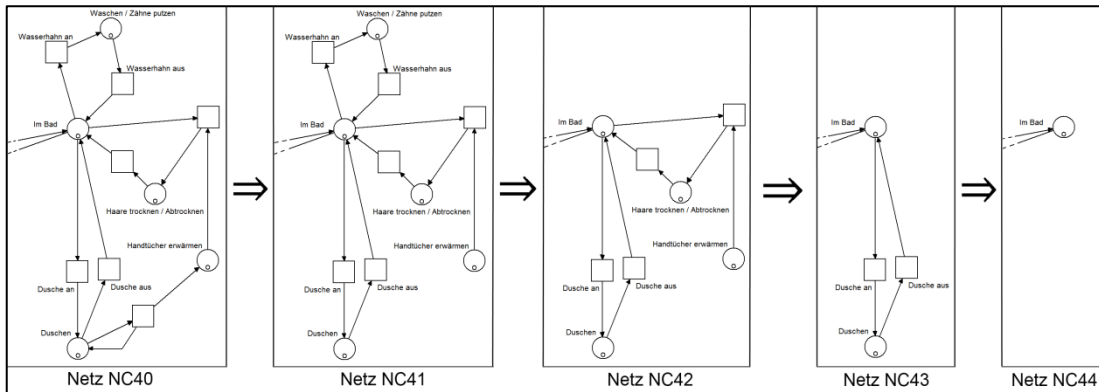
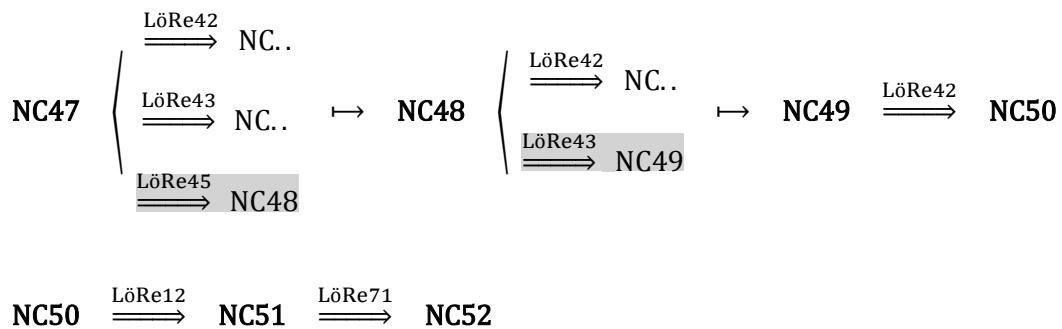


Abbildung 4.18: Netzausschnitte von Netz NC40 - NC44

Auch dieses Szenario endet mit dem Verlassen der Wohnung. Wie bereits angesprochen wurde (Abschnitt 4.3.1), kann sich das Netz fast komplett abbauen und bis auf eine Stelle alle Netzkomponenten löschen. Die folgende Darstellung der letzten Regelanwendungen gibt die Situation wieder.



Bei den letzten Übergängen stehen nur noch Löschregeln zur Auswahl. Alle Löschregeln, die für den kompletten Abbau des Netzes nötig sind, können am Ende angewendet werden. Da sich das Netz in NC47 bereits in einem Endzustand¹⁶ befindet, ist das Anwenden von Regeln oder das Schalten von Transitionen nicht mehr möglich. Das Netz hat nur noch die Auswahl zwischen unterschiedlichen Löschregeln, so dass kurz vor dem Ende nur noch vier Stellen im Netz enthalten sind (siehe Abbildung 4.19).

¹⁶ Endzustand im Sinne einer Verklemmung. Es gibt keine aktivierten Transitionen.

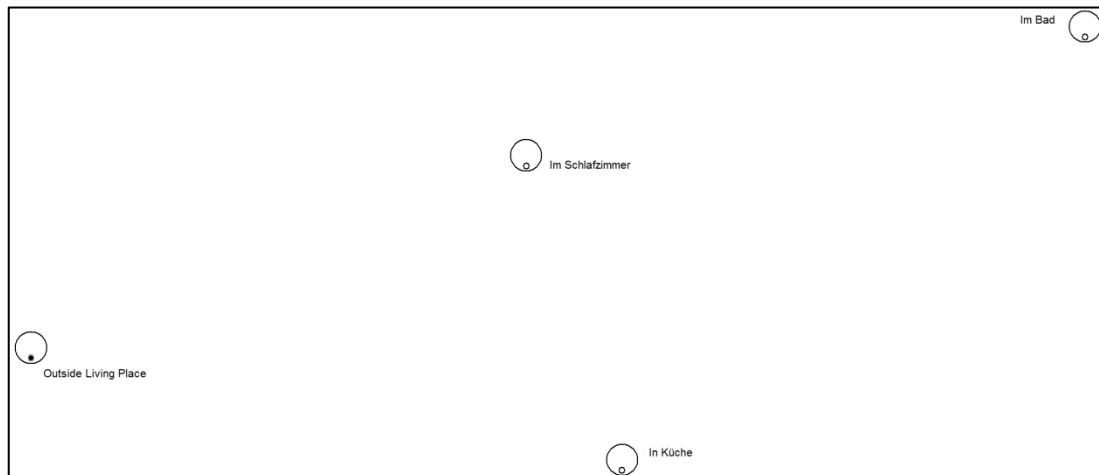


Abbildung 4.19:Netz NC51

Der Netzzustand, der in Netz NC51 gezeigt wird, soll den Aktionsraum des Bewohners repräsentieren (siehe Abschnitt 4.2.1). Bevor die „Räume“ der Wohnung gelöscht werden, müssen alle anderen Netzelemente entfernt worden sein. Löschregel LöRe71 kann nur dann angewendet werden, wenn sich diese Netzsituation ergeben hat. Egal welches Szenario stattfindet oder auf welche Art das Netz verändert wird, jeder mögliche Ablauf endet in Netz NC51 bzw. in Netz NC52 (vorausgesetzt das Netz wird komplett abgebaut).

4.4 Bewertung der Modellierung

Bei der Entwicklung dieser Morgen-Szenarien stand die Interaktion zwischen dem Bewohner und dem Living Place im Vordergrund. Der Bewohner soll in seinem morgendlichen Ablauf von der Wohnung „intelligent“ unterstützt werden. Die Wohnung erkennt Kontexte und Aktionen vom Bewohner, um daraufhin, möglichst ohne wahrgenommen zu werden, Unterstützung zu liefern.

Wie in den Modellierungen der Szenarien gezeigt wurde, lassen sich nebenläufige Prozesse erkennen. Während der Bewohner z.B. am Duschen ist, startet die Wohnung die Kaffeemaschine in der Küche. Möchte man also einerseits die Handlungen des Bewohners und andererseits die resultierenden, unterstützenden Aktionen der Wohnung analysieren, dann können sich unabhängige, nebenläufige Vorgänge ergeben. Es konnte gezeigt werden, dass eine Umsetzung dieser Modellierung möglich ist und speziell rekonfigurierbare Petrinetze für solche Prozesse geeignet sind.

Auch die Eigenschaft der Netzstrukturveränderung von RKN bietet sich für die Modellierung der Morgen-Szenarien an. Durch die Netztransformationen kann das Netz sich immer dem Kontext anpassen und zudem Teilnetze entfernen, die nicht mehr benötigt

werden. So bleibt die Größe des Netzes variabel und kann je nach Anforderung vergrößert oder verkleinert werden. Dazu sind besonders die bei dieser Modellierung angewandten Löseregeln interessant, die die Netzgröße während der Schaltdynamik reduzieren können. So bleibt das Netz übersichtlich.

Im Vergleich zu einer Modellierung mit klassischen S/T-Netzen steigt bei rekonfigurierbaren Petrinetzen die Komplexität bei der Modellierung. Es muss nicht nur eine Netzstruktur im Sinne des zu entwickelnden Systems modelliert werden, sondern eine große Schwierigkeit liegt zudem in der Formulierung der Regeln. Diese Problematik wurde auch in dieser Arbeit deutlich. Insbesondere durch die negativen Anwendungsbedingungen war eine ständige, akkurate Beobachtung der einzelnen Entwicklungsschritte entscheidend. Beim Löschen von Stellen, Transitionen und vor allen Dingen bei Kanten musste aufmerksam deren Anbindung und Kontakt zu anderen Elementen überprüft werden.

5 Analysemöglichkeiten

In Kapitel 4 wurde eine Modellierung der Morgen-Szenarien vorgestellt. Es wurde gezeigt, dass die Modellierung mit rekonfigurierbaren, dekorierten S/T-Netzen geeignet ist, um ein abstrahiertes Modell von der Interaktion eines Bewohners mit einer intelligenten Wohnung während des morgendlichen Ablaufs zu konstruieren.

Das folgende Kapitel soll einen Blick auf einige, verschiedene Möglichkeiten zur Analyse der modellierten Netze werfen. Dabei werden keine konkreten Verfahren präsentiert, sondern es wird vielmehr versucht, ein paar Ansätze zur Analyse der rekonfigurierbaren, dekorierten S/T-Netzen zu liefern.

5.1 Einführung

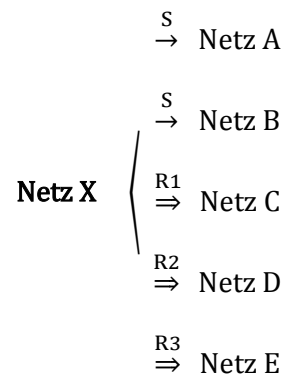
Petrinetze sind neben ihrer Modellierungsmöglichkeit auch deshalb beliebt, weil sie unter bestimmten Randbedingungen sehr gut zu analysieren sind [vgl. [Kleuker \(2009\)](#)]. Es gibt eine Vielzahl von verschiedenen Methoden und Werkzeugen, um eine mathematisch fundierte Analyse der Netze zu erstellen. Ein Grund für die Analyse von modellierten Systemen, ist das Erkennen von möglichen Problemen, die sich im Systemverlauf ergeben können. Identifiziert man die Probleme während der Analyse, kann noch vor der Realisierung des Systems entscheidend reagiert werden. Dieser Aspekt macht eine Analyse besonders bei großen Systemen unabdingbar und spiegelt ihren Stellenwert wider.

Der Hauptzweck der Modellierung von Systemen durch mathematische Strukturen ist die Untersuchung ihrer Verhaltensmöglichkeiten [vgl. [Baumgarten \(1990\)](#)]. Bereits in Abschnitt 2.4 wurden einige Eigenschaften von Petrinetzen beschrieben, die sich während des Tokenspiels ergeben können oder die das Verhalten der Petrinetze beschreiben. Wie auch schon erwähnt wurde, werden solche Eigenschaften häufig zu Analysezielen genutzt. Diese Analysemöglichkeiten sollen nun zum Teil auch auf die in dieser Arbeit vorgestellten rekonfigurierbaren Petrinetze angewendet werden. Dabei soll der Versuch unternommen werden, anhand der Modellierung von Abschnitt 4.3, einige konkrete Beispiele, die die

beschriebenen Eigenschaften aufweisen, herauszuarbeiten und Ansätze zur Analyse zu entwickeln.

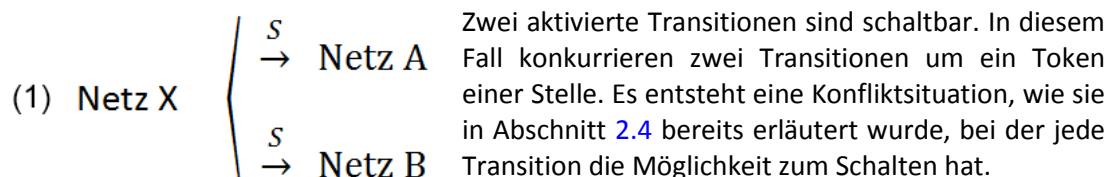
5.2 Konfluenz

Ein RKPN kann sich unter zwei verschiedenen Umständen verändern. Durch Schalten von Transitionen kann sich die Markierung des Netzes ändern und durch die regelbasierte Netztransformation kann die Netzstruktur modifiziert werden. Die formale Darstellung der Netzveränderung, wie sie in Abschnitt 4.3.1 eingeführt wurde, soll vor allem den Nichtdeterminismus hervorheben, der sich beim Anwenden von mehreren Regeln oder der Möglichkeit des Schaltens von mehreren Transitionen ergibt. *Netz X* kann durch das Schalten zweier Transitionen in *Netz A* oder in *Netz B* oder durch Anwenden der Regeln *R1*, *R2*, *R3* in die *Netze C*, *D* und *E* überführt werden.



Tritt nun eine solche Situation ein, bei der es mehrere Möglichkeiten zur Fortsetzung gibt, stellt sich die interessante Frage: Inwieweit beeinflusst die Auswahl den späteren Verlauf? Auf RKPN bezogen kann man diese Frage auf folgenden Analyseansatz vertiefen: Ist die Auswahl unabhängig voneinander, weil sie danach ohnehin zum selben Resultat (bzw. zum selben S/T-Netz) führt? Oder anders formuliert: Schließt die eine Auswahl die andere Auswahl aus?

Im Hinblick auf rekonfigurierbare Petrinetze können sich drei verschiedene Situationen ergeben.



(2) Netz X	$\left\{ \begin{array}{l} R1 \\ \Rightarrow \text{Netz A} \\ \\ R2 \\ \Rightarrow \text{Netz B} \end{array} \right.$	Die Anwendung von zwei Regeln ist möglich. Das Netz hat die Auswahl zwischen zwei Netztransformationen. Die Netzstruktur kann durch die beiden Transformationen verändert werden.
(3) Netz X	$\left\{ \begin{array}{l} S \\ \rightarrow \text{Netz A} \\ \\ R1 \\ \Rightarrow \text{Netz B} \end{array} \right.$	Es kann aber auch zu Situationen kommen, bei dem eine Transition schaltbar und eine Anwendung einer Regel möglich ist. Diese Konstellation vereinigt beide Veränderungsarten, die Veränderung der Markierung des Netzes und die Veränderung der Netzstruktur.

Diese Überlegungen von Unabhängigkeit führen zu dem Begriff der *lokalen Konfluenz*. Nach [Ehrig u.a. (2007)] ist die parallele Unabhängigkeit von (1) und (2) bereits bekannt und wurde auch schon nachgewiesen. Insbesondere bei zwei Netztransformationen (2) ist diese Situation verwandt mit dem *Church-Rosser-Theorem* [vgl. Ehrig u.a. (2010)]. Diese Situation kann kritisch für die *lokale Konfluenz* werden, wenn die Regeln sich gegenseitig beeinflussen. Beispielsweise ist *lokale Konfluenz* nicht mehr gegeben, wenn die eine Regel die Anwendbarkeit der anderen zerstört. Bei dieser Analyse der Konfluenz wird häufig der Begriff *Kritische-Paar-Analyse* verwendet.

Bei rekonfigurierbaren Petrinetzen ist im Hinblick auf deren Hauptmerkmal besonders interessant, unter welchen Bedingungen das Schalten einer Transition und die Anwendung einer regelbasierten Netztransformation unabhängig sind, siehe Situation (3). Während die Situationen (1) und (2) bereits aus anderen Bereichen der Graphentheorie bekannt sind, ergibt sich durch die Struktur von (3) ein anderer Analyseansatz. Dieser wird durch die besondere Eigenschaft der RKPZ gegeben. Die *lokale Konfluenz* soll anhand eines Beispiels verdeutlicht werden.

In Abbildung 5.1 wird eine Situation aus dem Szenario C in Ausschnitten dargestellt. Der Bewohner trinkt den Kaffee am Küchentresen (Punkt 7).

Da beim Szenario C auch die Löschregeln einbezogen sind, hat das Netz die Auswahl zwischen der Regelanwendung *LöRe54* und dem Schalten der Transition *t* (= „Frühstück fertig“). In der formalen Darstellung vom Szenario Verlauf (siehe Abschnitt 4.3.7) wird zuerst die Löschregel *LöRe54* angewendet und anschließend die Transition *t* geschaltet, dargestellt auf der rechten Seite der Abbildung. Dies entspricht den Netzübergängen von Netz NC9 bis NC11. Das resultierende Netz NC11 ist unten zu erkennen. Es wäre auch denkbar, dass zuerst die Transition *t* geschaltet und danach Löschregel *LöRe54* angewendet wird, dargestellt auf der linken Seite. Da in diesem Fall die Ausführungsreihenfolge von *t* und *LöRe54* ebenfalls zum selben Netz führt, sind die Transition *t* und die Löschregel *LöRe54* parallel unabhängig bzw. *lokal konfluent*.

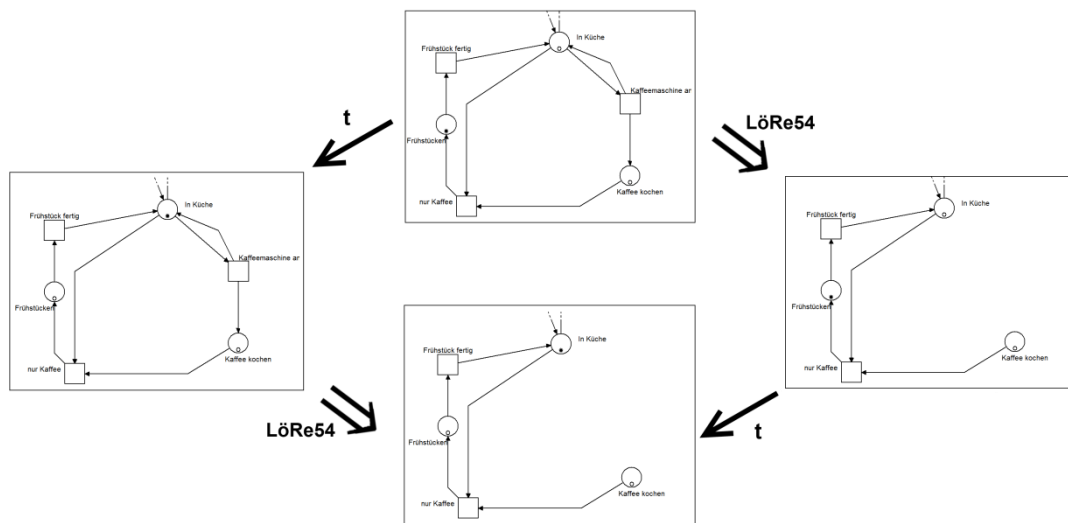


Abbildung 5.1: Lokale Konfluenz

Unter folgenden Bedingungen ist *lokale Konfluenz* gegeben (bezogen auf das Beispiel in Abbildung 5.1):

1. Die Transition t darf von der Regel $LöRe54$ nicht gelöscht werden.
2. Nach dem Schalten der Transition t muss die Markierung des Netzes groß genug sein (genügend Token vorliegen), damit ein Netzmorphismus zwischen der linken Seite der Regel und dem Netz existiert, um die Löschregel anwenden zu können.

In [Ehrig u.a. (2007)] wurde untersucht, unter welchen hinreichenden Bedingungen eine Netztransformation und das Schalten einer Transition parallel und sequentiell voneinander unabhängig sind. Dieser Ansatz wurde weiterentwickelt und auf die Untersuchung von notwendigen Bedingungen fortgeführt [vgl. Hoffmann u.a. (2010)].

Lokale Konfluenz ist, wie die Bezeichnung schon erkennen lässt, nur unter bestimmten Bedingungen lokal gegeben. Möchte man Konfluenz nicht nur lokal untersuchen, könnte man weiterhin analysieren, unter welchen Bedingungen mehrere Schalt- bzw. Regelschritte unabhängig wären.

Dieser Sachverhalt soll in Abbildung 5.2 veranschaulicht werden. Die Abbildung gibt eine Situation aus dem Szenario C wieder (Punkt 21 - 23). Da in dieser Abbildung viele Netzausschnitte abgebildet werden, sind die markanten Veränderungen gelb markiert, um die Ausdruckskraft zu erhöhen. Der Bewohner ist mit dem Duschen fertig und steht im Bad, abgebildet im linken Netz. Nun gibt es zwei Möglichkeiten für den weiteren Netzverlauf, die den nichtdeterministischen Kontext repräsentieren.



Abbildung 5.2: Konfluenz bei mehreren Schalt- und Regelschritten

Das Netz hat die Auswahl zwischen einer Regelanwendung (unterer Pfad) oder dem Schalten einer Transition (oberer Pfad). Der untere Pfad gibt den Ablauf wieder, so wie er im Szenario beschrieben wird (Netz NC34 bis Netz NC39). Eine andere Variante zeigt der obere Pfad. In beiden Pfaden wird dieselbe Regel angewendet und dieselben Transitionen geschaltet, jedoch in unterschiedlicher Reihenfolge. Beide Pfade führen zum selben Resultat, dargestellt durch das rechte Netz. Die Schalt- und Regelschritte sind also unabhängig voneinander. Diese Situation spiegelt auch *Konfluenz* wider, diesmal jedoch über mehrere Schritte.

Während sich die *lokale Konfluenz* auf eine Regel und auf einen Schaltvorgang bezieht, wird bei der *globalen Konfluenz* das gesamte Netz betrachtet. Abbildung 5.3 zeigt eine Skizze, die die globale Konfluenz darstellt.

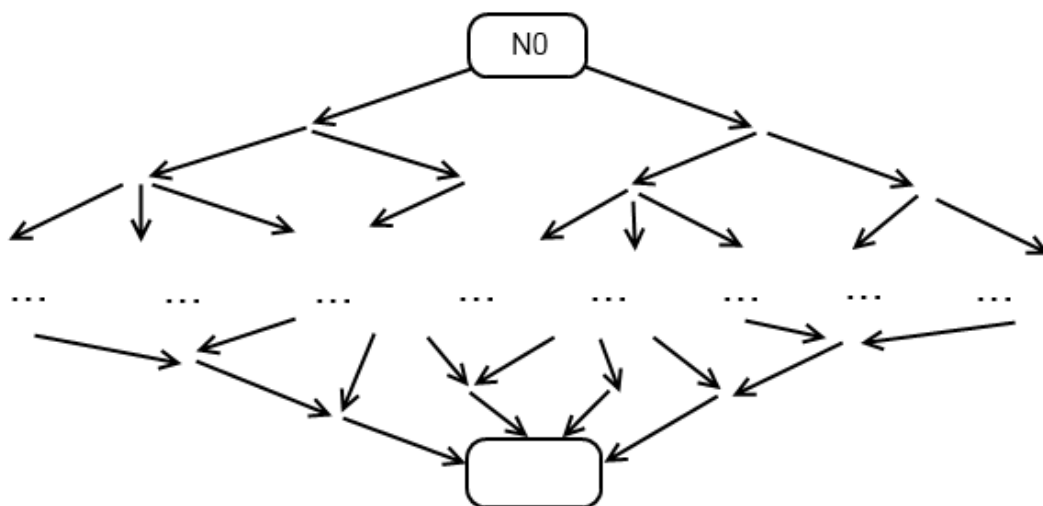


Abbildung 5.3: Globale Konfluenz (Skizze)

Jedes Szenario in dieser Arbeit beginnt mit dem Basisnetz NO. Die RKPn können entweder am Ende der Szenarien mit den Lösregeln, umgesetzt in Szenario A und B, oder während der Schaltdynamik abgebaut werden, umgesetzt in Szenario C. In allen drei Varianten endet das Netz in einem Zustand, bei dem nur noch die Stelle „Outside Living Place“ vorhanden ist, belegt mit einem Token. Werden also die RKPn in jedem Szenario per Lösregeln komplett wieder abgebaut, dann resultiert aus dieser Gegebenheit eine *globale Konfluenz*. Denn es ist völlig irrelevant, auf welche Weise sich das Netz verändert. Die verändernden Entscheidungen sind, „global gesehen“, alle unabhängig voneinander, da jeder Pfad am Ende zum selben Netz resultiert.

5.3 Erreichbarkeitsanalyse

Eine der konventionellen Analysetechniken zur Verifikation von Petrinetzen ist die Klasse der enumerativen Methoden. Diese Zählverfahren basieren auf der Konstruktion des Erreichbarkeitsgraphen [vgl. [Girault und Valk \(2003\)](#)]. In Abschnitt 2.4 wurde bereits der Erreichbarkeitsbegriff erläutert und auch die Konstruktion des Erreichbarkeitsgraphen beschrieben. Mit Blick auf den rechnerischen Aufwand, kann die Konstruktion des Erreichbarkeitsgraphen sehr aufwendig sein, weil er im Verhältnis zur Größe des Netzes exponentiell viele Knoten enthalten kann [vgl. [Valk und Moldt \(2006\)](#)].

Dennoch ist die *Erreichbarkeitsanalyse* ein sehr beliebtes Mittel, um ein Petrinetz zu analysieren und zu verifizieren. Und wie schon in Abschnitt 2.4 notiert wurde, ist laut [[Priese und Wimmel \(2008\)](#)] eine der Kernfragen zu Petrinetzen, ob ein Zustand zur Erreichbarkeitsmenge eines Petrinetzes gehört. Dies zeigt, welchen Stellenwert die Erreichbarkeitsanalyse bei Petrinetzen hat. Wie bereits in der Einführung erwähnt wurde, sollen die Analysemöglichkeiten der Petrinetze auch auf die RKPn angewendet werden. Denn obwohl RKPn eine Erweiterung der klassischen Petrinetze sind, unterscheiden sie sich von ihnen.

Vor allem durch die dynamische Veränderung der Netzstruktur grenzen sie sich von den klassischen Petrinetzen ab. Da sich die klassischen Petrinetze und die RKPn jedoch in der Basis gleichen, bietet sich also die Gelegenheit, bekannte Analysemöglichkeiten der klassischen Petrinetze auch bei den RKPn anzuwenden. Dieser Ansatz soll auch in diesem Abschnitt mit der Erreichbarkeitsanalyse von RKPn verfolgt werden.

Der Erreichbarkeitsgraph eines beschränkten Petrinetzes kann mittels Algorithmus berechnet und konstruiert werden [vgl. [Girault und Valk \(2003\)](#)]. Er liefert viele Informationen über den Zustand des Netzes, einschließlich aller erreichbaren Markierungen. Es kann z.B. analysiert werden, ob das Netz verklemmungsfrei ist oder ob es lebendig ist. Auch bei der Nutzung von RKPn ist eine solche Analyse wünschenswert.

Doch im Vergleich zu den klassischen Petrinetzen ist die Erreichbarkeitsanalyse von rekonfigurierbaren Petrinetzen nicht auf diese Weise anwendbar.

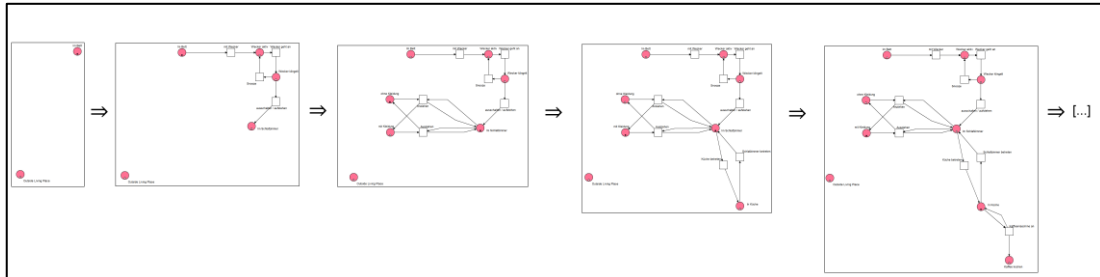


Abbildung 5.4: Iterative Veränderung der Netzstruktur (Auszug aus Szenario A)

Die in Abbildung 5.4 veranschaulichte, schrittweise Veränderung der Netzstruktur, bedingt durch Netztransformationen, soll den Unterschied verdeutlichen. In einem Erreichbarkeitsgraphen wird die Anzahl der Token von jeder Stelle notiert. Bei RKPn kann sich jedoch die Anzahl der Stellen durch die Transformationen verändern, so wie in der Abbildung dargestellt. Zu Beginn sind nur zwei Stellen vorhanden, nach einer Transformation sind es fünf Stellen und vier Transformationsschritte später sind es bereits neun Stellen (farblich illustriert). Analog dazu kann die Anzahl der Stellen durch Transformationen auch sinken.

Ein erster intuitiver Ansatz zur Konstruktion eines Erreichbarkeitsgraphen für RKPn wäre eine Darstellung der Zustandsübergänge durch Regelanwendungen und geschalteten Transitionen.

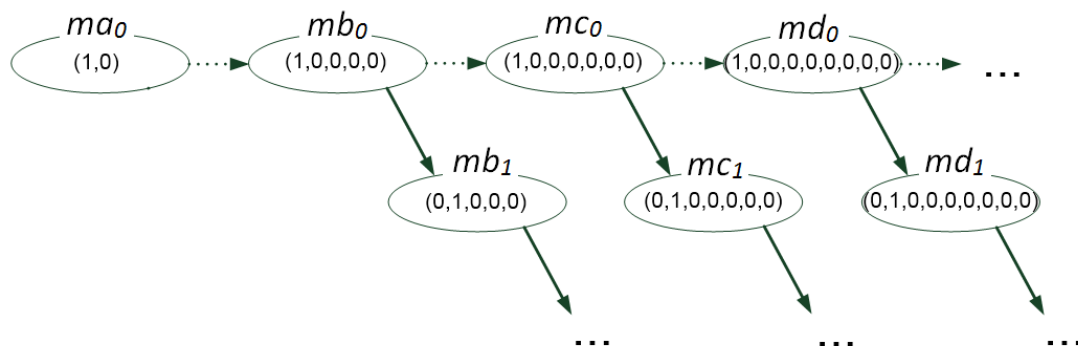


Abbildung 5.5: Hypothetischer Ansatz eines Erreichbarkeitsgraphen

In Anlehnung an den Auszug aus Szenario A (siehe Abbildung 5.4) ist dieser Ansatz in Abbildung 5.5 dargestellt, wobei einerseits „Regel-Zustandsübergänge“ (gepunktete Pfeile)

und andererseits die bekannten „Transitions-Zustandsübergänge“ (durchgezogene Pfeile) abgebildet werden. Da die Stellenanzahl in diesem Beispiel bei jeder Regelanwendung steigt, wird auch die Markierung des Netzes „größer“. Beim ersten „Regel-Zustandsübergang“ verändert sich die Markierung von $(1, 0)$ auf $(1, 0, 0, 0, 0)$. Ein solcher Erreichbarkeitsgraph wäre von Hand erstellbar, würde aber zu folgendem Problem führen: Bei einem Erreichbarkeitsgraphen wird jede Stelle an einer festgelegten Position (oder Index, siehe Abschnitt 2.4) in allen Knoten des Graphen notiert. Da in dem Ansatz von Abbildung 5.5 die Stellenanzahl nicht nur steigen, sondern auch schrumpfen kann, können die Indizes der Knoten durch immer neue Stellen belegt werden. Somit wäre eine klare Zuordnung von Stelle und Position im Knoten nicht mehr eindeutig gegeben. Die Identifizierung der Stellen bzgl. ihrer Indexpositionen im Erreichbarkeitsgraphen müsste eindeutig definiert werden.

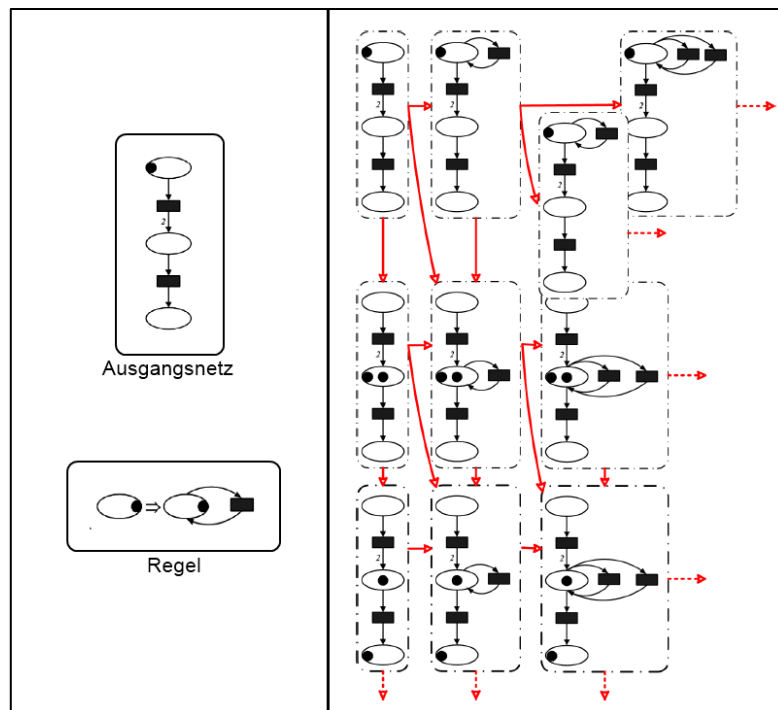


Abbildung 5.6: Abstrakter Erreichbarkeitsgraph aus [Padberg (2012)]

Eine Konstruktion auf Basis der konventionellen Erreichbarkeitsgraphen ist folglich also nicht praktikabel. In [Padberg (2012)] ist ein anderer Ansatz zur Konstruktion eines abstrakten Erreichbarkeitsgraphen vorgestellt worden. Hier wird der Erreichbarkeitsgraph (siehe Abbildung 5.6) auf ähnliche Weise dargestellt, wie der soeben formulierte Ansatz. Der grundlegende Unterschied zeigt sich allerdings in der Repräsentation der Knoten. In der Abbildung sind die Knoten des Graphen nicht mehr durch die Markierung des Netzes dargestellt, sondern das Netz wird zusammen mit der aktuellen Markierung als Zustandsbeschreibung genutzt und im Graphen als Knoten abgebildet. Dadurch können

sowohl die Anwendung von Regeln als auch geschaltete Transitionen als Kanten dargestellt werden. Das Ausgangsnetz ist links oben in der rechten Hälfte der Abbildung zu sehen. Immer wenn eine Regelanwendung erfolgt, wird dies horizontal nach rechts im nächsten Knoten abgebildet. Geschaltete Transitionen hingegen werden vertikal nach unten abgebildet. Durch die gegebene Netzstruktur und der Regel ist dies nur ein Ausschnitt des Erreichbarkeitsgraphen, da dieser unendlich groß ist.

Bezogen auf die modellierten RKN dieser Arbeit, lässt sich eine Parallele erkennen. Bei der Beschreibung der Szenarien (siehe Abschnitt 4.1.2) wurde bereits erläutert, dass die modellierten Szenarien *genau einen* möglichen Ablauf von vielen darstellen. Weiterhin wurden die Szenarien in Abschnitt 4.3 beschrieben und modelliert. Dabei sind alle Veränderungen der Netze, die sich während der Szenarien durch Regelanwendungen oder geschalteten Transitionen ergeben haben, in einem eigenen Netz abgebildet worden. Gewissermaßen sind also alle „Zwischenschritte“ als Netze dargestellt. Aufgrund der Größe und Vielzahl dieser Netze ist in jedem Szenario nur eine Auswahl abgebildet worden. Der komplette Satz aller Netze ist auf der beiliegenden CD zu finden.

Und genau hier ist die oben erwähnte Parallele zu erkennen. Jeder komplette Satz dieser Netze (von einem Szenario) spiegelt einen Pfad des abstrakten Erreichbarkeitsgraphen wider. Anhand der schon aus Abbildung 5.3 bekannten Graphik der globalen Konfluenz, soll der Sachverhalt veranschaulicht werden. Der gelb markierte Pfad (siehe Abbildung 5.7) ist ein möglicher Ablauf der morgendlichen Handlungen des Bewohners innerhalb der intelligenten Wohnung (= Szenario). Würde dieser Pfad inklusive aller Zwischenschritte als Netze und inklusive aller Zustandsübergänge dargestellt werden, wäre es gleichermaßen auch ein Pfad im abstrakten Erreichbarkeitsgraphen. Und für alle drei Szenarien ist ein solcher Pfad als kompletter Satz im Anhang bzw. auf der beiliegenden CD zu finden.

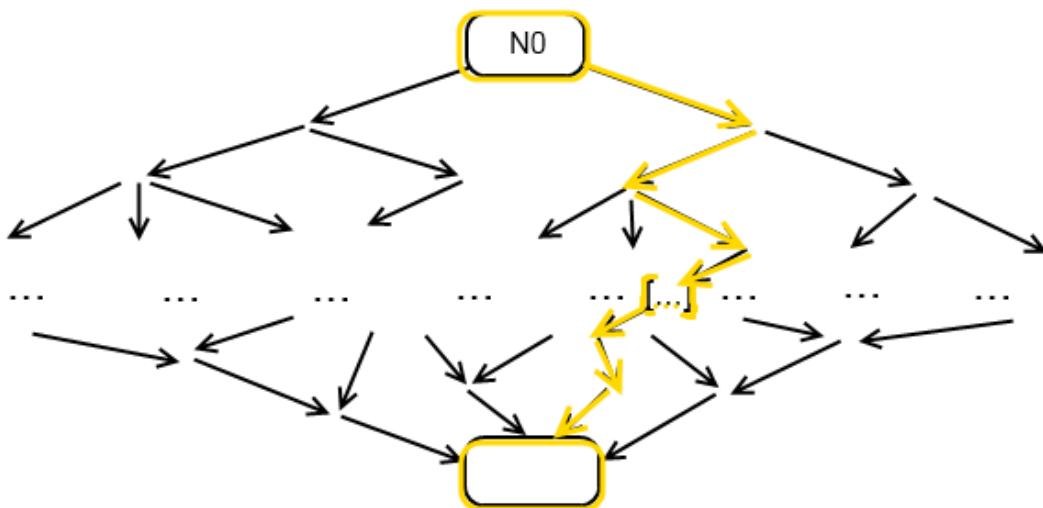


Abbildung 5.7: Ein möglicher Pfad (=Szenario)

Die Konstruktion des kompletten, abstrakten Erreichbarkeitsgraphen wäre bei dieser Modellierung algorithmisch machbar, weil der abstrakte Erreichbarkeitsgraph in den Szenarien dieser Arbeit endlich und somit auch darstellbar ist. Ohne entsprechende Programme wäre es allerdings wenig praktikabel. Das liegt hauptsächlich an den Modellierungsentscheidungen.

Ein Grund ist die Beschränktheit der Netze. Jede Stelle hat eine endliche Kapazität, demnach sind die RKPN also beschränkt. Daraus folgt, dass der (abstrakte) Erreichbarkeitsgraph, bezüglich der Transitionsschaltungen, endlich ist. Ein weiterer Grund ist der Entwurf der Regeln. Zum großen Teil sind die Regeln mit negativen Anwendungsbedingungen (NAC) konstruiert worden. Die Regeln in dieser Arbeit sind so entworfen worden, dass sie ein mehrfaches Anwenden der Regel ausschließen, so dass unendliches Anwenden unterbunden wird. Insofern kann der abstrakte Erreichbarkeitsgraph, auch bezüglich der Regelanwendungen, nicht unendlich groß werden.

Sind diese Eigenschaften im Netz nicht gegeben, kann der Erreichbarkeitsgraph unendlich groß werden. Im Gegensatz zu den klassischen Petrinetzen kann in diesem Fall der Überdeckungsgraph (siehe Abschnitt 2.4) bei einem unendlich großen, abstrakten Erreichbarkeitsgraphen nicht erzeugt werden. Grund dafür ist die oben erwähnte potentielle Veränderung der Stellenanzahl, die eine Überdeckung der Markierungen nicht zulässt.

Ist der abstrakte Erreichbarkeitsgraph unbeschränkt und unendlich groß, könnte der Aufbau nur bis zu einem bestimmten Schritt begrenzt werden. Dazu könnte man die Konstruktion nach x Schritten abbrechen und diesen partiellen Graphen analysieren.

5.4 Statischer Gesamtüberblick

Ein weiterer möglicher Ansatz von Erreichbarkeitsanalyse wäre denkbar. Dieser Ansatz der Analyse wäre allerdings sehr abstrakt, nicht zuverlässig, nur unter gewissen Umständen anwendbar und würde nur einen ersten „groben Eindruck“ der Erreichbarkeit liefern. Die Idee basiert auf der Kombination aller Regeln zu einem Netz.

Abbildung 5.8 zeigt ein Petrinetz, das alle anwendbaren Regeln beinhaltet. Es ist gewissermaßen ein Überblick, der alle Vorkommnisse der Szenarien abbildet und einen statischen Gesamtüberblick widerspiegelt. Jede Regel aus den Szenarien, mit Ausnahme der Löschregeln, ist in diesem Netz kombiniert worden. Diese Darstellung ist nur dann sinnvoll, wenn sich die Teilnetze der Regeln nicht gegenseitig löschen. Alternativ könnte man das Netz nur mit einer Auswahl von Regeln erweitern. Da dieser Ansatz der Analyse ohnehin nur einen abstrakten Überblick verschaffen kann, wäre eine freie Auswahl an Regeln denkbar.

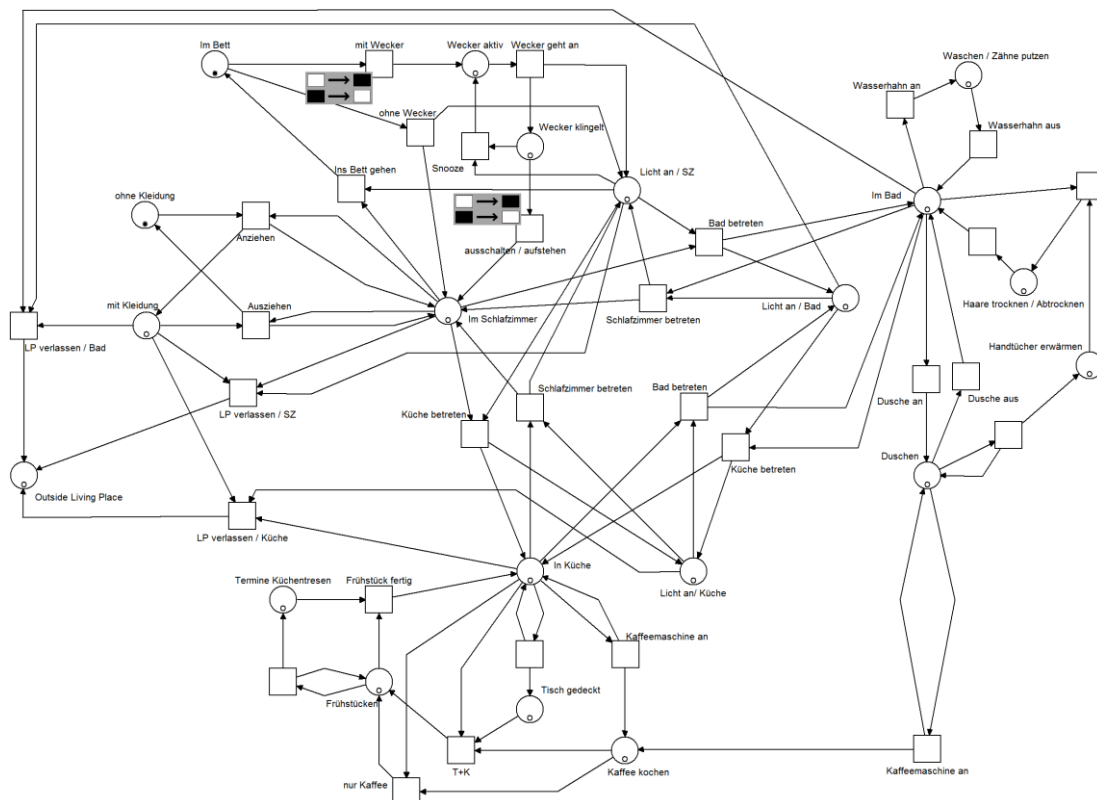


Abbildung 5.8: Statischer Gesamtüberblick aller anwendbaren Regeln

Wie bereits angemerkt wurde, ist dieser Ansatz sehr informell und keinesfalls eine zuverlässige Methode, um eine genaue Analyse anzufertigen. Dennoch könnte sie in bestimmten Situationen angemessen sein, um einen groben Eindruck vom erzeugten System zu bekommen.

6 Schluss

6.1 Zusammenfassung

In dieser Arbeit konnte eine Modellierung realisiert werden, die die Interaktion eines Bewohners mit einer intelligenten Wohnung beschreibt. Dabei wurden rekonfigurierbare Petrinetze zur Umsetzung dieser Modellierung angewendet. Anhand von drei ausgewählten Szenarien des Living Place konnte gezeigt werden, dass sich RKPN zur Beschreibung dieser Szenarien eignen.

Die eingeführten rekonfigurierbaren Petrinetze basieren auf einer Erweiterung der klassischen S/T-Netze, den dekorierten S/T-Netzen. Zusammen mit regelbasierten Netztransformationen können sie während des Tokenspiels ihre Netzstruktur verändern und dienen deshalb der Modellierung von nebenläufigen Systemen, die sich selbst dynamisch verändern.

Zur Modellierung wurden drei verschiedene, kontextabhängige Szenarien entwickelt, die den morgendlichen Ablauf eines fiktiven Bewohners im Living Place beschreiben. Dabei stand besonders die Interaktion des Bewohners mit der intelligenten Wohnung im Vordergrund. Durch die Beschreibung dieser Interaktion konnten nebenläufige, kontextwechselnde Strukturen geschaffen werden, die prädestiniert für die Anwendung von RKPN sind.

Nach der Erläuterung grundlegender Prinzipien und Entscheidungen hinsichtlich der Modellierung, wurden die Szenarien mit RKPN konkret modelliert. Durch das Anwenden von unterschiedlichen Regeln konnten verschiedene Netzsituationen konstruiert und aufgezeigt werden, an denen man die spezifischen Eigenschaften der rekonfigurierbaren Petrinetze identifizieren konnte.

Zum Abschluss wurden nach der Modellierung einige Analysemöglichkeiten vorgestellt. Eine tatsächliche Analyse der entwickelten Netzmodelle war nicht Teil dieser Arbeit. Es wurde vielmehr ein Blick auf interessante Stellen der Modellierung gerichtet, die charakteristisch für RKPN sind und die im Unterschied zu anderen Petrinetzarten andere Eigenheiten und Merkmale besitzen.

6.2 Ausblick

Die Ausarbeitungen dieser Arbeit haben gezeigt, dass eine Modellierung von Szenarien des Living Place Hamburg mit rekonfigurierbaren, dekorierten S/T-Netzen realisierbar ist. Auch darüber hinaus scheint ein Einsatz dieser Petrinetzart in anderen Bereichen nicht nur möglich, sondern teilweise zweckmäßiger zu sein. Entscheidend hierbei sind die Art des zu modellierenden Systems und seine Eigenschaften. Spielen Nebenläufigkeit, dynamische Anpassungen, verteilte Systeme oder viele unterschiedliche Kontexte eine wichtige Rolle, kann der Gebrauch von RKPN eine attraktive Lösung darstellen.

In Bezug auf die weiteren Forschungen und Entwicklungen der rekonfigurierbaren, dekorierten S/T-Netze ist jedoch die Analyse dieser Netze ausschlaggebend. Die Modellierung von Prozessen oder von Systemen bietet zwar einen guten Überblick, doch weitaus wichtiger ist die Verifikation bzgl. der Spezifikation. In [Padberg (2012)] wurde bereits ein abstrakter Erreichbarkeitsgraph vorgestellt. Dieser bietet eine erste Möglichkeit, um eine Erreichbarkeitsanalyse durchzuführen. Da die Erreichbarkeitsgraphen der Szenarien in dieser Arbeit begrenzt sind (siehe Abschnitt 5.3), wäre eine Erreichbarkeitsanalyse möglich.

Wie lässt sich jedoch eine Analyse auf einen unbegrenzten Erreichbarkeitsgraphen anwenden? Dieses Problem ist bei den rekonfigurierbaren, dekorierten S/T-Netzen noch nicht gelöst worden.

Eine erste Implementierung des abstrakten Erreichbarkeitsgraphen soll in dem Werkzeug *ReConNet* [vgl. Ede u.a. (2012)] umgesetzt werden. *ReConNet* ist ein Studentisches Projekt an der HAW-Hamburg. Hier wird ein gleichnamiges Tool zur Modellierung rekonfigurierbarer, dekorierte S/T-Netze entwickelt. Während derzeit die Modellierung von Netzen und die Anwendung von regelbasierten Netztransformationen möglich ist, soll in naher Zukunft auch die Analyse dieser Netze ergänzt werden.

Diese Arbeit hat sich ausschließlich mit Morgen-Szenarien des Living Place beschäftigt. Im Vergleich zu einem dauerhaften Leben in einer intelligenten Wohnung ist dies nur ein kleiner Ausschnitt. Um einen gesamten Tag zu modellieren, sind viele andere kontextabhängige Szenarien denkbar. Die Kombination unterschiedlicher Szenarien, mit unterschiedlichen Kontexten, könnte die Modellierung von einem gesamten Tagesablauf mit RKPN ermöglichen. Diese Arbeit hat einen Ansatz zu dieser Möglichkeit geliefert.

Literaturverzeichnis

- [Allen (1983)] ALLEN, James F.: Maintaining knowledge about temporal intervals. In: *Commun. ACM* 26 (1983), November, S. 832–843. – URL <http://doi.acm.org/10.1145/182.358434>. – ISSN 0001-0782
- [Baumgarten (1990)] BAUMGARTEN, Bernd: *Petri-Netze - Grundlagen und Anwendungen*. Mannheim : BI Wissenschaftsverlag, 1990. – ISBN 978-3-411-14291-0
- [Biermann u.a. (2008)] BIERMANN, Enrico ; ERMEL, Claudia ; MODICA, Tony ; SYLOPP, Peggy: Implementing Petri Net Transformations using Graph Transformation Tools. In: *ECEASST* (2008). – URL <http://journal.ub.tu-berlin.de/eceasst/article/download/189/180>. – ISSN 1863-2122
- [Ede u.a. (2012)] EDE, Marvin ; HOFFMANN, Kathrin ; OELKER, Gerhard ; PADBERG, Julia: ReConNet: A Tool for Modeling and Simulating with Reconfigurable Place/Transition Nets. In: *ECEASST* (2012). – URL <http://www.easst.org/eceasst/>. – ISSN 1863-2122
- [Ehrig u.a. (2010)] EHRIG, Hartmut ; EHRIG, Karsten ; PRANGE, Ulrike ; TAENTZER, Gabriele: *Fundamentals of Algebraic Graph Transformation*. 1st. Springer Publishing Company, Incorporated, 2010. – ISBN 3642068316, 9783642068317
- [Ehrig u.a. (2007)] EHRIG, Hartmut ; HOFFMANN, Kathrin ; PADBERG, Julia ; PRANGE, Ulrike ; ERMEL, Claudia: Independence of net transformations and token firing in reconfigurable place/transition systems. In: *Proceedings of the 28th international conference on Applications and theory of Petri nets and other models of concurrency*. Berlin, Heidelberg : Springer-Verlag, 2007 (ICATPN'07), S. 104–123. – URL <http://dl.acm.org/citation.cfm?id=1769053.1769064>. – ISBN 978-3-540-73093-4

- [Ellenberg (2010)] ELLENBERG, Jens: Ein Wecker in einem ubicom Haus / HAW Hamburg. HAW Hamburg, 2010. – Forschungsbericht. – URL <http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master09-10-aw1/Ellenberg/bericht.pdf>
- [Ellenberg (2011)] ELLENBERG, Jens: *Ontologiebasierte Aktivitätserkennung im Smart Home Kontext*, HAW Hamburg, Diplomarbeit, 2011. – URL <http://users.informatik.haw-hamburg.de/~ubicomp/arbeiten/master/ellenberg.pdf>
- [Ellenberg u.a. (2011)] ELLENBERG, Jens ; KARSTAEDT, Bastian ; VOSKUHL, Sören ; LUCK, Kai von ; WENDHOLT, Birgit: An Environment for Context-Aware Applications in Smart Homes / Hamburg University of Applied Sciences. URL <http://users.informatik.haw-hamburg.de/~ubicomp/arbeiten/papers/IPIN2011.pdf>, 2011. – Forschungsbericht
- [Girault und Valk (2003)] GIRAULT, Claude ; VALK, Rüdiger: *Petri Nets for Systems Engineering - A Guide to Modeling, Verification, and Applications*. 1. Aufl. Berlin, Heidelberg : Springer, 2003. – ISBN 978-3-540-41217-5
- [Hardenack (2011)] HARDENACK, Frank: - *Das intelligente Bett - Sensorbasierte Detektion von Schlafphasen*, HAW Hamburg, Diplomarbeit, 2011. – URL <http://users.informatik.haw-hamburg.de/~ubicomp/arbeiten/master/hardenack.pdf>
- [Hoffmann u.a. (2010)] HOFFMANN, K. ; EHRIG, H. ; HERMANN, F.: Flexible Independence of Net Transformations and Token Firing in the Cospan DPO Approach. In: *Proc. 3rd International Symposium of Multiagent Systems (MAS), Robotics and Cybernetics: Theory and Practice, 2009, 2010*
- [Karstaedt (2012)] KARSTAEDT, Bastian: *Kontextinterpretation in Smart Home auf Basis 3D semantischer Gebäudemodelle*, HAW Hamburg, Diplomarbeit, 2012. – URL <http://users.informatik.haw-hamburg.de/~ubicomp/arbeiten/master/karstaedt.pdf>
- [Kleuker (2009)] KLEUKER, Stephan: *Formale Modelle Der Softwareentwicklung - Model-Checking, Verifikation, Analyse Und Simulation*. 2009. Auflage 1. Wiesbaden : Vieweg Teubner Verlag, 2009. – ISBN 978—3-8-34-80-6
- [von Luck u.a. (2010)] LUCK, Prof. Dr. K. von ; KLEMKE, Prof. Dr. G. ; GREGOR, Sebastian ; RAHIMI, Mohammad A. ; VOGT, Matthias: *Living Place Hamburg - A place for concepts of IT based modern living* / Hamburg University of Applied Sciences. URL http://livingplace.informatik.haw-hamburg.de/content/-LivingPlaceHamburg_en.pdf, Mai 2010. – Forschungsbericht
- [Otto und Voskuhl (2010)] OTTO, Kjell ; VOSKUHL, Sören: *Entwicklung einer Architektur für das Living Place Hamburg* / HAW Hamburg. HAW Hamburg, 2010. –

- Forschungsbericht. – URL <http://users.informatik.haw-hamburg.de/~ubicomp/-projekte/master2010-proj/otto-voskuhl.pdf>
- [Padberg (2011)] PADBERG, Julia: *Rekonfigurierbare Petrinetze - Skript*. 2011
- [Padberg (2012)] PADBERG, Julia: Abstract Interleaving Semantics for Reconfigurable Petri Nets. In: *ECEASST (2012)*. – URL <http://www.easst.org/eceasst/>. – ISSN 1863-2122
- [Petri (1962)] PETRI, Carl A.: *Kommunikation mit Automaten*, Dissertation, 1962
- [Priese und Wimmel (2008)] PRIESE, Lutz ; WIMMEL, Harro: *Petri-Netze*. 2. Aufl. Berlin : Springer DE, 2008. – ISBN 978-3-540-76970-5
- [Rahimi und Vogt (2011)] RAHIMI, Mohammad A. ; VOGT, Matthias: *Seamless Interaction - Natürliche Interaktionen in Smart Living Umgebungen*, HAW Hamburg, Diplomarbeit, 02 2011. – URL <http://users.informatik.haw-hamburg.de/~ubicomp/-arbeiten/master/rahimi-vogt.pdf>
- [Rein u.a. (2008)] REIN, Alexander ; PRANGE, Ulrike ; LAMBERS, Leen ; HOFFMANN, Kathrin ; PADBERG, Julia: Negative Application Conditions for Reconfigurable Place/Transition Systems. In: *ECEASST 10 (2008)*. – URL <http://eceasst.cs.tu-berlin.de/index.php/eceasst/article/view/140>. – ISSN 1863-2122
- [Reisig (2010)] REISIG, Wolfgang: *Petrinetze: Modellierungstechnik, Analysemethoden, Fallstudien*. 2010. Aufl. Vieweg+Teubner Verlag, 15 July 2010 (Leitfäden der Informatik). – ISBN 978-3-8348-1290-2
- [Schneider (2010)] SCHNEIDER, Marco: *Entwicklung und Realisierung eines Sensornetzwerkes für das Living Place Hamburg*, HAW Hamburg, Diplomarbeit, 2010. – URL http://users.informatik.haw-hamburg.de/~ubicomp/arbeiten/-bachelor/schneider_marco.pdf
- [Tennstedt (2011)] TENNSTEDT, Sven: *Szenarien basierte Interpretationen eines Smart-Homes*, HAW Hamburg, Diplomarbeit, 2011. – URL <http://users.informatik.haw-hamburg.de/~ubicomp/arbeiten/master/tennstedt.pdf>
- [Valk und Moldt (2006)] VALK, Rüdiger ; MOLDT, Daniel: *Modellierung und Analyse paralleler und verteilter Systeme - Skript*. 2006

- [Voskuhl (2012)] VOSKUHL, Sören: *Modellunabhängige Kontextinterpretation in einer Smart Home Umgebung*, HAW Hamburg, Diplomarbeit, 2012. – URL <http://users.informatik.haw-hamburg.de/~ubicomp/arbeiten/master/voskuhl.pdf>
- [Wang (1998)] WANG, Jiacun: *Timed Petri Nets - Theory and Application*. 1. Aufl. Berlin, Heidelberg : Springer, 1998. – ISBN 978-0-792-38270-6

Anhang

A Regeln

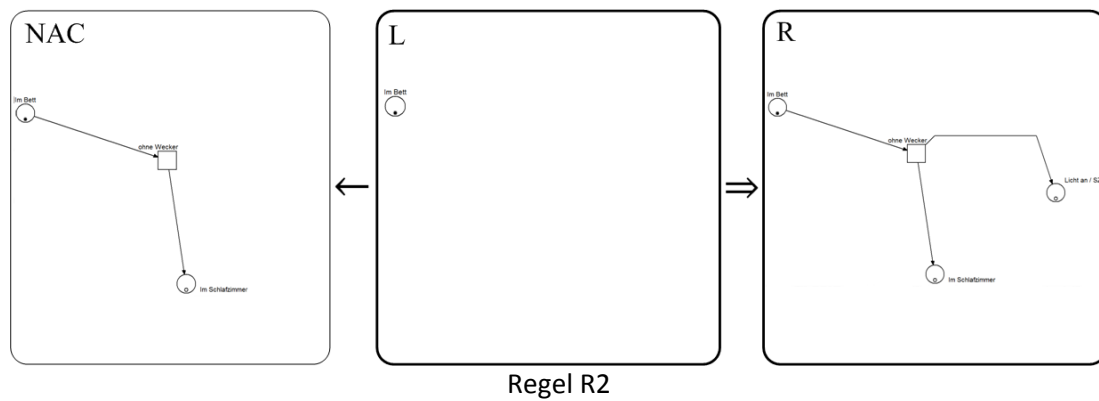
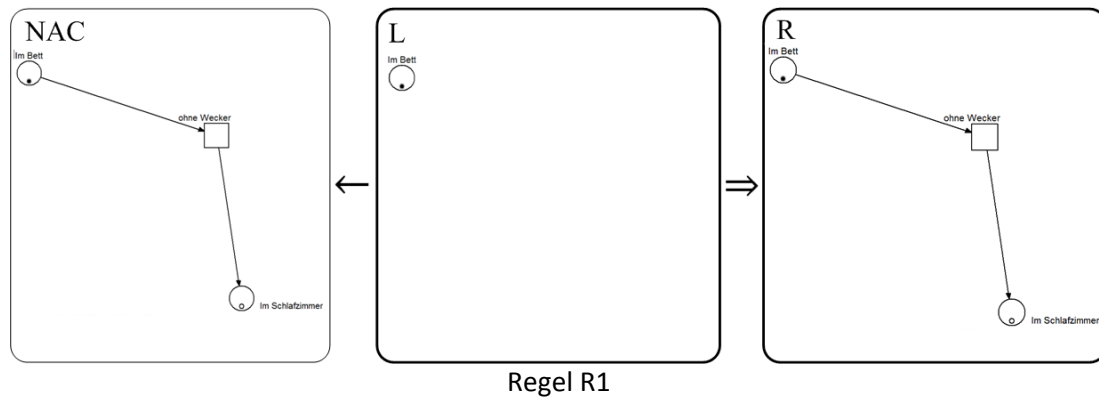
A.1 Übersicht aller Regeln

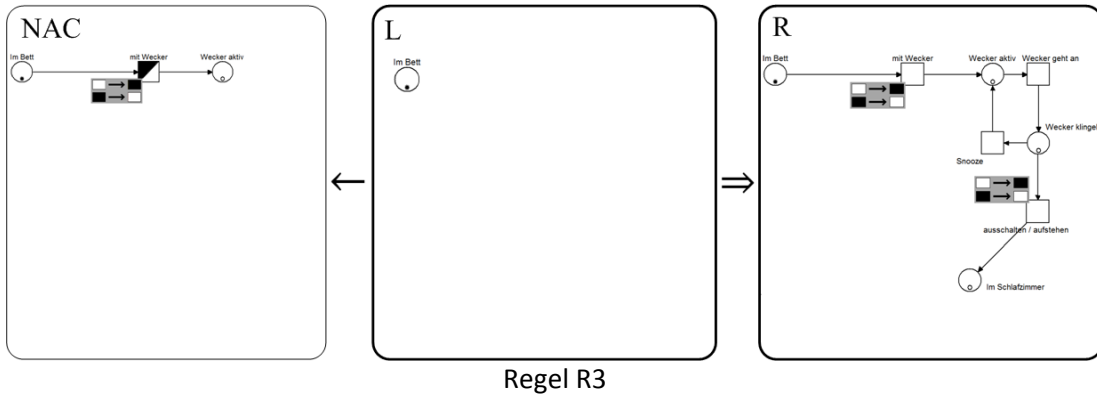
Regeln	Erläuterung	Wo? Stelle?	Löschregeln
R1	Aufstehen, ohne Wecker ohne Licht	Im Bett	LöRe36
R2	Aufstehen, ohne Wecker mit Licht	Im Bett	LöRe37
R3	Aufstehen, mit Wecker ohne Licht	Im Bett	LöRe38
R4	Aufstehen, mit Wecker mit Licht	Im Bett	LöRe39
R5	ins Bett gehen ohne Licht	SZ	LöRe40
R6	ins Bett gehen mit Licht	SZ	LöRe41
R7	Kleidung anziehen/ausziehen	SZ	LöRe42
R8	in Küche gehen ohne Licht	SZ	LöRe43
R9	in Küche gehen mit Licht	SZ	LöRe44
R10	ins Bad gehen ohne Licht	SZ	LöRe45
R11	ins Bad gehen mit Licht	SZ	LöRe46
R12	LP verlassen ohne Licht	SZ	LöRe47
R13	LP verlassen mit Licht	SZ	LöRe48
R14	ins SZ gehen ohne Licht	Küche	LöRe49
R15	ins SZ gehen mit Licht	Küche	LöRe50
R16	ins Bad gehen ohne Licht	Küche	LöRe51
R17	ins Bad gehen mit Licht	Küche	LöRe52
R18	Tisch decken	Küche	LöRe53
R19	Kaffeemaschine an (Küche)	Küche	LöRe54
R20	Frühstücken, nur Kaffee trinken	Küche	LöRe55
R21	Frühstücken, T+K	Küche	LöRe56
R22	LP verlassen ohne Licht	Küche	LöRe57
R23	LP verlassen mit Licht	Küche	LöRe58
R24	Termine Küchentresen	Küche, Frühstück	LöRe59
R25	ins SZ gehen ohne Licht	Bad	LöRe60
R26	ins SZ gehen mit Licht	Bad	LöRe61
R27	in Küche gehen ohne Licht	Bad	LöRe62
R28	in Küche gehen mit Licht	Bad	LöRe63
R29	Duschen	Bad	LöRe64
R30	Waschen / Zähne putzen	Bad	LöRe65
R31	Haare trocknen / Abtrocknen	Bad	LöRe66
R32	LP verlassen ohne Licht	Bad	LöRe67
R33	LP verlassen mit Licht	Bad	LöRe68
R34	Kaffeemaschine an (Bad)	Bad, Dusche	LöRe69
R35	Handtücher erwärmen	Bad, Dusche	LöRe70
	Alle "Räume" löschen	Outside LP	LöRe71

Übersicht aller Regeln. Grau markierte Regeln sind nicht in Szenario A enthalten.

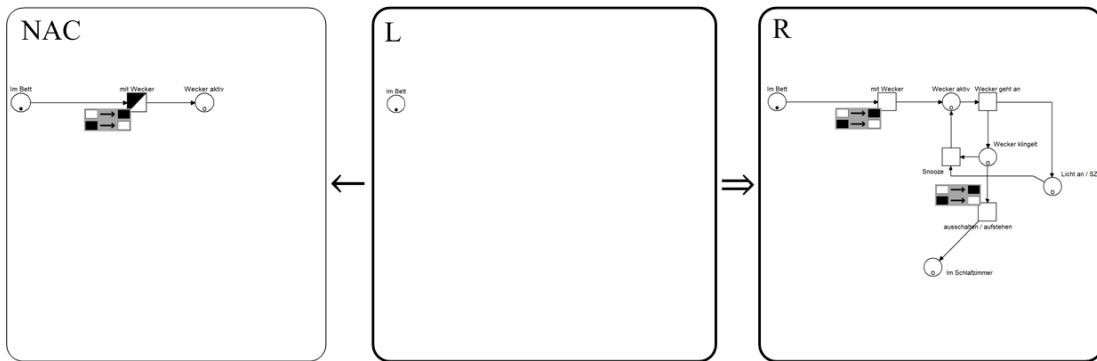
A.2 Satz aller Regeln (exkl. Löschrregeln) aus Szenario A

Im Folgenden soll der Satz aller Regeln aus Szenario A abgebildet werden. Die entsprechenden Löschrregeln sind dabei nicht einbezogen.

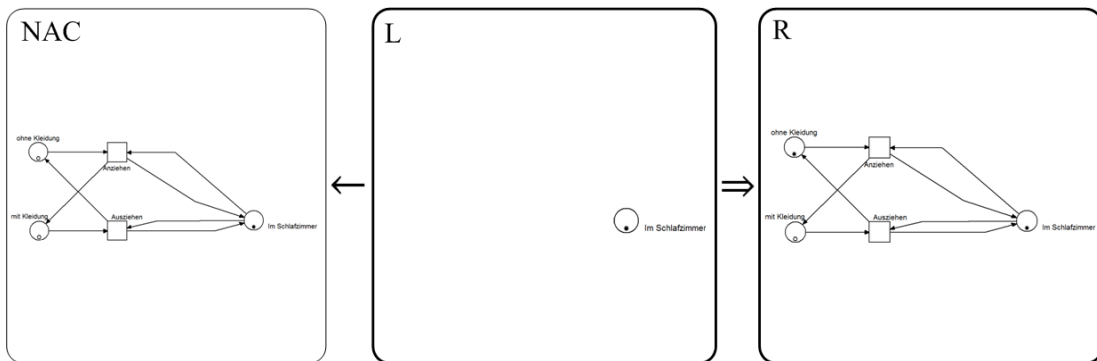




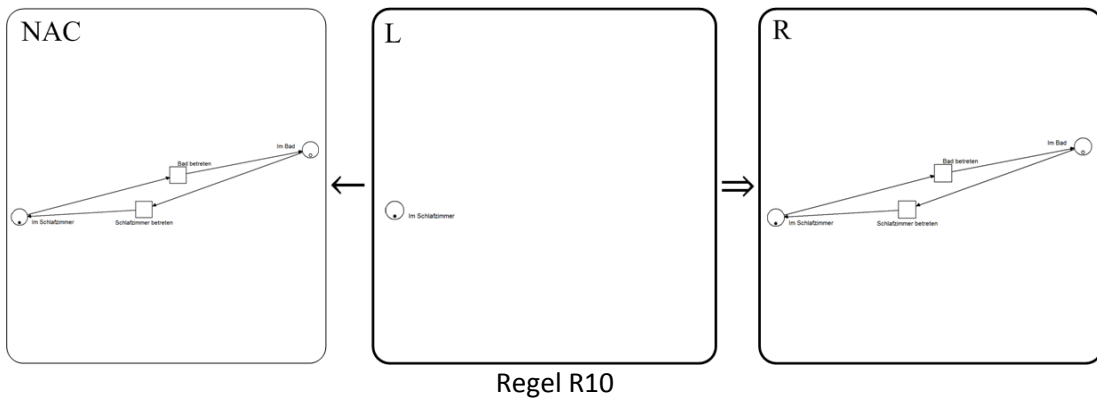
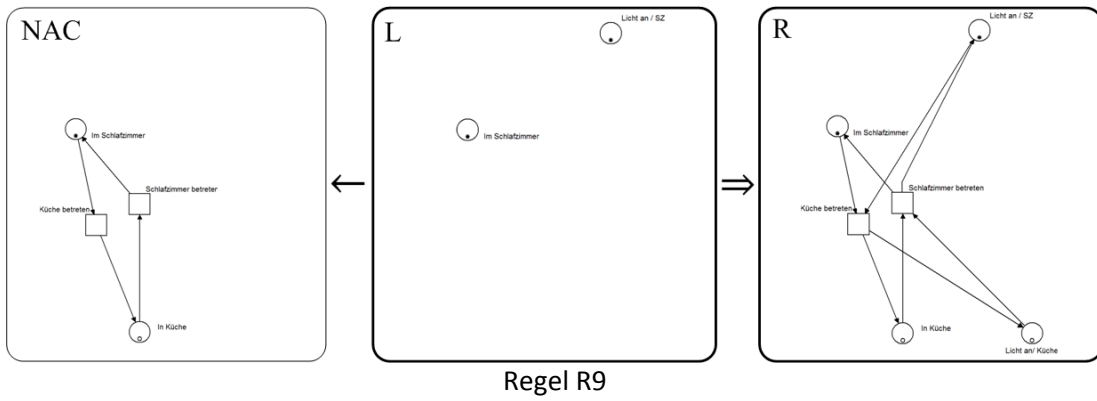
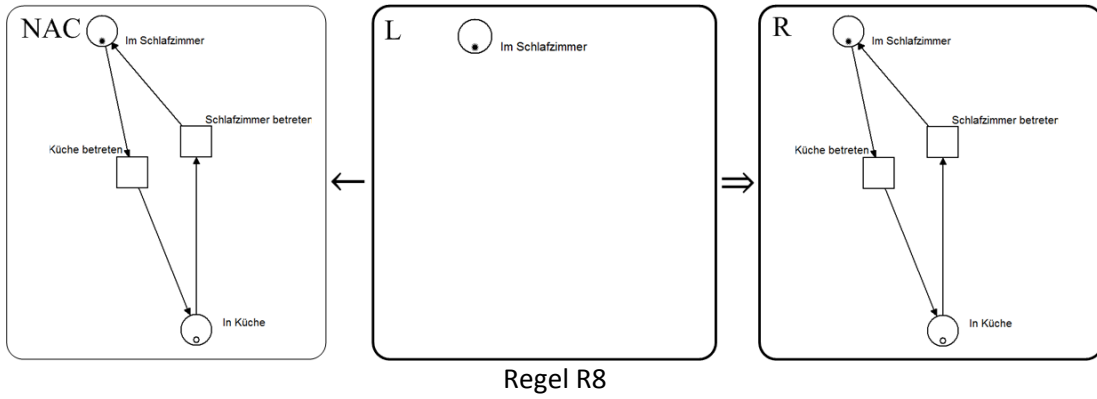
Regel R3

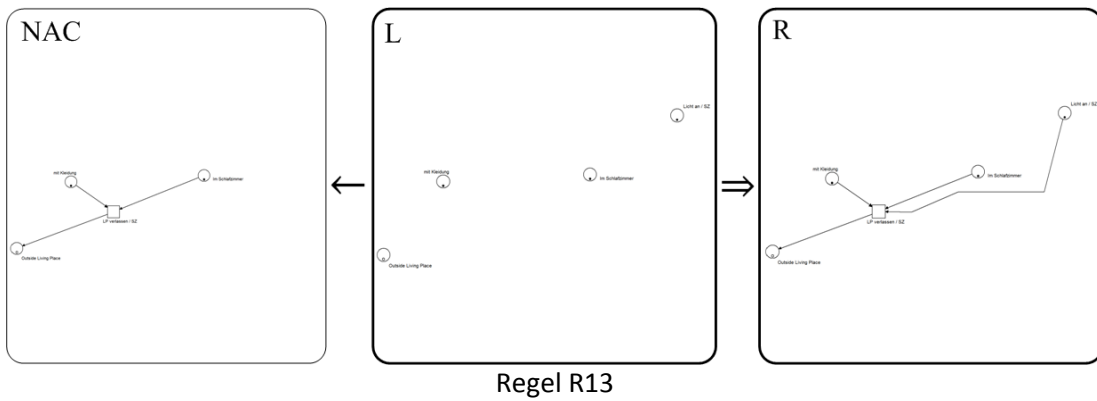
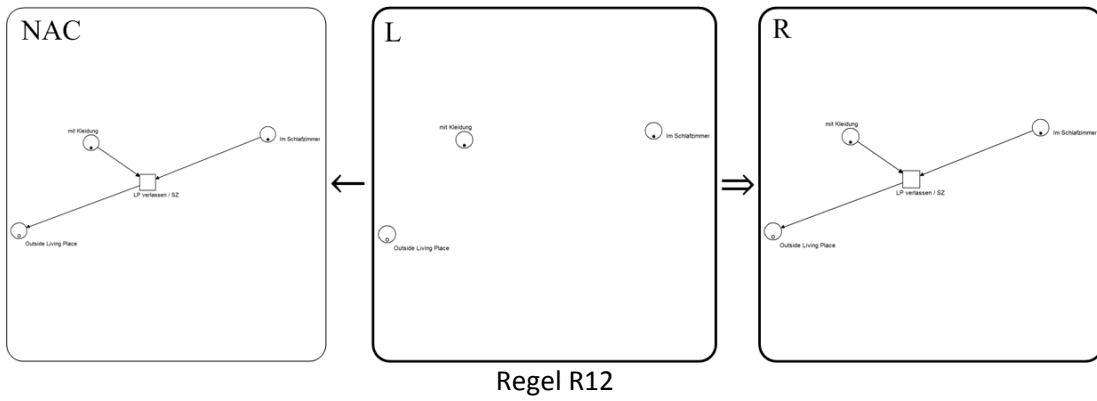
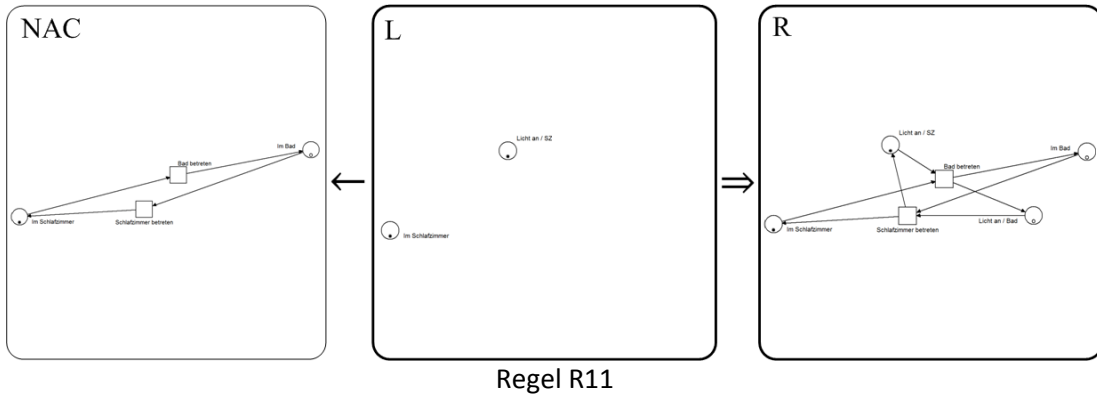


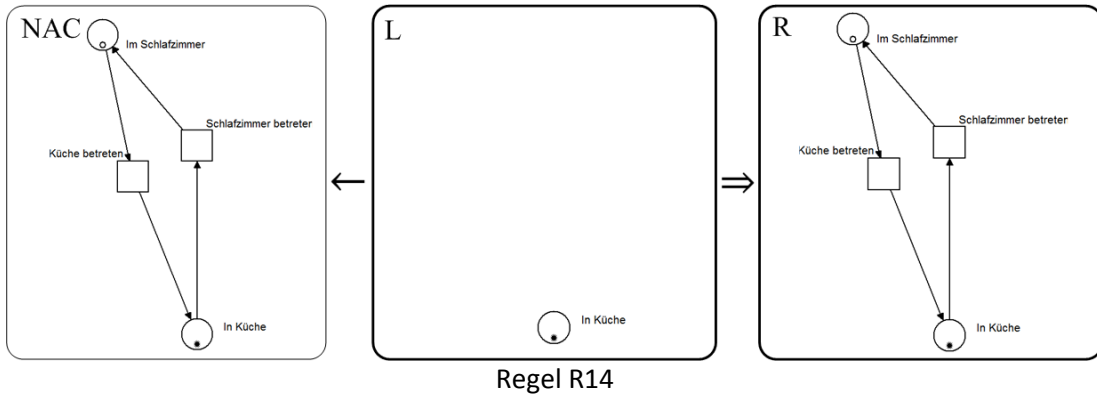
Regel R4



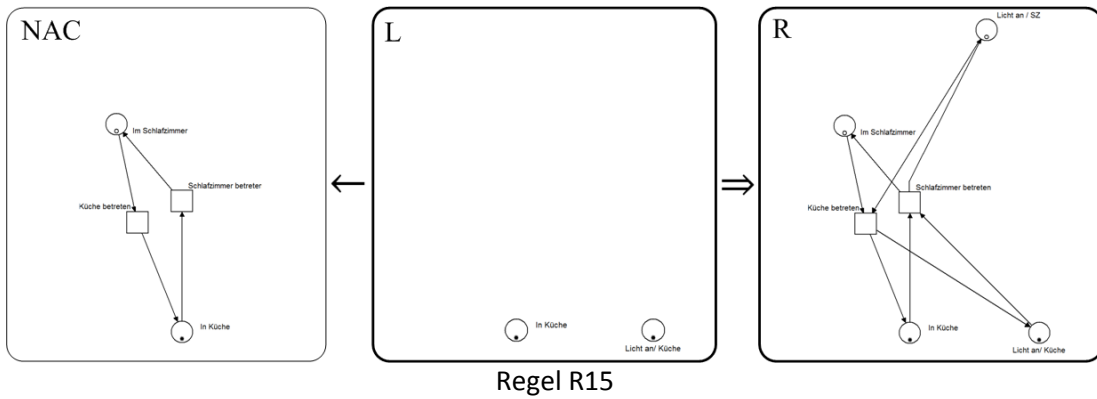
Regel R7



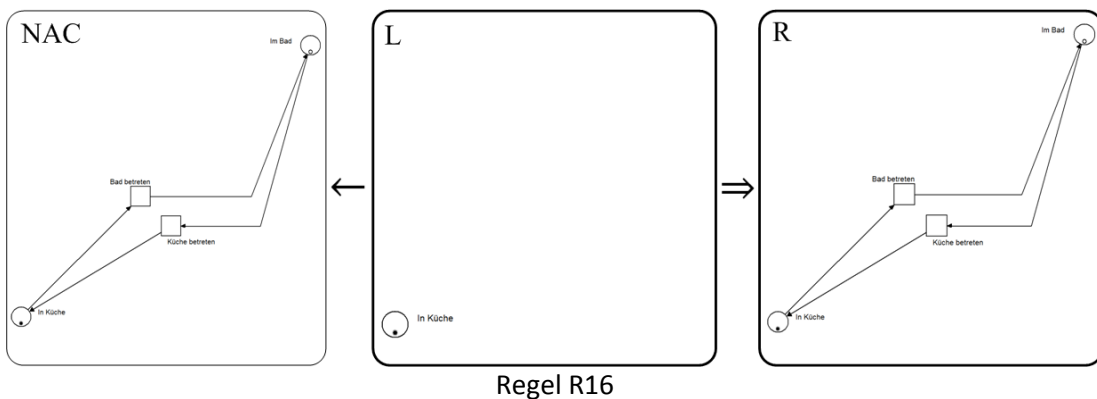




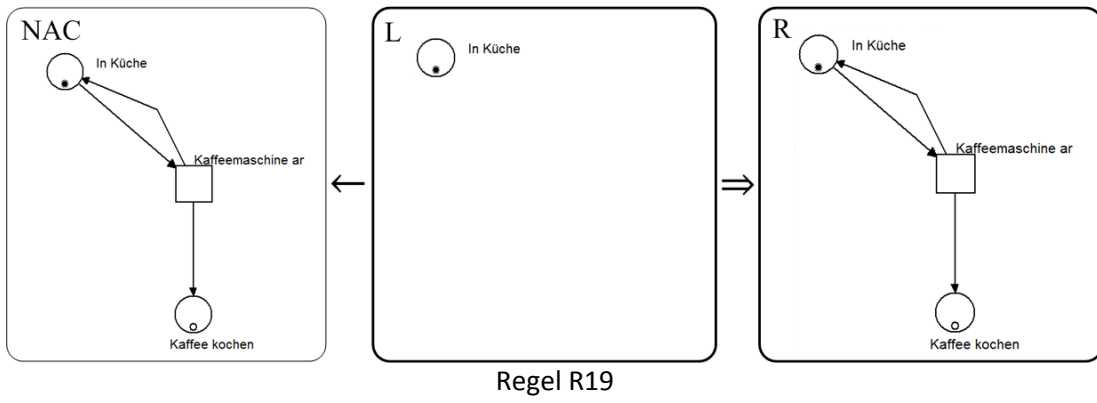
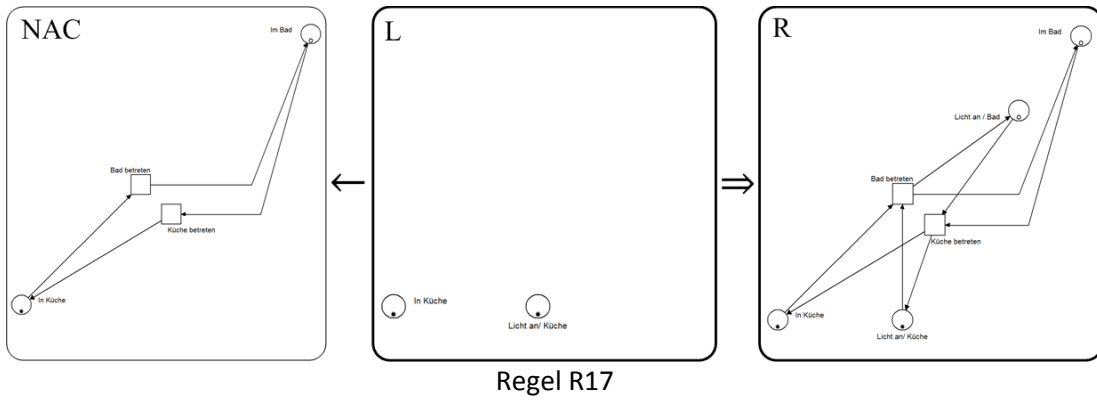
Regel R14

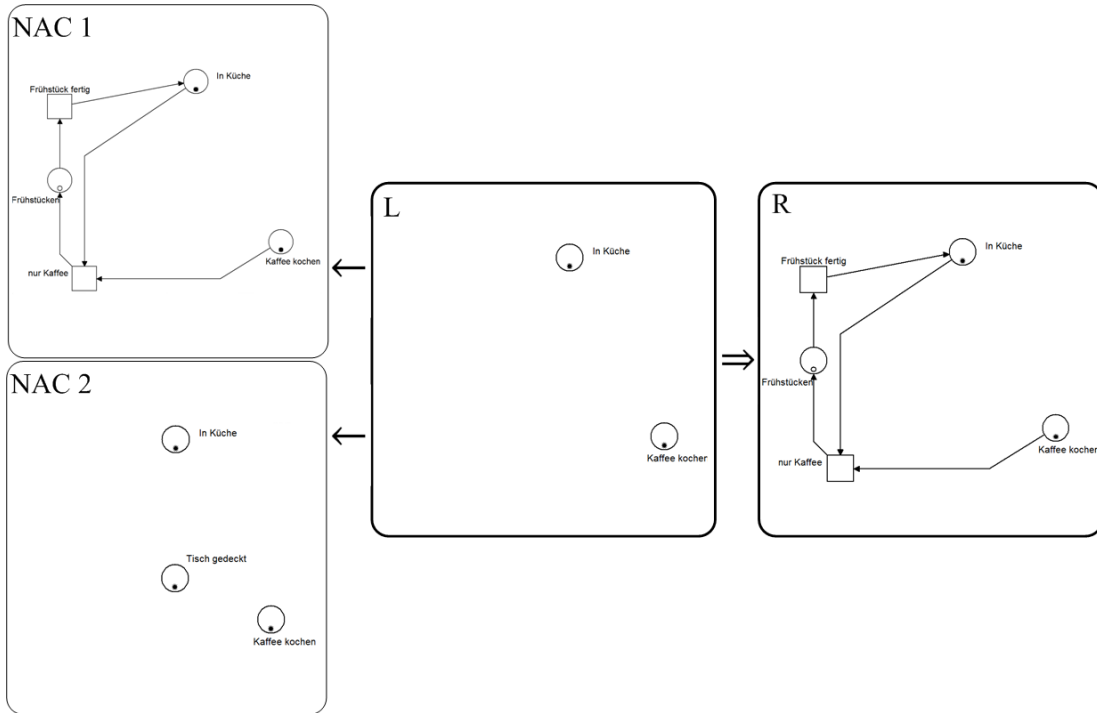


Regel R15

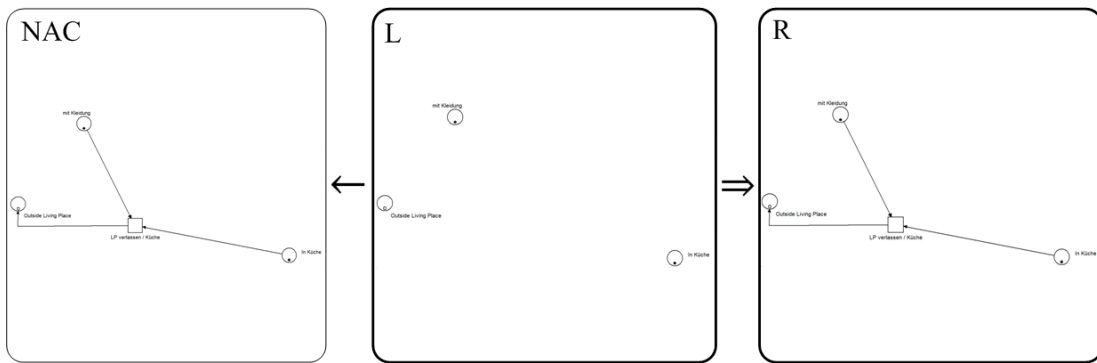


Regel R16

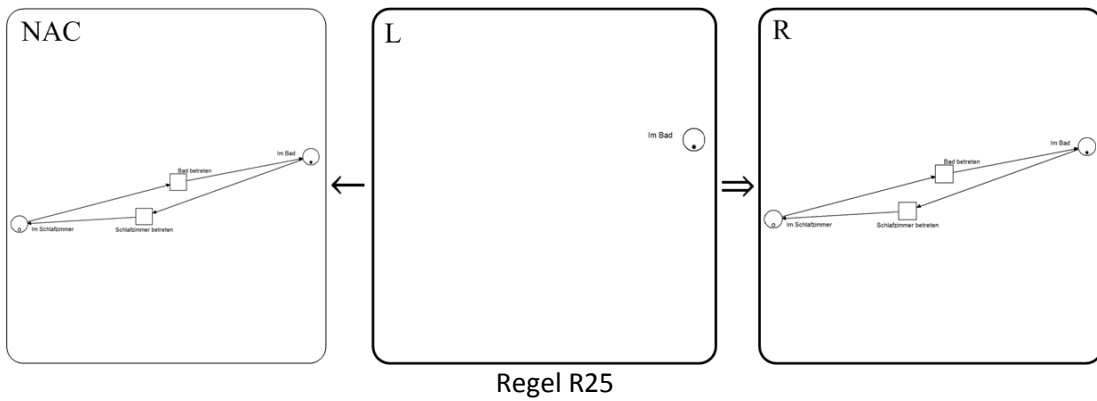
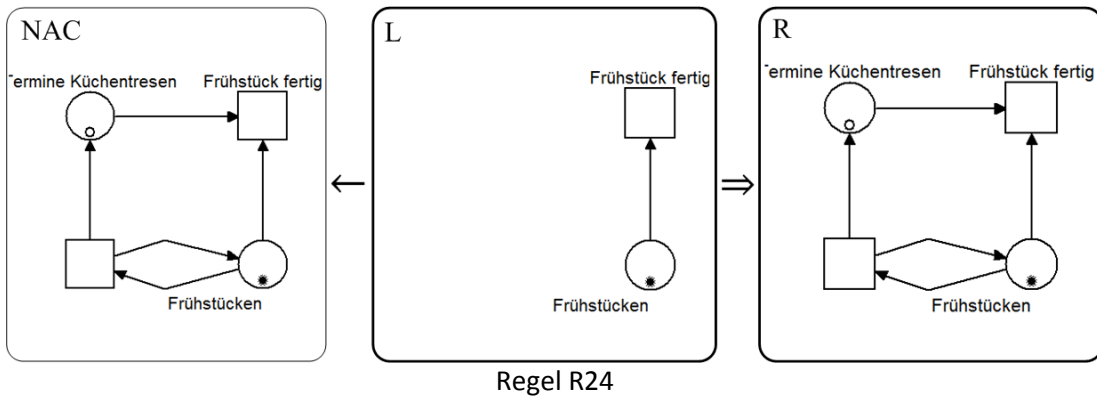
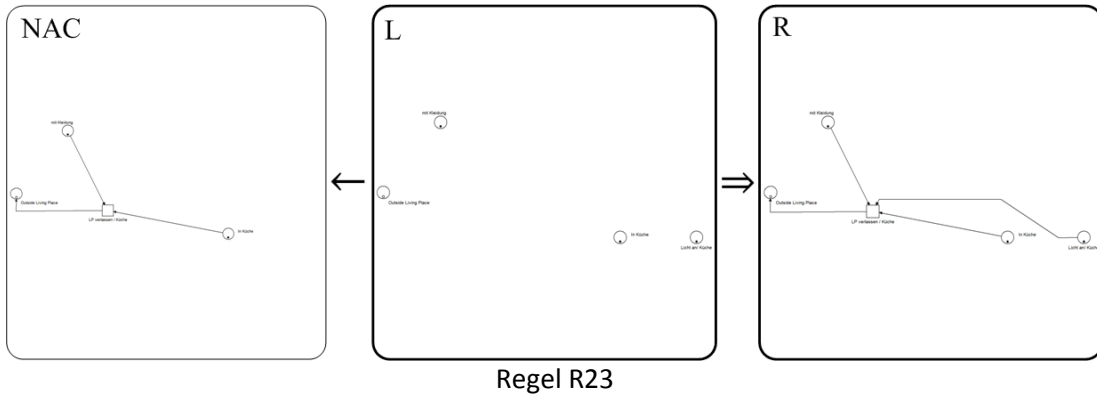


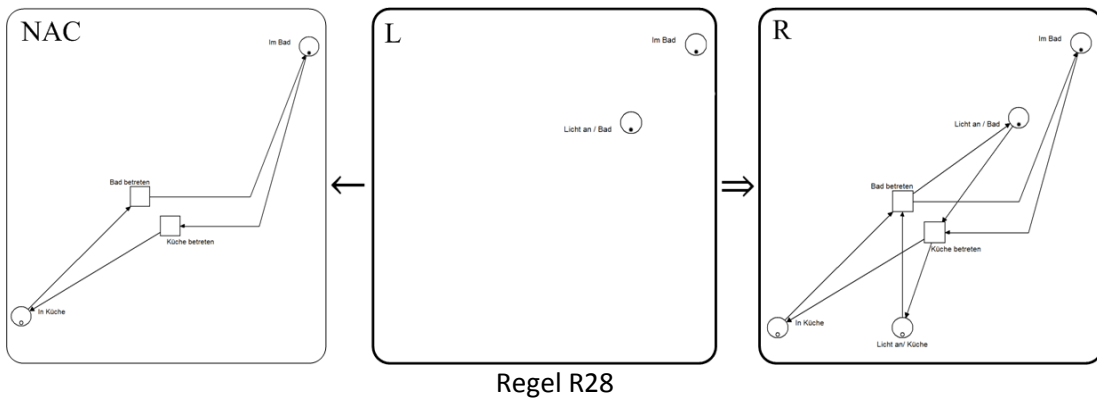
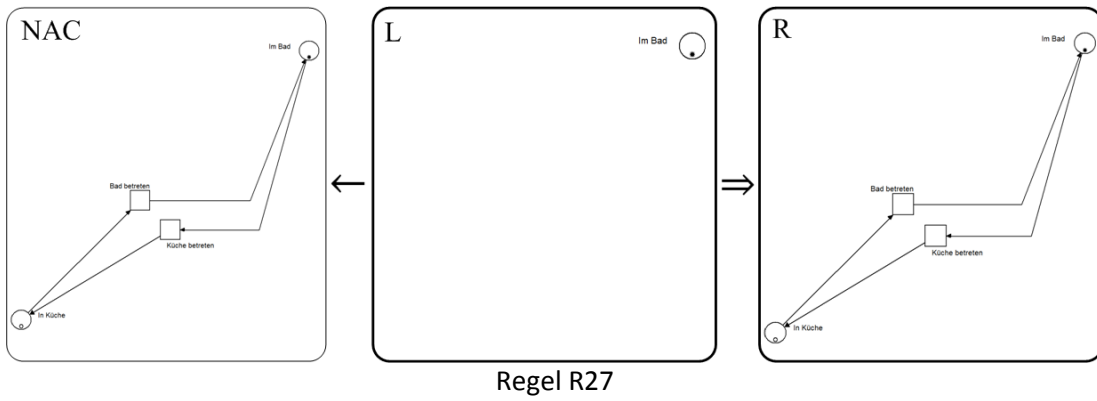
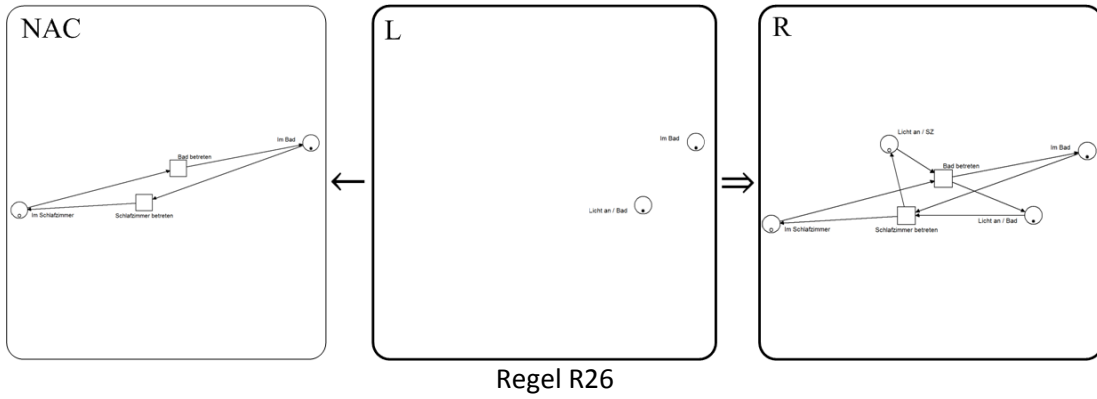


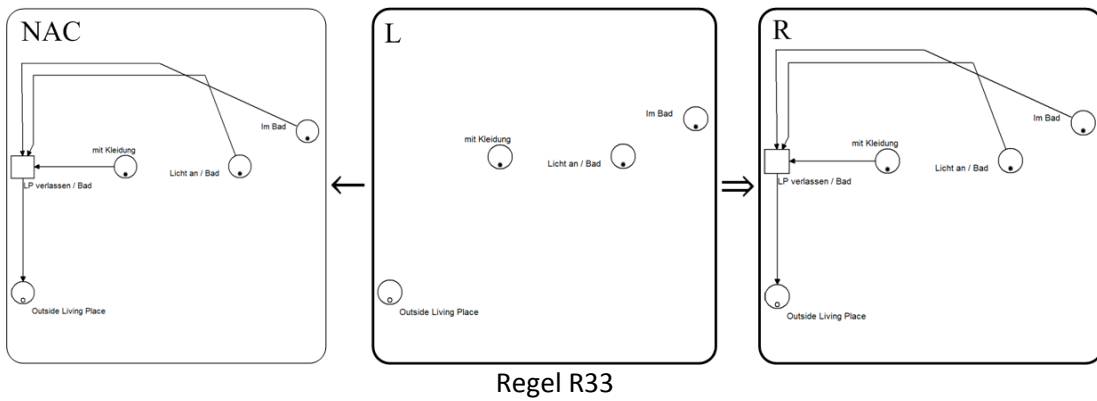
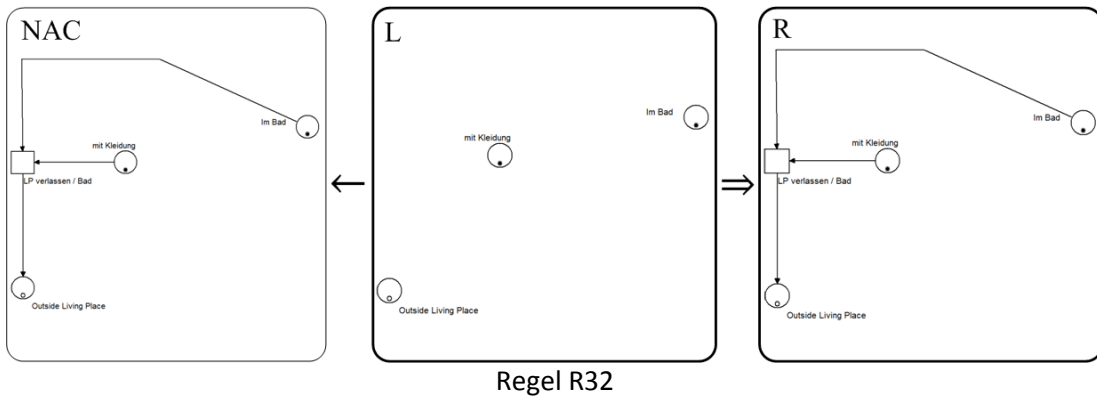
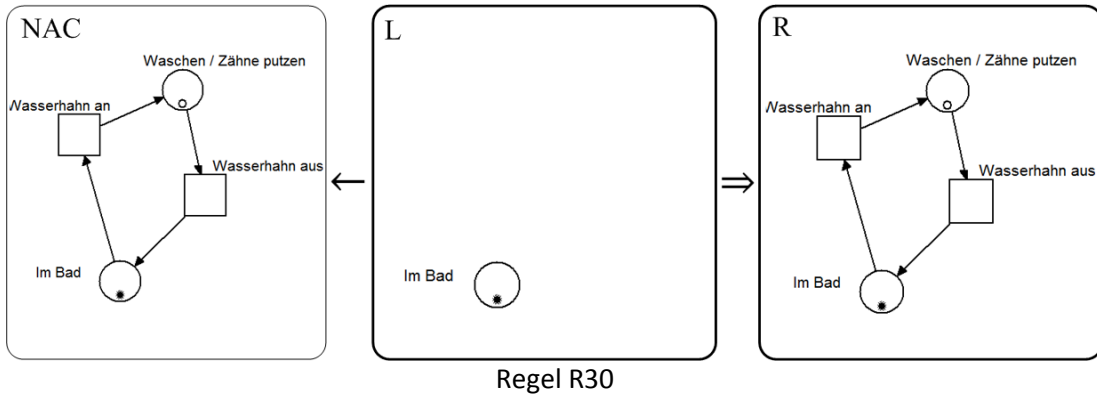
Regel R20



Regel R22





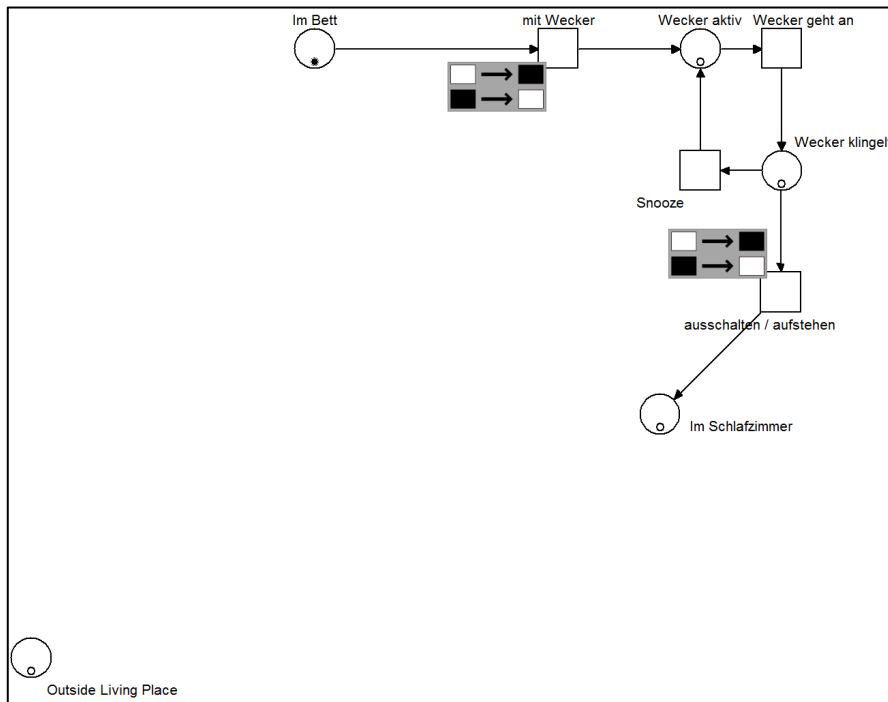


B Netze

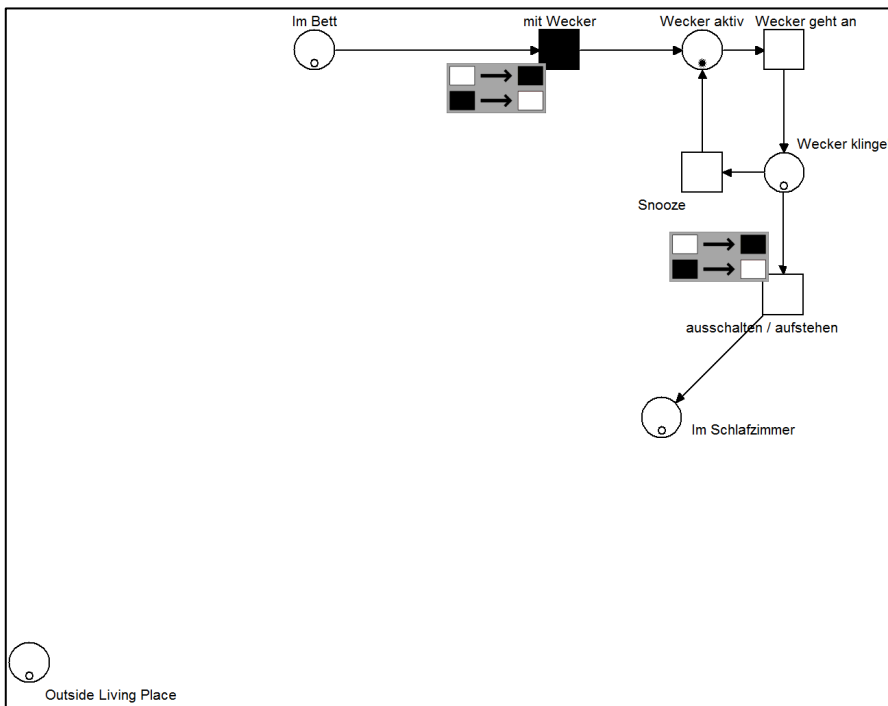
B.1 Alle Netze aus Szenario A



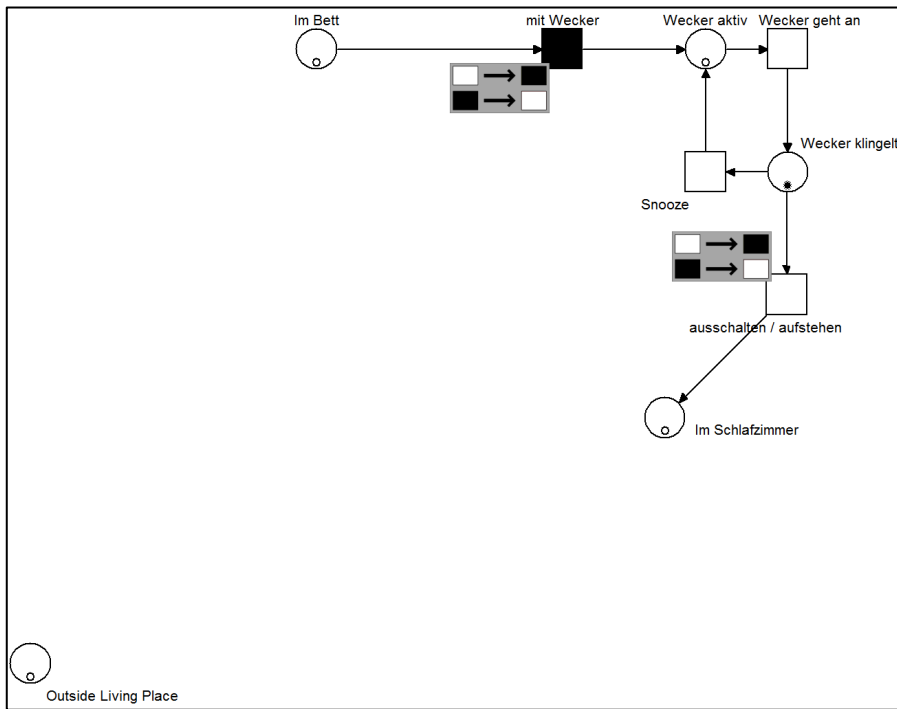
Netz NA0



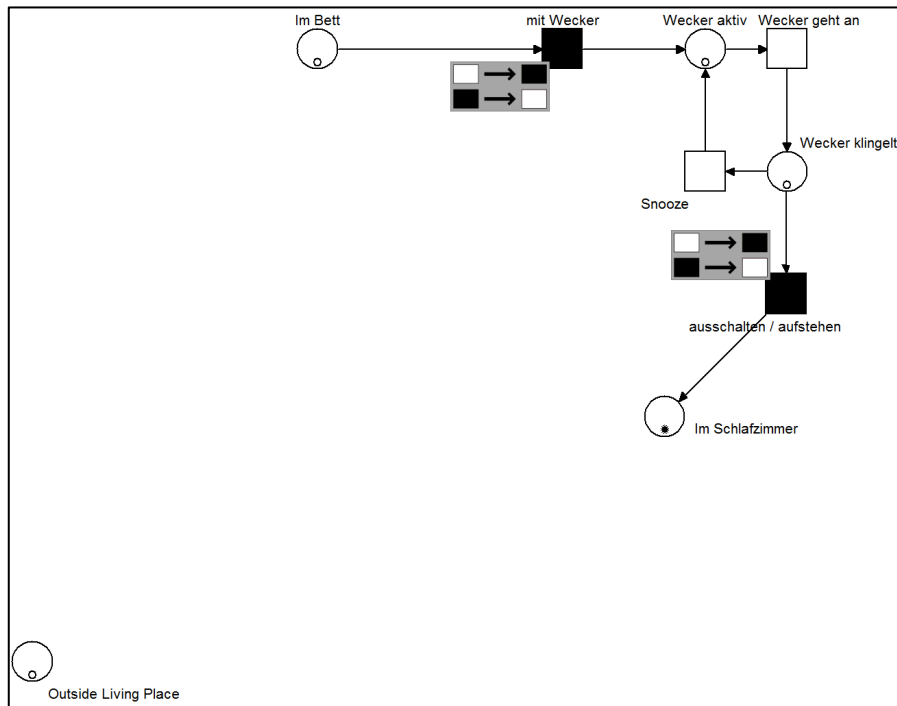
Netz NA1



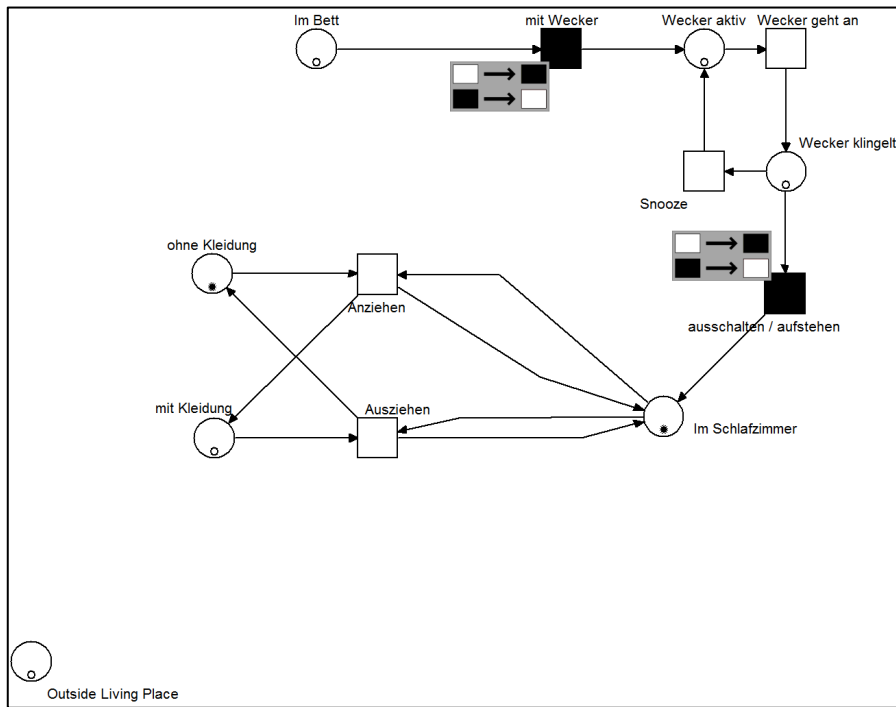
Netz NA2



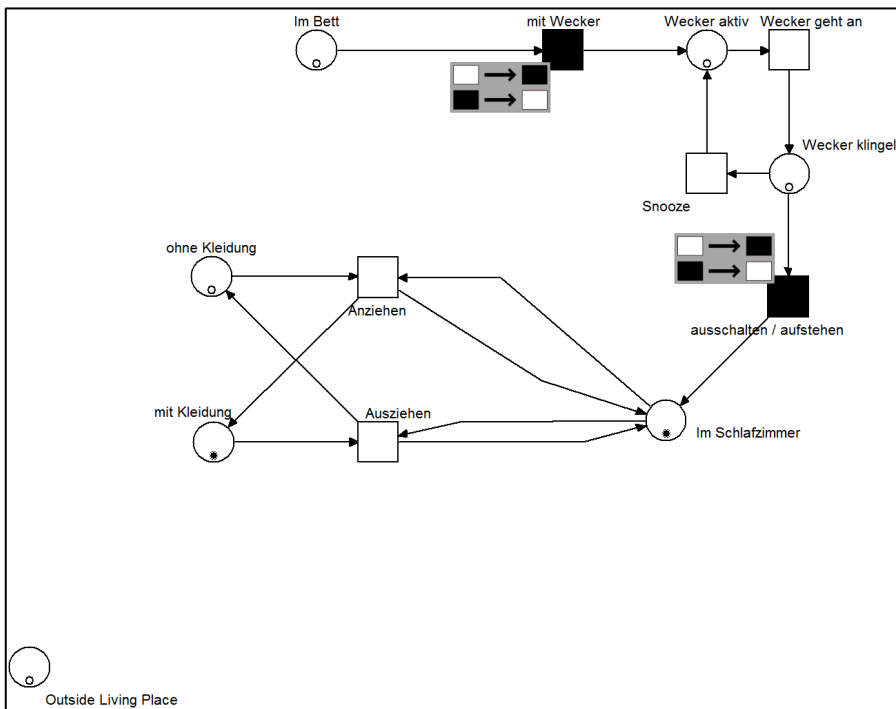
Netz NA3



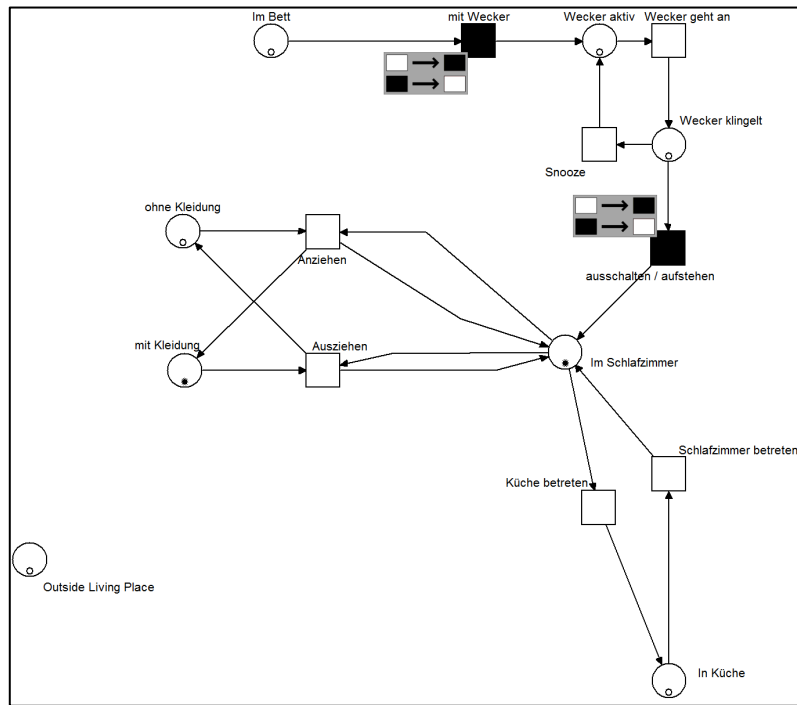
Netz NA4



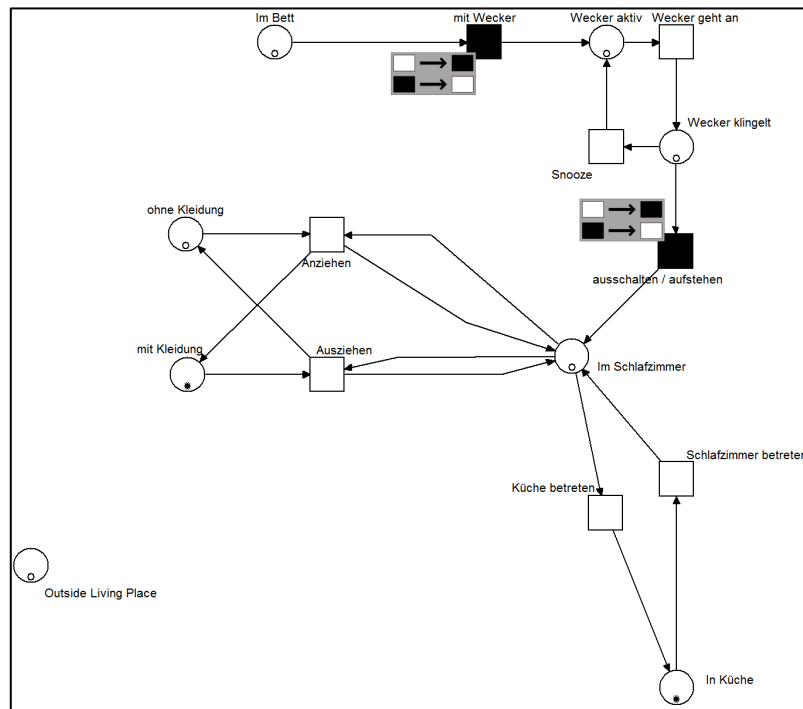
Netz NA5



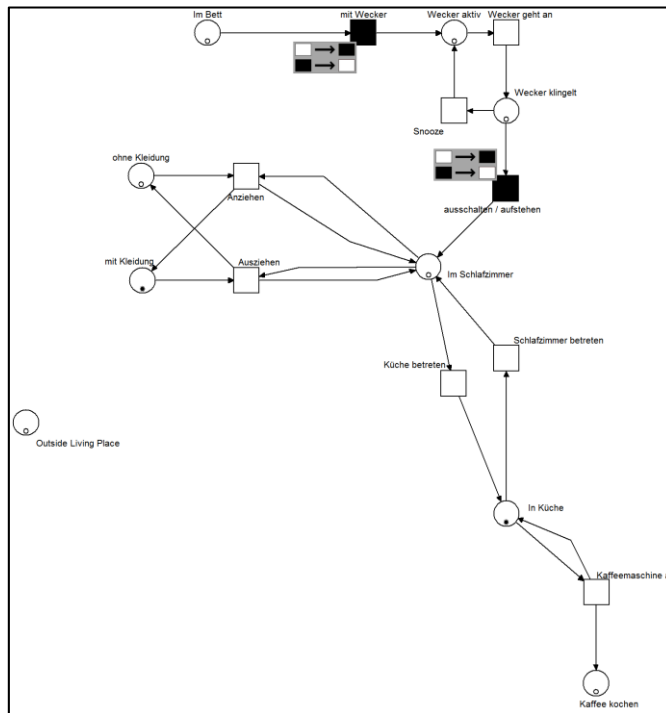
Netz NA6



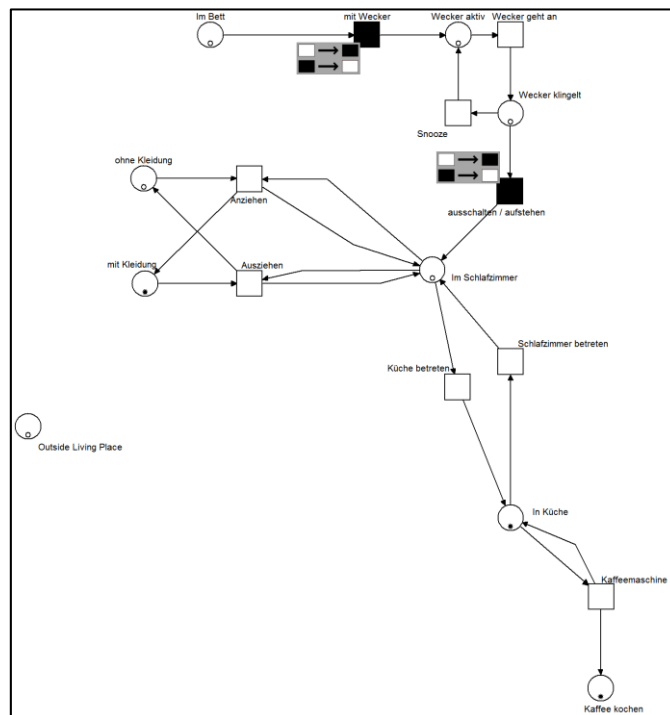
Netz NA7



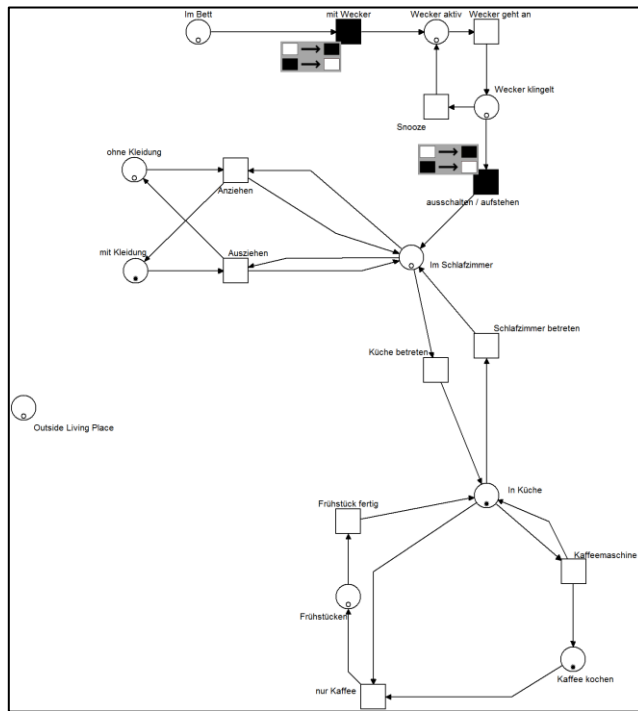
Netz NA8



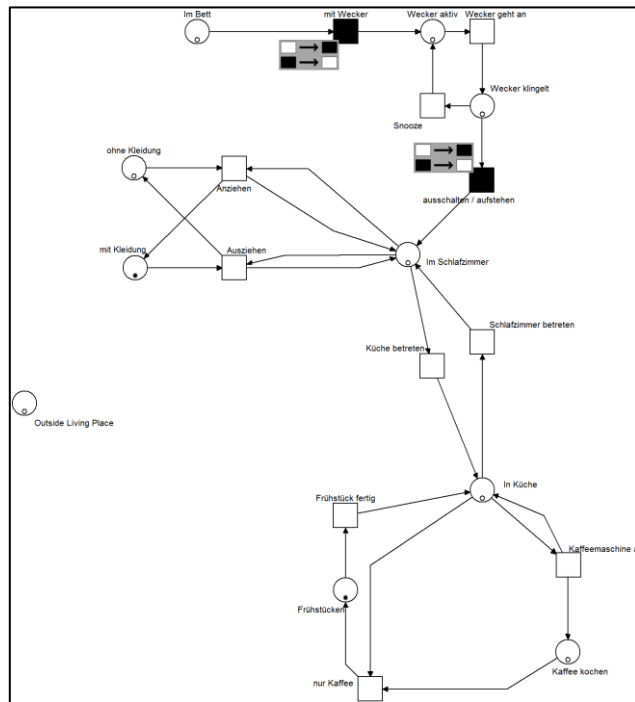
Netz NA9



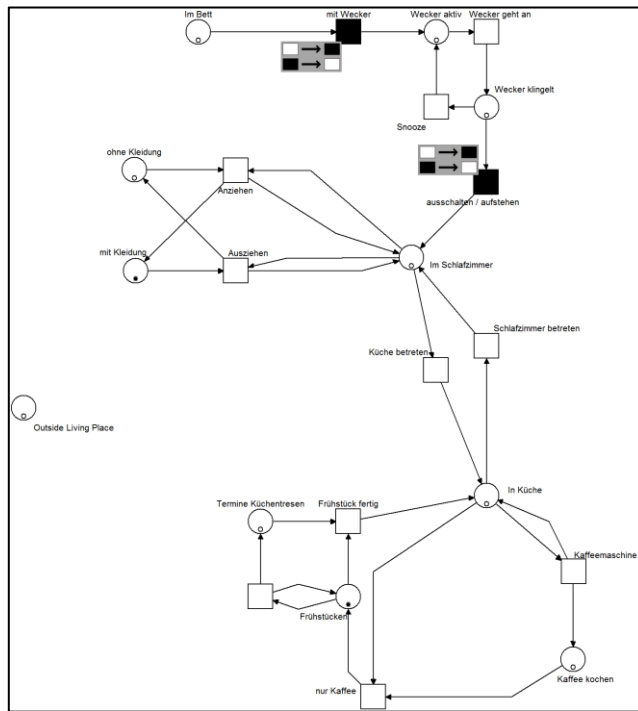
Netz NA10



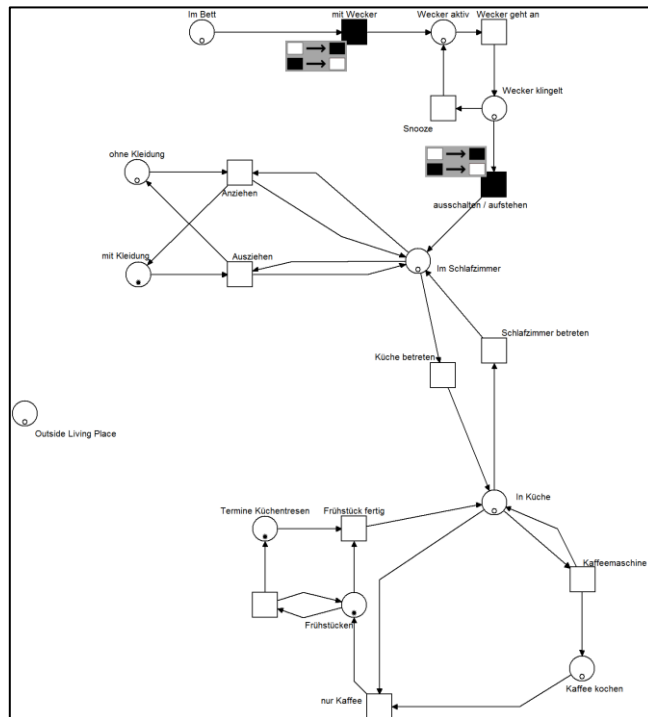
Netz NA11



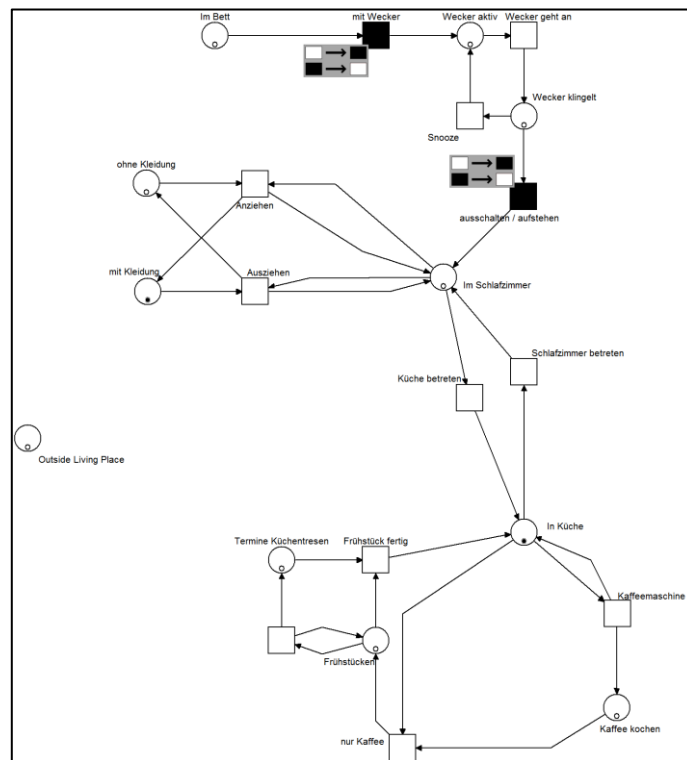
Netz NA12



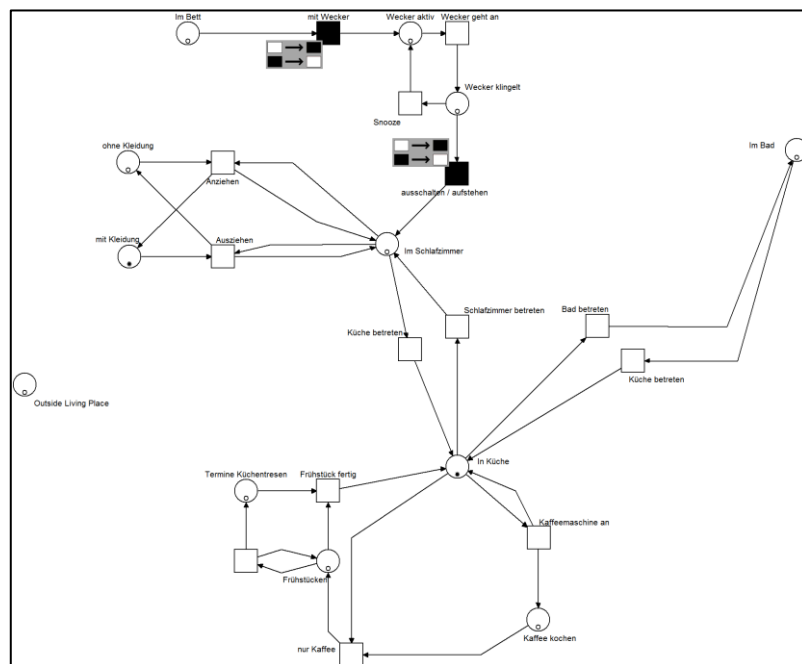
Netz NA13



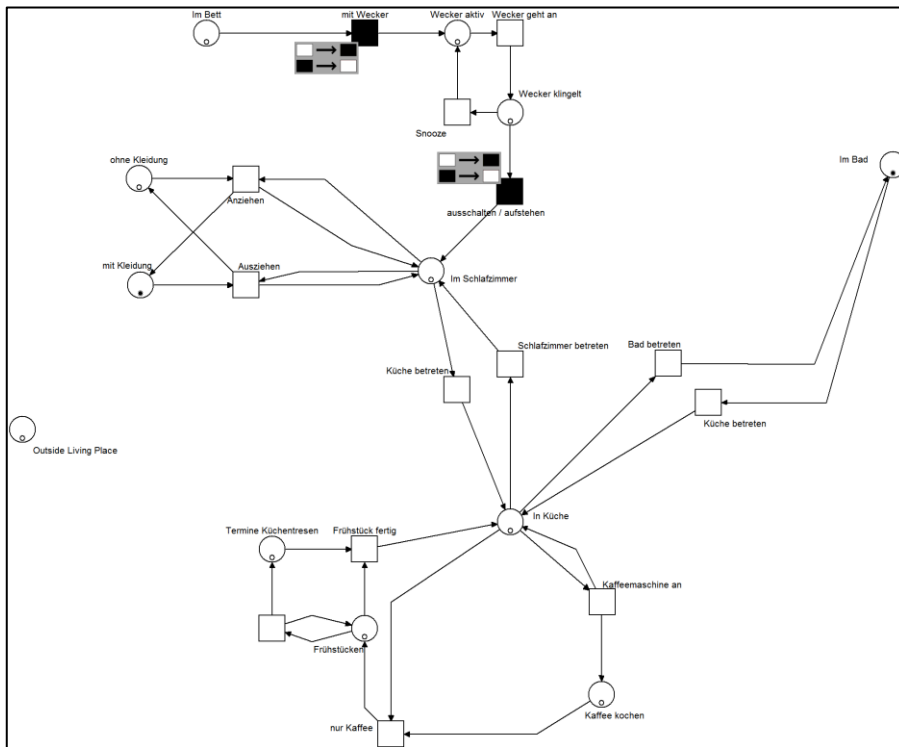
Netz NA14



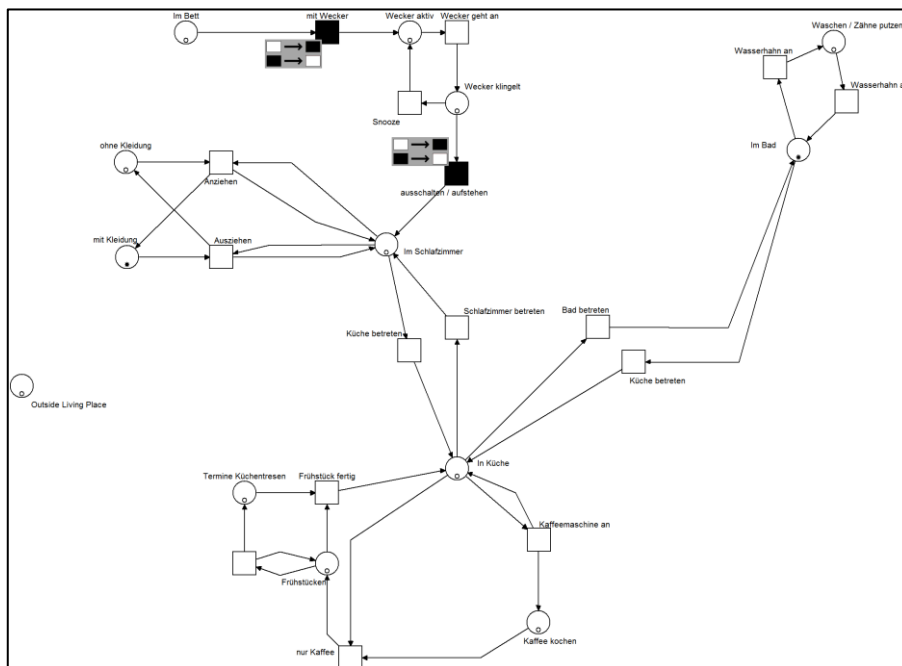
Netz NA15



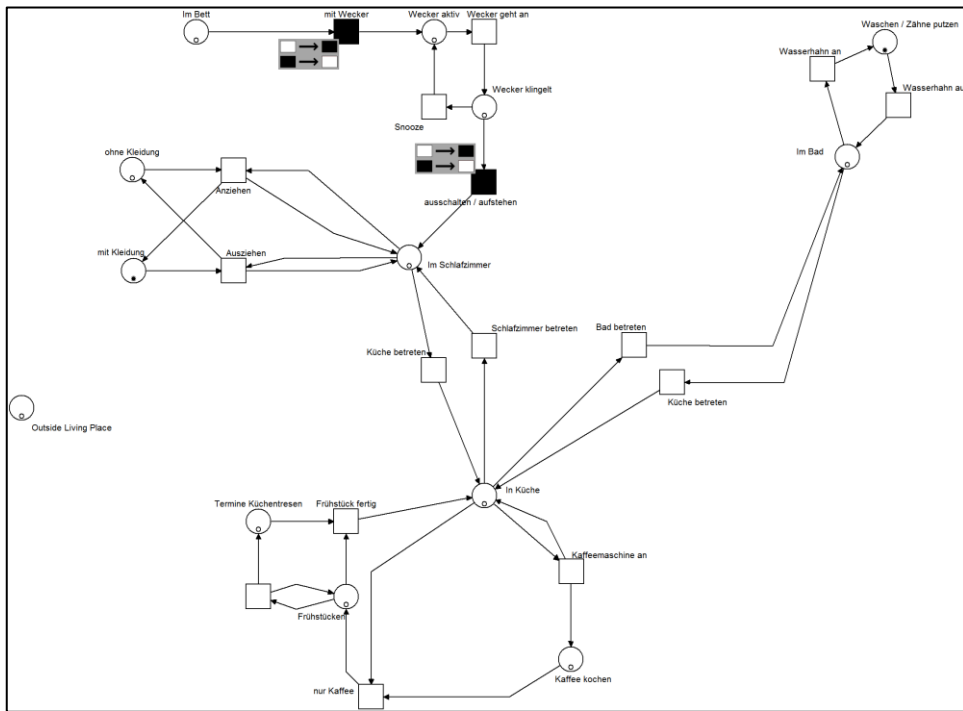
Netz NA16



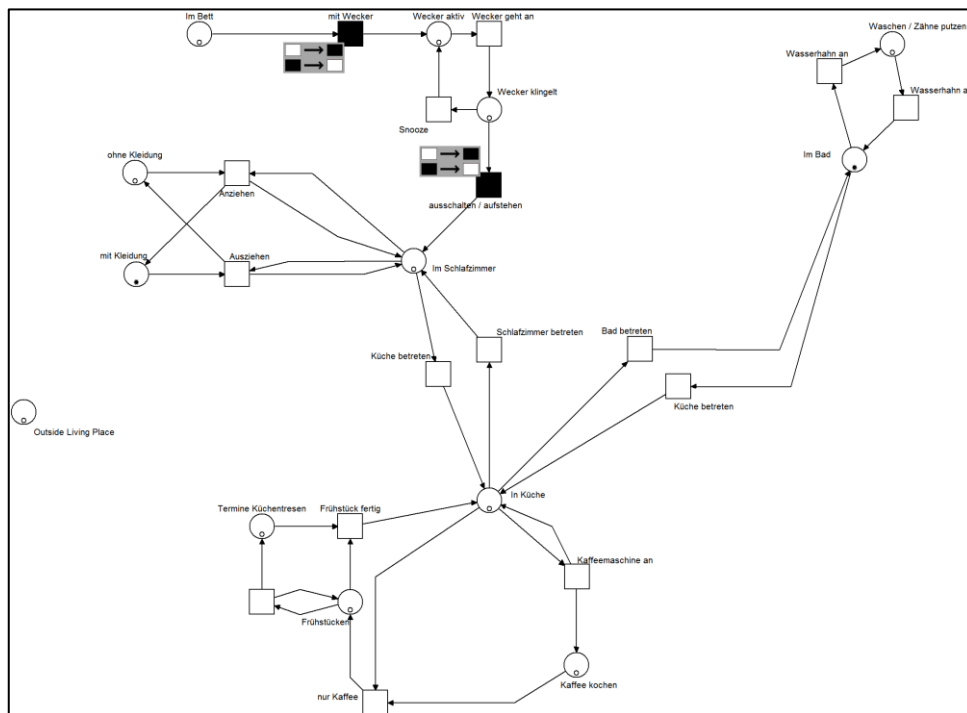
Netz NA17



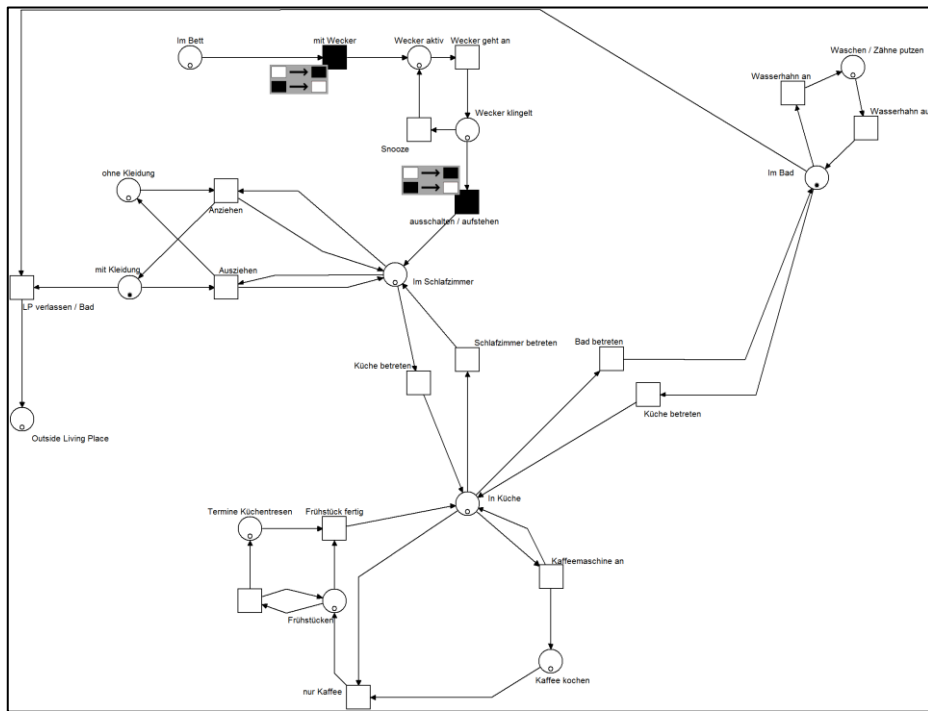
Netz NA18



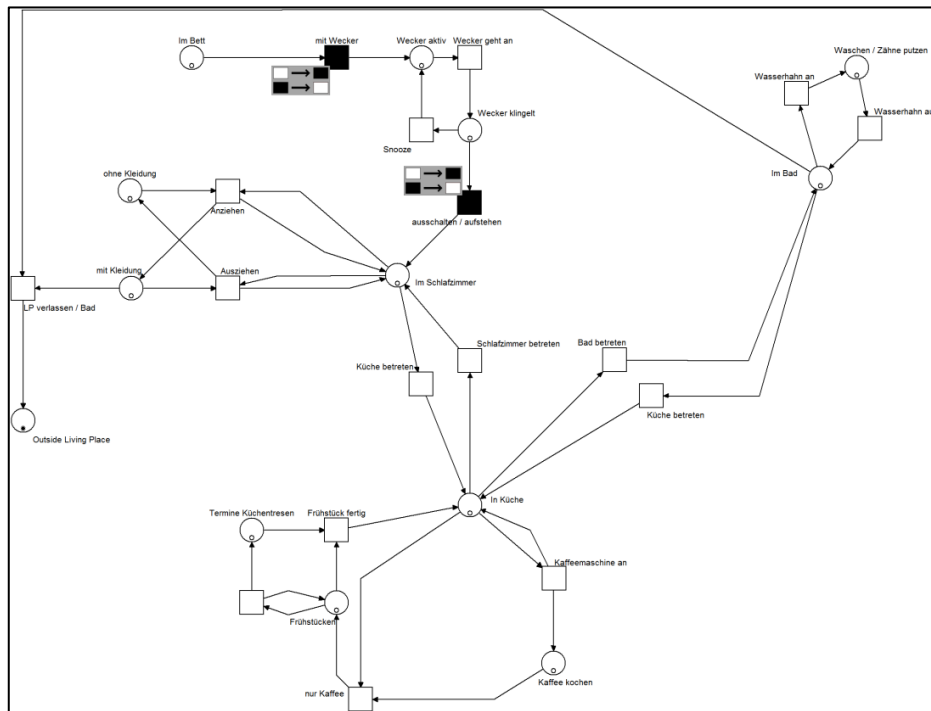
Netz NA19



Netz NA20



Netz NA21



Netz NA22

C Inhalt der CD

Eine CD ist dieser Arbeit beigelegt. Darin sind folgende Daten enthalten:

- Diese Bachelorarbeit als PDF
- Formale Darstellung aller Szenarien
- Eine Übersicht aller Regeln
- Alle Regeln (R1 – R35)
- Alle Löschregeln (LöRe36 – LöRe71)
- Die Netze aller drei Szenarien
- Das Petrinetz-Tool *Netlab*

für Piet Matthis

Versicherung über Selbstständigkeit

Hiermit versichere ich, dass ich die vorliegende Arbeit ohne fremde Hilfe selbstständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.

Hamburg, den _____