Strukturelle Stationarität

Roland Meyer

roland.meyer@liafa.jussieu.fr

Abstract: In dynamischen Netzwerken ändern sich zur Laufzeit sowohl die Zahl der Komponenten als auch deren Verbindungstruktur. Während die Literatur verschiedene abstraktionsbasierte Verifikationsverfahren vorschlägt, ist die Ausdrucksmächtigkeit dynamischer Netzwerke kaum verstanden. Die Dissertation liefert eine Klassifikation dynamischer Netzwerke, die auf engen Beziehungen zu Automatenmodellen beruht. Zugleich formen die resultierenden endlichen Darstellungen die Basis für Entscheidbarkeitsresultate und, aufbauend, vollständige automatische Verifikationsalgorithmen. Die Schlüsselbeobachtung der Theorie sind *Verbindungsmuster*, Regularitäten in der Netzwerkstruktur.

1 Einleitung

"Finally the checker has to verify that the process comes to an end."(Alan Turing 1949). In dem Artikel *Checking a Large Routine* studiert Turing ein Programm zur Berechnung der Fakultät und moniert die klaffende Lücke zwischen der mathematischen Formulierung der Funktion und ihrer Berechnung mit den in einem Computer verfügbaren Befehlen [MJ84]. Um die Korrektheit des Programms zu garantieren, sei ein Beweis notwendig, folgert Turing. Das Programm solle mit Zusicherungen versehen werden, aus denen ein Mathematiker, den er *Checker* nennt, die Korrektheit ableiten könne.

Turings bahnbrechende Arbeit legt den Grundstein für die Programmverifikation, einen heute etablierten Bereich der Informatik. Ihr prominentester Zweig, die computergestützte Verifikation, hat seinen Mathematiker längst durch einen Algorithmus ersetzt. Diese noch immer als *(Model-)Checker* bezeichneten Verfahren untersuchen weiterhin die Korrektheit von Programmen. Allerdings hat sich die Problemklasse geändert. An die Stelle der sequentiellen Programme sind verteilte getreten, und neben Terminierung sind temporale Abhängigkeiten zwischen den Befehlen der Komponenten einzuhalten. Außerdem wächst der Grad an Nebenläufigkeit mit neuen Technologien stetig.

2010 werden selbst kritische Anwendungen wie das Homebanking über Internetschnittstellen realisiert. Ähnlich wie verteilte Programme bestehen diese *dynamischen Netzwerke* aus interagierenden Komponenten. Allerdings ändern sich in dynamischen Netzwerken sowohl deren Anzahl als auch deren Verbindungsstruktur zur Laufzeit. Als Konsequenz hängt die Korrektheit dynamischer Netzwerke von der Verbindungsstruktur ab, nicht mehr allein von der Synchronisation der Komponenten. Zum Beispiel ist beim Homebanking vor jeder Überweisung ein sicherer Kanal zwischen Client und Server aufzubauen.

Auch für Verifikationsalgorithmen hat der Unterschied eine entscheidende Konsequenz. Während verteilte Programme endlich viele Threads verwenden, gilt eine solche Endlichkeitsannahme für dynamische Netzwerke nicht. Oft greifen sehr viele Clients auf einen Bankserver zu, so dass sich keine Schranke für ihre Zahl rechtfertigen lässt. Dynamische Netzwerke sind *zustandsunendlich*. Automatisierte Verfahren sind daher von endlichen Darstellungen des Zustandsraums abhängig, die sie erschöpfend durchsuchen können.

In der Literatur zur Verifikation wird die Zustandsunendlichkeit dynamischer Netzwerke als Berechnungsvollständigkeit interpretiert, um die Untersuchung abstraktionsbasierter Analysemethoden zu rechtfertigen. Neben unvollständigen Verfahren hat Abstraktion den Nachteil, die wichtigen Lebendigkeitseigenschaften zu verlieren. Meine Dissertation [Mey09] hinterfragt die Annahme der Berechnungsvollständigkeit. Das Ergebnis ist eine Klassifikation zustandsunendlicher Netzwerke, die auf folgender Beobachtung beruht.

Trotz unbeschränkter Komponentenzahl verwenden viele Netzwerke eine endliche Anzahl wiederkehrender Verbindungsmuster. Netzwerke, die dieser Beschränkung genügen, werden als *strukturell stationär* bezeichnet. Diese Zusammenfassung bespricht in vier Kapiteln die wichtigsten Ergebnisse zu strukturell stationären Netzwerken aus [Mey09].

Die Eigenschaft erlaubt eine kanonische Darstellung des Netzwerks als endliches Stellen-Transitions-Petrinetz [Rei85], die als korrekt und optimal nachgewiesen wird. Der Hauptsatz von Kapitel 2 ist eine Charakterisierung struktureller Stationarität. Sie verallgemeinert bestehende Entscheidbarkeitsresultate in der Literatur.

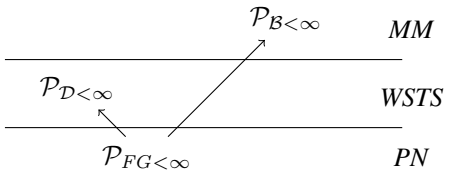

Abbildung 1: Beziehungen zwischen den Klassen dynamischer Netzwerke und zu Automatenmodellen. Die Bezeichnungen sind im Text erläutert. Pfeile repräsentieren Inklusionen.

In Abschnitt 3 werden die Vorteile der Petrinetzdarstellung für die automatische Verifikation aufgezeigt. Neben algorithmischen Aspekten der Übersetzung werden effiziente Analysemethoden skizziert.

Die Ausdrucksmächtigkeit strukturell stationärer Netzwerke ist das Thema von Kapitel 4. Die Eigenschaft zerfällt in die zwei Dimensionen beschränkter Tiefe und beschränkter Breite. Der Hauptsatz zeigt, dass die Transitionssysteme von Netzwerken beschränkter Tiefe wohlstrukturiert sind [FS01]. Die Konsequenz sind Entscheidbarkeitsresultate, die unter beschränkter Breite unmöglich sind. Diese Netzwerke sind tatsächlich berechnungsvollständig [Min67]. Abbildung 1 illustriert den Zusammenhang. Strukturell stationäre Netzwerke liegen in $\mathcal{P}_{FG<\infty}$, $\mathcal{P}_{\mathcal{D}<\infty}$ fordert beschränkte Tiefe und $\mathcal{P}_{\mathcal{B}<\infty}$ beschränkte Breite. Das Kürzel *PN* bezeichnet Petrinetze, *WSTS* wohlstrukturierte Transitionssysteme und *MM* die berechnungsvollständigen Minsky-Maschinen.

In Kapitel 5 wird die Grenze zwischen dynamischen Netzwerken und Petrinetzen herausgearbeitet, ein trotz seiner Bedeutung ungelöstes Problem. Die neue Übersetzung wird um die Darstellung von Nebenläufigkeit erweitert, bis ein Unentscheidbarkeitsbeweis weitere Verallgemeinerungen ausschließt. Die Konstruktionen benötigen nur beschränkte Tiefe.

Die Darstellung der Ergebnisse in dieser Zusammenfassung ist allgemein gehalten. Meine Dissertation untersucht strukturelle Stationarität für den π-Kalkül [MPW92], die Schlüsseldefinitionen lassen sich aber für Graphersetzungssysteme, Objekt-orientierte Programme oder allgemeine graphbeschriftete Transitionssysteme formulieren.

2 Strukturelle Stationarität

Auch bei unendlichen Zustandsräumen erlauben Petrinetze mächtige Entscheidbarkeitsresultate und bieten außerdem ausgereifte Analysewerkzeuge. Zur Verifikation verteilter Programme werden zwei Klassen von Petrinetzdarstellungen genutzt. Nebenläufigkeitssemantiken bilden die Interaktionen der Komponenten ab, kausale Semantiken spiegeln die Abhängigkeiten zwischen den Befehlen wider. Die Verbindungsstruktur geht verloren. Als entsprechend ungeeignet erwiesen sich die Ansätze zur Netzwerkanalyse.

Meine Dissertation schlägt vor, die Verbindungsstruktur als eine zu Nebenläufigkeit und Kausalität orthogonale Dimension dynamischer Netzwerke aufzufassen. Netzwerke sollten als Linkgraphen verstanden werden, nicht als unstrukturierte Gruppen von Threads. Die Schlüsselbeobachtung ist, dass die Graphen nur wenige *Verbindungsmuster* nutzen. So wird beim Homebanking jeder Client mit einem Thread des Bankservers verbunden. Diese Muster bilden die Stellen des Petrinetzes $\mathcal{N}[\![P]\!]$, der *strukturellen Semantik des Netzwerks* P. Die technische Basis der Übersetzung ist eine neue Normalform für die Terme des π-Kalküls. In dieser *Restriktionsform* erscheinen die Verbindungsmuster als Subterme, die im Folgenden *Fragmente* genannt werden.

Die strukturelle Semantik spiegelt das Verhalten des Netzwerks exakt wider. Die Transitionssysteme sind isomorph und auch das Inverse des Isomorphismus ist bekannt. Es erlaubt, die Zustände des Netzwerks aus den Markierungen des Petrinetzes zu bestimmen.

Theorem 1 (Full Retrievability)
$\mathcal{T}(P) = \mathcal{T}(\mathcal{N}[\![P]\!])$ und auch die Zustände des Netzwerks bleiben erhalten.

Die Übersetzung bewahrt also neben dem Transitionsverhalten die erreichbaren Verbindungen des Netzwerks. Der Erhalt der Netzwerkstruktur erlaubt es, Eigenschaften in Verbindungslogiken anhand der Petrinetzdarstellung nachzuweisen [CC03].

Je mehr Information eine Übersetzung bewahrt, desto mehr Speicherplatz benötigt sie. Die strukturelle Semantik enthält genau die zum Prüfen korrekter Verbindungen erforderliche Information. Die Beschreibungsmächtigkeit von Verbindungslogiken stimmt mit struktureller Kongruenz überein. Zwei Netzwerke P und Q erfüllen genau dann dieselben Formeln der Verbindungslogik, wenn ihre Strukturen in der Relation $\equiv$ übereinstimmen [Hir04]. Auch die strukturelle Semantik entspricht dieser Relation.

Theorem 2 (Full Abstraction)
$\mathcal{N}[\![P]\!] = \mathcal{N}[\![Q]\!]$ genau dann, wenn $P \equiv Q$.

Die Kombination dieses Resultats mit der zitierten logischen Äquivalenz zeigt, dass zwei Netzwerke genau dann dieselben Eigenschaften der Verbindungslogik erfüllen, wenn sie auf dasselbe Petrinetz abgebildet werden. Bewahrte die Übersetzung weniger Information, gäbe es Eigenschaften, die sich an ihr nicht nachweisen ließen. In diesem Sinn ist die strukturelle Semantik für den Nachweis korrekter Verbindungen optimal.

Während der π-Kalkül berechnungsvollständig ist, sind bei endlichen Petrinetzen wichtige Eigenschaften entscheidbar. Entsprechend muss die strukturelle Semantik einige Netzwerke auf unendliche Petrinetze (mit unendlicher Stellen- oder Transitionszahl) abbilden. Unter der strukturellen Semantik werden genau die Netzwerke endlich dargestellt, deren Zustände sich einer endlichen Menge an Fragmenten bedienen. Sie werden als *strukturell stationär* bezeichnet und gehören der Klasse $\mathcal{P}_{FG<\infty}$ an.

Lemma 3 (Endlichkeitscharakterisierung)
$\mathcal{N}[\![P]\!]$ ist genau dann endlich, wenn $P \in \mathcal{P}_{FG<\infty}$.

Das Lemma hat überraschende Konsequenzen. Anders als bisher angenommen impliziert die Kombination von unbeschränkter Thread- und unbeschränkter Linkerstellung *nicht* die Notwendigkeit einer unendlichen automatentheoretischen Darstellung des Netzwerks.

Da sich die Definition struktureller Stationarität auf die Form der erreichbaren Fragmente stützt, ist der Nachweis der Eigenschaft schwierig. Das Problem lässt sich auf die Existenz einer Schranke reduzieren. Ein Netzwerk ist genau dann strukturell stationär, wenn die Anzahl der sequentiellen Komponenten in jedem Fragment beschränkt ist. Bezeichnet man die beschränkten Netzwerke mit $\mathcal{P}_{S<\infty}$, gilt folgende Charakterisierung.

Theorem 4 (Charakterisierung struktureller Stationarität I)
$\mathcal{P}_{FG<\infty} = \mathcal{P}_{S<\infty}$.

Der Beweis bedient sich einer neuen Theorie der *Ableitungen* eines π-Kalkül-Terms. Aus dieser endlichen Menge lassen sich bei gegebener Schranke alle Fragmente bestimmen.

Theorem 4 ist ein mächtiges Werkzeug zum Beweis struktureller Stationarität. Es zeigt zum Beispiel, dass $\mathcal{P}_{FG<\infty}$ zwei bekannte Klassen der Literatur verallgemeinert. Sowohl Netzwerke mit statischer Komponentenzahl und dynamischer Linkerstellung [Dam96] als auch solche mit dynamischer Komponentenerstellung und statischer Verbindungsstruktur [AM02] sind strukturell stationär. Für beide Klassen waren endliche Darstellungen und Entscheidbarkeiten bekannt, jedoch fehlte ein einheitliches Verständnis der Ergebnisse. Strukturelle Stationarität ist das benötigte Bindeglied.

Zur Modellierung von Client-Server-Architekturen wie der Homebanking-Anwendung wird die Klasse der Netzwerke mit *beschränktem Server* vorgeschlagen. Der Zugang zu einem lokalen Netzwerk geschieht über einen Server, der die Anzahl der Teilnehmer begrenzt. Das Kommunikationsprotokoll ist beliebig. Mit einer geeigneten Invariante folgt strukturelle Stationarität aus Theorem 4.

Zur Komplexitätsabschätzung wird die Kodierung von Petrinetzen zurück in strukturell stationäre Netzwerke untersucht. Sie zeigt Unentscheidbarkeit sowie EXPSPACE-Härte

einiger Verifikationsprobleme für strukturell stationäre Netzwerke. Ferner wird bewiesen, dass die Größe der strukturellen Semantik nicht durch eine primitiv-rekursive Funktion in der Größe des Netzwerks beschränkt ist. Das Theorem bedient sich einer ausgefeilten Konstruktion. Um dennoch die Praktikabilität des vorgeschlagenen übersetzungsbasierten Verifikationsansatzes zu belegen, werden industrielle Fallstudien betrachtet.

3 Anwendung in der Verifikation

Tim Strazny implementierte die strukturelle Semantik innerhalb des *Petruchio*-Werkzeugs [MS10]. Die deklarative Definition lässt das Problem der Berechenbarkeit der strukturellen Semantik $\mathcal{N}[\![P]\!]$ offen. Mit dem klassischen Blick der denotationellen Semantik kann sie als kleinster Fixpunkt einer Funktion ϕ_P auf einer besonderen Klasse von Petrinetzen aufgefasst werden, $\mathcal{N}[\![P]\!] = \mathit{lfp}(\phi_P)$. Die Stetigkeit der Funktion ermöglicht eine unkonventionelle Kleene-Iteration zum Fixpunkt, die ihrerseits Überdeckbarkeitsanfragen stellt. Sie terminiert genau dann, wenn die Menge der Fragmente stationär wird.

Mit Hilfe des Werkzeugs wurden drei industrielle Fallstudien verifiziert und korrigiert: ein bekanntes Modell des GSM-Mobilfunknetzes [OP92], eine vereinfachte Version eines Autobahnkontrollsystems [HESV91] sowie ein vollständiges Modell eines automatisierten Produktionssystems [BR01]. Bei 195 Zeilen π-Kalkül-Code ist eine manuelle Verifikation des letzten Modells nicht mehr möglich. Es enthielt in der Tat einen subtilen Fehler, der mit der neuen Verifikationstechnik aufgedeckt und behoben werden konnte.

Für die Fallstudien wurden nicht-triviale Eigenschaften untersucht, die sich in drei Klassen unterteilen lassen. *Topologischen* Bedingungen fassen die Korrektheit der Verbindungsstruktur. *Temporalen* Eigenschaften fordern gewisse Ereignisfolgen und verbieten andere. *Quantitative* Constraints garantieren Relationen zwischen den Komponentenanzahlen.

Zur Verifikation vieler Aussagen sind effiziente Algorithmen, die allein die Graphstruktur des Petrinetzes untersuchen, ausreichend. Sie vermeiden aufwendige Zustandsraumberechnungen und umgehen daher die Zustandsraumexplosion, das Hauptproblem in der automatischen Verifikation. Die Verfahren stützen sich in hohem Maße auf die Definition der strukturellen Semantik. Zum Beispiel gibt eine Stelle im Petrinetz alle dynamischen Links der enthaltenen Komponenten an. Außerdem werden geschlossene Netzwerke ohne statische Verbindungen in die Teilklasse der kommunikationsfreien Petrinetze übersetzt, für die besonders effiziente Analysemethoden verfügbar sind.

Proposition 5
Falls $P \in \mathcal{P}_{FG<\infty}$ ein geschlossenes Netzwerk ist, so ist $\mathcal{N}[\![P]\!]$ kommunikationsfrei.

Netzwerke mit statischer Komponentenanzahl und dynamischer Linkerstellung [Dam96] liefern unter der strukturellen Semantik Petrinetze mit sehr kleinen Schranken. Genauer werden die Gleichungen, die das Verhalten einer Komponente definieren, als ihr *Orbit* bezeichnet. Der maximale Schnitt dieser Orbits sei $\|P\|_\cap$.

Theorem 6 (Schrankensatz)
$\mathcal{N}[\![P]\!]$ ist $\|P\|_\cap$-beschränkt, falls $P \in \mathcal{P}_{FG<\infty}$ eine feste Komponentenzahl besitzt.

Während eines Aufenthalts an der Universität Newcastle wurde, basierend auf Theorem 6, eine Werkzeugkette zur Analyse von Netzwerken mit statischer Threadzahl entwickelt. Sie berechnet aus der strukturellen Semantik eine sogenannte Entfaltung, die das Verhalten des Petrinetzes kompakter repräsentiert als das Transitionssystem. Anschließend wird die Entfaltung in ein SAT-Problem überführt, welches die Erreichbarkeit von Zuständen charakterisiert [KKY04].

4 Tiefe und Breite

Um zu verstehen, welche Netzwerke sich nicht mit Hilfe der strukturellen Semantik und den aufbauenden Techniken verifizieren lassen, wird nach einer intuitiven Erklärung struktureller Stationarität gesucht. Das erste Haupresultat ist eine Charakterisierung, die die Eigenschaft als Beschränktheit zweier Funktionen formuliert. Die *Tiefe* misst die wechselseitige Abhängigkeit von Restriktionen eines Terms, die *Breite* ermittelt ihre Verteilung. Die Klassen der entsprechend beschränkten Netzwerke sind $\mathcal{P}_{\mathcal{D}<\infty}$ und $\mathcal{P}_{\mathcal{B}<\infty}$.

Theorem 7 (Charakterisierung struktureller Stationarität II)
$\mathcal{P}_{FG<\infty} = \mathcal{P}_{\mathcal{D}<\infty} \cap \mathcal{P}_{\mathcal{B}<\infty}$.

Aus theoretischer Sicht ist interessant, dass sich diese Charakterisierung auf den Restriktionsoperator stützt, während die erste den Paralleloperator des π-Kalküls nutzt. Allerdings hat Theorem 7 auch wichtige praktische Konsequenzen, da es obige Frage beantwortet. Sobald Tiefe oder Breite nicht beschränkt sind, ist ein Netzwerk nicht mehr strukturell stationär. Die Unendlichkeit der Übersetzung hat also genau zwei mögliche Ursachen.

Ein Verständnis von Tiefe und Breite ist leider schwierig, da sich die Definitionen auf alle Terme einer strukturellen Kongruenzklasse beziehen. In der Dissertation werden intuitive Charakterisierungen von deren Beschränktheit erarbeitet, die sich die Interpretation von Termen als Hypergraphen zunutze machen [MPW92]. Ein Netzwerk ist genau dann in der Tiefe beschränkt, wenn es eine Schranke für die Länge der einfachen Pfade in den Hypergraphen gibt. Einfache Pfade wiederholen keine Hyperkanten, die so beschränkten Netzwerke liegen in $\mathcal{P}_{lsp<\infty}$. Für die Breite gilt sogar ein stärkeres Resultat, sie ist gleich dem maximalen Grad der Hyperkanten. Das folgende Theorem gibt nur das Korollar für die entsprechend beschränkten Netzwerke in $\mathcal{P}_{deg<\infty}$ an.

Theorem 8 (Charakterisierungen beschränkter Tiefe und Breite)
$\mathcal{P}_{\mathcal{D}<\infty} = \mathcal{P}_{lsp<\infty}$ und $\mathcal{P}_{\mathcal{B}<\infty} = \mathcal{P}_{deg<\infty}$.

Der Beweis der ersten Gleichheit ist ein tiefes Resultat, das sich einer strengen Normalform unter struktureller Kongruenz bedient. Diese *verankerten Fragmente* ermöglichen

erst die folgenden Entscheidbarkeitsresultate für Netzwerke beschränkter Tiefe.

Mit der Äquivalenz zu struktureller Stationarität besitzen viele Netzwerktypen Schranken für die Tiefe und die Breite. Es gibt aber auch zwei entscheidbare π-Kalkül Klassen, die wichtige Funktionen dynamischer Netzwerke fassen und nicht strukturell stationär sind. Amadio betrachtet Modelle für Internetanwendungen [Ama00]. Busi und Gorrieri untersuchen Netzwerke mit einer beschränkten Anzahl dynamischer Links [BG09]. Obwohl nicht strukturell stationär, ist die Tiefe in beiden Klassen beschränkt. Diese Beobachtung motiviert die Suche nach Entscheidbarkeiten in den größeren Klassen $\mathcal{P}_{\mathcal{D}<\infty}$ und $\mathcal{P}_{\mathcal{B}<\infty}$.

Der zweite Hauptbeitrag dieses Abschnitts sind Entscheidungsverfahren für die Terminierung und die Endlichkeit des Zustandsraums in Netzwerken beschränkter Tiefe. Mit obiger Argumentation ist $\mathcal{P}_{\mathcal{D}<\infty}$ daher die bislang ausdrucksmächtigste und gleichzeitig entscheidbare Klasse dynamischer Netzwerke. Sie umfasst alle in der Literatur als entscheidbar nachgewiesenen Typen. Wie bereits von Turing beobachtet, ist Terminierung ein bedeutendes Problem. In der computergestützten Verifikation ist es die Grundlage der automatentheoretischen Analyse von Lebendigkeitseigenschaften [Var91].

Die Entscheidbarkeit folgt aus einem allgemeineren Ergebnis. Die Transitionssysteme von Netzwerken beschränkter Tiefe sind *wohlstrukturiert* [FS01]. Das Rahmenwerk der Wohlstrukturiertheit verallgemeinert Entscheidbarkeitsresultate für Petrinetze und Lossy-Channel-Systeme. Formal heißt ein Transitionssystem $(S, \to)$ wohlstrukturiert, falls die Zustände mit einer Ordnung $\leq \subseteq S \times S$ versehen werden können, die mit der Transitionsrelation $\to \subseteq S \times S$ verträglich ist. Abhängig von der Ordnung und der Verträglichkeit liefert die Instanziierung Entscheidungsverfahren für Terminierung oder Überdeckbarkeit.

Theorem 9 (Wohlstrukturiertheit von $\mathcal{P}_{\mathcal{D}<\infty}$)
Das Transitionssystem $(Reach(P)/_{\equiv}, \to, \preceq_{\mathcal{P}})$ eines in der Tiefe beschränkten Netzwerks $P \in \mathcal{P}_{\mathcal{D}<\infty}$ ist wohlstrukturiert. In $\mathcal{P}_{\mathcal{D}<\infty}$ sind Terminierung und die Endlichkeit des Zustandsraums entscheidbar.

Der technische Beitrag ist die neue Ordnungsrelation $\preceq_{\mathcal{P}}$ auf den Termen des π-Kalküls. Auf $\mathcal{P}_{\mathcal{D}<\infty}$ ist sie sogar eine *Wohlquasiordnung*, in jeder unendlichen Folge von Termen existieren zwei mit $\preceq_{\mathcal{P}}$ vergleichbare. Der Beweis verknüpft die Restriktionsform mit der Theorie der Ableitungen und den verankerten Fragmenten, spannt also einen Bogen über die Kernbeiträge der Dissertation. Da $\preceq_{\mathcal{P}}$ zugleich eine Simulation ist, ist die Ordnung in einem strengen Sinn mit der Transitionsrelation verträglich.

Die beiden Eigenschaften werden über ein induktives Argument auf einem endlichen Präfix des Berechnungsbaums entschieden. Ein kürzlich erschienener Artikel von Wies, Zuffrey und Henzinger erweitert die Theorie der tiefenbeschränkten Netzwerke [WZH10]. Sie beschreiben die Grenzwerte von Zustandsfolgen, womit sich ein neuer Algorithmus zum Prüfen der Überdeckbarkeit von Zuständen ergibt.

In beschränkter Breite ist eine Kodierung von Minsky-Maschinen möglich [Min67].

Theorem 10 (Berechnungsvollständigkeit von $\mathcal{P}_{\mathcal{B}<\infty}$)
$\mathcal{P}_{\mathcal{B}<\infty}$ ist berechnungsvollständig.

Die Darstellung von Minsky-Maschinen hat interessante Konsequenzen. Es ist unentscheidbar, ob ein gegebenes Netzwerk strukturell stationär ist. Mit der Äquivalenz aus Theorem 7 folgt unmittelbar die Unentscheidbarkeit der Tiefenbeschränktheit. Auch die Entscheidbarkeit der Breitenbeschränktheit wird negativ beantwortet.

5 Grenze zu Petrinetzen

Strukturell stationäre Netzwerke bedienen sich aller Verifikationstechniken für Petrinetze. Die Entscheidbarkeitsresultate für tiefenbeschränkte Netzwerke sind auf nur zwei Eigenschaften begrenzt und stellen keine Werkzeugunterstützung zur Verfügung. Da die bisherigen Ergebnisse die Existenz einer Petrinetzdarstellung dieser größeren Klasse nicht ausschließen, wird hier die Beziehung von tiefenbeschränkten Netzwerken zu Petrinetzen untersucht.

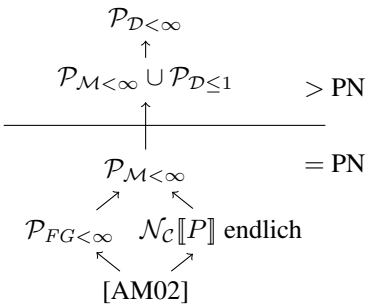

Abbildung 2: Beziehung von tiefenbeschränkten Netzwerken und Petrinetzen.

Das Hauptresultat ist eine Verallgemeinerung der strukturellen Semantik. Leider wird die gesamte Klasse $\mathcal{P}_{\mathcal{D}<\infty}$ der tiefenbeschränkten Netzwerke als ausdrucksmächtiger als Petrinetze bewiesen. Das Erreichbarkeitsproblem ist bereits bei einer Tiefe von eins unentscheidbar. Die Erweiterung der Übersetzung und die Schranke der Unentscheidbarkeit liegen nahe beieinander, so dass die Grenze zwischen den Modellen gefunden ist. Der Ansatz, die strukturelle Semantik zu erweitern, ist folgender.

Wie in Kapitel 2 diskutiert, sind Struktur und Nebenläufigkeit orthogonale Aspekte eines Netzwerks. Um beide Dimensionen zu verknüpfen, wird eine *Nebenläufigkeitssemantik* $\mathcal{N}_{\mathcal{C}}[\![P]\!]$ definiert, die sich einer Endlichkeitscharakterisierung ähnlich Lemma 3 erfreut und zugleich ein bisimulationsäquivalentes Transitionssystem besitzt (vgl. Theorem 1).

Theorem 11 (Korrektheit und Endlichkeit der Nebenläufigkeitssemantik)
$\mathcal{T}(P) \approx \mathcal{T}(\mathcal{N}_{\mathcal{C}}[\![P]\!])$. Ferner ist $\mathcal{N}_{\mathcal{C}}[\![P]\!]$ genau dann endlich, wenn eine Schranke für die Anzahl der erstellten dynamischen Links existiert.

Das Theorem löst ein offenes Problem in der Nebenläufigkeitstheorie. In der Literatur wurden verschiedene Nebenläufigkeitssemantiken vorgeschlagen [BG95, Eng96, BG09], die entweder unendliche Petrinetze verwenden oder inkorrekt im Sinne fehlender Bisimulationsäquivalenz sind. Der Schlüssel zur Definition von $\mathcal{N}_{\mathcal{C}}[\![-]\!]$ ist ein neues Transitionssystem, das die Ordnung der dynamisch erstellten Links bewahrt.

Getypte Restriktionen erlauben die Verbindung der Semantiken. Ungetypte Restriktionen werden von der strukturellen, getypte von der Nebenläufigkeitssemantik übersetzt. Die resultierende *gemischte Semantik* $\mathcal{N}_{\mathcal{M}}[\![P]\!]$ ist wieder bisimulationsäquivalent. Ferner stellt

sie die *gemischt-beschränkten* Netzwerke in $\mathcal{P}_{\mathcal{M}<\infty}$ endlich dar. Ungetypte Links bilden endlich viele Fragmente, von den getypten Links werden nur endlich viele erstellt.

Theorem 12 (Korrektheit und Endlichkeit der gemischten Semantik)
$\mathcal{T}(P) \approx \mathcal{T}(\mathcal{N}_{\mathcal{M}}[\![P]\!])$. Ferner ist $\mathcal{N}_{\mathcal{M}}[\![P]\!]$ genau dann endlich, wenn $P \in \mathcal{P}_{\mathcal{M}<\infty}$.

Die gemischte Semantik erweitert die anderen beiden auf strenge Weise.

Theorem 13 (Konservative Erweiterung)
Für ein vollständig getyptes Netzwerk $P \in \mathcal{P}_{\mathcal{M}<\infty}$ gilt $\mathcal{N}_{\mathcal{M}}[\![P]\!] = \mathcal{N}_{\mathcal{C}}[\![P]\!]$. Für ein ungetyptes Netzwerk gilt $\mathcal{N}_{\mathcal{M}}[\![P]\!] = \mathcal{N}[\![P]\!]$.

Natürlich existieren Netzwerke, die unter der gemischten Semantik endlich dargestellt werden, aber nicht unter einer der beiden anderen. Außerdem sind gemischt-beschränkte Netzwerke in der Tiefe beschränkt, $\mathcal{P}_{\mathcal{M}<\infty} \subseteq \mathcal{P}_{\mathcal{D}<\infty}$.

Überraschend konnte gezeigt werden, dass die gemischte Semantik die Grenze zwischen tiefenbeschränkten Netzwerken und Petrinetzen markiert. Werden die Netzwerke in $\mathcal{P}_{\mathcal{M}<\infty}$ minimal erweitert, wird die Erreichbarkeit von Zuständen unentscheidbar.

Theorem 14
In $\mathcal{P}_{\mathcal{D}\leq 1}$ ist das Erreichbarkeitsproblem unentscheidbar.

Abbildung 2 fasst die Zusammenhänge dieses Kapitels zusammen. Die Einbettung von Petrinetzen findet sich in Kapitel 2.

Danksagung Ich danke meinem Doktorvater Ernst-Rüdiger Olderog für die ausgezeichnete Anleitung und die kostbaren Ratschläge zur wissenschaftlichen Arbeit.

Literatur

[AM02] R. Amadio und C. Meyssonnier. On Decidability of the Control Reachability Problem in the Asynchronous π-Calculus. *Nordic J. Comp.*, 9(1):70–101, 2002.

[Ama00] R. Amadio. On Modelling Mobility. *TCS*, 240(1):147–176, 2000.

[BG95] N. Busi und R. Gorrieri. A Petri Net Semantics for π-Calculus. In *CONCUR*, Band 962 der LNCS, Seiten 145–159. Springer, 1995.

[BG09] N. Busi und R. Gorrieri. Distributed semantics for the π-calculus based on Petri nets with inhibitor arcs. *J. Logic and Algebraic Programming*, 78(1):138–162, 2009.

[BR01] A. Braatz und A. Ritter. Referenzfallstudie Produktionstechnik. Bericht, IFF Universität Stuttgart und Fraunhofer IPA Stuttgart, 2001.

[CC03] L. Caires und L. Cardelli. A Spatial Logic for Concurrency (Part I). *Inf. and Comp.*, 186(2):194–235, 2003.

[Dam96] M. Dam. Model Checking Mobile Processes. *Inf. and Comp.*, 129(1):35–51, 1996.

[Eng96] J. Engelfriet. A Multiset Semantics for the pi-Calculus with Replication. *TCS*, 153(1-2):65–94, 1996.

[FS01] A. Finkel und Ph. Schnoebelen. Well-Structured Transition Systems Everywhere! *TCS*, 256(1-2):63–92, 2001.

[HESV91] A. Hsu, F. Eskafi, S. Sachs und P. Varaiya. Design of Platoon Maneuver Protocols for IVHS. Path research report, University of California, Berkeley, 1991.

[Hir04] D. Hirschkoff. An Extensional Spatial Logic for Mobile Processes. In *CONCUR*, Band 3170 der LNCS, Seiten 325–339. Springer, 2004.

[KKY04] V. Khomenko, M. Koutny und A. Yakovlev. Detecting State Encoding Conflicts in STG Unfoldings Using SAT. *Fund. Inf.*, 62(2):221–241, 2004.

[Mey09] R. Meyer. *Structural Stationarity in the π-Calculus*. Dissertation, Department für Informatik, Carl von Ossietzky Universität Oldenburg, 2009.

[Min67] M. Minsky. *Computation: Finite and Infinite Machines*. Prentice Hall, 1967.

[MJ84] F. L. Morris und C. B. Jones. An Early Program Proof by Alan Turing. *IEEE Annals of the History of Computing*, 6(2):139–143, 1984.

[MPW92] R. Milner, J. Parrow und D. Walker. A Calculus of Mobile Processes (Part I). *Inf. and Comp.*, 100(1):1–40, 1992.

[MS10] R. Meyer und T. Strazny. Petruchio: From Dynamic Networks to Nets. In *CAV*, LNCS. Springer, 2010. Zur Publikation akzeptiert.

[OP92] F. Orava und J. Parrow. An Algebraic Verification of a Mobile Network. *Formal Aspects of Computing*, 4(6):497–543, 1992.

[Rei85] W. Reisig. *Petri nets: An Introduction*. Monographs in Theoretical Computer Science. An EATCS Series. Springer, 1985.

[Var91] M. Y. Vardi. Verification of Concurrent Programs: The Automata-Theoretic Framework. *Annals of Pure and Applied Logic*, 51(1-2):79–98, 1991.

[WZH10] T. Wies, D. Zuffrey und T. A. Henzinger. Forward Analysis of Depth-Bounded Processes. In *FoSSaCS*, Band 6014 der LNCS, Seiten 94–108. Springer, 2010.

Roland Meyer, geboren 1981, absolvierte von 2001 bis 2005 ein Studium der Informatik mit Nebenfach Mathematik an der Carl von Ossietzky Universität Oldenburg. Nach dem Abschluss als Diplom-Informatiker im September 2005 begann er ein Promotionsstudium, das im Rahmen des Graduiertenkollegs *Vertrauenswürdige Software-Systeme* mit einem Stipendium der DFG gefördert wurde. Seine Arbeit über *Strukturelle Stationarität im π-Kalkül* beendete er im Februar 2009 mit der Note *summa cum laude*. Seit März 2009 forscht Roland Meyer als CNRS geförderter Post-Doc im LIAFA-Institut der Universität Paris Diderot - Paris 7. Seine Dissertation wurde im Jahr 2009 mit dem Gerhard-Wachsmann-Preis der Carl von Ossietzky Universität ausgezeichnet. Roland Meyer ist Mitglied des Programmkomitees der ACSD-Konferenz zur Anwendung von Nebenläufigkeit im Systemdesign.