Formale Modellierung interagierender autonomer und reaktiver Komponenten verteilter Systeme mit I–Systemen: Formale Basis und Beiträge zur Theorie

Dissertation

zur Erlangung des Grades eines

Doktors der Naturwissenschaften

der Universität Dortmund

am Fachbereich Informatik

von

Arnim Wedig

Dortmund

2004

Tag der mündlichen Prüfung: 07.01.2004

Dekan: Prof. Dr. Bernhard Steffen

Gutachter: Prof. Dr. Horst F. Wedde (Universität Dortmund)
Prof. Dr. Ingo Wegener (Universität Dortmund)
Prof. Dr. Eike Best (Universität Oldenburg)

Kurzfassung

Gegenstand der Arbeit ist die formale Modellierung des Systemverhaltens verteilter Systeme vom Standpunkt der Systemkontrolle. In dem früher entwickelten formalen Ansatz der Interaktionssysteme (I–Systeme) war das Kern-Prinzip, aus der expliziten Darstellung von lokalen Einflüssen ihre lokalen und globalen Effekte abzuleiten. Dies geschieht durch zwei Typen restriktiver bilateraler Interaktionsbeziehungen. Auf diese Weise scheint das Systemverhalten geprägt durch den Autonomiegrad der Systemkomponenten.
Die Arbeit beschäftigt sich mit der Strukturierung des Modellierungskonzeptes und liefert Definitionen und Interpretationen zu Syntax, Dynamik und Semantik von I–Systemen.
Ein offenes Problem bestand bisher in der Spezifikation geeigneter Semantiken von I–Systemen, über die sich Schlüsselphänomene im Systemverhalten interagierender Systemkomponenten dokumentieren lassen. Als Lösungsansatz werden hierzu neue Formen von Trace-Semantiken von I–Systemen präsentiert und Unterschiede in deren Ausdruckskräften, auch im Vergleich mit Zustandsgraphen, herausgearbeitet.
Die Komplexität in der formalen Modellierung praxisrelevanter verteilter Systeme erzwingt den Einsatz inkrementeller und modularer Entwurfs- und Analysemethoden. Mit Bezug auf die eingeführten Semantiken werden solche Methoden für I–Systeme vorgestellt. Hierbei werden auch I–System-Transformationen mit dem Ziel behandelt, I–Systeme in „handhabbarere“ (bzgl. Entwurf/Analyse/Implementierung) äquivalente I–Systeme umzuformen.
Insgesamt liefert diese Arbeit die formale Basis für die Weiterentwicklung der Theorie der I–Systeme.

MEINEN ELTERN
HERBERT UND INGRID

Danksagung

Diese Arbeit entstand während meiner Zeit am Lehrstuhl Informatik III der Universität Dortmund unter der Betreuung von Prof. Dr. Horst F. Wedde.

Mein Dank gilt Prof. Dr. Horst F. Wedde, der mir immer wieder konstruktive Anregungen gab und diese Arbeit mit großem Interesse verfolgt und betreut hat. Ich danke Prof. Dr. Ingo Wegener für sein Engagement als zweiter Gutachter dieser Arbeit. Seine Anmerkungen waren mir eine wertvolle Hilfe. Prof. Dr. Eike Best hat sich freundlicherweise als dritter Gutachter zur Verfügung gestellt. Ich bedanke mich bei ihm für die Mühen, die mit der eingehenden Durchsicht der Arbeit verbunden waren. Prof. Dr. Heiko Krumm und Dr. Oliver Rüthing danke ich für ihre Mitwirkung am Promotionsverfahren.
Ich sage Dank allen Teilnehmern an den LS III-Teepausen. Die gute kollegiale Atmosphäre hat mir sehr geholfen. Ein herzliches Dankeschön geht an Marcus Rattay für das gewissenhafte „syntaktische“ Korrekturlesen der Arbeit aus der Sicht des fachfremden Lesers.
Mein besonderer Dank gebührt schließlich meinen lieben Eltern für ihre unermüdliche, uneingeschränkte Unterstützung. Ich kann mich in allen Lebenslagen auf ihre Fürsorge verlassen und sie geben mir den familiären Rückhalt, der für mich notwendig ist, damit solch eine Arbeit entstehen kann. In diesem Sinne danke ich auch dem Rest meiner Sippe.

Inhaltsverzeichnis

Abbildungsverzeichnis

Kapitel 1

Einleitung

1.1 Motivation

Der Fortschritt in der Informationstechnologie zeichnet sich aktuell aus durch eine zunehmende Verteiltheit von Informationen und informationsverarbeitenden Prozessen. Als Beispiele hierzu seien nur die Nutzungsvielfalt des Internets, die Bereitstellung mobiler Kommunikationseinheiten innerhalb von Wireless-LANs, Heimautomatisierungsysteme in „Vernetzten Häusern" und verteilte partiell autonome Produktionssysteme genannt. Verteiltheit tritt dabei sowohl geographisch als auch funktionell auf. Vom Standpunkt der Systemkontrolle existiert keine ausgezeichnete zentrale Kontrolleinheit, die Aufgaben an die Systemkomponenten global verteilt, sondern die Systemkomponenten stehen innerhalb von Teilsystemen in Wechselwirkungen und kontrollieren kooperativ ihre lokale, meist heterogene Systemumgebung. *Ein adäquates formales Modell sollte Schlüsselphänomene solcher realen Systeme in elementare Konzepte und Eigenschaften abbilden.*

Man betrachte exemplarisch das folgende Beispiel eines *Leitsystems für Fußgänger im Straßenverkehr* (Personal Navigation System for Human Outdoor Activities) - im folgenden kurz LS genannt - aus [35]. Im Blickpunkt stehen Fußgänger, ausgestattet mit kompakten Notebooks, die durch den Straßenverkehr geleitet werden sollen. Die Anforderungen an die Art des Leitens variieren. So kann es unter anderem vorkommen, dass ein Fußgänger einen bestimmten Zielort ansteuern möchte (z.B. ein ortsunkundiger Tourist eine Sehenswürdigkeit) oder aber ein konkreter Zielort nicht vorgegen wird (z.B. beim Stadtbummel) und hingegen die Vermeidung von Gefahrensituation einer besonderen Beachtung bedarf (z.B. bei sehbehinderten Personen). Das LS umfasst eine Vielzahl unterschiedlich gearteter quasi-autonomer Systemkomponenten, die miteinander kommunizieren, um die erforderlichen Leitinformationen zu generieren. Komponenten sind:

- Fußgänger mit Notebooks: Das LS wird mit unterschiedlichen Benutzergruppen konfrontiert und muss auf deren jeweilige Voraussetzungen und Bedürfnisse reagieren können. Die Bedürfnisse können sich dynamisch ändern (z.B. neuer Zielort, Verspätungen von öffentlichen Verkehrsmitteln, zusätzliche Begleitpersonen, Termine). Erhalten Fußgänger Anweisungen vom LS, können sie autonom entscheiden, inwieweit sie den Anweisungen folgen. Z.B. muss ein Weg-Vorschlag des LS nicht angenommen werden. Der Fußgänger hat die Möglichkeit, Anfragen an das LS zu stellen, die je nach Dringlichkeit unverzüglich oder im Laufe der Zeit vom LS bearbeitet werden müssen.

- Andere Verkehrsteilnehmer (Autos, Busse, Züge,...): Um die Fußgänger optimal leiten zu können, müssen sich alle Verkehrsteilnehmer an dieser Aufgabe beteiligen. So müssen sich die öffentlichen Verkehrsmittel „absprechen", welche Beförderungsalternativen den Fußgängern angeboten werden, abhängig von der jeweiligen Verkehrssituation und Auslastung. Gegebenenfalls müssen Plätze freigehalten werden (z.B. für schwangere Frauen). Autos müssen zum Stoppen gezwungen werden können, wenn vor ihnen ein (sehbehinderter) Fußgänger unerwartet die Straße betritt, vorausgesetzt, dass die umgebende Verkehrslage dieses zulässt.

- Positionsbestimmung: Die Positionsbestimmung ist dafür zuständig, die aktuelle Position aller Verkehrsteilnehmer zu ermitteln und anderen Komponenten diese Informationen zur

Verfügung zu stellen. Des Weiteren müssen Positionsvorhersagen berechnet werden. Je nach (Gefahren-)Situation sind die zeitlichen Intervalle der Positionsdatenupdates geeignet zu wählen. Ebenfalls situationsbedingt ist der erforderliche Grad an Exaktheit der Positionsdaten. Die Positionsbestimmung ist stark abhängig von den technischen Gegebenheiten, z.B. GPS-Systemen, optische Messverfahren.

- Verkehrskontrolle: Um einen reibungslosen Verkehrsfluss zu erreichen, muss das Verhalten der einzelnen Verkehrsteilnehmer koordiniert werden in Bezug auf Wegwahl und Geschwindigkeit. Dabei müssen die Ziele, aktuelle Positionen, Fahrpläne und spezielle Bedürfnisse (z.B. Behinderten gerechte Wege) berücksichtigt werden. Zu den Aufgaben der Verkehrskontrolle gehören das Setzen von Tempo-Limits und das Steuern von Ampelschaltungen. Die Verkehrskontrolle interagiert mit den Verkehrsteilnehmern und fordert diese zur Einhaltung der Verkehrsregeln auf. Des Weiteren meldet sie Staus und wertet Notrufe aus.

- Informationssystem: Damit Anfragen von Fußgängern an das Leitsystem beantwortet werden können, müssen die notwendigen Informationen beschafft werden. Dabei kann es sich um lokale Informationen (z.B. den Standpunkt der nächsten Telefonzelle) oder globale Informationen (z.B. die letzten Lottozahlen) handeln. Das Informationssystem beschäftigt sich mit der (verteilten) Informationsbeschaffung, Selektion, Aufbereitung und audiovisuellen Umsetzung. Dabei sind unterschiedliche Dringlichkeiten bei den Anfragen zu beachten.

- Alarmsystem: Eine wichtige Aufgabe ist die Erkennung von Gefahrensituationen und die Einleitung geeigneter Notfallmaßnahmen. Droht ein Fußgänger in Gefahr zu geraten (z.B. nähert er sich S-Bahnschienen, ohne langsamer zu werden), muss das Alarmsystem unverzüglich autonom entscheiden, ob es den Fußgänger durch Warnungen noch stoppen kann, oder ob es Einfluss auf die Umgebung des Fußgängers nehmen muss, um die Situation zu entschärfen (andere Personen zur Hilfeleistung auffordern, S-Bahn abbremsen). Fallen einzelne Komponenten des LS aus (z.B. durch technische Probleme oder Zerstörung), muss das Alarmsystem dafür Sorge tragen, dass die Ausfälle durch Zusammenwirken anderer Komponenten so gut wie möglich kompensiert werden.

- Technische Komponenten: Hierbei handelt es sich um reaktive oder partiell autonome Einheiten, die den Fußgängern spezielle Services anbieten. Beispiele sind Ticket-Automaten, die unter Beachtung des Zieles und der (durch andere Komponenten) ausgewählten Verkehrsmittel automatisch die richtigen Fahrscheine ausdrucken, oder Fahrstühle, die einen Fußgänger erwarten, oder Eingangstüren, die bei Annäherung aufgehen, sofern die Person gegebenenfalls authentifiziert und autorisiert ist.

- usw.

Schon die obige skizzenhafte informelle Beschreibung der Komponenten des LS verdeutlicht die große Komplexität des Gesamtsystems. Eine *formale Modellierung* im Ganzen ist praktisch nicht durchführbar. Trotzdem ist es wünschenswert, Systeme wie das des Beispiels formal zu erfassen, um diese dann anhand des formalen Modells (toolunterstützt) zu analysieren und Systemanforderungen zu verifizieren. Ein Lösungsansatz besteht darin, von Teilen des Systemverhaltens zu abstrahieren und den Systementwurf *vom Standpunkt der Kontrolle* aus zu betrachten. Anhand des Beispiels lassen sich dann folgende *Schlüsselphänomene* festhalten:

- ▸ Die Kontrolle ist *dezentral*. Eine zentrale Einheit (z.B. Laptop des Fußgängers) könnte die Masse an anfallenden, in vielen Fällen dynamischen Daten (Ziele, Positionen, Fahrpläne, Geschwindigkeiten, Verkehrsdichten, allgemeine angefragte Informationen, Geräteüberwachungsdaten,...) unter Beachtung aller lokalen Anforderungen (zum Teil zeitkritisch) gar nicht termingerecht verarbeiten.

- ▸ Es gibt *autonome* und *reaktive* Komponenten. Reaktive Komponenten zeichnen sich dadurch aus, dass Ereignisse in ihnen ausschließlich Auswirkungen von externen Einflüssen anderer (autonomer oder reaktiver) Komponenten sind (z.B. das Stoppen und Starten eines GPS-Empfängers erfolgt durch die Positionsbestimmung). Dahingegen können autonome Komponenten auch eigenständig Aktionen anstoßen, oder aber auch nicht. Sie repräsentieren einen positiven Grad an Entscheidungsfreiheit (z.B. kann ein Fußgänger einer Weg-Empfehlung folgen oder nicht).

- Die Komponenten unterliegen *gegenseitigen Einflüssen*. Um kooperativ ein sicheres Geleiten eines Fußgängers durch den Straßenverkehr zu erreichen, muss gewährleistet sein, dass einzelne Komponenten des LS in anderen Komponenten (insbesondere bei reaktiven Komponenten) Aktionen anstoßen/erzwingen können. So bewirkt z.B. eine Anfrage eines Fußgängers an das Informationssystem, dass dort ein Suchprozess gestartet wird. Die Einfluss-Beziehungen zwischen den Komponenten sind in vielen Fällen *nicht symmetrisch*. Z.B. übt eine rote Ampel einen Einfluss auf einen Fußgänger aus, anzuhalten, wo hingegen der Fußgänger keinen Zwang auf die rote Ampel zum Wechsel des Rotlichtes ausüben können darf. Somit gibt es neben der Forderung von Einflüssen/Zwängen auch die explizite Forderung nach deren Nicht-Existenz.

- Es treten *Fortpflanzungen von Wirkungen* von Einflüssen über mehrere Komponenten hinweg auf. Das Anstoßen einer Aktion in einer Nachbarkomponente kann oder wird dazu führen, dass diese Komponente ebenfalls eine Aktion in einer ihrer Nachbarkomponenten anstößt, usw. Z.B.: Aufgrund eines Unfalls stoppt das Alarmsystem die S-Bahn. Diese interagiert mit der Verkehrskontrolle und bewirkt die Schaltung einer Grün-Ampelphase für die Feuerwehr. Die Verkehrskontrolle informiert das Laptop eines Fußgängers, um die Berechnung eines neuen Weges, der den Unfallort umgeht, zu veranlassen.

- Ereignisse (als Kontrollzustandswechsel) in den Komponenten lassen sich in *erzwungene* und *freie Ereignisse* unterteilen. Die Ursache eines Zwanges, die zum Eintreten eines erzwungenen Ereignisses führt, kann zum einen in externen (propagierten) Einflüssen liegen, so wie sie in den zwei vorherigen Punkten beschrieben worden sind, oder aber in intern vorgegebenen Ablaufstrukturen. Der Übergang in einen Backup-Zustand ist ein Beispiel für eine intern erzwungene Aktion beim Informationssystem. Das Informationssystem führt im Eigeninteresse diese Aktion regelmäßig aus, ohne Nachdruck von außen. Erzwungene Ereignisse sind Ereignisse, die eintreten *werden*, sofern dies nicht durch andere externe Einflüsse verhindert wird. Demgegenüber stehen freie Ereignisse, die nur in autonomen Komponenten auftreten können und aus Kontrollentscheidungen zur Durchführung von Zustandswechseln resultieren. Freie Ereignisse *können* eintreten (oder auch nicht).

- Bei der Beschreibung bzw. der formalen Modellierung des Systems wird man konfrontiert mit *unvollständigen Informationen* über das endgültige Systemverhalten. Die Unvollständigkeit (hier des LS) ergibt sich zum einem aus der bereits aufgezeigten Komplexität des Gesamtsystems. Es sind eine Vielzahl von Funktionalitäten zu integrieren, deren Abdeckung der Gesamtaufgabe (das sichere Leiten eines Fußgängers) von vornherein nicht offensichtlich ist. Im Zuge der Entwicklung des Systems können sich Notwendigkeiten der Einbringung zusätzlicher oder der Erweiterung bestehender Komponenten ergeben (z.B. die Hinzunahme einer lokalen Wettervorhersage). Die Unvollständigkeit basiert des Weiteren auf einer Ungewissheit über die genauen Wechselwirkungen zwischen den einzelnen Komponenten (Gibt es z.B. Verbindungen zwischen dem Informationssystem und der Positionsbestimmung?). Aufgrund der verteilten Kontrolle können durch fortgepflanzte Wirkungen (siehe Punkt 4) unvorhersehbare, im negativen Fall unerwünschte Propagierungseffekte auftreten. Solche unerwünschten Effekte müssen durch lokale System-Anpassungen beseitigt werden. Unvollständigkeit lässt sich zudem interpretieren als Offenheit gegenüber zusätzlichen Anforderungen resultierend aus praktischen Tests oder der Einbeziehung neuester technologischer Entwicklungen.

Die Theorie der I–Systeme wurde entwickelt, um genannte Schlüsselphänomene in elementare Konzepte und Eigenschaften eines formalen Modells abzubilden. Es hat sich gezeigt, dass traditionelle formale Modelle wie z.B. Petri-Netze oder Statecharts diesbezüglich bisher keine adäquaten Lösungen anbieten.
Teil der Modellierungsmethodologie der I–Systeme – das „I“ hat seinen Ursprung in dem Wort Interaktion – ist es, von der Komplexität der verteilten Kontrollstruktur aus zu starten und zu versuchen, Kontrollaspekte lokal zu modellieren unter Einbeziehung der Umgebung. Dabei erfolgt die Behandlung lokaler Anforderungen und Bedürfnisse ausschließlich auf Basis *zweier elementarer bilateraler Interaktionsbeziehungen* zwischen Komponenten zur expliziten (graphischen) Darstellung wechselseitiger lokaler Einflüsse. Dieser Ansatz wirkt unvollständig, es stellt sich jedoch heraus, dass er als Modellierungswerkzeug sehr mächtig und überschaubar ist. Insbesondere

werden die oben skizzierten Schlüsselphänomene abgedeckt, z.B.: es wird zwischen autonomen und reaktiven Komponenten (so genannte autonome und träge Bereiche) unterschieden; es lassen sich Einflüsse zwischen Bereichen innerhalb einer möglicherweise asymmetrischen Einflussstruktur explizit darstellen, ebenso wie die Nicht-Existenz von Einflüssen; Einflüsse erwirken erzwungene Ereignisse (so genannte erzwungene Phasentransitionen); in autonomen Komponenten können zudem freie Phasentransitionen auftreten; Fortpflanzungen von Wirkungen werden nachvollziehbar. Gerade in letzterem Punkt versagen in der Regel traditionelle formale Ansätze, da dort Einflüsse, deren Ursachen und (propagierte) Wirkungen unüberschaubar sind.
Die Theorie der I–Systeme verfolgt das Prinzip des *inkrementellen restriktiven Modellierens*. Bei einem realen verteilten System alle erlaubten (nebenläufigen) Abläufe zu beschreiben ist zu aufwendig. Besser ist es, alles zu erlauben und nur die Situationen auszuschließen, die die Systemsicherheit gefährden oder der Erfüllung der Anwendungsaufgabe entgegenstehen. Z.B. wäre es undenkbar, einem Verkehrsteilnehmer das Verhalten in jeder möglichen Verkehrssituation vorzuschreiben. Dahingegen ist die Restriktion „Kein Überqueren einer roten Ampel“ eine Vorgabe, die zur Aufrechterhaltung der Verkehrssicherheit notwendig ist und die Handlungsfreiheit des Verkehrsteilnehmers gezielt einschränkt. Man braucht aus der sehr großen Menge von Systemsituationen in der Regel nur vergleichsweise wenige für solche Korrekturüberlegungen zu berücksichtigen, und man kann Maßnahmen dieser Art schrittweise verschärfen, ohne dass frühere Maßnahmen wieder rückgängig gemacht werden müssen. Bei I–Systemen werden Restriktionen durch elementare Interaktionsbeziehungen realisiert, d.h. zusätzliche Beschränkungen des Systemverhaltens ergeben sich durch Wirkungen neu installierter lokaler Einflüsse. Für eine inkrementelle Entwurfsstrategie ist es wichtig, dass durch die Hinzunahme von Einflüssen die bisherige Spezifikation unverändert bleibt, in dem Sinne, dass ursprüngliche Einflüsse erhalten bleiben. Auch an dieser Stelle versagen die traditionellen Ansätze aufgrund der Unüberschaubarkeit der vorhandenen, in der Regel aber nicht explizit dargestellten Einflüsse. Eine inkrementelle Entwurfsstrategie ist bei I–Systemen das Mittel für die Arbeit mit unvollständigen Informationen zur Design-Zeit (siehe letztes oben skizzierte Schlüsselphänomen). Auf neue unvorhergesehene Anforderungen kann schrittweise reagiert werden, ohne einen kompletten Neuentwurf bzw. eine umfassende Systemanalyse starten zu müssen. Dabei ist es möglich, propagierte Einflüsse (vgl. drittes Schlüsselphänomen) zurückzuverfolgen, um gegebenenfalls Korrekturen an den Ursachen vorzunehmen.
Parallel zur inkrementellen Herangehensweise erlauben I–Systeme einen *modularen Entwurf* und eine *modulare Analyse*. Die Komplexität der Anwendungssysteme (siehe z.B. LS) erzwingt solche Techniken, bei denen es darum geht, die Arbeitsweise von Teilsystemen und deren Kooperation mit dem Gesamtsystem zu verstehen.

Obwohl das Entwurfskonzept der I–Systeme und dessen Mächtigkeit (gemäß obigen Ausführungen) im Wesentlichen klar sind, liegt in der Definition geeigneter *Semantiken*, die als Grundlage für eine umfassende Systemanalyse oder Systemverifikation dienen, ein offenes nicht-triviales Problem. Es stellt sich die Frage nach der Darstellungsform einer Semantik, die sich aus den lokalen Wechselwirkungen ergibt. Je nach Analyseabsicht sollten ein Teil oder alle der ersten vier genannten Schlüsselphänomene über die Semantik nachvollziehbar sein. Dazu gehört insbesondere die Dokumentierung, wie Zwänge aus Einflüssen entstehen und warum sie entstehen. Da zu erwarten ist, dass ein Mehr an Ausdruckskraft einer Semantik einen Anstieg der Darstellungskomplexität (d.h. des Speicherplatzbedarfs einer minimalen Datenstruktur für die Semantik) mit sich bringt, sollten alternative Semantiken unterschiedlicher Ausdruckskraft und Darstellungskomplexität, die je nach Analyse-/Verifikationsziel eingesetzt werden können, angeboten werden. Die Eignung der Semantiken für eine inkrementelle modulare Entwurf- und Analysestrategie ist dabei eine notwendige Voraussetzung. Des Weiteren sollten sich so genannte *I–System-Transformationen* auf sie beziehen können. Ein Ziel von I–System-Transformationen, und als Spezialfall von I–System-Reduktionen, ist es, I–Systeme derart umzuformen, dass sich eine Systemanalyse bzw. Systemverifikation effizienter durchführen lässt. Durch die Umformungen dürfen sich relevante Systemeigenschaften natürlich nicht ändern. Formal bedeutet das, dass eine Äquivalenz von I–Systemen vom Standpunkt der Semantik aus festgelegt werden muss und Transformationen nur innerhalb semantisch äquivalenter I–Systeme erfolgen dürfen. Diese Arbeit beschäftigt sich mit der Lösung des Semantik-Problems.

Ein Ergebnis der bisherigen Arbeit mit I–Systemen ist die Erkenntnis, dass der formalen Modellierung eine Strukturierung in folgende *Modellierungsebenen* unterliegt.

- *Anwendungsebene*: wird repräsentiert durch ein reales verteiltes System, das formal modelliert werden soll.
- *Formale Ebene*: umfasst mathematische Formalismen wie z.B. Syntax und Semantiken der I–Systeme, sowie Systemanforderungsspezifikationen.
- *Algorithmische Ebene*: spezifiziert die Systemdynamik, beschreibt Kommunikationsabläufe und bietet Implementierungsansätze.
- *Anschauungsebene*: dient der Interpretation der Systemdynamik im Kontext der Einfluss/Wirkungen-Beziehungen.

Wenn es gelingt, diese Strukturierung herauszuarbeiten und als Bestandteil des formalen Modells zu etablieren, lassen sich Vorteile erwarten im Hinblick auf Design- und Analysesystematisierung sowie Modell-Handhabbarkeit/-Adaptierbarkeit/-Erweiterbarkeit. Vergleichbare Strukturierungskonzepte haben sich bereits bei anderen Entwurfsdisziplinen (z.B. in der Softwaretechnologie) bewährt.

1.2 Ziele

Das Ziel dieser Arbeit ist es, die Theorie der I–Systeme weiterzuentwickeln, so dass ein formaler Systementwurf ermöglicht wird in Anlehnung an die Motivation. Um dies zu erreichen, sollen folgende *drei Schwerpunkte* bearbeitet werden:

- Realisierung der Modellierungsebenen;
- Lösung des Semantik-Problems;
- Durchführung von modularem Entwurf/Analyse und I–System-Transformationen.

Der erste Schwerpunkt beinhaltet eine Überarbeitung der bisherigen formalen Strukturen der I–Systeme. Zum Zweck der Allgemeingültigkeit soll dabei von einer konkreten Anwendung abstrahiert und sollen insbesondere die Inhalte der formalen Ebene, der algorithmischen Ebene und der Anschauungsebene sowie Schnittstellen zwischen den Ebenen spezifiziert werden.

Der zweite Schwerpunkt zielt auf die Lösung des bereits in der Motivation benannten offenen Problems der Definition angemessener Semantiken für I–Systeme. Hierbei erfolgt die Konzentration auf zwei anerkannte, bewährte Darstellungsformen zur Beschreibung von Systemaktivitäten:

1. *Trace-Semantiken*, als Mengen von Folgen von Systemzuständen, wobei jede Folge eine mögliche Systemausführung beschreibt;
2. *Zustandsgraphen*, bei denen die Knoten Systemzustände und die Kanten Systemzustandsübergänge repräsentieren.

Mit der Auswahl dieser Darstellungsformen ist die Erwartung verbunden, von der Vielzahl existierender Methoden und Tools, die auf Trace-Semantiken oder Zustandsgraphen aufsetzen, profitieren zu können. Es ist das Ziel, neuartige Trace-Semantik- und Zustandsgraph-Varianten für I–Systeme zu definieren, unter Beachtung der Modellierungsebenen. Die einzelnen Varianten unterscheiden sich in ihrer Ausdruckskraft (bezüglich der Dokumentation der Schlüsselphänomene aus der Motivation) und in ihrer Darstellungskomplexität. Charakteristika der Trace-Semantiken und Zustandsgraphen sollen erarbeitet, Unterschiede und Abhängigkeiten aufgezeigt werden.

Beim dritten Schwerpunkt geht es um die Präsentation von Methoden für einen inkrementellen modularen Systementwurf, für eine modulare Systemanalyse und für I–System-Transformationen, jeweils unter Einsatz der eingeführten Semantiken. Speziell gehören zu diesem Punkt die Eingrenzung von Entwurfs- und Analyseeinheiten (so genannte Strukturbausteine), die Beschreibung von Verifikationtechniken (Eigenschaft-/Anforderungsspezifikation und Beweisstrategien) und die Definition von I–System-Äquivalenzen.

Es ist zu erwarten, dass sich aus der erfolgreichen Bearbeitung der genannten Schwerpunkte präzise Ansatzpunkte für aufbauende Arbeiten ergeben.

1.3 Bisherige Arbeiten

Den Ausgangspunkt der Entwicklung von I–Systemen bilden die so genannten Lose Gekoppelten Systeme (Loosely Coupled Systems, LCS). Bei ihnen handelt es sich um einen Formalismus, der neben einer formalen auch eine graphische Repräsentation der Einschränkungen liefert, denen Prozesse in verteilten Systemen durch wechselseitigen Ausschluss einiger Zustände der Systemkomponenten unterliegen. Eine ausführliche Darstellung der Theorie der LCS findet sich in [16, 65]. Im Gegensatz zu I–Systemen sind bei LCS noch keine Konzepte vorgesehen, die es erlauben, Einflüsse zu modellieren, die explizit von einer Komponente auf eine andere ausgeübt werden. Eine Differenzierung bei den Ereignissen in z.B. „erzwungene“ und „freie“ Phasentransitionen (vgl. Abschnitt 1.1) ist nicht möglich. LCS beschreiben ausschließlich symmetrische Ereignisstrukturen. Um die Möglichkeiten der LCS dahingehend auszubauen, dass Einflüsse/Zwänge und deren Wirkungen ebenfalls dargestellt werden können, wurden so genannte Interaktionssysteme als formales Modell eingeführt und diskutiert. Das Modellierungskonzept basiert im Wesentlichen auf der Verwendung zweier elementarer bilateraler Interaktionsbeziehungen (Kopplungs- und Erregungsrelation) zwischen Zuständen (Phasen) benachbarter Komponenten (Bereichen) eines verteilten Systems zur Beschreibung von wechselseitigem Ausschluss und Einflüssen/Zwängen. Mehrere Arbeiten beschäftigen sich mit der Motivation, Anwendungsbeispielen, formalen Ansätzen [13, 14, 56, 57, 84, 85, 90]. Im Gegensatz zu LCS lassen sich mit Interaktionssystemen auch gerichtete Ereignisstrukturen modellieren. Der Übergang von LCS zu Interaktionssystemen brachte eine neue Spezifikation der Dynamik mittels axiomatischer Verhaltensbeschreibungen mit sich. Die Dynamik wurde im Laufe der Entwicklung immer wieder angepasst.
Auf den Modellierungskonzepten der Interaktionssysteme bauen die I–Systeme mit dem Ziel auf, die Theorie weiter zu entwickeln. In diesem Sinne wurden erste Trace-Semantiken eingeführt und verstärkt Methoden zum modularen inkrementellen Design und zur modularen Analyse betrachtet [89, 91]. In dieser Arbeit erfolgt die formale Ausarbeitung der in den bisherigen Arbeiten vorgestellten formalen Ansätze unter Einbeziehung neuer Konzepte (z.B. Modellierungsebenen).

Andere bekannte formale Modelle zur Beschreibung nebenläufiger Abläufe, die sich durch eine eingängige graphische Repräsentation auszeichnen, sind Petri-Netze [72] und Statecharts [39]. Beide Modelle verfolgen eine monitormäßige, hierarchische statt inkrementelle Designstrategie. Die Ablaufsteuerung erfolgt durch Festlegung aller möglichen Ereignisabfolgen und nicht durch Verhaltensbeschränkung wie bei I–Systemen. Beide Modelle sind nicht dafür entwickelt worden, wechselseitige Einflüsse in einem System interagierender Komponenten explizit darzustellen und bieten keine darauf ausgelegten Modellierungsmittel an. Somit sind Petri-Netze und Statecharts nur eingeschränkt zur Modellierung von Szenarien geeignet, wie sie in der Motivation beschrieben sind. Notwendig werden Erweiterungen (Zusatzbeschriftungen, zusätzliche Komponenten) der ursprünglichen Modelldefinitionen, die sich natürlich nicht nachteilig auf die Handhabbarkeit des jeweiligen Modells auswirken sollten.
Eine solche Erweiterung bei Petri-Netzen präsentiert Reisig in [75] und liefert damit den einzigen bekannten vergleichbaren Ansatz zu I–Systemen. Reisig betrachtet so genannte System Nets, bei denen die Transitionsmenge in „progressive“ und „quiescent“ Transitionen aufgeteilt ist. In der Terminologie der I–Systeme entsprechen progressive Transitionen erzwungenen und quiescent Transitionen freien Phasentransitionen. Über die Ursachen des Schaltens bzw. Nicht-Schaltens dieser beiden Arten von Transitionen lassen sich folgende Aussagen machen: Das Schalten einer quiescent Transition ist abhängig von Informationen, die am Modell nicht ablesbar sind. Die Ursache dafür, dass eine quiescent Transition nicht schaltet (z.B. weil eine Vorbedingung aufgrund

einer zu restriktiven Synchronisationsbedingung nie erfüllt ist) kann ebenfalls am Modell nicht zurückverfolgt werden. Die Ursache eines zum Schalten einer progressive Transition führenden notwendigen Zwanges kann nur mittels entsprechender Stellen- oder Transitionsbeschriftungen deutlich gemacht werden. Hier zeigt sich der Unterschied zu I–Systemen, bei denen die Klarstellung der Ursache von Zwängen zur Darstellung gehört und bei denen erkennbar gemacht wird, wo keine Einflüsse vorliegen. Die Unterscheidung der Transitionen wurde bei Reisig nicht in die Darstellungsformen der Semantiken (Interleaved Runs, Concurrent Runs) aufgenommen. Für das Anliegen von Reisig ist dies allerdings auch nicht notwendig. Fortschritt an sich soll dokumentiert werden, nicht Ursachen dafür.

In der Motivation wurde deutlich gemacht, dass sich I–Systeme im Gegensatz zu traditionellen formalen Modellen zur Modellierung autonomer dezentraler Systeme anbieten. Die besondere Beachtung solcher Systeme spiegelt sich in einer seit 1996 stattfindenden Konferenzserie wider [47]. In dem Beitrag [7] werden der Begriff der Autonomie und Abstufungen davon diskutiert. Abstufungen in der Autonomie realisiert man bei I–Systemen einerseits durch die Unterscheidung zwischen reaktiven und autonomen Komponenten (träge und autonome Bereiche) und andererseits durch die Auferlegung unterschiedlich strenger Verhaltensrestriktionen mittels der beiden bilateralen Interaktionsbeziehungen.

Es wurde bereits darauf hingewiesen, dass Strukturierungskonzepte vergleichbar den Modellierungsebenen bei I–Systemen ein etabliertes Prinzip z.B. im Bereich der Softwaretechnologie sind. Dort dient die Unterscheidung von Designebenen der Strukturierung von Softwareentwicklungsprozessen. Details hierzu finden sich in [74].

Weitere Literaturquellen zu den in dieser Arbeit behandelten Themengebieten (z.B. Semantik, Modularisierung, Transformationen) sind themenbezogen in den einzelnen Kapiteln angegeben.

1.4 Gliederung der Arbeit

In *Kapitel 2* wird die Syntax der I–Systeme und deren graphische Darstellung vorgestellt. Es werden grundlegende Begriffe wie z.B. Nachbarschaften eingeführt.

Kapitel 3 behandelt die Dynamik der I–Systeme. Es werden zwei Typen von Systemzuständen unterschiedlicher Ausdruckskraft auf der formalen Ebene definiert und axiomatisch Zustandsübergänge auf algorithmischer Ebene spezifiziert. Dazu wird jedem I–System ein so genanntes $\mathrm{V_I}$System zugeordnet, das vorgegebene verteilte Algorithmen, so genannte Aktionen, abarbeitet. Die relevanten Systemzustände und Übergänge lassen sich aus den Ausführungen des $\mathrm{V_I}$Systems ableiten. Zur Veranschaulichung der programmiersprachlichen Notation der Aktionen werden Interpretationen der einzelnen Instruktionen angegeben, die sich auf wechselseitige Einflüsse zwischen interagierenden Komponenten und auf die Auswirkungen der Einflüsse beziehen.

Unter Rückgriff auf die Dynamik der I–Systeme werden in *Kapitel 4* drei Varianten von Trace-Semantiken für I–Systeme eingeführt, die sich in Ausdruckskraft und Darstellungskomplexität voneinander unterscheiden, abhängig davon welche Informationen (z.B. Systemzustandstyp) aus dem dynamischen Verhalten herausgelesen und in die Semantik aufgenommen werden. Für jede Semantik wird ein zentraler Charakterisierungs-Satz formuliert und bewiesen, der die speziellen (elementaren) Eigenschaften der I–Systeme bezogen auf die jeweilige Semantik herausstellt. Abhängigkeiten zwischen den drei Semantiken werden deutlich gemacht. Ein besonderes Kennzeichen der Trace-Semantiken für I–Systeme ist deren Nicht-Präfixabgeschlossenheit, die in diesem Kapitel gezeigt wird. Über die Semantiken lassen sich auf der formalen Ebenen Systemereignisse definieren, wie z.B. freie und erzwungene Phasentransitionen.

In *Kapitel 5* erfolgt eine Fokussierung auf den Interleaving Anteil der in Kapitel 3 vorgestellten Semantiken. Nebenläufige Ereignisse in unterschiedlichen Bereichen eines I–Systems bleiben unberücksichtigt. Untersucht wird das Informationspotential der so genannten Interleaving-Trace-

Semantiken im Vergleich mit den normalen Trace-Semantiken.

Mit Zustandsgraphen befasst sich *Kapitel 6*. Diese endlichen gerichteten Graphen sind neben Trace-Semantiken die zweite Darstellungsform zur Beschreibung von Systemaktivitäten, die in dieser Arbeit behandelt werden. In diesem Kapitel werden drei Typen von Zustandsgraphen definiert und deren Ausdruckskraft wird mit der von Trace-Semantiken verglichen.

Kapitel 7 beschreibt die Modellierungsmethodik bei I–Systemen und bezieht sich dabei auf die Konzepte/Ergebnisse aus den Kapiteln 2 bis 6. Hierzu gehören die Präsentation der Modellierungsebenen, deren Inhalte und Schnittstellen. Des Weiteren erfolgt eine Zusammenstellung der eingeführten Semantiken und Zustandsgraphen sowie aller Abhängigkeiten.

Lose Gekoppelte Systeme sind syntaktisch betrachtet ein Spezialfall der I–Systeme. In *Kapitel 8* wird untersucht, ob diese Einbettung auch semantisch durchgängig ist.

Der Entwurf verteilter Systeme (ohne zentrale Kontrolle) orientiert sich an lokalen Interessen und Anforderungen. Hierbei lassen sich die Bereiche eines I–Systems in so genannte Relevanz- und Kontrollbereiche unterteilen, je nach Aufgabe im Rahmen der Systementwicklung. Details hierzu finden sich in *Kapitel 9*. Um formal den Blick auf das Systemverhalten innerhalb einer lokalen Umgebung vollziehen zu können, werden in diesem Kapitel Projektionen der bisher definierten Semantiken und Graphen auf ausgezeichnete Teilsysteme eingeführt. Als umfassendes Beispiel wird die Modellierung lokaler Ereignisstrukturen in einem Bereich eines I–Systems behandelt.

Kapitel 10 beschäftigt sich mit der Modularität bei Entwurf und Analyse von I–Systemen. Als Modellierungsmodule werden so genannte Strukturbausteine verwendet. Bei ihnen handelt es sich um (parametrisierte) I–System-Teilstrukturen mit bewiesenen Interaktionseigenschaften. Je nach Beweisstrategie wird zwischen elementaren und abgeleiteten Struktureigenschaften unterschieden. Exemplarisch werden einige Beispiele für Strukturbausteine präsentiert. Als Beispiel für die Kombination von Strukturbausteinen und eine modulare Systemverifikation wird das Modell eines synchronen Kommunikationsmechanismus betrachtet. Anhand eines weiteren Beispiels wird demonstriert, wie sich mit Hilfe von Strukturbausteinen ein I–System inkrementell entwerfen lässt. Hierzu wird ein nicht-triviales Synchronisationsproblem zwischen (Betriebssystem-)Prozessen durch das schrittweise Einbringen von restringierenden Interaktionsbeziehungen gelöst. Die Korrektheit der Gesamtkonstruktion ergibt sich einfach aus der Vereinigung der Eigenschaften der verwendeten Strukturbausteine.

In *Kapitel 11* wird auf I–System-Transformationen (siehe Ende Abschnitt 1.1) eingegangen. Es werden verschiedene Transformationsziele aufgelistet (z.B. Minimierung der Anzahl der Kontrollbereiche) und die Vorteile für eine Systemanalyse oder modellnahe Implementierung beschrieben. Um die Korrektheit von Regelschemata für I–System-Transformationen zeigen zu können, muss festgelegt sein, wann I–Systeme als gleichwertig in ihrem Systemverhalten (gegebenenfalls bei Blick auf ein Teilsystem) anzusehen sind. Dieses Kapitel liefert formale Definitionen für Äquivalenzen auf der Menge der I–Systeme. Anhand eines längeren Beispiels wird die Zweckmäßigkeit von I–System-Transformationen verdeutlicht.

Kapitel 12 fasst die Ergebnisse dieser Arbeit zusammen und liefert einen Ausblick auf weiterführende Arbeiten.

Zur besseren Lesbarkeit der Arbeit sind lange Beweise von Sätzen in *Anhang A* aufgeführt.

Kapitel 2

I–Systeme

Als Mittel zur formalen Modellierung von verteilten Systemen, so wie sie in der Einleitung beschrieben sind, werden in dieser Arbeit die so genannten I–Systeme betrachtet. Sie dienen zur Darstellung der Systemtopologie, zur Beschreibung des Verhaltens in sowie der Interaktion zwischen einzelnen Systemkomponenten. Im ersten Teil dieses Kapitels wird die Syntax der I–Systeme vorgestellt und es werden die Bezüge zur Anwendungsebene verdeutlicht.
Ein wichtiges Merkmal von I–Systemen ist deren anschauliche und leicht verständliche graphische Darstellungsform. Hierauf wird im zweiten Abschnitt eingegangen.
In der Einleitung wurde betont, dass Modularität ein entscheidender Aspekt bei der Arbeit mit verteilten Systemen ist (siehe auch [2, 15, 20]). Das Verhalten des Gesamtsystems, das in der Regel so komplex ist, dass es in seiner Gesamtheit nicht erfasst werden kann, ergibt sich aus dem Verhalten und der Interaktion formal beherrschbarer Teilsysteme, die aus mehreren benachbarten Komponenten bestehen können. Das Verhalten einer Komponente ist dabei sowohl abhängig von eigenen lokalen Gegebenheiten (z.B. interne Zwänge, Entscheidungsspielräume) wie auch von Einflüssen, die von Nachbarkomponenten ausgeübt werden. Die Nachbarschaft ergibt sich dabei aus der Möglichkeit zur wechselseitigen Interaktion. Der letzte Abschnitt liefert erste Definitionen.

2.1 Formale Struktur

Ein I–System wird durch ein 5-Tupel beschrieben, dessen Komponenten bestimmte Elemente eines verteilten Systems repräsentieren. So werden die Systemkomponenten/Knoten des verteilten Systems durch so genannte *Bereiche* modelliert. Jeder Bereich zeichnet sich aus durch eine endliche Anzahl von für die Modellierung des Systemverhaltens relevanter lokaler Zustände, die *Phasen* genannt werden. Das einzige, was über die Bereiche angenommen wird, ist, dass sie sich zu jeder Zeit in genau einer Phase befinden. In diesem Sinne kann das Verhalten in jedem Bereich als endlicher Automat (vgl. [77]) aufgefasst werden. Um die Interaktion zwischen Systemkomponenten geeignet darstellen zu können, werden zwei Relationen zwischen Phasen eingeführt. Die *Kopplungsrelation* spezifiziert eine Menge von Paaren von wechselseitig ausgeschlossenen Phasen. Zwei Bereiche können sich nicht zum gleichen Zeitpunkt (vom Standpunkt der Systemumgebung) in Phasen befinden, die in der Kopplungsrelation stehen. Die *Erregungsrelation* vermittelt Einflüsse, die eine Systemkomponente auf eine andere Systemkomponente ausübt oder ausüben kann, um letztere dazu zu bewegen, eine bestimmte Phase zu verlassen. Befinden sich zwei Bereiche zum gleichen Zeitpunkt in Phasen, die in der Erregungsrelation stehen, so erregt die erste Phase die zweite. Die genauen Auswirkungen der beiden Relationen auf die Dynamik eines I–Systems, speziell die Formulierung der notwendigen Interaktionsmechanismen zur Aufrechterhaltung der durch die Interpretation der Relationen gegebenen Verhaltenseinschränkungen (z.B. wechselseitiger Ausschluss von Phasen), werden in Kapitel 3 behandelt.
Bei I–Systemen unterscheidet man zwischen *trägen* und *autonomen* Bereichen. Diese Unterscheidung resultiert aus der Beobachtung, dass es in einem verteilten System Komponenten geben kann, deren lokale Aktivitäten ausschließlich durch äußere Einflüsse, d.h. durch Interaktionen mit anderen Systemkomponenten, angestoßen werden (siehe Kapitel 1.1). Reaktive Systeme

und reaktive Software-Komponenten sind spezielle Beispiele dafür [2, 17, 63]. Solche reaktiven Komponenten werden durch träge Bereiche modelliert. Den Gegensatz zu den reaktiven bilden die autonomen Systemkomponenten eines verteilten Systems, bei denen lokale Aktionen von der Umgebung betrachtet auch selbsttätig auftreten können, ohne extern angestoßen worden zu sein. Die entsprechenden Bereiche bei einem I–System werden als *autonom* klassifiziert. Der Einfluss von Trägheit und Autonomie auf die Spezifikation der Dynamik eines I–Systems wird in Kapitel 3 aufgezeigt. Zusammenfassend ergibt sich folgende Struktur.

Definition 2.1 (I–System). Ein 5-Tupel $IS = (P, B, \underline{B}, K, E)$ heißt *I–System*, wenn gilt:

(1) P ist eine endliche Menge von *Phasen*.

(2) B ist eine Menge von *Bereichen*, wobei B eine Partition von P ist, d.h. es gelten:

  a) $\forall b \in B : b \subseteq P$,

  b) $\bigcup_{b \in B} b = P$,

  c) $\forall b_1, b_2 \in B, b_1 \neq b_2 : b_1 \cap b_2 = \emptyset$.

(3) $\underline{B} \subseteq B$ ist eine ausgezeichnete Menge von *trägen Bereichen*.

(4) $K \subseteq P \times P$ ist die *Kopplungsrelation von* IS mit:

  a) K ist symmetrisch,

  b) $\forall b \in B, \forall p_1, p_2 \in b, p_1 \neq p_2 : (p_1, p_2) \in K$.

(5) $E \subseteq P \times P$ ist die *Erregungsrelation von* IS mit:

  a) $E \cap (E^{-1} \cup K) = \emptyset$.

Für $p \in P$ bezeichnet $b(p)$ den Bereich von p. Es gilt: $b(p) = b'$ gdw. $p \in b' \in B$.
$AB(IS) := B \setminus \underline{B}$ ist die *Menge der autonomen Bereiche*.
Mit $ISystem$ wird die Menge aller I–Systeme bezeichnet. □

2.2 Graphische Darstellung

Eine Eigenschaft der I–Systeme ist die Möglichkeit deren graphischer Darstellung. Gerade im Zusammenhang mit verteilten Systemen und Nebenläufigkeit hat eine graphische Darstellung Vorteile gegenüber einer rein textuellen.
„Die Vorstellung von „Verteiltheit“ und „Nebenläufigkeit“ ist wohl immer räumlich*, selbst dann, wenn potentiell nebenläufige Prozesse in einigen Anwendungen im Endeffekt auf nur einem Prozessor ausgeführt werden. Räumliche Vorstellungen lassen sich jedoch in Graphiken auf natürliche Weise auch räumlich darstellen, was zu einer kleinen „artikulatorischen Distanz“ zwischen dem Gedankenmodell des Benutzers und der Darstellung (auf dem Rechner) führt. Diese Distanz klein zu halten, ist erwünscht, weil dann der Benutzer keine großen gedanklichen Anstrengungen benötigt, um seine Vorstellungen umzusetzen, und sich voll auf das eigentliche, inhaltliche Problem konzentrieren kann.“* [29]
Insbesondere bei der Kommunikation mit Dritten (z.B. dem Auftraggeber oder späterem Benutzer eines zu erstellenden Software-Systems) sind rein textorientierte formale Darstellungen häufig nicht brauchbar, weil diese nicht oder nur schwer von allen Beteiligten verstanden werden. Der Vorteil graphischer Darstellungen hat sich bereits bei anderen formalen Modellen gezeigt, wie z.B. bei Statecharts [39] oder Petri-Netzen [72].

Für ein gegebenes I–System $IS = (P, B, \underline{B}, K, E)$ gibt es folgende graphische Repräsentation: Jede Phase $p \in P$ wird durch einen Kreis dargestellt, der mit dem Phasennamen bezeichnet wird. Die disjunkte Einteilung der Phasen in die Menge der Bereiche B wird durch rechteckige abgerundete Umrandungen verdeutlicht. Bereichsnamen können die Bereiche bezeichnen. Die trägen Bereiche aus $\underline{B}$ werden durch Schraffierung/Einfärbung der umrandeten Fläche oder/und Unterstreichung des Bereichsnamens hervorgehoben. Für jedes Element (p, p') der Erregungsrelation E wird ein Pfeil von dem Kreis der Phase p zu dem Kreis der Phase p' gezogen. Jedes Element (q, q') der

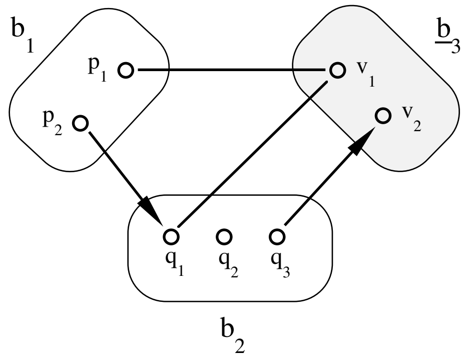


Abbildung 2.1: Graphische Darstellung des I–Systems IS_1

Kopplungsrelation K wird durch eine Linie, die die Kreise von q und q' verbindet, dargestellt. Die Kopplungsrelation innerhalb eines Bereiches wird nicht explizit aufgezeichnet.

Beispiel 2.2. Abbildung 2.1 zeigt das I–System $IS_1 := (\{p_1, p_2, q_1, q_2, q_3, v_1, v_2\}, \{\{p_1, p_2\}, \{q_1, q_2, q_3\}, \{v_1, v_2\}\}, \{\{v_1, v_2\}\}, \{(p_1, v_1), (v_1, p_1), (q_1, v_1), (v_1, q_1)\}, \{(p_2, q_1), (q_3, v_2)\})$. □

2.3 Lokale Umgebungen

Ein Kennzeichen verteilter Systeme ist die *verteilte Kontrolle* der Aktivitäten in den Systemkomponenten. Es gibt keinen ausgezeichneten Master-Knoten, der das Verhalten der einzelnen Komponenten koordiniert. Jede Komponente steuert sich selbst (z.B. durch einen Node Manager wie in [87]) und kennt dabei nur die Komponenten, mit denen sie direkt durch Kommunikationskanäle verbunden ist. Die Informationen über das restliche System müssen, sofern überhaupt erforderlich, mittels Interaktion mit der direkten Umgebung, d.h. den Nachbarkomponenten, ermittelt werden. Die Nachbarkomponenten stehen ihrerseits ausschließlich in Kommunikation mit ihren Nachbarkomponenten, usw. Anfragen/Anweisungen/Rückmeldungen einer Komponente an eine entfernte (nicht benachbarte) Komponente werden folglich über die dazwischenliegenden (im Sinne der Nachbarschaften) Komponenten propagiert. Der Begriff *Nachbarschaft* ist somit als ein zentraler Begriff bei der Modellierung verteilter Systeme anzusehen. Auf der Ebene der I–Systeme wird die Möglichkeit zur wechselseitigen Interaktion von Komponenten gleichgesetzt mit der Existenz von Kopplungs- oder Erregungskanten, die die entsprechenden Bereiche in der graphischen Darstellung verbinden. Die folgenden zwei Definitionen definieren Nachbarschaften von Phasen und Bereichen eines I–Systems auf der Basis der Kopplungs- und Erregungsrelation.

Definition 2.3 ($R(p)$, $R^{-1}(p)$, $R\langle b_1, b_2\rangle$). Sei $IS = (P, B, \underline{B}, K, E)$ ein I–System, $p \in P$, $b_1, b_2 \in B$. Sei $R \subseteq P \times P$ (z.B. $R = E$ oder $R = K$).

a) $R(p) := \{q \in P \mid (p, q) \in R\}$

b) $R^{-1}(p) := \{q \in P \mid (q, p) \in R\}$

c) $R\langle b_1, b_2\rangle := R \cap (b_1 \times b_2)$ □

Definition 2.4 (Nachbarbereich/-phase). Sei $IS = (P, B, \underline{B}, K, E)$ ein I–System und $b, b' \in B$ mit $b \neq b'$, $p \in b$, $m \subseteq b$, $v \in b'$.

a) v heißt *Nachbarphase von* p gdw. $v \in (K(p) \setminus b) \cup E(p) \cup E^{-1}(p)$.

b) v heißt *K-Nachbarphase von* p gdw. $v \in K(p) \setminus b$.

c) v heißt *E_{out}-Nachbarphase von* p gdw. $v \in E(p)$.

d) v heißt *E_{in}-Nachbarphase von* p gdw. $v \in E^{-1}(p)$.

e) v heißt *(K-, E_{out}-, E_{in}-) Nachbarphase von m* gdw. $\exists p' \in m : v$ ist eine (K-, E_{out}-, E_{in}-) Nachbarphase von p'.

f) b' heißt *(K-, E_{out}-, E_{in}-) Nachbarbereich von p* gdw. $\exists v' \in b' : v'$ ist eine (K-, E_{out}-, E_{in}-) Nachbarphase von p.

g) b' heißt *(K-, E_{out}-, E_{in}-) Nachbarbereich von m* gdw. $\exists v' \in b', \exists p' \in m : v'$ ist eine (K-, E_{out}-, E_{in}-) Nachbarphase von p'. □

Es sei bemerkt, dass der Fall $m = b$ gemäß obiger Definition zugelassen ist. In Beispiel 2.2 gilt somit zum Beispiel: b_2 ist Nachbarbereich von b_1 und von b_3, b_2 ist K-Nachbarbereich von b_3, b_2 ist E_{in}-Nachbarbereich von v_2, q_1 ist E_{out}-Nachbarphase von p_2, v_1 ist K-Nachbarphase von b_1 und $\{q_1, q_2\}$. Die Unterscheidung der Nachbarbereiche/-phasen in K-, E_{out}- und E_{in}-Nachbarbereiche/ -phasen kommt bei der Spezifikation der Dynamik (Kapitel 3) zum Tragen. Abhängig vom Typ wird dort die Art der Interaktion zwischen den betroffenen Komponenten festgelegt.

Notation 2.5. Für eine einelementige Menge $\{p\}$ wird auch p geschrieben, wenn der Zusammenhang eindeutig ist. Für die Mengenoperatoren $\setminus$ (Subtraktion), $\cap$ (Durchschnitt) und $\cup$ (Vereinigung) wird festgelegt, dass $\setminus$ stärker bindet als $\cap$ und $\cup$, und dass $\cap$ stärker bindet als $\cup$. D.h. z.B. für Mengen A, B, C, D wird statt $(A \setminus B) \cup (C \cap D)$ auch $A \setminus B \cup C \cap D$ geschrieben.

Da komplexe verteilte Systeme, die aus hunderten oder mehr Systemkomponenten bestehen, weder als ein Gesamtes konstruiert noch analysiert werden können, verlagert man die Bearbeitung auf *Teilsysteme*, die aus mehreren benachbarten Komponenten bestehen. Die Auswahl der Teilsysteme unterliegt dabei keinem allgemein gültigen Mechanismus. Sie ergibt sich in der Regel aus dem funktionellen Kontext, der Wiederverwendbarkeit und dem Analyseaufwand.

Definition 2.6 (Teilsystem). Sei $IS = (P, B, \underline{B}, K, E)$ ein I–System. Sei $M \subseteq B$ und $Q := \bigcup\limits_{b \in M} b$. Das *Teilsystem $IS\lfloor_M$ von IS* ist festgelegt als $IS\lfloor_M := (Q, M, \underline{B} \cap M, K', E')$ mit $K' = \{(p,q) \in K \mid \{p,q\} \subseteq Q\}$ und $E' = \{(p,q) \in E \mid \{p,q\} \subseteq Q\}$. □

Das Teilsystem $IS\lfloor_M$ ergibt sich anschaulich im zugehörigen Graphen durch Löschen aller Bereiche von IS, die nicht zu M gehören, sowie aller Kopplungs- und Erregungskanten, die mit Phasen außerhalb der Bereiche aus M verbunden sind. Aus der Definition ist ersichtlich:

Bemerkung 2.7. Jedes Teilsystem eines I–Systems ist selbst ein I–System.

Auf speziell aufgebaute Teilsysteme wird bei der semantischen Analyse von I–Systemen zurückgegriffen (ab Kapitel 9). Für die Teilsysteme werden gewisse Eigenschaften formuliert und bewiesen, die dann auch im Verbund, unter Beachtung der Einflüsse der Umgebung, gültig sind.

Kapitel 3

Dynamik

Bei der Modellierung von verteilten Systemen mit I–Systemen stehen nicht nur strukturelle Aspekte, d.h. die Aufteilung der verteilten Systeme in Teilsysteme bzw. Komponenten, im Blickpunkt des Interesses, sondern man möchte auch Anforderungen an die Aktivitäten in den Komponenten und an die Interaktionen zwischen den Komponenten formal spezifizieren und anhand der Modelle verifizieren können. Dazu ist es notwendig, die mathematische Struktur des I–Systems aus Kapitel 2.1 um dynamische Elemente zu erweitern. Aus der gleichen Intention heraus werden z.B. bei Petri-Netzen das Tokenspiel und bei Automaten oder Statecharts Zustandsübergangsfunktionen eingeführt.
In Kapitel 1.1 wurde als ein Schlüsselphänomen von verteilten Systemen festgehalten, dass sich Ereignisse in den Komponenten, aus denen sich die Aktivität des Gesamtsystems ergibt, in zwei Typen unterteilen lassen (siehe auch [88]). Zum einen gibt es *erzwungene* Ereignisse, die eintreten *werden*, sofern dies nicht durch externe Einflüsse verhindert wird. Ursachen für erzwungene Ereignisse in einer Komponente können einerseits externe Einflüsse von Nachbarkomponenten, andererseits interne Zwänge aufgrund lokaler Verhaltensvorschriften sein. Zum anderen gibt es *freie* Ereignisse, die eintreten *können* oder auch nicht, d.h. insbesondere liegt kein Zwang vor, eine Aktion durchzuführen. Freie Ereignisse können nur in autonomen Komponenten auftreten und von außen betrachtet obliegt es dem Entscheidungswillen der Komponente, ob und wann welche Aktion ausgeführt wird. Die unterschiedlichen Typen von Ereignissen und die Ursachen für deren Eintreten oder auch deren Nicht-Eintreten sollen innerhalb der Dynamik zum Ausdruck kommen.
Bei I–Systemen wird ein zustandsbasierter Ansatz bei der Spezifikation der Dynamik verwendet. Dazu werden im nächsten Abschnitt zwei Varianten vorgestellt, Systemzustände eines I–Systems zu beschreiben. Übergänge zwischen Systemzuständen repräsentieren dann Ereignisse in dem System, aus denen sich das Gesamtverhalten zusammensetzt (vgl. [55]). Da, um eine Anwendung korrekt zu modellieren, in der Regel nicht jeder beliebige Übergang von einem Zustand aus erlaubt ist, bedarf es Regeln zur Bestimmung der möglichen Folgezustände. Die notwendigen Formalismen werden im verbleibenden Teil dieses Kapitels behandelt.

3.1 Globale Systemzustände

Die Spezifikation der Dynamik eines I–Systems beruht auf der Beschreibung von Veränderungen von Systemzuständen. In einem verteilten System ergibt sich ein *globaler Systemzustand* aus den *lokalen* Systemzuständen der einzelnen Komponenten [8, 23, 30]. Ein globaler aktueller Systemzustand ist aufgrund der Verteiltheit der Kontrolle in der Regel lokal nicht bekannt und auch lokal nicht bestimmbar [31, 68]. Trotzdem sind Anforderungen an ein verteiltes System meistens mit Blick auf das Gesamtsystem formuliert, also global [12, 31, 78]. Hierzu gehören z.B. Sicherheits- oder Fairnesskriterien. In den folgenden beiden Unterabschnitten werden zwei unterschiedliche Arten von globalen Systemzuständen definiert, deren Unterschiede in der Ausdrucksstärke über das lokale Verhalten in den Komponenten liegen.

3.1.1 Cases

In Kapitel 2.1 wurde erwähnt, dass die Kopplungsrelation K eines I–Systems IS eine Menge von Paaren wechselseitig ausgeschlossener Phasen spezifiziert. Die Bedingung 4.b aus Definition 2.1 impliziert somit, dass ein Bereich sich nicht gleichzeitig in zweien seiner Phasen befinden kann. Setzt man nun zur Präzisierung der Aktivität eines Bereiches von IS voraus, dass sich dieser in genau einer Phase befindet, so definiert jede Phase für sich betrachtet einen lokalen Zustand des zugehörigen Bereiches. Ausgehend von den lokalen Zuständen der Bereiche verwenden wir die Beziehungen, die durch die Kopplungsrelation gegeben sind, um globale Systemsituationen zu definieren, die als Cases bezeichnet werden.

Definition 3.1 (Case). Sei $IS = (P, B, \underline{B}, K, E)$ ein I–System. Ein *Case* von IS ist eine Menge $c \subseteq P$ von Phasen mit:

(1) $\forall b \in B : |c \cap b| = 1$

(2) $\forall p_1, p_2 \in c : (p_1, p_2) \notin K$

$Case(IS)$ bezeichnet die Menge aller Cases von IS.
Typische Bezeichnungen für Cases sind c, c_1, c'. □

3.1.2 Aktivitätszustände

In der Einleitung dieses Kapitels wurde erwähnt, dass man bei den Ereignissen in einer Komponente eines verteilten Systems zwei Typen unterscheiden kann: erzwungene Ereignisse (die eintreten werden) und freie Ereignisse (die eintreten können). Abhängig ist das Verhalten im Wesentlichen von externen Einflüssen, lokalen Zwängen, lokalen Entscheidungen. Um solche Phänomene formal zu erfassen, werden den einzelnen Phasen bestimmte Aktivitätsqualitäten (*Phasenqualitäten*) zugewiesen. Genauer, ist ein Bereich b in einer Phase p, unterscheidet man zwischen drei verschiedenen Zuständen von p:

a) Die Phasenqualität F verdeutlicht das Wirken von Einflüssen von Nachbarbereichen. Der Bereich b unterliegt einem Zwang, die Phase p zu verlassen, und führt bestimmte Aktionen aus, um dieses (sofern möglich) zu erreichen.

b) Ereignisse, die nicht durch Einflüsse erzwungen sind, werden interpretiert als das Ergebnis einer Kontrollentscheidung in dem Bereich. Die Phasenqualität q bezeichnet eine Entscheidung von b, einen Phasenwechsel nach q durchführen zu wollen.

c) Die Phasenqualität $\mathsf{1}$ bezeichnet einen Zustand, in dem b Aktivitäten ausführt, die der Phase p zugeordnet sind, aber es liegt weder eine Entscheidung vor, in eine andere Phase einzutreten, noch sind Einflüsse von Nachbarbereichen wirksam, die es notwendig machen, p zu verlassen.

Die trägen Bereiche eines I–Systems sind in Kapitel 2.1 durch reaktive Systeme und Software motiviert worden. Sie spiegeln das Verhalten von Komponenten wider, in denen *jedes* Ereignis das Resultat externer Einflüsse ist. Für Bereiche, die solche Komponenten repräsentieren, entfällt Punkt b), d.h. träge Bereiche können keine Entscheidung treffen.
Zusammenfassend definiert sich ein *lokaler Aktivitätszustand* eines Bereiches wie folgt:

Definition 3.2 (Lokaler Aktivitätszustand). Sei $IS = (P, B, \underline{B}, K, E)$ ein I–System und $b \in B$. Seien $\mathsf{0}$, $\mathsf{1}$ und F ausgezeichnete Symbole mit $P \cap \{\mathsf{0}, \mathsf{1}, \mathsf{F}\} = \emptyset$.

Eine Abbildung $z\langle b\rangle : b \longrightarrow b \cup \{\mathsf{0}, \mathsf{1}, \mathsf{F}\}$, für die gilt:

(1) $\exists! p \in b : z\langle b\rangle(p) \neq \mathsf{0}$

(2) $\forall p_1, p_2 \in b : (z\langle b\rangle(p_1) = p_2) \Rightarrow (p_1 \neq p_2) \wedge b(p_1) \in AB(IS))$

heißt *lokaler Aktivitätszustand* von b.

Die Elemente des Wertebereichs eines lokalen Aktivitätszustandes werden *Phasenqualitäten* genannt. Für eine Phase p in einem Bereich b werden die Phasenqualitäten wie folgt interpretiert:

$$z\langle b\rangle(p) = \begin{cases} q: b \text{ ist in } p. \ b \text{ hat die Entscheidung getroffen, nach } q \text{ zu wechseln.} \\ \mathsf{F}: b \text{ ist in } p. \ b \text{ ist instabil in } p. \\ 1: b \text{ ist in } p. \ b \text{ ist stabil in } p. \\ 0: b \text{ ist nicht in } p. \end{cases}$$

$LZustand(b)$ ist die Menge aller lokalen Aktivitätszustände von b.
Typische Bezeichnungen für lokale Aktivitätszustände von b sind $z\langle b\rangle$, $z'\langle b\rangle$, $z_1\langle b\rangle$. □

Man beachte, dass die Voraussetzung $P \cap \{0, 1, \mathsf{F}\} = \emptyset$ durch geeignete Umbenennung der Phasen in P immer erreicht werden kann. Ohne Einschränkung gehen wir fortan davon aus, dass diese Voraussetzung bei jedem I–System erfüllt ist.

Ein *globaler Aktivitätszustand* eines I–Systems setzt sich aus lokalen Aktivitätszuständen der Bereiche zusammen. Dabei wird gefordert, dass der wechselseitige Ausschluss zwischen Phasen, der durch die Kopplungsrelation spezifiziert wird, eingehalten wird.

Definition 3.3 (Globaler Aktivitätszustand). Sei IS ein I–System wie in Definition 3.2 und $B = \{b_1, \dots, b_n\}$, $n \in I\!N$. Seien $z\langle b_i\rangle \in LZustand(b_i)$, $i = 1, \dots, n$, lokale Aktivitätszustände der einzelnen Bereiche.

Eine Abbildung $z : P \longrightarrow P \cup \{0, 1, \mathsf{F}\}$, für die gilt:

(1) $z|_{b_i} = z\langle b_i\rangle$

(2) $\forall p_1, p_2 \in P : (z(p_1) \neq 0 \wedge z(p_2) \neq 0) \Rightarrow (p_1, p_2) \notin K$

heißt *globaler Aktivitätszustand* von IS.

$z|_{b_i}$ ist dabei die Einschränkung von z auf den Definitionsbereich b_i. Die Festlegung und Interpretation von Phasenqualitäten gilt analog zu Definition 3.2.

$GZustand(IS)$ ist die Menge aller globalen Aktivitätszustände von IS.
Typische Bezeichnungen für globale Aktivitätszustände von IS sind $z\langle IS\rangle$, $z'\langle IS\rangle$, $z_1\langle IS\rangle$, bzw. z, z', z_1, wenn das zugehörige I–System eindeutig ist. Für $\tilde{z} \in GZustand(IS)$ bezeichnen $\tilde{z}\langle b_1\rangle, \dots, \tilde{z}\langle b_n\rangle$ die zugrunde liegenden lokalen Aktivitätszustände, d.h. $\tilde{z}\langle b_i\rangle = \tilde{z}|_{b_i}$ für $i = 1, \dots, n$. □

Da alle Bereiche (als Mengen von Phasen) eines I–Systems disjunkt sind (Definition 2.1) und der Definitionsbereich ganz P umfasst, stellen die globalen Aktivitätszustände durch die Hinzunahme von Phasenqualitäten eine Verfeinerung der Cases dar. Ein globaler Aktivitätszustand z ist dabei als eine Verfeinerung eines Cases c anzusehen, wenn für jede Phase p gilt: $p \in c$ gdw. $z(p) \neq 0$. Jeder globale Aktivitätszustand ist eindeutig bestimmt durch die Phasen, deren Phasenqualität ungleich 0 ist. Die folgende Notation vernachlässigt die Phasen, die auf 0 abgebildet werden.

Notation 3.4. Ein globaler Aktivitätszustand z wird beschrieben durch ein Tupel $[p_1{<}\alpha_1{>}, p_2{<}\alpha_2{>}, \dots, p_n{<}\alpha_n{>}]$ genau dann, wenn gilt: $\{p_1, p_2, \dots, p_n\} = \{p \mid z(p) \neq 0\}$, $n \in I\!N$ und $z(p_i) = \alpha_i$ für $i = 1, \dots, n$.

Im späteren Verlauf werden lokale und globale Aktivitätszustände, bei denen als Phasenqualitäten nur 0 und 1 auftreten, eine besondere Rolle spielen. Sie dienen insbesondere als Initialisierungszustände bei der Modellierung von dynamischen Abläufen. Um diese Zustände bezeichnen zu können, wird für sie ein eigener Begriff eingeführt.

Definition 3.5 (Stabilität). Sei $IS = (P, B, \underline{B}, K, E)$ ein I–System mit $B = \{b_1, \dots, b_n\}$, $n \in I\!N$.

a) Ein lokaler Aktivitätszustand $z\langle b_i\rangle$ von b_i, $i \in \{1, \dots, n\}$, heißt *stabil*, wenn für alle $p \in b_i$ gilt: $z\langle b_i\rangle(p) \neq 0 \Rightarrow z\langle b_i\rangle(p) = 1$.
$StabLZustand(b_i)$ bezeichnet die Menge aller stabilen lokalen Aktivitätszustände von b_i.

b) Ein globaler Aktivitätszustand $z\langle IS\rangle$ von IS heißt *stabil* genau dann, wenn die zugeordneten lokalen Aktivitätszustände $z\langle b_i\rangle$ für $i = 1, \dots, n$ stabil sind.
$StabGZustand(IS)$ bezeichnet die Menge aller stabilen globalen Aktivitätszustände von IS. □

Mit den Cases und den globalen Aktivitätszuständen eines I–Systems sind zwei Arten von Repräsentanten der (für die Modellierung relevanten) globalen Systemzustände eines modellierten verteilten Systems eingeführt worden. Wie schon erwähnt wurde, stellen Letztere dabei eine Verfeinerung von Ersteren dar. Die Beziehungen zwischen beiden Zustandsarten werden formal durch die zwei folgenden Abbildungen präzisiert. Die Funktion $cz(\cdot)$ bettet die Menge der Cases in die Menge der globalen Aktivitätszustände ein, derart, dass jede Phase eines Cases beim zugewiesenen globalen Aktivitätszustand auf den Phasenwert 1 abgebildet wird. Alle verbleibenden Phasen erhalten den Wert $\mathsf{0}$. Der resultierende Zustand ist somit stabil. Die Funktion $zc(\cdot)$ projiziert globale Aktivitätszustände auf Cases. Eine Phase, deren Phasenqualität beim globalen Aktivitätszustand ungleich $\mathsf{0}$ ist, wird in den zugeordneten Case übernommen. Auf diese Weise geht die Information über den genauen Wert der ursprünglichen Phasenqualität (z.B. $\mathsf{1}$, F oder q) verloren.

Definition 3.6 (zc, cz). Sei $IS = (P, B, \underline{B}, K, E)$ ein I–System mit $B = \{b_1, \ldots, b_n\}$, $n \in I\!N$.

a) Die Funktion zc ordnet einem globalen Aktivitätszustand $z \in GZustand(IS)$ einen Case der Menge $Case(IS)$ zu in der Form:
$zc(z) := \{p \in P \mid z(p) \neq \mathsf{0}\}$

b) Die Funktion cz ordnet einem Case $c \in Case(IS)$ einen stabilen globalen Aktivitätszustand zu in der Form:
$cz(c) : P \longrightarrow P \cup \{\mathsf{0}, \mathsf{1}, \mathsf{F}\}$ mit $cz(c)(p) = \begin{cases} \mathsf{1} \text{ falls } p \in c \\ \mathsf{0} \text{ sonst} \end{cases}$ □

Es bleibt, die Wohldefiniertheit der Funktionen zc und cz zu überprüfen.

z ist nach Voraussetzung ein globaler Aktivitätszustand von IS. Damit gilt nach den Definitionen 3.2.2 und 3.3: (Der Definitionsbereich von z ist ganz P) und $(\forall b \in B\, \exists!\, p_b \in b : z|_b(p_b) \neq \mathsf{0})$ und $(\forall p_1, p_2 \in P$ mit $z(p_1) \neq \mathsf{0}$ und $z(p_2) \neq \mathsf{0}$ gilt: $(p_1, p_2) \notin K)$. Mit der angegebenen Konstruktion von $zc(z)$ erhält man direkt: $(zc(z) \subseteq P)$ und $(\forall b \in B : |zc(z) \cap b| = 1)$ und $(\forall p_1, p_2 \in zc(z) : (p_1, p_2) \notin K)$. Damit gilt nach Definition 3.1: $zc(z) \in Case(IS)$.

c ist nach Voraussetzung ein Case von IS, also gilt nach Definition 3.1: $(c \subseteq P)$ und $(\forall b \in B : |c \cap b| = 1)$ und $(\forall p_1, p_2 \in c : (p_1, p_2) \notin K)$. Mit der angegebenen Konstruktion von $cz(c)$ ergibt sich: (Der Definitionsbereich von z ist P) und $(\forall b \in B : |\{p \mid cz(c)|_b(p) \neq \mathsf{0}\}| = 1)$ und $(\forall p_1, p_2 \in P$ mit $cz(c)(p_1) \neq \mathsf{0}$ und $cz(c)(p_2) \neq \mathsf{0}$ gilt: $(p_1, p_2) \notin K)$ und $(\forall p \in P : cz(c)(p) \neq \mathsf{0} \Rightarrow cz(c)(p) = 1)$. Damit gilt nach den Definitionen 3.3 und 3.5: $cz(c) \in GZustand(IS)$ und $cz(c)$ ist stabil.

Nach der Spezifikation der Beziehungen zwischen globalen Aktivitätszuständen und Cases durch die Funktionen zc und cz stellt sich die Frage nach der Hintereinanderausführung der beiden Funktionen. Erhält man denselben Aktivitätszustand zurück, wenn man auf ihn die Funktion zc und auf das Ergebnis die Funktion cz anwendet? Gilt gleiches für Cases und die Hintereinanderausführung von cz und zc? Der folgende Satz verdeutlicht die Zusammenhänge.

Satz 3.7 (zc, cz). Sei IS ein I–System.

a) Für einen stabilen globalen Aktivitätszustand $z \in StabGZustand(IS)$ gilt:
$cz(zc(z)) = z$

b) Für einen Case $c \in Case(IS)$ gilt:
$zc(cz(c)) = c$

Beweis. Der Satz folgt direkt aus den Definitionen 3.1, 3.5 und 3.6 sowie den vorangegangenen Betrachtungen zur Wohldefiniertheit von zc und cz. □

Es sei bei a) noch einmal auf die Notwendigkeit der Forderung nach Stabilität von z hingewiesen. Durch die Funktion zc geht die Information über die Phasenqualitäten verloren. cz erzeugt stabile globale Aktivitätszustände, sodass zur Erfüllung der Umkehreigenschaft Stabilität vorausgesetzt werden muss.

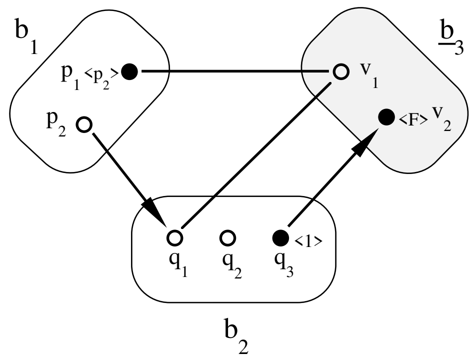


Abbildung 3.1: Globaler Aktivitätszustand von IS_1

Die Kopplungsrelation ist motiviert worden als Spezifikationsmittel für wechselseitigen Ausschluss und die Erregungsrelation zur expliziten Modellierung von Einflüssen zwischen Komponenten. Die folgende Definition beschreibt Eigenschaften, die sich auf einen globalen Zustand (Case oder globaler Aktivitätszustand) beziehen in Verbindung mit der Kopplungs- und Erregungsrelation. *Freiheit* bedeutet, dass eine Phase im aktuellen Zustand nicht aufgrund eines wechselseitigen Ausschlusses durch eine andere Phase blockiert wird und somit als Folgephase bei einem anstehenden Phasenwechsel in Frage kommt. *Erregung* beschreibt aufgebaute Einflüsse. Es gibt eine erregende und eine erregte Phase, die beide in benachbarten Komponenten liegen. Die Komponente der erregenden Phase übt einen Einfluss auf die Komponente der erregten Phase auf, diese wenn möglich zu verlassen.

Definition 3.8 (frei, erregt). Seien IS ein I–System mit Phasenmenge P und $c \in Case(IS)$ und $z \in GZustand(IS)$.

a) *p ist frei in c* gdw. $p \notin c \wedge \forall v \in K(p) : v \notin c$.

b) *p ist frei in z* gdw. $z(p) = \mathsf{0} \wedge \forall v \in K(p) : z(v) = \mathsf{0}$.

c) *p erregt v in c* gdw. $p \in c \wedge v \in c \wedge (p, v) \in E$.

d) *p erregt v in z* gdw. $z(p) \neq \mathsf{0} \wedge z(v) \neq \mathsf{0} \wedge (p, v) \in E$.

Die einzelnen Punkte der Definition sind in dem Sinne konsistent zu den Beziehungen zwischen Cases und globalen Aktivitätszuständen gehalten, dass p ist frei (erregt v) in c genau dann gilt, wenn p ist frei (erregt v) in z und $zc(z) = c$ gilt. Die Bedeutung dieser Eigenschaften für das Systemverhalten wird im nächsten Abschnitt bei der Formalisierung der dynamischen Abläufe zum Ausdruck gebracht.

3.1.3 Graphische Darstellung

Die graphische Darstellung eines globalen Aktivitätszustandes z eines I–Systems erfolgt, ausgehend von der Darstellung des I–Systems (siehe Kapitel 2.2), durch (schwarze) Einfärbung aller Kreise der Phasen p mit $z(p) \neq \mathsf{0}$ und durch zusätzliche Beschriftung mit „$<\alpha>$“ wobei gilt: $\alpha = z(p)$. Alle eingefärbten Kreise bilden den zugeordneten Case $zc(z)$. Ist nur der Case von Interesse, kann auf die Zusatzbeschriftung verzichtet werden.
Abbildung 3.1 zeigt einen globalen Aktivitätszustand $z := [p_1{<}p_2{>}, q_3{<}1{>}, v_2{<}\mathsf{F}{>}]$ für das I–System IS_1 aus Beispiel 2.2. Der zugeordnete Case ist $c := \{p_1, q_3, v_2\}$.
Mit den Bezeichnungen aus Definition 3.8 gilt: p_2, q_1, q_2 sind frei in c (in z) und q_3 erregt v_2 in c (in z). Es gilt hingegen nicht: v_1 ist frei in c (in z) oder p_2 erregt q_1.

3.2 V$_I$Systeme

Nach der Definition von globalen Systemzuständen in Form von Cases oder globalen Aktivitätszuständen sollen nun Regeln angegeben werden, die festlegen, welche aufeinander folgenden Systemzustände Ereignisse in einem gegebenen I–System repräsentieren und welche nicht. Das Regelsystem soll dabei die Interpretationen der Phasenqualitäten bei globalen Aktivitätszuständen (siehe Abschnitt 3.1.2) anschaulich umsetzen. Weiterhin soll es ermöglichen festzulegen, welche Ereignisse eintreten werden und welche eintreten können.
Um die zuvor aufgeführten Kriterien zu erfüllen, wird ein Ansatz gewählt, bei dem die Bereiche des betrachteten I–Systems als Komponenten eines konkreten verteilten Systems angesehen werden mit der Möglichkeit, verteilte Algorithmen, die das lokale Verhalten der Komponenten spezifizieren, auszuführen. Die Komponenten interagieren dabei mittels Nachrichtenaustausch. Da aber ein I–System auf der formalen Ebene definiert ist und als mathematisches Konstrukt diese Fähigkeiten nicht besitzen kann, wird ihm ein eindeutiges konkretes verteiltes System, genannt $V_I System$, mit axiomatischen Verhaltensbeschreibungen zugeordnet. Die Verhaltensbeschreibungen sind aus Sicht einer einzelnen Komponente formuliert und gelten für alle Komponenten gleich. Durch dieses Lokalitätsprinzip bleibt die Spezifikation der Dynamik unabhängig von der Komplexität des Gesamtmodells.
Die Spezifikation der Dynamik ist komplexer als z.B. die von Petri-Netzen oder Statecharts, erlaubt aber durch die Anlehnung an ein konkretes verteiltes System ein anschauliches realitätsbezogenes Verständnis der verteilten Abläufe und notwendigen Synchronisationsmechanismen.

Definition 3.9 (V$_I$System). Sei $IS = (P, B, \underline{B}, K, E)$ ein I–System mit $B = \{b_1, \ldots, b_n\}, n \in I\!N$. $V_I System(IS)$ ist ein IS eindeutig zugeordnetes verteiltes System kommunizierender Komponenten $V_{b_1}, \ldots, V_{b_n}$. V_{b_i} hat b_i als Zustandsmenge $(i = 1, \ldots, n)$. V_{b_j} ist eine Nachbarkomponente (d.h. es besteht die Möglichkeit zur wechselseitigen Kommunikation) von V_{b_i} genau dann, wenn b_j ein Nachbarbereich von b_i ist $(i, j \in \{1, \ldots, n\}, i \neq j)$. Das Verhalten von $V_I System(IS)$ ist durch die unter 3.2.1 angegebenen Verhaltensaxiome festgelegt. $V_I System(IS)$ wird als $V_I System$ klassifiziert. □

3.2.1 Verhaltensaxiome

VA1 (Verhalten)
Das Verhalten von $V_I System(IS)$ ergibt sich aus dem nebenläufigen Verhalten der einzelnen Komponenten. Das Verhalten einer Komponente V_b ist spezifiziert durch die Aktionen unter 3.2.2. Die Aktivität beginnt dabei jeweils mit Aktion A1.

VA2 (Kommunikation zwischen Komponenten)
Die Kommunikation zwischen Komponenten erfolgt durch Nachrichtenaustausch. Nachrichten treffen sicher in der Reihenfolge ein, in der sie gesendet wurden. Es werden alle eingetroffenen Nachrichten in der Reihenfolge ihres Empfanges abgearbeitet. Nachrichten bleiben so lange gespeichert, bis sie bearbeitet wurden. Auf alle zwischenzeitlich eingetroffenen Nachrichten wird reagiert, entweder vor Ausführung der nächsten Aktion ungleich A6, A7, A13 oder während einer Aktion, solange auf eintreffende Nachrichten gewartet wird. Nachrichten, die erwartet werden, treffen schließlich auch ein.

VA3 (Zeit)
Es existieren eine Lokalzeit für jede Komponente sowie eine Globalzeit (Systemumgebungszeit) für das Gesamtsystem. Die Ausführungs- und Kommunikationszeiten sind relevant und endlich, aber sowohl lokal als auch global nicht bekannt. Variablenzuweisungen sind zeitlich atomar.

Erläuterungen. VA1 verdeutlicht die lokale Unabhängigkeit der einzelnen Komponenten, die zwar alle dieselben Aktionen ausführen, bei denen aber einer Komponente nicht bekannt ist, was sich genau in den anderen zur gleichen Zeit abspielt. Die einzige Möglichkeit gegenseitiger Einflussnahme ist die Übermittlung von Daten in Form von Nachrichten. Die durch VA2 gewährleistete Sicherheit des Nachrichtenaustausches dient ausschließlich der Vereinfachung des Aktionensystems. Die Komplexität der Aktionen soll minimiert und damit das Verständnis erleichtert werden. Es ist aber durchaus möglich, mittels bekannter Verfahren die Aktionen zu erweitern, um Fehlertoleranz und Deadlockfreiheit zu garantieren, ohne dieses explizit axiomatisch festzulegen. Spezielle Kommunikationsprotokolle, die z.B. mit Timeouts und Nachrichtenwiederholungen arbeiten, bieten entsprechende Mechanismen, die in verteilten Systemen Anwendung finden [24, 36, 49]. Die Existenz eines Zeitrahmens, geregelt durch VA3, ist notwendig, um globale Systemzustände während der nebenläufigen Ausführungen der Aktionen ableiten zu können. In Abschnitt 3.2.3 wird detailliert darauf eingegangen. Wir gehen hier von einem kontinuierlichen Zeitmodell aus, bei dem Zeitpunkte durch reellwertige Zahlenwerte repräsentiert werden und Zeitperioden durch reelle Intervalle (vgl. [41, 51]). Die Kommunikationszeiten sind nicht vorhersehbar, das Verhalten einer Komponente kann entsprechend der Reihenfolge des Eintreffens von Nachrichten variieren. Der Zusammenhang zwischen Lokalzeiten und Global-/Umgebungszeit bei verteilten Systemen ist Schwerpunkt umfangreicher wissenschaftlicher Arbeiten und findet unter anderem praktische Anwendung beim Lamport Algorithmus [60] oder Heartbeat Methoden [79]. Naturgemäß kann dabei keine Sicherheit über Synchronität gegeben werden, sondern nur im gewissen Rahmen. Die Atomarität der Variablenzuweisungen dient ausschließlich der Vereinfachung späterer Aussagen über das Systemverhalten und deren Beweise. Diese Festlegung führt zu keiner Einschränkung bei den Ausführungen der Aktionen.

3.2.2 Aktionen einer Komponente

Die Aktivität einer Komponente V_b wird durch eine Art von verteilten Algorithmen spezifiziert. Die Notation ist angelehnt an [64] und die Anweisungen in Pseudo-Code sind verbunden mit den üblichen Bedeutungen aus dem Gebiet der Programmiersprachen. Das Verhalten der Komponente ist dabei im Wesentlichen abhängig von den Werten spezieller *lokaler Variablen*, die nachfolgend aufgeführt sind. Die ersten drei Variablen sind boolesche Variablen, die in Abhängigkeit eintreffender Nachrichten für eine bestimmte Teilmenge von Phasen gesetzt werden. Sie beschreiben das „Wissen“ von V_b über die *Nachbarkomponenten*. Dabei wird $V_{b'}$ als Nachbarkomponente von V_b betrachtet, wenn b' ein Nachbarbereich von b ist. Die nächsten drei booleschen Variablen werden nicht explizit gesetzt, sondern gelten entsprechend den angegebenen Abhängigkeiten. Sie beschreiben bestimmte Zustände lokaler Phasen. Die letzte Variable ordnet jeder lokalen Phase von V_b eine Phasenqualität zu, deren Bedeutung mit der Interpretation der Phasenqualitäten bei lokalen Aktivitätszuständen übereinstimmt. Die folgende Aufzählung benennt die Variablen und spezifiziert daneben in eckigen Klammern den Typ, die Art des Updates und den in Frage kommenden Definitionsbereich. Zusätzlich gibt es zu jeder Variablen eine anschauliche Interpretation im Kontext wechselseitiger Einflüsse und derer Wirkungen, die in Anführungszeichen mit angegeben ist.

Lokale Variablen der Komponente V_b:

$\_mark(v)$ [boolesch, Update explizit, $v \in P \setminus b$]
„Die Komponente von v führt Aktionen aus, um v zu verlassen oder in v einzutreten.“

$\_in(v)$ [boolesch, Update explizit, $v \in P \setminus b$]
„Die Komponente von v ist in v oder (falls $\_mark(v)$ gilt) führt Aktionen aus, um v zu verlassen.“

$\_s(q)$ [boolesch, Update explizit, $q \in P$]
„Die Komponente von q beeinflusst Nachbarkomponenten zum Verlassen von mit q wechselseitig ausgeschlossenen Phasen.“

$\_e_{in}(p)$ [boolesch, Update implizit, $p \in b$]
$\_e_{in}(p) = true$ gdw. eine Phase x existiert mit $(x,p) \in E$ und $\_in(x)$.
„Ist V_b in p, dann wird p erregt.“

$\_e_{out}(p)$ [boolesch, Update implizit, $p \in b$]
$\_e_{out}(p) = true$ gdw. eine Phase x existiert mit $(p,x) \in E$ und $\_in(x)$.
„Ist V_b in p, dann ist p Erreger.“

$\_k(p)$ [boolesch, Update implizit, $p \in b$]
$\_k(p) = true$ gdw. eine Phase x existiert mit $(p,x) \in K$ und $\_in(x)$.
„p ist nicht frei.“

$\_z(p)$ [in $b \cup \{0, 1, \mathsf{F}\}$, Update explizit, $p \in b$]
„Phasenqualität von p mit folgenden Interpretationen:

$$\_z(p) = \begin{cases} q : V_b \text{ ist in } p, V_b \text{ hat die Entscheidung getroffen, nach } q \text{ zu wechseln.} \\ \mathsf{F} : V_b \text{ ist in } p. V_b \text{ ist instabil in } p. \\ 1 : V_b \text{ ist in } p. V_b \text{ ist stabil in } p. \\ 0 : V_b \text{ ist nicht in } p.\text{“} \end{cases}$$

Aktionen von V_b:

Wenn immer es möglich ist, führt V_b eine der folgenden Aktionen aus. Die Möglichkeit ergibt sich aus einer erfüllten Vorbedingung und der Beachtung von VA1 und VA2. In Anführungszeichen ist eine anschauliche Interpretation der Abläufe angegeben.

A1 (Initialisierung)
Vorbedingung:
Es wird ausgegangen von einer Startbelegung mit $\_z(p) = 1$ für genau ein $p \in b$ und $\_z(p') = 0$ für alle $p' \in b \setminus \{p\}$. Alle booleschen Variablen sind $false$.
Aktion:
Sende an alle $V_{b'}$, b' ist Nachbarbereich von b, die Nachricht $reqinit$.
Empfange jeweils die Nachricht $ackinit(v)$ mit $v \in P$.
$\_in(v) := true$, für jedes $ackinit(v)$.
Update-Aufruf (A13).
■
„V_b erkundigt sich bei jeder Nachbarkomponente nach dessen aktuellen Phase und führt am Ende ein Phasenqualitätsupdate der eigenen aktuellen Phase durch.“

A2 (Reaktion auf *reqinit*)
Vorbedingung:
Empfang der Nachricht $reqinit$ von $V_{b'}$.
Es gilt $\_z(p) \neq 0$ für ein $p \in b$.
Aktion:
Sende an $V_{b'}$ die Nachricht $ackinit(p)$.
■
„V_b antwortet auf eine Anfrage nach der aktuellen Phase.“

A3 (Autonome Entscheidung treffen)
Vorbedingung:
b ist autonom und $\_z(p) = 1$ für ein $p \in b$.
Aktion:
Falls $not\ \_e_{out}(p)$ und $not\ \_mark(v)$ für eine $\mathrm{E_{out}}$-Nachbarphase v von p gilt, dann <u>kann</u> $\_z(p) := q$, mit $q \in b \setminus \{p\}$, ausgeführt werden.
■
„Ist p eine stabile Phase von V_b und kein Erreger, dann kann p eine Entscheidung zur Phasentransition nach q treffen, vorausgesetzt, V_b ist nicht träge.“

A4 (Phasentransition wegen Entscheidung)
Vorbedingung:
Es gilt $\_z(p) = q$ für ein $p \in b$.
Es handelt sich um den ersten Aufruf nach A1 oder lokale Variablen haben sich seit dem letzten Aufruf verändert.
Aktion:
Sende an alle $V_{b'}$, b' ist Nachbarbereich von $\{p, q\}$, die Nachricht $reqmark(\{p, q\})$.
Empfange jeweils die Nachricht $ackmark$.
Fälle:
a) Es gilt inzwischen $\_z(p) = 1$.
b) $\_mark(v)$ für eine $\mathrm{E_{out}}$-Nachbarphase v von p.
c) $not\ \_k(q)$ und $not\ \_mark(v)$ für jede K-Nachbarphase v von q.

Verhalten:
Bei a) oder b):
Sende an alle $V_{b'}$, b' ist Nachbarbereich von $\{p,q\}$, die Nachricht *break*.
Bei *not* a) und *not* b) und c):
Die Phasentransition $p \to q$ wird atomar ausgeführt.
(d.h. $\_z(p) := 0$, $\_z(q) := 1$ gleichzeitig)
$\_s(q) := false$.
Update-Aufruf (A13).
Sende an alle $V_{b'}$, b' ist Nachbarbereich von $\{p,q\}$, die Nachricht *done*$(p \to q)$.
Bei *not* a) und *not* b) und *not* c):
Sende an alle $V_{b'}$, b' ist Nachbarbereich von $\{p,q\}$, die Nachricht *break*.
Falls weiterhin $\_z(p) = q$ gilt: Solicitation-Aufruf bzgl. $\{q\}$ (A6).

■

„Es wird versucht, eine Phasentransition von p nach q durchzuführen. V_b bittet diesbezüglich um eine Bestätigung der Kenntnisnahme von den Nachbarkomponenten. Nach Erhalt der letzten Bestätigung können drei Fälle eintreten. Erstens kann p inzwischen ein Erreger sein, die Aktion wird dann abgebrochen. Zweitens kann die Phasentransition durchgeführt werden, sofern p definitiv kein Erreger und q frei ist. Danach erfolgt ein Phasenqualitätsupdate der dann aktuellen Phase q. Drittens können Einflüsse auf Nachbarkomponenten ausgeübt werden, sofern p kein Erreger und q nicht frei ist. Durch die Einflüsse soll erreicht werden, dass Nachbarkomponenten Phasen verlassen, die q blockieren (wechselseitiger Ausschluss). Die Nachbarkomponenten werden jeweils über den eingetretenen Fall informiert.“

A5 (Phasentransition wegen Erregung)
Vorbedingung:
Es gilt $\_z(p) = \mathsf{F}$ für ein $p \in b$.
Es handelt sich um den ersten Aufruf nach A1 oder lokale Variablen haben sich seit dem letzten Aufruf verändert.
Aktion:
Sende an alle $V_{b'}$, b' ist Nachbarbereich von b, die Nachricht *reqmark*(b).
Empfange jeweils die Nachricht *ackmark*.
Sei $M_1 := \{q' \in b \setminus \{p\} \mid$ *not* $\_k(q')$, *not* $\_e_{in}(q')$ und *not* $\_mark(v)$ für jede K- oder $\mathrm{E_{in}}$-Nachbarphase v von q' $\}$.
Sei $M_2 := \{q' \in b \setminus \{p\} \mid$ *not* $\_k(q')$ und *not* $\_mark(v)$ für jede K-Nachbarphase v von q' $\}$.
Fälle:
a) Es gilt inzwischen $\_z(p) = 1$.
b) $\_mark(v)$ für eine $\mathrm{E_{out}}$-Nachbarphase v von p.
c) $M_1 \neq \emptyset$ oder $M_2 \neq \emptyset$.
Verhalten:
Bei a) oder b):
Sende an alle $V_{b'}$, b' ist Nachbarbereich von b, die Nachricht *break*.
Bei *not* a) und *not* b) und c):
Wähle beliebiges q aus M_1 falls $M_1 \neq \emptyset$, aus M_2 falls $M_1 = \emptyset$.
Die Phasentransition $p \to q$ wird atomar ausgeführt.
(d.h. $\_z(p) := 0$, $\_z(q) := 1$ gleichzeitig)
$\_s(p') := false$ für alle $p' \in b \setminus \{p\}$.
Update-Aufruf (A13).
Sende an alle $V_{b'}$, b' ist Nachbarbereich von b, die Nachricht *done*$(p \to q)$.
Bei *not* a) und *not* b) und *not* c):
Sende an alle $V_{b'}$, b' ist Nachbarbereich von b, die Nachricht *break*.
Solicitation-Aufruf bzgl. $b \setminus \{p\}$ (A6).

■

„Da p instabil ist, versucht V_b eine Phasentransition durchzuführen. Als potentielle Zielphase q kommen alle freien Phasen in Frage, bevorzugt werden aber solche, die nach einer Transition nicht erregt werden. V_b bittet diesbezüglich um eine Bestätigung der Kenntnisnahme von den Nachbarkomponenten. Nach Erhalt der letzten Bestätigung können drei Fälle eintreten. Erstens kann p inzwischen ein Erreger sein, die Aktion wird dann abgebrochen. Zweitens kann die

Phasentransition durchgeführt werden, sofern p definitiv kein Erreger ist und eine einnehmbare Folgephase existiert. Danach erfolgt ein Phasenqualitätsupdate der dann aktuellen Phase q. Drittens können Einflüsse auf Nachbarkomponenten ausgeübt werden, sofern p kein Erreger und keine freie Phase bei V_b existiert. Durch die Einflüsse soll erreicht werden, dass Nachbarkomponenten Phasen verlassen, die mögliche Folgephasen blockieren (wechselseitiger Ausschluss). Die Nachbarkomponenten werden jeweils über den eingetretenen Fall informiert.“

A6 (Solicitation)
Vorbedingung:
Aufruf bzgl. einer Phasenmenge $M \subseteq b$.
Aktion:
Sende an alle $V_{b'}$, b' ist K-Nachbarbereich von M, die Nachricht $solicit(M)$.
$\_s(q) := true$, für alle $q \in M$.
■
„Alle K-Nachbarkomponenten der Phasen aus M werden aufgefordert, eine Phasentransition durchzuführen, sofern die aktuelle Phase zu einer der Phasen aus M in der Kopplungsrelation steht.“

A7 (Cancellation)
Vorbedingung:
Aufruf.
Aktion:
Sei $M := \{q \in b \mid \_s(q)\}$.
Sende an alle $V_{b'}$, b' ist K-Nachbarbereich von M, die Nachricht $cancel(M)$.
$\_s(q) := false$, für alle $q \in M$.
■
„Die mittels $solicit(\cdot)$ erfolgten Aufforderungen zur Phasentransition (A6) werden zurückgenommen.“

A8 (Reaktion auf *reqmark*)
Vorbedingung:
Empfang der Nachricht $reqmark(M)$ von $V_{b'}$.
Aktion:
$\_mark(v) := true$, für alle $v \in M$.
Sende an $V_{b'}$ die Nachricht $ackmark$.
■
„Es wird registriert, dass bei der Nachbarkomponente $V_{b'}$ versucht wird, eine Phasentransition durchzuführen. Bis zum Erhalt weiterer Informationen wird jede Phase aus M als potentiell aktuell betrachtet. V_b bestätigt $V_{b'}$ die Kenntnisnahme.“

A9 (Reaktion auf *break*)
Vorbedingung:
Empfang der Nachricht $break$ von $V_{b'}$.
Aktion:
$\_mark(v) := false$, für alle $v \in b'$.
■
„Es wird registriert, dass die Nachbarkomponente $V_{b'}$ den Versuch abgebrochen hat, eine Phasentransition durchzuführen. Die bisherige aktuelle Phase bleibt dort auch weiterhin aktuell.“

A10 (Reaktion auf *done*)
Vorbedingung:
Empfang der Nachricht $done(v \to w)$ von $V_{b'}$.
Aktion:
$\_in(v) := false$, $\_in(w) := true$.
$\_s(v') := false$ und $\_mark(v') := false$, für alle $v' \in b'$.
Für p mit $\_z(p) \neq 0$:
  Update-Aufruf (A13).
■

„Es trifft die Nachricht ein, dass bei der Nachbarkomponente $V_{b'}$ eine Phasentransition von v nach w stattgefunden hat. V_b merkt sich w als neue aktuelle Phase von $V_{b'}$. Alle von $V_{b'}$ eingetroffenen Aufforderungen zur Phasentransition (mittels A6) können als nicht mehr relevant betrachtet werden. Der Phasenaktivitätswechsel bei $V_{b'}$ erfordert ein abschließendes Phasenqualitätsupdate der eigenen aktuellen Phase.“

A11 (Reaktion auf *solicit*)
Vorbedingung:
Empfang der Nachricht $solicit(M)$ von $V_{b'}$.
Aktion:
$\_s(v) := true$, für alle $v \in M$.
Für p mit $\_z(p) \neq 0$:
  Update-Aufruf (A13).
■
„Es trifft von der K-Nachbarkomponente $V_{b'}$ die Aufforderung ein, eine Phasentransition durchzuführen, sofern die eigene aktuelle Phase zu einer der Phasen aus M in der Kopplungsrelation steht. Die Auswirkungen dieser Aufforderung schlagen sich in einem Phasenqualitätsupdate der aktuellen Phase nieder.“

A12 (Reaktion auf *cancel*)
Vorbedingung:
Empfang der Nachricht $cancel(M)$ von $V_{b'}$.
Aktion:
$\_s(v) := false$, für alle $v \in M$.
Für p mit $\_z(p) \neq 0$:
  Update-Aufruf (A13).
■
„Die K-Nachbarkomponente $V_{b'}$ nimmt die Aufforderung, eine Phasentransition durchzuführen (A11), zurück. Als Reaktion erfolgt ein Phasenqualitätsupdate der eigenen aktuellen Phase.“

A13 (Update)
Vorbedingung:
Aufruf während einer Aktion. Es gilt $\_z(p) \neq 0$ für ein $p \in b$.
Aktion:
Fallunterscheidung:
i) $\_z(p) = \mathsf{F}$ und ($\_e_{out}(p)$ oder (*not* $\_e_{out}(p)$ und *not* $e_{in}(p)$ und *not* $\_s(v)$ für eine K-Nachbarphase v von p.)) :
  Cancellation-Aufruf (A7).
  $\_z(p) := 1$.
ii) $\_z(p) = q$ und $\_e_{out}(p)$:
  Cancellation-Aufruf (A7).
  $\_z(p) := 1$.
iii) $\_z(p) = q$ und *not* $\_e_{out}(p)$ und ($\_e_{in}(p)$ oder $\_s(v)$ für eine K-Nachbarphase v von p) :
  $\_z(p) := \mathsf{F}$.
iv) $\_z(p) = 1$ und *not* $\_e_{out}(p)$ und ($\_e_{in}(p)$ oder $\_s(v)$ für eine K-Nachbarphase v von p) :
  $\_z(p) := \mathsf{F}$.
v) Keiner der Fälle i)-iv) trifft zu:
  $\_z(p) := \_z(p)$.
■
„Die Phasenqualität der aktuellen Phase p wird, entsprechend den neuesten lokalen Informationen über die Zustände der Nachbarkomponenten, angepasst. Ein Wechsel von instabil nach stabil erfolgt, wenn p Erreger geworden oder kein Einfluss zum Phasenwechsel auf V_b mehr wirksam ist. Ein Wechsel von einer getroffenen Entscheidung nach stabil erfolgt, wenn p Erreger geworden ist. In beiden Fällen werden bestehende Aufforderungen an K-Nachbarkomponenten zur Phasentransition zurückgenommen. Ein Wechsel von einer getroffenen Entscheidung nach instabil sowie von stabil nach instabil erfolgt, wenn eine Erregung von p oder ein Einfluss auf V_b zum Phasenwechsel wirksam ist, vorausgesetzt, dass p kein Erreger ist.“

3.2.3 Ausführungen

Die Verifikation eines I–Systems beruht auf der formalen Spezifikation von Systemanforderungen (je nach zugrunde liegender Anwendung) und der Überprüfung dieser Anforderungen am Modell selbst. Damit auf das Verhalten eines I–Systems, was bedeutet auf die Aktivitäten des zugeordneten V_ISystems, formal Bezug genommen werden kann, ist es notwendig, eine begriffliche Schnittstelle zum axiomatisch/algorithmischen Verhalten zu definieren. Dabei ist es wichtig, den globalzeitlichen Rahmen mit einzubeziehen, um temporale Bezüge zwischen Aktionen einzelner Komponenten herstellen zu können. *Ausführungen* des V_ISystems sollen diesen Zweck erfüllen.

Definition 3.10 (Ausführung). Verhält sich das einem I–System IS zugeordnete verteilte System $V_ISystem(IS)$ wie folgt:

(1) Die Axiome VA1, VA2, VA3 zur Dynamik (siehe Abschnitt 3.2.1) werden respektiert.

(2) Eine andauernde Variablenbelegung $\_z(p) = q$ bei einer Komponente V_b von $V_ISystem(IS)$ führt dort schließlich zu einem Verhalten gemäß Aktion A4, Fall *not* b), und eine andauernde Variablenbelegung $\_z(p) = \mathsf{F}$ schließlich zu einem Verhalten gemäß Aktion A5, Fall *not* b).

Dann nennt man das eine *Ausführung* von $V_ISystem(IS)$.

Jede Ausführung beginnt zu einem ausgezeichneten Start(global)zeitpunkt und hat eine unendliche Ausführungsdauer. Typische Bezeichnungen für Ausführungen sind Π, Π', Π_1. □

Punkt (1) der Definition setzt die korrekte Anwendung der Axiome zur Dynamik voraus und daraus resultierend (laut Axiom VA1) die richtige Bearbeitung der Aktionen aus Abschnitt 3.2.2. Punkt (2) der Definition fordert eine bestimmte Art von Fairness bei den Ausführungen. Es soll verhindert werden, dass eine Komponente immer wieder die Aktion zur Durchführung einer Phasentransition (A4, A5) abbrechen muss, ohne jemals eine Phasentransition durchzuführen oder zumindest Einflüsse auf Nachbarkomponenten auszuüben. Fairness-Voraussetzungen finden sich auch bei anderen formalen Modellen, bei denen eine Systemdynamik spezifiziert wird, wieder [12, 19, 61, 66, 70]. Einen Überblick über unterschiedliche Definitionen von Fairness im Kontext von TLA liefert [62]. Im formalen Modell der I–Systeme kommt der Punkt (2) der Definition bei den semantischen Definitionen und Sätzen, die in Kapitel 4 behandelt werden, zum Tragen.
Die Festlegung der Unendlichkeit der Ausführungsdauer dient der Modellvereinfachung. Es kann vorkommen, dass von einem bestimmten Zeitpunkt an keine Variablenveränderungen in lokalen Komponenten mehr auftreten und kein Nachrichtenaustausch mehr stattfindet. Dieser Zeitpunkt ließe sich alternativ als Ausführungsende definieren. Er tritt allerdings nicht bei jeder Ausführung auf. Es gibt V_ISysteme mit Ausführungen, bei denen immer wieder neue Variablenbelegungen auftreten. Diese Ausführungen sind auch weiterhin von unendlicher Dauer. Zur Vereinheitlichung wird deshalb festgelegt, dass, einmal begonnen (mit Aktion A1), sich ein V_ISystem unendlich lange im Zustand der Abarbeitung der Axiome/Aktionen zur Dynamik befindet. Ein vergleichbarer Ansatz, eine Menge von Anweisungen unendlich oft zu bearbeiten und Fixpunkte bei den Zuständen als „Quasi"-Ende zu betrachten, findet sich in Unity [19].

Das Verhalten der einzelnen Komponenten ist abhängig von den lokalen Variablenbelegungen zu bestimmten Zeitpunkten einer Ausführung des V_ISystems. Möchte man das Verhalten analysieren, ist es notwendig, sich auf die Variablenbelegungen in Abhängigkeit der Globalzeit zu beziehen. Die folgende Definition liefert Notationen, die in dieser Arbeit durchgehend Verwendung finden.

Definition 3.11 (Belegung). Sei $IS = (P, B, \underline{B}, K, E)$ ein I–System mit $b \in B$ und $p \in b$. Sei $\alpha \in \{mark, in, s, e_{in}, e_{out}, k, z\}$ der Bezeichner einer lokalen Variablen $\_\alpha(p)$ bei der Komponente V_b des zugeordneten V_ISystems $V_ISystem(IS)$ entsprechend Abschnitt 3.2.2. Sei Π eine Ausführung von $V_ISystem(IS)$ und t ein Globalzeitpunkt der Ausführung.

a) $\alpha^{\Pi,t}\langle V_b\rangle(p)$ liefert den Wert der Variablen $\_\alpha(p)$ bei V_b zum Globalzeitpunkt t bezogen auf Π.

b) Die Funktion $\alpha^{\Pi,t}\langle V_ISystem(IS)\rangle$ mit Definitionsbereich P ist definiert durch $\alpha^{\Pi,t}\langle V_ISystem(IS)\rangle(p) = \alpha^{\Pi,t}\langle V_{b(p)}\rangle(p)$. Die Funktion heißt *$\alpha$-Globalbelegung.* □

Die Existenz der Globalzeit für $V_I Sytem(IS)$ ist festgelegt durch das Verhaltensaxiom VA3 in Abschnitt 3.2.1. Die durch den Satz eingeführten Begriffe $\alpha^{\Pi,t}\langle V_b\rangle(p)$ und $\alpha^{\Pi,t}\langle V_I System(IS)\rangle$ sind damit wohldefiniert. Der Wertebereich der α-Globalbelegung ergibt sich aus den möglichen Belegungen für $\_\alpha(p)$. Für $\alpha \in \{mark, in, s, e_{in}, e_{out}, k\}$ ist der Wertebereich somit $\{true, false\}$, für $\alpha = z$ ist er $P \cup \{0, 1, \mathsf{F}\}$.

Notation 3.12. Ist die Ausführung Π im jeweiligen Kontext eindeutig, schreiben wir auch $\alpha^t\langle V_b\rangle(p)$ statt $\alpha^{\Pi,t}\langle V_b\rangle(p)$ und $\alpha^t\langle V_I System(IS)\rangle$ statt $\alpha^{\Pi,t}\langle V_I System(IS)\rangle$. Ist das I–System IS eindeutig, schreiben wir auch $\alpha^{\Pi,t}$ statt $\alpha^{\Pi,t}\langle V_I System(IS)\rangle$ bzw. α^t statt $\alpha^t\langle V_I System(IS)\rangle$.

Die α-Globalbelegungen, wie sie in Definition 3.11 angegeben sind, werden im nächsten Kapitel benutzt, um auf der formalen Ebene Semantiken für I–Systeme zu definieren. Veränderungen der Belegungen werden durch die Aktionen A1-A13 festgelegt, immer unter Berücksichtigung des übergeordneten Axiomensystems VA1-VA3. Um später eine eindeutige Semantik festlegen zu können, ist es notwendig zu zeigen, dass die Axiome und Aktionen in sich stimmig sind, was die Behandlung der lokalen Variablen angeht.

Satz 3.13 (Widerspruchsfreiheit der Axiome und Aktionen). Sei IS ein I–System und b ein Bereich von IS.
Die Axiome VA1-VA3 (Abschnitt 3.2.1) in Verbindung mit dem Aktionensystem A1-A13 (Abschnitt 3.2.2) sind widerspruchsfrei in dem Sinne, dass ausgeschlossen ist, dass zu einem Zeitpunkt einer Ausführung von $V_I System(IS)$ eine Änderung der Belegung lokaler Variablen bei der Komponente V_b einerseits möglich, andererseits aber auch ausgeschlossen wird.

Beweis. Um einen Widerspruch, wie er in dem Satz beschrieben ist, auszuschließen, reicht es zu zeigen:

a) Die Zuweisungen innerhalb einer Aktion sind eindeutig.

b) Es wird immer nur die Abarbeitung einer eindeutig bestimmbaren Aktion zugelassen.

Zu a). Die Variablenzuweisungen erfolgen innerhalb der Aktionen A1-13. Die Aktionen selbst sind sequentielle Abfolgen von Anweisungen mit der operationalen Semantik prozeduraler Programmiersprachen. Bei alternativen Fällen innerhalb einer Aktion (A4, A5, A13) sind die Fallvoraussetzungen eindeutig. Es können nicht zwei Fälle gleichzeitig eintreten. Folglich kann es innerhalb einer Aktion nicht zu Unstimmigkeiten bei den Zuweisungen kommen.

Zu b). Die Vorbedingungen der Aktionen sind alle unterschiedlich und hängen ab von den Werten der lokalen Variablen bzw. vom Typ der zu bearbeitenden Nachricht. Dadurch ist zu jeder Zeit einer Ausführung von $V_I System(IS)$ immer nur eine Aktion ausführbar. Die Ausführungsreihenfolge ist eindeutig festgelegt bei Beachtung von VA1 und VA2. □

Das Ziel dieses Kapitels ist es, eine Dynamik für I–Systeme zu spezifizieren, in denen Systemereignisse durch aufeinander folgende globale Systemzustände repräsentiert werden. Als globale Systemzustände für I–Systeme sind bisher Cases und als Verfeinerung globale Aktivitätszustände definiert worden (siehe Abschnitt 3.1). Der folgende Satz beschreibt, wie globale Systemzustände aus den Ausführungen des zugeordneten V_ISystems abgeleitet werden können. Das Prinzip liegt in dem Auslesen der lokalen $\_z(\cdot)$-Variablen, zusammengefasst als z-Globalbelegung. Widerspruchsfreiheit ist hierbei durch Satz 3.13 gewährleistet. Als Initialisierung wird ein stabiler globaler Aktivitätszustand vorausgesetzt, wodurch erreicht wird, dass die Vorbedingung der Startaktion A1 für jede Komponente des V_ISystems erfüllt ist.

Satz 3.14 (Globale Aktivitätszustände bei V_ISystemen). Seien IS ein I–System und Π eine Ausführung von $V_I System(IS)$ mit Startzeitpunkt t_0. Vorausgesetzt werde $z^{\Pi,t_0}\langle V_I System(IS)\rangle \in StabGZustand(IS)$.

Dann gilt: $\forall t \geq t_0 : z^{\Pi,t}\langle V_I System(IS)\rangle \in GZustand(IS)$

Beweis. Sei $IS = (P, B, \underline{B}, K, E)$ und Π eine Ausführung von $V_I System(IS)$ mit Startzeitpunkt t_0.

Schreibe z^t für $z^{\Pi,t}\langle V_I System(IS)\rangle$ und $z^t\langle V_b\rangle(p)$ für $z^{\Pi,t}\langle V_b\rangle(p)$, d.h. es wird insbesondere der Bezug auf Π vorausgesetzt.

Seien $t_0, t_1, t_2, \ldots$ die Zeitpunkte der Ausführung mit $t_i < t_{i+1}$, $z^{t_i} \neq z^{t_{i+1}}$ und $\forall t_x,\ t_i \leq t_x < t_{i+1} : z^{t_x} = z^{t_i}$ für $i = 0, 1, 2, \ldots$. Es gelte $z^{t_0} \in StabGZustand(IS)$.

Zu zeigen: Für alle $i = 0, 1, 2, \ldots$ ist z^{t_i} ein globaler Aktivitätszustand von IS, d.h. es gilt gemäß Definition 3.2 und Definition 3.3:

a) Für alle $b \in B$ gilt: $\exists!\, p \in b : z^{t_i}(p) \in \{1, \mathsf{F}\} \cup b \setminus \{p\}$ und $\forall p' \in b \setminus \{p\} : z^{t_i}(p') = \mathsf{0}$.

b) Für alle $b \in B$ gilt: $\forall p_1, p_2 \in b : (z^{t_i}(p_1) = p_2) \Rightarrow (p_1 \neq p_2 \wedge b(p_1) \in AB(IS))$.

c) $\forall p, v \in P$ mit $z^{t_i}(p) \neq \mathsf{0}$ und $z^{t_i}(v) \neq \mathsf{0}$ gilt: $(p, v) \notin K$.

Zu a) und b). Sei $b \in B$ beliebig. Es reicht zu zeigen (beachte Definition 3.11):

a′) Für alle $i = 0, 1, 2, \ldots$ gilt: $\exists!\, p \in b : z^{t_i}\langle V_b\rangle(p) \in \{1, \mathsf{F}\} \cup b \setminus \{p\}$ und $\forall p' \in b \setminus \{p\} : z^{t_i}\langle V_b\rangle(p') = \mathsf{0}$.

b′) $\forall p_1, p_2 \in b : (z^{t_i}\langle V_b\rangle(p_1) = p_2) \Rightarrow (p_1 \neq p_2 \wedge b(p_1) \in AB(IS))$.

Beweis durch Induktion über i:

IA: z^{t_0} ist ein (stabiler) globaler Aktivitätszustand nach Voraussetzung. Damit ist $z^{t_0}\langle b\rangle$ ein (stabiler) lokaler Aktivitätszustand von b und es gelten a′) und b′) für $i = 0$ direkt aufgrund der Definition 3.2.

IV: a′) und b′) gelten für $i > 0$.

IS: Fallunterscheidung:
Fall 1). Für alle $p \in b$ gilt $z^{t_{i+1}}\langle V_b\rangle(p) = z^{t_i}\langle V_b\rangle(p)$
$\Rightarrow$ Bei V_b findet beim Übergang von i nach $i+1$ keine Veränderung von $\_z(\cdot)$ statt. a′) und b′) gelten für $i+1$ nach IV.

Fall 2). Es existiert ein $p \in b$ mit $z^{t_{i+1}}\langle V_b\rangle(p) \neq z^{t_i}\langle V_b\rangle(p)$
$\Rightarrow$ Eine Veränderung von $\_z(p)$ beim Übergang von i nach $i+1$ erfolgt durch eine der Aktionen A1-A13. Die Art der möglichen Veränderungen wird im Folgenden untersucht anhand der Aktionsbeschreibungen.

A1, A2, A6, A7, A8, A9, A10, A11, A12: Es erfolgt entweder keine direkte Veränderung von $\_z(\cdot)$ oder nur durch einen Aufruf von A13. Diese Aktionen brauchen deshalb nicht betrachtet zu werden. A13 wird gesondert betrachtet.

A3: Gemäß der Vorbedingung von A3 gilt $z^{t_i}\langle V_b\rangle(p) = 1$, wobei b als autonom vorausgesetzt wird, d.h. $b(p) \in AB(IS)$. Nach IV ist p zum Ausführungszeitpunkt t_i die einzige Phase bei V_b mit $\_z(p) \neq 0$. p bleibt während der ganzen Aktion die einzige Phase mit $\_z(p) \neq 0$. Zwei Fälle können auftreten: Erstens, es findet keine $\_z(\cdot)$-Zuweisung innerhalb von A3 statt, wodurch a′) und b′) automatisch weiterhin gelten. Zweitens, $\_z(p)$ wechselt von 1 auf q mit $q \in b$, $q \neq p$. Folglich gelten a′) und b′) für $i+1$.

A4: Gemäß der Vorbedingung von A4 gilt $z^{t_i}\langle V_b\rangle(p) = q$ mit $q \neq p$ und $b \in AB(IS)$ nach IV. Nach IV ist p zum Ausführungszeitpunkt t_i die einzige Phase bei V_b mit $\_z(p) \neq 0$. Eine Veränderung von $\_z(p)$ innerhalb von A4 tritt nur ein im Fall der Ausführung der Phasentransition $p \to q$. Durch die Ausführung wird q zum Zeitpunkt t_{i+1} einzige Phase mit $\_z(q) \neq 0$. $\_z(q)$ wechselt von $\mathsf{0}$ auf $\mathsf{1}$ und $\_z(p)$ wechselt *gleichzeitig* von q nach $\mathsf{0}$. Folglich gelten a′) und b′) für $i+1$.

A5: Gemäß der Vorbedingung von A5 gilt $z^{t_i}\langle V_b\rangle(p) = \mathsf{F}$. Nach IV ist p zum Ausführungszeitpunkt t_i die einzige Phase bei V_b mit $\_z(p) \neq 0$. Eine Veränderung von $\_z(p)$ innerhalb von A5 tritt nur ein im Fall der Ausführung der Phasentransition $p \to q$. Dabei gilt $q \in b \setminus \{p\}$ gemäß der Auswahl der Folgephase. Durch die Ausführung wird q zum Zeitpunkt t_{i+1} einzige Phase mit $\_z(q) \neq 0$. $\_z(q)$ wechselt von $\mathsf{0}$ auf $\mathsf{1}$ und $\_z(p)$ wechselt *gleichzeitig* von q nach $\mathsf{0}$. Folglich gelten a′) und b′) für $i+1$.

A13: Gemäß der Vorbedingung von A13 gilt $z^{t_i}\langle V_b\rangle(p) \neq 0$. Nach IV ist p zum Ausführungszeitpunkt t_i die einzige Phase bei V_b mit $\_z(p) \neq 0$. Tritt, bezogen auf die Aktionsbeschreibung von A13, i) ein, findet ein Wechsel bei $\_z(p)$ von F nach 1 statt, bei ii) ein Wechsel von $q \in b$ nach 1, bei iii) ein Wechsel von $q \in b$ nach F und bei iv) ein Wechsel von 1 nach F. In den Fällen i)-iv) gilt somit, dass p direkt nach der Wertezuweisung, d.h. zum Zeitpunkt t_{i+1}, weiterhin die einzige Phase mit $\_z(q) \neq 0$ ist. Somit gilt hier a$'$) für $i+1$ und auch b$'$), da nur F und 1 als Werte für $z^{t_{i+1}}\langle V_b\rangle(p)$ in Frage kommen. Im Fall A13.v) bleibt die Variablenbelegung gleich, und die Aussagen a$'$) und b$'$) für $i+1$ gelten nach IV.

Es sind damit alle Möglichkeiten überprüft und die Gültigkeit von a$'$) und b$'$) im Einzelnen gezeigt worden. Der Induktionsschluss für $i+1$ ist wahr und folglich gelten a$'$) und b$'$) für $i+1$ im Ganzen und damit auch a) und b).

Zu c). Beweis durch Induktion über i.

IA: Nach Satzvoraussetzung gilt $z^{t_0} \in StabGZustand(IS)$. Da $StabGZustand(IS) \subseteq GZustand(IS)$ folgt c) für $i = 0$ aus Definition 3.3.2.

IV: c) gilt für $i > 0$.

IS: Beweis durch Widerspruch.

Annahme: Es existieren $q, w \in P$ mit $z^{t_{i+1}}(q) \neq 0$ und $z^{t_i}(w) \neq 0$ und $(q, w) \in K$.

Unter Beachtung der IV für i gilt dann:
($\forall p, v \in P$ mit $z^{t_i}(p) \neq 0$ und $z^{t_i}(v) \neq 0$ gilt: $(p, v) \notin K$) und ($\exists q, w \in P$ mit $z^{t_{i+1}}(q) \neq 0$, $z^{t_{i+1}}(w) \neq 0$ und $(q, w) \in K$).

Für z^{t_i} und $z^{t_{i+1}}$ gilt bereits die Aussage a) vom ersten Teil des Beweises. Somit ist jede Komponente von $V_ISystem(IS)$ in immer genau einer Phase, und es muss eine der beiden folgenden Aussagen (i oder ii) gelten.

i) $\exists b, b' \in B$, $b \neq b'$ und $\exists p, q \in b$, $v \in b'$ mit $(q, v) \in K$ und es gilt: $z^{t_i}\langle V_b\rangle(p) \neq 0$, $z^{t_i}\langle V_b\rangle(q) = 0$, $z^{t_i}\langle V_{b'}\rangle(v) \neq 0$, $z^{t_{i+1}}\langle V_b\rangle(p) = 0$, $z^{t_{i+1}}\langle V_b\rangle(q) \neq 0$, $z^{t_{i+1}}\langle V_{b'}\rangle(v) \neq 0$.

ii) $\exists b, b' \in B$, $b \neq b'$ und $\exists p, q \in b$, $v, w \in b'$ mit $(q, w) \in K$ und es gilt: $z^{t_i}\langle V_b\rangle(p) \neq 0$, $z^{t_i}\langle V_b\rangle(q) = 0$, $z^{t_i}\langle V_{b'}\rangle(v) \neq 0$, $z^{t_i}\langle V_{b'}\rangle(w) = 0$, $z^{t_{i+1}}\langle V_b\rangle(p) = 0$, $z^{t_{i+1}}\langle V_b\rangle(q) \neq 0$, $z^{t_{i+1}}\langle V_{b'}\rangle(v) = 0$, $z^{t_{i+1}}\langle V_{b'}\rangle(w) \neq 0$.

Bei i) folgt : $\exists b, b' \in B$, $b \neq b'$ und $\exists p, q \in b$, $v \in b'$ mit $(q, v) \in K$, und es gilt:
Zum Zeitpunkt t_{i+1} findet eine Phasentransition $p \to q$ bei V_b statt, und gleichzeitig ist $V_{b'}$ in v.
Bei ii) folgt: $\exists b, b' \in B$, $b \neq b'$ und $\exists p, q \in b$, $v, w \in b'$ mit $(q, w) \in K$, und es gilt:
Zum Zeitpunkt t_{i+1} findet eine Phasentransition $p \to q$ bei V_b statt und gleichzeitig die Phasentransition $v \to w$ bei $V_{b'}$.

Es sei noch einmal darauf hingewiesen, dass Phasentransitionen zeitlich atomar erfolgen. Zeitpunkte und der Begriff der Gleichzeitigkeit beziehen sich bei den obigen und auch folgenden Formulierungen immer auf die Globalzeit. Es folgen nun zwei allgemein gültige Beobachtungen über das Verhalten von $V_ISystem(IS)$, auf die beim Abschluss des Beweises zurück gegriffen wird.

Beobachtung 1:

Findet bei einer Komponente V_b von $V_ISystem(IS)$ eine Phasentransition $p \to q$ statt (Aktion A4 oder A5), gilt zur gleichen Globalzeit bei allen Nachbarkomponenten $\_mark(p) = true$ und $\_mark(q) = true$.

Präzisierung: Vor Ausführung einer Phasentransition $p \to q$ bei V_b wird, gemäß den Aktionsbeschreibungen von A4 und A5, eine Nachricht $reqmark(M)$, mit $p, q \in M$, an alle Nachbarkomponenten von V_b geschickt und jeweils auf die Rückmeldung $ackmark$ gewartet. Die Nachbarkomponenten reagieren auf $reqmark(M)$, indem sie bei sich unter anderem $\_mark(p)$ und $\_mark(q)$ auf $true$ setzen und erst dann $ackmark$ als Bestätigung zurückschicken (A8). Die Nachbarkomponenten setzen $\_mark(p)$ und $\_mark(q)$ erst wieder auf $false$ nach Erhalt und Bearbeitung der $done(p \to q)$ Nachricht von V_b (A10), also nach Ausführung der Phasentransition. $\square$

Beobachtung 2:

Nach Abschluss der Initialisierung (A1) gilt bei jeder Komponente V_b von $V_I System(IS)$ zu jedem Zeitpunkt einer Ausführung: Ist v eine Nachbarphase von b und $\_z(v) \neq 0$ bei $V_{b(v)}$, dann gilt zur gleichen Globalzeit $\_in(v) = true$ oder $\_mark(v) = true$ bei V_b.
Gilt bei einer Komponente V_b $\_in(v) = true$ und $\_mark(v) = false$, dann gilt zur gleichen Globalzeit $\_z(v) \neq 0$ bei der Nachbarkomponente $V_{b(v)}$, d.h. $V_{b(v)}$ ist in v.
Gilt bei V_b $\_in(v) = true$ und gleichzeitig $\_mark(v) = true$, dann ist entweder $V_{b(v)}$ in v oder $V_{b(v)}$ ist in einer Phase $v' \in b(v)$, für die bei V_b $\_mark(v') = true$ gilt.

Präzisierung: $\_in(v) := true$ wird bei V_b gesetzt einerseits am Ende der Initialisierung (A1) nach Empfang der Nachricht $ackinit(v)$ oder andererseits als Reaktion auf den Empfang einer Nachricht $done(x \to v)$ mit $x \in b(v)$ (A10). Im ersten Fall ist das Setzen das Ergebnis der Nachfrage bei den Nachbarkomponenten, mittels $reqinit$, nach den Phasen v, in denen sich die Komponenten aktuell befinden, d.h. für die $\_z(v) \neq 0$ gilt. Die Nachbarkomponenten antworten durch $ackinit(v)$-Nachrichten. Da alle Komponenten ihre Aktivität mit A1 beginnen (Axiom VA1), gilt während der Initialisierung $not\ \_mark(\cdot)$, eine Vorbedingung von A1. Im zweiten Fall bedeutet der Empfang der Nachricht $done(x \to v)$, dass die Nachbarkomponente $V_{b(v)}$ die Phasentransition $x \to v$ durchgeführt hat (mittels A4 oder A5), wodurch v die aktuelle Phase von $V_{b(v)}$ wurde, d.h. dass dort $\_z(v) \neq 0$ gilt. V_b reagiert mit Setzen von $\_in(v) := true$ und $\_in(x) := false$ (A10). $\_mark(v) := true$ wird bei V_b gesetzt als Reaktion auf die Nachricht $reqmark(M)$ mit $v \in M$ (A8), wodurch festgehalten wird, dass die Komponente $V_{b(v)}$ eine Phasentransition $x \to v$ oder $v \to x$ durchführen möchte und dass die Aktivität von v, aus der Sicht von V_b, bis zum Erhalt weiterer Nachrichten ungewiss ist. $\_mark(v)$ wird auf $false$ zurückgesetzt, sobald V_b Nachricht darüber hat, dass $V_{b(v)}$ die Phasentransition erfolgreich durchgeführt (A10) oder die Aktion abgebrochen hat (A9). Da die Nachrichtenübertragung als sicher vorausgesetzt wird (Axiom VA2), ist gewährleistet, dass eine weitere Phasentransition $v \to y$ bzw. $x \to y$ bei $V_{b(v)}$ erst ausführt werden kann, nachdem V_b Kenntnis über die Ausführung oder den Abbruch der vorherigen hat. Genauer, eine $reqmark(\cdot)$-Nachricht von $V_{b(v)}$ nach V_b kann eine vorher abgeschickte $done(\cdot)$- oder $break$-Nachricht von $V_{b(v)}$ nach V_b nicht überholen. Das Setzen der $\_mark(\cdot)$- und $\_in(\cdot)$-Variablen verläuft somit in der korrekten Reihenfolge. □

Fortsetzung des Beweises von c):

Phasentransitionen erfolgen durch die Aktionen A4 oder A5. Damit eine Phasentransition $p \to q$ bei V_b durchgeführt werden kann, müssen bestimmte Vorbedingungen von Variablenbelegungen bei V_b erfüllt sein, die durch die Aktionsbeschreibungen festgelegt sind. Insbesondere muss not $\_mark(x)$ gelten für jede K-Nachbarphase x von q, sowie $not\ \_k(q)$.

Für den Fall der Gültigkeit von i) ergibt sich demnach:
Zum Zeitpunkt t_{i+1} gilt bei V_b $not\ \_mark(v)$ und $not\ \_k(q)$.
$\Rightarrow$ {Definition von $\_k(q)$ (siehe Abschnitt 3.2.2)}
Zum Zeitpunkt t_{i+1} gilt bei V_b $not\ \_mark(v)$ und $not\ \_in(v)$.

Das ist ein Widerspruch zur Beobachtung 2, da v während der Phasentransition eine Phase mit $\_z(v) \neq 0$ (bei $V_{b(v)}$) ist und folglich bei V_b $\_in(v)$ oder $\_mark(v)$ gilt.

Für den Fall der Gültigkeit von ii) ergibt sich:
Zum Zeitpunkt t_{i+1} gilt bei V_b $not\ \_mark(w)$ und $not\ \_k(q)$.
$\Rightarrow$ {Definition von $\_k(q)$}
Zum Zeitpunkt t_{i+1} gilt bei V_b $not\ \_mark(w)$ und $not\ \_in(w)$.

Das ist ein Widerspruch zur Beobachtung 1, da wegen der Phasentransition bei V_b $\_mark(w)$ gilt.

Da für beide Fälle, die auftreten können, ein Widerspruch gezeigt wurde, ist die Annahme falsch. Folglich gilt der Induktionsschluss für $i + 1$ und damit auch c). □

Betrachtet man alle möglichen Ausführungen eines $\mathrm{V_I}$Systems und die Menge aller aus diesen Ausführungen ableitbaren globalen Aktivitätszustände, so stimmt diese Menge in der Regel nicht mit der Menge aller für das zugrunde liegende I–System definierten globalen Aktivitätszustände überein. Sie besitzt meistens weniger Elemente. Die aus den Ausführungen ableitbaren globalen Aktivitätszustände werden als *relevant* bezeichnet. Sie sind vergleichbar mit erreichbaren Markierungen bei Petri-Netzen [80].

Definition 3.15 (Relevante globale Aktivitätszustände). Ergibt sich ein globaler Aktivitätszustand eines I–Systems IS als z-Globalbelegung während einer Ausführung von $V_I System(IS)$, so nennt man ihn *relevant*. $RelGZustand(IS)$ bezeichnet die Menge aller relevanten globalen Aktivitätszustände von IS, d.h. $RelGZustand(IS) = \{z \in GZustand(IS) \mid$ Es existieren eine Ausführung Π von $V_I System(IS)$ und ein Zeitpunkt t in Π mit $z = z^{\Pi,t}\langle V_I System(IS)\rangle$ $\}$. □

Betrachtet man das I–System IS_1 aus Beispiel 2.2, so gilt dort z.B. (ohne Beweis): $[p_1{<}p_2{>}, q_3{<}1{>}, v_2{<}\mathsf{F}{>}] \in RelGZustand(IS_1)$ aber $[p_1{<}1{>}, q_2{<}\mathsf{F}{>}, v_2{<}1{>}] \notin RelGZustand(IS_1)$. In beiden Fällen handelt es sich um globale Aktivitätszustände von IS_1, d.h. um Elemente aus $GZustand(IS)$.

Da sich Cases aus globalen Aktivitätszuständen über die Funktion $zc(\cdot)$ ableiten lassen, kann Satz 3.14 auf einfache Weise so umformuliert werden, dass man eine Aussage darüber erhält, wie Cases eines I–Systems aus den Ausführungen des zugeordneten $\mathrm{V_I}$Systems abgeleitet werden können.

Korollar 3.16 (Cases bei $\mathbf{V_I}$Systemen). Seien IS ein I–System und Π eine Ausführung von $V_I System(IS)$ mit Startzeitpunkt t_0. Vorausgesetzt werde $z^{\Pi,t_0}\langle V_I System(IS)\rangle \in StabGZustand(IS)$.

Dann gilt: $\forall t \geq t_0 : zc(z^{\Pi,t}\langle V_I System(IS)\rangle) \in Case(IS)$

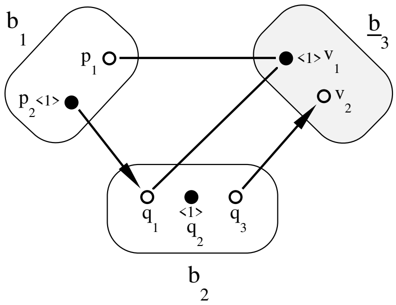
Beweis. Der Satz folgt direkt aus Satz 3.14 und der Wohldefiniertheit der Funktion $zc(\cdot)$ aus Definition 3.6. □

Die Ausführungen eines $\mathrm{V_I}$Systems eignen sich nach Satz 3.14 und Korollar 3.16 dazu, über die z-Globalbelegungen globale Aktivitätszustände bzw. Cases des zugrunde liegenden I–Systems zu bestimmen. Da sich die z-Globalbelegung während einer Ausführung in der Regel ändert, repräsentieren diese Änderungen, die folglich von einem globalen Aktivitätszustand (bzw. Case) zu einem globalen Folge-Aktivitätszustand (bzw. Folge-Case) führen, Ereignisse in dem I–System. Die über die Ausführungen herleitbaren Folgen von nacheinander unterschiedlichen Aktivitätszuständen (bzw. Cases) werden im nächsten Kapitel dazu verwendet, auf der formalen Ebene Semantiken für I–Systeme zu definieren.

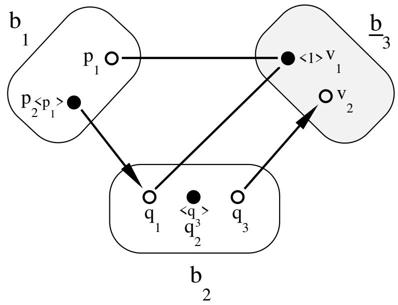
3.3 Beispiel

Die Festlegungen zur Dynamik von I–Systemen sollen anhand von I–System IS_1 demonstriert werden. Betrachtet wird hierzu eine Ausführung Π von $V_I System(IS_1)$. Zum Start-Globalzeitpunkt t_0 gelte $\_z(p_2) = 1$ bei V_{b_1}, $\_z(q_2) = 1$ bei V_{b_2} und $\_z(v_1) = 1$ bei $V_{\underline{b}_3}$. Die z-Globalbelegung $z^{\Pi,t_0}\langle V_I System(IS_1)\rangle$ liefert somit einen globalen Start-Aktivitätszustand z_0, so wie er in der folgenden Abbildung (links) graphisch dargestellt ist.

z_0:



z_1:

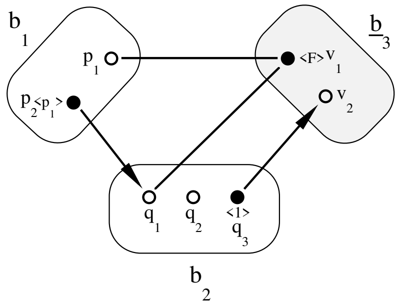


Es werde nun angenommen, dass V_{b_1} und V_{b_2} nach Abarbeitung der Initialisierungsaktionen A1 und A2 (aus Abschnitt 3.2.2) gleichzeitig zu einem Globalzeitpunkt $t_1 > t_0$, aber unabhängig voneinander, eine autonome Entscheidung treffen, eine Phasentransition durchzuführen (jeweils

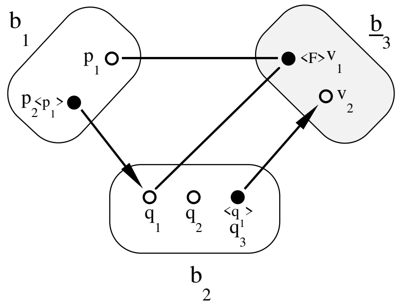
Aktion A3). Angenommen, V_{b_1} möchte nach p_1 und V_{b_2} möchte nach q_3 wechseln, dann ergeben sich die Variablenbelegungen $\_z(p_2) = p_1$ bei V_{b_1} und $\_z(q_2) = q_3$ bei V_{b_2}. $V_{\underline{b}_3}$ kann eine Entscheidung zum Phasenwechsel nicht treffen, da es sich bei $\underline{b}_3$ um einen trägen Bereich handelt. Als neuer globaler Aktivitätszustand, an dem die getroffenen Entscheidungen ablesbar sind, ergibt sich $z_1 := z^{\Pi,t_1}\langle V_I System(IS_1)\rangle$ (obige Abbildung, rechts).

Als nächstes führen V_{b_1} und V_{b_2} jeweils die Aktion A4 aus. Innerhalb dieser Aktion findet bei V_{b_2} zu einem Globalzeitpunkt $t_2 > t_1$ die Phasentransition $q_2 \to q_3$ statt. Es liegen keine Einflüsse vor, die dies verhindern. V_{b_2} kann keine Phasentransition nach p_1 ausführen, da $V_{\underline{b}_3}$ in v_1 ist und aufgrund des wechselseitigen Ausschlusses von p_1 und v_1 p_1 nicht frei ist. Deshalb schickt V_{b_2} mittels Aktion A6 eine $solicit(\cdot)$-Nachricht an $V_{\underline{b}_3}$, um hierdurch einen Einfluss auf $V_{\underline{b}_3}$ auszuüben, die Phase v_1 zu verlassen. $V_{\underline{b}_3}$ empfängt die Nachricht (Aktion A11) und geht durch Setzen von $\_z(v_1) := \mathsf{F}$, in diesem Beispiel zum Zeitpunkt t_2, in der anschließenden Update-Aktion A13 in einen instabilen Zustand über. Es ergibt sich ein globaler Aktivitätszustand $z_2 := z^{\Pi,t_2}\langle V_I System(IS_1)\rangle$ (folgende Abbildung, links).

z_2:



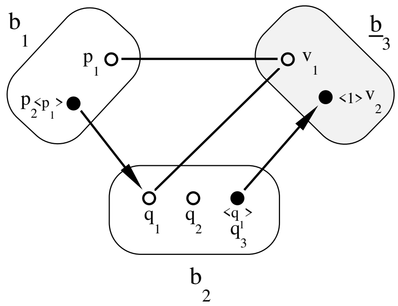
z_3:



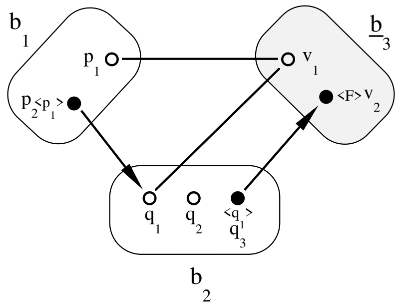
Nach der Phasentransition bei V_{b_2} treffe V_{b_2} zu einem Globalzeitpunkt $t_3 > t_2$ erneut eine autonome Entscheidung, die aktuelle Phase zu wechseln, diesmal nach q_1. Der globale Aktivitätszustand $z_3 := z^{\Pi,t_3}\langle V_I System(IS_1)\rangle$ (obige Abbildung, rechts) dokumentiert diesen Vorgang in Form einer Phasenqualität q_1 bei q_3.

Die Instabilität von $V_{\underline{b}_3}$ (d.h. die Phasenqualität F bei v_1) ist der Grund dafür, dass $V_{\underline{b}_3}$ die Aktion A5 ausführt, woraus dann schließlich eine Phasentransition $v_1 \to v_2$ in die einzige freie Phase v_2 resultiert. Dies geschähe zu einem Globalzeitpunkt $t_4 > t_3$. Direkt nach der Phasentransition erhält man $z_4 := z^{\Pi,t_4}\langle V_I System(IS_1)\rangle$ (folgende Abbildung, links). Es werde angenommen, dass in der Zwischenzeit bei V_{b_2} die anstehende Aktion A4 noch nicht angegangen wurde. Diese Annahme ist erlaubt, da die lokalen Ausführungszeiten nicht bekannt sind (gemäß Verhaltensaxiom VA3 aus Abschnitt 3.2.1).

z_4:



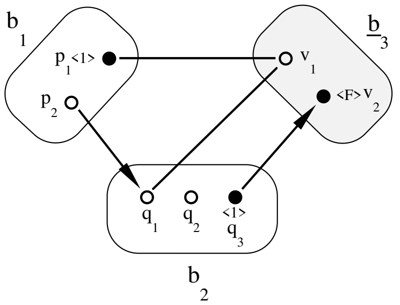
z_5:



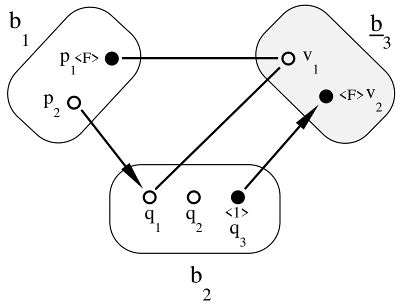
Die Aktionsbeschreibung von A5 beinhaltet einen Update-Aufruf (A13) nach Durchführung einer Phasentransition. Dieses Update bewirkt bei $V_{\underline{b}_3}$ das Setzen der $\_z(v_2)$-Variablen auf F zu einem Globalzeitpunkt $t_5 > t_4$, weil V_{b_2} in q_3 ist und als Folge davon bei $V_{\underline{b}_3}$ $e_{in}(v_2) = true$ vorliegt. Anschaulich übt V_{b_2} über die Erregungsbeziehung (q_3, v_2) einen Zwang auf $V_{\underline{b}_3}$ aus, die Phase v_2 zu verlassen. Dies ist die Ursache einer Instabilität von $V_{\underline{b}_3}$, so wie es bei $z_5 := z^{\Pi,t_5}\langle V_I System(IS_1)\rangle$ (obige Abbildung, rechts) abzulesen ist.

Ebenfalls beinhaltet die Aktionsbeschreibung von A5, dass Nachbarbereiche über eine durchgeführte Phasentransition mittels einer $done(\cdot)$-Nachricht informiert werden. Auf diese Weise erhält V_{b_2} eine $done(v_1 \rightarrow v_2)$-Nachricht von $V_{\underline{b}_3}$ und führt schließlich Aktion A10 mit abschließendem Update-Aufruf (A13) aus. Das Update führt zu einem Globalzeitpunkt $t_6 > t_5$ bei V_{b_2} zu $\_z(q_3) := 1$ (da dort $e_{out}(q_3) = true$ gilt), oder anschaulich ausgedrückt, V_{b_2} wird stabil in v_2, da q_3 eine erregende Phase ist und die Erregung so lange aufrecht erhalten wird, wie $V_{\underline{b}_3}$ v_2 nicht verlassen hat. Parallel zu den Aktivitäten von V_{b_2} und $V_{\underline{b}_3}$ führt V_{b_1} zum Zeitpunkt t_6 die Phasentransition $p_2 \rightarrow p_1$ aus (Aktion A4). Durch das durch V_{b_1} erzwungene Verlassen von v_1 wurde p_1 eine freie Phase und der Phasentransition steht nichts mehr entgegen. Als neuer globaler Aktivitätszustand ergibt sich $z_6 := z^{\Pi,t_6}\langle V_I System(IS_1)\rangle$ (folgende Abbildung, links).

z_6:



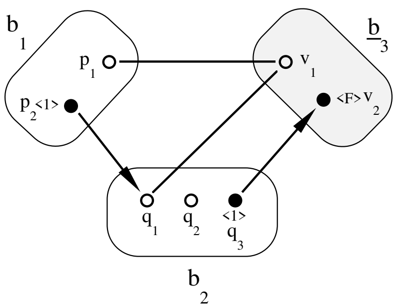
z_7:



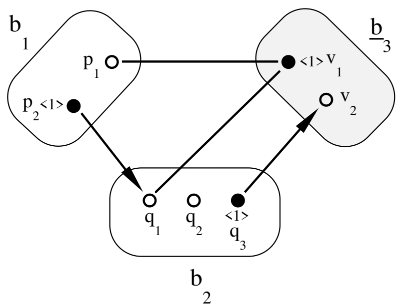
$V_{\underline{b}_3}$ wird gezwungenermaßen als nächstes die Aktion A5 ausführen, mit dem Ziel, v_2 zu verlassen. Die einzige potentielle Folgephase ist v_1, die allerdings aufgrund des wechselseitigen Ausschlusses mit p_1 und der Tatsache, dass sich V_{b_1} bereits in p_1 befindet, nicht frei ist, d.h. als lokale Variablenbelegung gilt bei $V_{\underline{b}_3}$ $\_k(v_1) = true$. Eine Phasentransition $v_2 \rightarrow v_1$ kann bei $V_{\underline{b}_3}$ somit nicht eintreten. Stattdessen erfolgt ein Solicitation-Aufruf bzgl. $\{v_1\}$ (A6) und $V_{\underline{b}_3}$ schickt eine $solicit(\cdot)$-Nachricht an V_{b_1}, um dort einen Phasenwechsel zu erzwingen. Als Reaktion auf die Nachricht führt V_{b_1} die Aktion A11 aus, und der abschließende Update-Aufruf (A13) bewirkt eine Instabilität von V_{b_1} in p_1. Der Globalzeitpunkt sei $t_7 > t_6$. Am globalen Aktivitätszustand $z_7 := z^{\Pi,t_7}\langle V_I System(IS_1)\rangle$ (obige Abbildung, rechts) ist diese Instabilität an der Phasenqualität F im Bereich b_1 erkennbar.
Zurückblickend liegt die Ursache der Instabilität von V_{b_1} in p_1 in der Erregung von v_2 durch q_3. Der Einfluss von V_{b_2} auf $V_{\underline{b}_3}$, um dort eine Phasentransition zu erzwingen, wird nach V_{b_1} propagiert und erwirkt dort die Instabilität. Solange $V_{\underline{b}_3}$ in v_2 ist, kann V_{b_2} keine Entscheidung treffen, die Phase q_3 zu verlassen (siehe Vorbedingungen von A3), um dadurch die Einflusspropagierung im Ursprung zu unterbinden.

Die Instabilität von V_{b_1} in p_1 hat zur Folge, dass V_{b_1} die Aktion A5 ausführt, innerhalb derer zu einem Globalzeitpunkt $t_8 > t_7$ eine Phasentransition $p_1 \rightarrow p_2$ stattfindet, da p_2 frei und die einzige mögliche Folgephase ist. Als neuen globalen Aktivitätszustand erhält man $z_8 := z^{\Pi,t_8}\langle V_I System(IS_1)\rangle$ (folgende Abbildung, links).

z_8:



z_9:



Nachdem V_{b_1} die Phase p_1 verlassen hat, ist v_1 frei und $V_{\underline{b}_3}$ wird im Rahmen von Aktion A5 eine Phasentransition $v_2 \rightarrow v_1$ ausführen. Dies geschähe zu einem Globalzeitpunkt $t_9 > t_8$. Es ergibt sich ein stabiler globaler Aktivitätszustand $z_9 := z^{\Pi,t_9}\langle V_I System(IS_1)\rangle$ (obige Abbildung, rechts).

Wenn V_{b_1} oder V_{b_2} nicht erneut eine autonome Entscheidung treffen, die aktuelle Phase zu wechseln, dann wird sich die letzte z-Globalbelegung nicht mehr verändern, und mit z_9 ist ein Ende der Folge von globalen Aktivitätszuständen, die sich während der Ausführung Π ergeben, erreicht. Zum Treffen einer autonomen Entscheidung besteht für die autonomen Komponenten kein Zwang (siehe Aktionsbeschreibung von A3). Dieses Beispiel zeigt den Fall, dass keine Entscheidung mehr getroffen wird und dass keine Ereignisse im Gesamtsystem mehr zu verzeichnen sind. □

Notation 3.17. Eine Ausführung Π von $V_I System(IS)$ für ein I–System IS wird als Sequenz der auftretenden z-Globalbelegungen notiert, wobei die z-Globalbelegungen in der Notation 3.4 angegeben werden. Pfeile verdeutlichen die Abfolge der z-Globalbelegungen, und über den Pfeilen ist jeweils angegeben, welche Aktion in welcher Komponente eine Veränderung bewirkt. Z.B. bedeutet b_1.A3, dass die Komponente V_{b_1} die Aktion A3 ausführt. Die Update Aktion A13 wird der jeweils aufrufenden Aktion zugeordnet.

Beispiel: Die Ausführung Π aus dem vorangegangenen Beispiel lässt sich wie folgt aufschreiben:

$[p_2{<}1{>}, q_2{<}1{>}, v_1{<}1{>}] \xrightarrow{b_1.\mathrm{A3}, b_2.\mathrm{A3}} [p_2{<}p_1{>}, q_2{<}q_3{>}, v_1{<}1{>}] \xrightarrow{b_2.\mathrm{A4}, \underline{b}_3.\mathrm{A11}}$

$[p_2{<}p_1{>}, q_3{<}1{>}, v_1{<}\mathsf{F}{>}] \xrightarrow{b_2.\mathrm{A3}} [p_2{<}p_1{>}, q_3{<}q_1{>}, v_1{<}\mathsf{F}{>}] \xrightarrow{\underline{b}_3.\mathrm{A5}} [p_2{<}p_1{>}, q_3{<}q_1{>}, v_2{<}1{>}]$

$\xrightarrow{\underline{b}_3.\mathrm{A5}} [p_2{<}p_1{>}, q_3{<}q_1{>}, v_2{<}\mathsf{F}{>}] \xrightarrow{b_1.\mathrm{A4}, b_2.\mathrm{A10}} [p_1{<}1{>}, q_3{<}1{>}, v_2{<}\mathsf{F}{>}] \xrightarrow{b_1.\mathrm{A11}}$

$[p_1{<}\mathsf{F}{>}, q_3{<}1{>}, v_2{<}\mathsf{F}{>}] \xrightarrow{b_1.\mathrm{A5}} [p_2{<}1{>}, q_3{<}1{>}, v_2{<}\mathsf{F}{>}] \xrightarrow{\underline{b}_3.\mathrm{A5}} [p_2{<}1{>}, q_3{<}1{>}, v_1{<}1{>}]$

Kapitel 4

Trace-Semantiken

Nach der Spezifikation der formalen Struktur (Kapitel 2) sowie von lokalen/globalen Systemzuständen und dynamischen Abläufen (Kapitel 3) sind die Voraussetzungen gegeben, geeignete Semantiken für I–Systeme formulieren zu können. Die Semantik eines I–Systems bildet die Basis für eine Systemanalyse oder Systemverifikation auf der formalen Ebene, d.h. das Finden bzw. die Überprüfung von Systemeigenschaften der modellierten Anwendung im Rahmen der formalen Repräsentation. Die Wahl einer Semantik orientiert sich an folgenden Kriterien:

- Die Semantik soll ausreichend Ansatzpunkte für eine Systemanalyse/-verifikation bieten.
- Es sollen in Kapitel 1.1 genannte Schlüsselphänomene über die Semantik nachvollziehbar sein.
- Die Semantik soll einfach aus der formalen Struktur und der Dynamik ableitbar sein.

Es gibt eine Vielzahl von Literaturquellen, die sich mit Semantiken von verteilten, nebenläufigen oder sequentiellen Prozessen beschäftigen [3, 4, 6, 38, 58, 69]. Ein bewährter Ansatz besteht darin, Zustands- oder Ereignisfolgen (Traces), die die Prozessaktivitäten wiedergeben, zur Bildung von Semantiken zu verwenden. Solch ein Ansatz bietet sich auch für I–Systeme an, da sich aus den Ausführungen der zugeordneten $\mathrm{V_I}$Systeme (auf der algorithmischen Ebene) alle relevanten Zustandsfolgen (auf der formalen Ebene) direkt ableiten lassen. Wenn es erst einmal gelungen ist, auf Traces basierende Semantiken (Trace-Semantiken) für I–Systeme zu entwickeln, dann lässt sich bei der Weiterentwicklung der Theorie der I–Systeme auf die umfangreiche Theorie der Traces (siehe [27]) zurückgreifen. Neben dem Begriff „Traces“ werden in der Literatur auch die Begriffe „Runs“ [75] und „Streams“ [15] für Zustands- oder Ereignisfolgen verwendet.

Anhand von Trace-Semantiken lassen sich „klassische“ Systemeigenschaften wie Erreichbarkeit, Sicherheit, Lebendigkeit und Fairness überprüfen. Erreichbarkeit bedeutet, dass eine bestimmte Systemsituation eintreten kann; Sicherheit bedeutet, dass unter bestimmten Bedingungen ein Ereignis niemals auftritt; Lebendigkeit bedeutet, dass unter bestimmten Bedingungen ein Ereignis tatsächlich eintreten wird; Fairness bedeutet, dass unter bestimmten Bedingungen ein Ereignis unendlich oft auftreten wird. Präzisierungen finden sich in [9, 62]. Orientiert man sich an Methoden des Model-Checkings (vgl. [9, 22]) dann werden solche Systemeigenschaften in Form von temporallogischen Formeln spezifiziert. Die Gültigkeit von temporallogischen Formeln lässt sich über Trace-Mengen definieren und überprüfen. Somit können Trace-Semantiken als Schnittstelle zu existierenden Verifikationstechniken eingesetzt werden.
Neben den klassischen Systemeigenschaften ist man bei I–Systemen interessiert an einer Analyse der modellierten Einflussstrukturen, d.h. der Existenz und dem Fehlen von Einflüssen, Ursachen, Wirkungen, Fortpflanzungen. Erkenntnisse hieraus fließen ein in einen inkrementellen Entwurfsprozess (siehe Kapitel 10.4). Trace-Semantiken können für solch eine Analyse verwendet werden, sofern man die Elemente der Traces geeignet wählt. Die Wahl der Elemente bestimmt das Informationspotential (bzgl. der Schlüsselphänomene aus Kapitel 1.1) der Traces und damit den Umfang der Analysemöglichkeiten. Da in der Regel ein Mehr an Informationspotential bei einer Semantik einen Anstieg der Darstellungskomplexität mit sich bringt, sollten alternative

Semantiken unterschiedlicher Ausdruckskraft und Darstellungskomplexität angeboten werden, die je nach Analyseziel eingesetzt werden können.

Nach einigen allgemeinen Festlegungen werden mit dem *Verhalten*, der *Casetrace-Semantik* und der *Erweiterten Casetrace-Semantik* in diesem Kapitel drei Trace-Semantiken für I–Systeme vorgestellt und deren Unterschiede und Abhängigkeiten deutlich gemacht.

4.1 Allgemeine Festlegungen

Im Folgenden werden kurz Begriffe und Notationen aufgeführt, die in der weiteren Arbeit verwendet werden. Da es sich dabei um gängige Formalismen aus der theoretischen Informatik handelt, wird auf eine ausführliche Motivation und umfassende Definitionen an dieser Stelle verzichtet und auf [41, 46, 76] verwiesen.

Sei A eine endliche Menge von Symbolen. Eine *Trace über A* ist eine endliche oder unendliche Sequenz von Symbolen aus A. Die *Kleene'sche Hülle* A^* ist die Menge aller Traces über A, wobei die leere Trace (bezeichnet als ε) mit enthalten ist. Die *positive Hülle* A^+ ist die Menge aller Traces über A ohne die leere Trace. Es sei bemerkt, dass A^* und A^+ immer unendlich sind. Für Traces atr_1 und atr_2 bezeichnet $atr_1.atr_2$, oder alternativ $atr_1 atr_2$, die *Konkatenation* der beiden Traces. Der „.“ kann weggelassen werden, wenn die Symbole eindeutig sind. Traces können von endlicher oder unendlicher Länge sein. $a_1 a_2 \ldots a_n$ mit $n \in \mathbb{N}$ beschreibt eine *endliche* Trace der Länge n. $a_1 a_2 \ldots \infty$ beschreibt eine *unendlich* lange Trace. Mit $a_1 a_2 \ldots$ bleibt die Länge unbestimmt, also *endlich oder unendlich*. Diese Art der Bezeichnung wird auch für Aufzählungen (durch Kommata getrennte Symbole) übernommen, z.B bei $i = 1, 2, \ldots$ wird i auf endlich oder unendlich viele Werte gesetzt, je nach Kontext. Für eine Trace atr liefert $first(atr)$ das erste und $last(atr)$ das letzte Symbol der Trace.
Bei einem $\mathrm{V_I}$System gehen wir bei den lokalen Zeiten in den Komponenten sowie bei der globalen Systemumgebungszeit von einem linearen kontinuierlichem Zeitmodell aus (siehe Kapitel 3.2.1). In diesem Sinne werden Zeitpunkte durch reelle Zahlen repräsentiert und Zeitintervalle durch entsprechende reelle Intervalle. Zuweisungen innerhalb einer Ausführung eines $\mathrm{V_I}$Systems sind zeitlich atomar.
Bei Beweisführungen, die auf Fallunterscheidungen basieren, werden in den Bedingungen der Unterfälle (z.B. von einem Fall 2.1.3.1) die Bedingungen der übergeordneten Fälle (Fälle 2, 2.1, 2.1.3) vorausgesetzt, ohne sie explizit erneut aufzuführen. Verschiedene Unterfälle vermitteln alternative Fortführungen des Falles, aus dem sie hervorgehen.
Um Beweisschritte verständlich darzustellen, wird bei Implikations- oder Gleichheitsketten die Begründung eines jeweiligen Schritts stichwortartig nach dem zugehörigen Symbol aufgeführt. So bedeutet z.B. $A \Rightarrow \{Begr\} B$, dass B aus A folgt mit der Begründung $Begr$.

4.2 Verhalten

Das *Verhalten* eines I–Systems IS umfasst Folgen globaler Aktivitätszustände, die aus den Ausführungen des zugeordneten $\mathrm{V_I}$Systems $V_I System(IS)$ abgeleitet werden. Als Schnittstelle dienen die z-Globalbelegungen, d.h. die Werte der lokalen $\_z(\cdot)$-Variablen der Komponenten des $\mathrm{V_I}$Systems. Mit der folgenden Definition des Verhaltens wird das Ziel verfolgt, neben der Information, ob eine Phase in einem Bereich angenommen wird oder nicht, auch noch die durch die globalen Aktivitätszustände gegebenen Phasenqualitäten und deren Abfolgen in der Semantik zu dokumentieren. Bewegt man sich nun im bisherigen Anschauungskontext von Einflüssen, Zwängen, Entscheidungen, Trägheit, usw. wird durch die Definition des Verhaltens die Grundlage geschaffen, auf der formalen Ebene von z.B. Einflussbereichen, Entscheidungsspielräumen, Propagierung von Zwängen, erzwungenen und freien Ereignissen reden zu können und Analysen diesbezüglich zu betreiben.

Definition 4.1 (Verhalten eines I–Systems). Sei $IS \in ISystem$ ein I–System. Das *Verhalten* $\mathcal{V}[\![IS]\!]$ von IS ist definiert als:

$$\mathcal{V}[\![IS]\!] \subseteq GZustand(IS)^{+}$$

mit

$$\mathcal{V}[\![IS]\!] = \bigcup_{z_0 \in StabGZustand(IS)} \mathcal{V}[\![IS]\!](z_0)$$

Das Verhalten von IS *bzgl. eines stabilen globalen Start-Aktivitätszustandes* $z_0 \in StabGZustand(IS)$ ist dabei gegeben durch:

$$\mathcal{V}[\![IS]\!](z_0) =$$

$\{z^{\Pi,t_0}\langle V_I System(IS)\rangle.z^{\Pi,t_1}\langle V_I System(IS)\rangle.z^{\Pi,t_2}\langle V_I System(IS)\rangle.\ldots \in GZustand(IS)^{+} \mid$ Π ist eine Ausführung von $V_I System(IS)$ und $t_0, t_1, t_2, \ldots$ eine max. Folge von Globalzeitpunkten der Ausführung mit: t_0 ist Startzeitpunkt, $z^{\Pi,t_0}\langle V_I System(IS)\rangle = z_0$, $\forall i = 1, 2, \ldots : (t_{i-1} < t_i,\ z^{\Pi,t_{i-1}}\langle V_I System(IS)\rangle \neq z^{\Pi,t_i}\langle V_I System(IS)\rangle,$ $\forall t, t_{i-1} \leq t < t_i : z^{\Pi,t}\langle V_I System(IS)\rangle = z^{\Pi,t_{i-1}}\langle V_I System(IS)\rangle)\}$

□

Die Wohldefiniertheit des Verhaltens ist gewährleistet durch Satz 3.14, der garantiert, dass als z-Globalbelegungen zu jedem Zeitpunkt einer Ausführung Π nur globale Aktivitätszustände auftreten. Dabei wird als Initialisierung ein stabiler globaler Aktivitätszustand vorausgesetzt, d.h. es treten dort als Phasenqualitäten nur 0 und 1 auf. Die Voraussetzung ist in der obigen Definition mit z_0 zum Globalzeitpunkt t_0 erfüllt. Die Anfangsstabilität muss gefordert werden, um einen sinnvollen Ausführungsbeginn zu garantieren. Jede Komponente von $V_I System(IS)$ kann die Initialisierungsaktion A1 mit wahrer Vorbedingung passieren. Eine maximale Folge von Globalzeitpunkten ist als Festlegung notwendig, um die gesamte Ausführungszeit zu erfassen und alle während einer Ausführung auftretenden Aktivitätszustände zu protokollieren. Es sei bemerkt, dass das leere Verhalten, also $\mathcal{V}[\![IS]\!] = \emptyset$, auftreten kann, nämlich dann, wenn kein globaler Start-Aktivitätszustand z_0 existiert.

Bemerkung: Der Beschränkung auf die stabilen Aktivitätszustände als Ausgangspunkt der Ausführungen liegt die Eigenschaft (verteilter) Systeme zugrunde, Initialzustände zu besitzen, die für den Betrachter/Anwender eine Anfangsstabilität signalisieren. Ein Verarbeitungsprozess hat noch nicht begonnen, Kommunikation zur wechselseitigen Einflussnahme hat noch nicht stattgefunden oder hatte noch keine Auswirkungen. Reale Situationen in diesem Sinne sind z.B. ein ausgeschalteter Rechner, eine leere Printqueue, inaktive Signalleitungen oder ein noch nicht gestartetes verteiltes Programm. Zu beachten ist der Zusammenhang zwischen den Startzuständen auf der formalen semantischen Ebene des I–Systems und den Komponenteninitialisierungen auf der algorithmischen Ebene des $\mathrm{V_I}$Systems. Durch die Festlegung auf Stabilität der globalen Start-Aktivitätszustände beim Verhalten eines I–Systems wird nur ein Teil der Initialisierungen beim zugrunde liegenden $\mathrm{V_I}$System vorgegeben, nämlich die Belegung der $\_z(\cdot)$-Variablen. Die Anfangszustände der verbleibenden Variablen werden hingegen durch die Axiome und Aktionen zur Dynamik (Kapitel 3.2.1, 3.2.2) festgelegt.

Beispiel 4.2. Es sollen zwei Elemente des Verhaltens $\mathcal{V}[\![IS_1]\!]$ des I–Systems IS_1 aus Beispiel 2.2 bestimmt werden. Dazu werden zwei Ausführungen von $V_I System(IS_1)$ und die auftretenden globalen Aktivitätszustände betrachtet. (Die Ausführungen werden in der Notation 3.17 angegeben.)

Ausführung 1:

$$\overbrace{[p_2{<}1{>}, q_2{<}1{>}, v_2{<}1{>}]}^{z_0} \xrightarrow{b_1.\mathrm{A3}, b_2.\mathrm{A3}} \overbrace{[p_2{<}p_1{>}, q_2{<}q_3{>}, v_2{<}1{>}]}^{z_1} \xrightarrow{b_1.\mathrm{A4}} \overbrace{[p_1{<}1{>}, q_2{<}q_3{>}, v_2{<}1{>}]}^{z_2}$$

$$\xrightarrow{b_1.\mathrm{A3}, b_2.\mathrm{A4}} \overbrace{[p_1{<}p_2{>}, q_3{<}1{>}, v_2{<}1{>}]}^{z_3} \xrightarrow{\underline{b_3}.\mathrm{A10}} \overbrace{[p_1{<}p_2{>}, q_3{<}1{>}, v_2{<}\mathsf{F}{>}]}^{z_4} \xrightarrow{b_1.\mathrm{A4}}$$

$$\overbrace{[p_2{<}1{>}, q_3{<}1{>}, v_2{<}\mathsf{F}{>}]}^{z_5} \xrightarrow{\underline{b_3}.\mathrm{A5}} \overbrace{[p_2{<}1{>}, q_3{<}1{>}, v_1{<}1{>}]}^{z_6}$$

Ausführung 2:

$$\overbrace{[p_1{<}1{>}, q_1{<}1{>}, v_2{<}1{>}]}^{z'_0} \xrightarrow{b_1.\mathrm{A3}} \overbrace{[p_1{<}p_2{>}, q_1{<}1{>}, v_2{<}1{>}]}^{z'_1} \xrightarrow{b_1.\mathrm{A4}} \overbrace{[p_2{<}1{>}, q_1{<}1{>}, v_2{<}1{>}]}^{z'_2}$$
$$\xrightarrow{b_2.\mathrm{A10}} \overbrace{[p_2{<}1{>}, q_1{<}\mathsf{F}{>}, v_2{<}1{>}]}^{z'_3} \xrightarrow{b_2.\mathrm{A5}} \overbrace{[p_2{<}1{>}, q_2{<}1{>}, v_2{<}1{>}]}^{z'_4}$$

Es ergibt sich somit $z_0z_1z_2z_3z_4z_5z_6 \in \mathcal{V}[\![IS_1]\!]$ und $z'_0z'_1z'_2z'_3z'_4 \in \mathcal{V}[\![IS_1]\!]$. Setzt man die zweite mit der ersten Ausführung fort, möglich wegen $z'_4 = z_0$, erhält man zusätzlich $z'_0z'_1z'_2z'_3z'_4z_1z_2z_3z_4z_5z_6 \in \mathcal{V}[\![IS_1]\!]$. □

In Kapitel 1 wurde beschrieben, dass man bei den dynamischen Abläufen in einem verteilten System zwischen Ereignissen unterscheiden kann, die eintreten *werden*, und welchen, die eintreten *können*. Betrachtet man die Axiome und Aktionen zur Festlegung der Dynamik eines I–Systems, wird deutlich, dass das Fortschreiten des zugeordneten $\mathrm{V_I}$Systems durch die Aktionsbeschreibungen A1-A2 und A4-A13 garantiert wird. Variablenänderungen bzw. Kommunikationsanweisungen *werden* eintreten als Folge der Abarbeitung der einzelnen Befehlssequenzen. Die einzige Ausnahme bildet Aktion A3 (Autonome Entscheidung treffen). Hier ist explizit angegeben, dass eine Veränderung, nämlich $\_z(p) := q$, eintreten *kann*. Ob diese Anweisung tatsächlich ausgeführt wird, obliegt der „Entscheidungsfreiheit“ der jeweiligen Komponente. Von der technischen Realisierung der Entscheidungsfreiheit (z.B. mit Hilfe von Zufallszahlen oder der Auswertung vorher definierter spezieller Variablen) im Rahmen der operationellen Semantik des $\mathrm{V_I}$Systems wird abstrahiert. Die Komponente wird als autonom angenommen. Dieses erklärt auch den Begriff der autonomen Bereiche (Definition 2.1). Nur bei diesen Bereichen zugeordneter Komponenten kann A3 Anwendung finden, entsprechend der Vorbedingung der Aktionsbeschreibung. Die garantierte oder nur mögliche Ausführung von Aktionsanweisungen bei den Ausführungen eines $\mathrm{V_I}$Systems hat Auswirkungen auf die formale Ebene in Form von nicht präfix-abgeschlossenen Semantiken der I–Systeme, insbesondere des Verhaltens.

Definition 4.3 (präfix-abgeschlossen). Eine Trace-Semantik $S[\![IS]\!]$ eines I–Systems IS heißt *präfix-abgeschlossen*, wenn für jedes Element tr aus $S[\![IS]\!]$ gilt: Jedes von ε verschiedene Präfix von tr, das syntaktisch als Element von $S[\![IS]\!]$ in Frage kommt, gehört auch zu $S[\![IS]\!]$. □

Satz 4.4 (Nicht-Präfix-Abgeschlossenheit des Verhaltens). Für das Verhalten $\mathcal{V}[\![IS]\!]$ eines I–Systems IS ist Präfix-Abgeschlossenheit nicht garantiert.

Beweis. Anhand des I–Systems IS_1 aus Beispiel 2.2 soll demonstriert werden, warum das Verhalten eines I–Systems im Allgemeinen nicht präfix-abgeschlossen ist. In Beispiel 4.2 wurde bereits $z_0z_1z_2z_3z_4z_5z_6 \in \mathcal{V}[\![IS_1]\!]$ gezeigt (für die dort angegebenen globalen Aktivitätszustände $z_0, z_1, \ldots, z_6$).

Behauptung: $z_0z_1z_2z_3 \notin \mathcal{V}[\![IS_1]\!]$

Beweis. Sei eine beliebige Ausführung Π von $V_ISystem(IS_1)$ gegeben, die gemäß Definition 4.1 $z_0z_1z_2z_3$ erzeugt, und seien t_0, t_1, t_2, t_3 die durch die Definition gegebenen Globalzeitpunkte.

$\Rightarrow$

$z^{\Pi,t'}\langle V_ISystem(IS_1)\rangle(v_2) = 1$ für $t' \in [t_0, t_3]$, $z^{\Pi,t''}\langle V_ISystem(IS_1)\rangle(q_2) \neq \mathsf{0}$ für $t'' \in [t_0, t_3[$ und $z^{\Pi,t_3}\langle V_ISystem(IS_1)\rangle(q_3) = 1$.

$\Rightarrow$

Zum Zeitpunkt t_3 findet bei der Komponente V_{b_2} (atomar) eine Phasentransition $q_2 \to q_3$ statt, während $V_{\underline{b}_3}$ in der Phase v_2 ist.

Phasentransitionen können bei V_{b_2} mittels der Aktionen A4 oder A5 erfolgen. In beiden Fällen erfolgt danach das Senden einer Nachricht $done(q_2 \to q_3)$ von V_{b_2} nach $V_{\underline{b}_3}$, da $\underline{b}_3$ ein Nachbarbereich von b_2 ist.

Auf eine eintreffende $done(q_2 \to q_3)$-Nachricht *wird* $V_{\underline{b}_3}$ gemäß Aktionsbeschreibung A10 reagieren,

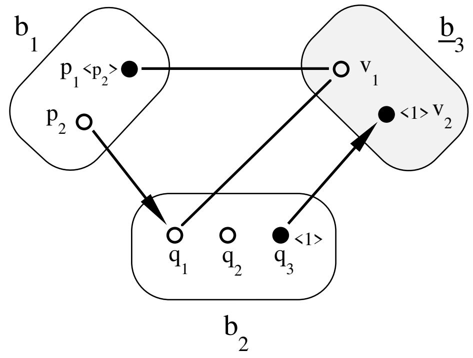


Abbildung 4.1: Globaler Aktivitätszustand z_3

d.h. es *wird* $\_in(q_3) := true$ und dann ein Update, mit v_2 als aktueller Phase, erfolgen (A13).

Aufgrund der gegebenen Struktur von IS_1, $E(v_2) = \emptyset$ und $\underline{b}_3$ ist ein träger Bereich, gilt bei $V_{\underline{b}_3}$ immer $\_e_{out}(v_2) = false$ sowie $\_z(v_2) \in \{0, 1, \mathsf{F}\}$. Folglich wird beim Update A13.iv ausgeführt mit $\_z(v_2) := \mathsf{F}$.

$\Rightarrow$

Es existiert ein Globalzeitpunkt t_x, $t_x > t_3$, mit $z^{\Pi, t_x} \langle V_I System(IS_1)\rangle(v_2) = \mathsf{F}$.

$\Rightarrow$

t_0, t_1, t_2, t_3 ist keine max. Folge von Globalzeitpunkten gemäß Definition 4.1.

$\Rightarrow$

$z_0 z_1 z_2 z_3 \notin \mathcal{V}[\![IS_1]\!]$.

Der zuvor dargestellte Ablauf lässt sich wie folgt interpretieren: Die Aktivität innerhalb der Komponenten ist noch nicht abgeschlossen, da es noch Auswirkungen von Erregungen, in diesem Fall der Erregung von v_2 durch q_3, gibt. Anschaulich zeigt sich die bestehende Erregung in Abbildung 4.1, in der der globale Aktivitätszustand z_3 graphisch dargestellt ist. Die Kreise zu q_3 und v_2 sind ausgefüllt, d.h. beide Phasen sind belegt in z_3. Der verbindende Pfeil verdeutlicht die Erregung. Als Auswirkung der Erregung wird $V_{\underline{b}_3}$ instabil in v_2, d.h. die Phasenqualität von v_2 wechselt von 1 nach F. Ein weiteres Ereignis, das noch eintreten wird, dessen Betrachtung für den Beweis aber nicht mehr erforderlich ist, ist eine Phasentransition $p_1 \rightarrow p_2$, die aus der bestehenden Entscheidung von V_{b_1}, diese Phasentransition durchzuführen, resultiert. $\square$

Die Eigenschaft der Nicht-Präfix-Abgeschlossenheit einer Semantik kann ausgenutzt werden, um Fortschrittsanforderungen an ein verteiltes System zu spezifizieren und zu verifizieren. Ein Beispiel mit System Nets, einer Variante von Petri-Netzen, wird in [75] vorgestellt. Dort werden spezielle temporale Logiken, die auf nicht präfix-abgeschlossenen Mengen von so genannten Interleaved oder Concurrent Runs operieren, verwendet, um Fortschrittseigenschaften zu spezifizieren bzw. verifizieren. Die Methoden können analog auf I–Systeme übertragen werden. Mit dem Verhalten $\mathcal{V}[\![\cdot]\!]$ ist die benötigte Referenzmenge geschaffen worden. Auf die technische Realisierung wird an dieser Stelle verzichtet.

Die Nicht-Präfix-Abgeschlossenheit des Verhaltens erlaubt es, Ereignisse zu identifizieren, die in Abhängigkeit von den auftretenden Endelementen der Traces entweder eintreten werden oder eintreten können. Aus der Nicht-Präfix-Abgeschlossenheit alleine ist hingegen nicht ableitbar, *warum* bestimmte Ereignisse eintreten oder *warum* sie *nicht* eintreten. Zur Bestimmung der Ursachen wird auf die Phasenqualitäten bei den globalen Aktivitätszuständen zurückgegriffen. In Abhängigkeit von den Phasenqualitäten repräsentieren zwei aufeinander folgende globale Aktivitätszustände in einer Trace des Verhaltens eine bestimmte Aktivität in den Bereichen eines I–Systems.

Definition 4.5 (Ereignisse). Sei IS ein I–System, b ein Bereich von IS und $p, q \in b$. Sei $z_0 z_1 \ldots z_{i-1} z_i \ldots \in \mathcal{V}[\![IS]\!]$ eine beliebige Trace des zugehörigen Verhaltens mit $z_j \in GZustand(IS)$, $j = 0, 1, 2, \ldots$ und $i \in I\!N$. Die Teiltrace $z_{i-1} z_i$ repräsentiert folgende Ereignisse:

a) Bei $z_{i-1}(p) = 1$ und $z_i(p) = q$ sagen wir, dass b *eine Entscheidung trifft*, nach q zu wechseln.

b) Bei $z_{i-1}(p) = 1$ und $z_i(p) = \mathsf{F}$ sagen wir, dass b *instabil wird*.

c) Bei $z_{i-1}(p) = q$ und $z_i(p) = 1$ sagen wir, dass die *Entscheidung* von b, nach q zu wechseln, *zurückgenommen wird*.

d) Bei $z_{i-1}(p) = q$ und $z_i(q) = 1$ sagen wir, dass eine *freie Phasentransition* von p nach q in b auftritt.

e) Bei $z_{i-1}(p) = \mathsf{F}$ und $z_i(q) = 1$ sagen wir, dass eine *erzwungene Phasentransition* von p nach q in b auftritt. □

Auf der algorithmischen Ebene ergeben sich die einzelnen Ereignisse aus den Aktionen A1-A13 des zugeordneten V_ISystems. Bei a) befindet man sich in A3, bei b) und c) in A13 als Aufruf aus A1, A4, A5, A10, A11 oder A12. Bei d) befindet man sich in A4 und bei e) in A5. Die Folgeaktivitäten ergeben sich dann aus den Axiomen und Aktionen des V_ISystems.

Alle Traces des Verhaltens eines I–Systems leiten sich ab aus den Ausführungen des zugeordneten V_ISystems über die z-Globalbelegungen (siehe Definition 4.1). Diese Belegungen ergeben sich während der Abarbeitung der Aktionen, die die Dynamik spezifizieren. Die Abarbeitung der Aktionen wird gesteuert durch die lokalen Variablenbelegungen in den Komponenten des V_ISystems, die ihrerseits abhängig sind von dem Aufbau des betrachteten I–Systems. Es stellt sich nun die Frage nach den direkten Beziehungen (auf der formalen Ebene) zwischen den Eigenschaften der Traces des Verhaltens und dem Aufbau des I–Systems.
Der folgende Satz beschreibt grundlegende Zusammenhänge zwischen der formalen Struktur eines beliebigen I–Systems IS und dessen Verhalten $\mathcal{V}[\![IS]\!]$. Insbesondere wird die Bedeutung der Kopplungs- und Erregungsrelation zum Ausdruck gebracht.

Satz 4.6 (Charakterisierung des Verhaltens). Sei $IS = (P, B, \underline{B}, K, E)$ ein I–System und $z_0 z_1 z_2 \ldots \in \mathcal{V}[\![IS]\!]$ ein Element des zugehörigen Verhaltens mit $z_j \in GZustand(IS), j = 0, 1, 2, \ldots$. Seien $p, q, v, w \in P$ mit $b(p) = b(q)$ und $b(v) = b(w)$. Dann gelten folgende Aussagen für alle $i = 1, 2, \ldots$:

a) *Stop*
$$((p,v) \in E \wedge z_{i-1}(p) \neq 0 \wedge z_{i-1}(v) \neq 0) \quad \Rightarrow \quad (z_i(p) \neq 0 \wedge \exists k, k \geq i-1 : z_k(p) = 1)$$

b) *Erregung*
$$((p,v) \in E \wedge z_{i-1}(p) \neq 0 \wedge z_{i-1}(v) \neq 0) \quad \Rightarrow \quad (\exists k, k \geq i-1 : (z_k(v) \in \{0, \mathsf{F}\} \vee (z_k(v) = 1 \wedge \exists x \in E(v) : z_k(x) \neq 0)))$$

c) *Nicht-Koinzidenz*
$$((p,w) \in K \wedge (q,v) \in K \wedge z_{i-1}(p) \neq 0 \wedge z_{i-1}(v) \neq 0) \quad \Rightarrow \quad (z_i(q) = 0 \vee z_i(w) = 0)$$

d) *Einflusspropagierung*
$$((q,v) \in K \wedge z_{i-1}(p) \in \{q, \mathsf{F}\} \wedge z_{i-1}(v) \neq 0) \quad \Rightarrow \quad (\exists k, k \geq i-1 : (z_k(p) \in \{0,1\} \vee z_k(v) \in \{0, \mathsf{F}\} \vee (z_k(v) = 1 \wedge \exists x \in E(v) : z_k(x) \neq 0)))$$

Beweis. Der Beweis basiert auf der Analyse einer beliebigen Ausführung Π von $V_I System(IS)$, die $z_0 z_1 z_2 \ldots$ gemäß Definition 4.1 erzeugt. Alle Möglichkeiten für das Aussehen von Π werden soweit per Fallunterscheidungen untersucht, bis die Gültigkeit der Satzaussagen gezeigt ist. Hierzu werden die Aktionen aus Kapitel 3.2.2 systematisch abgearbeitet.

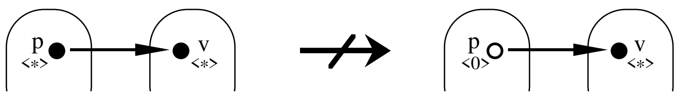
Der vollständige Beweis befindet sich im Anhang A.1 (Seite 161 ff.).

Hervorzugeben ist, dass die Kriterien (1) und (2) aus Definition 3.10 (Ausführung) in dem Beweis zum Tragen kommen. Das Kriterium (1) fordert für die Ausführung Π, dass die Verhaltensaxiome VA1, VA2 und VA3 (aus Kapitel 3.2.1) respektiert werden. Ein direkter Verweis auf VA1 und

VA2 findet sich im Beweis zu a) und zu b) in den Fällen ii.1), ii.1.2), ii.1.3), ii.2.3), iii.1) sowie im Beweis zu d) in den Fällen 1) und 1.3.2). Auf VA3 wird in dem Beweis nicht direkt verwiesen. Der Einfluss dieses Verhaltensaxioms spiegelt sich in der Anzahl der betrachteten Fälle wider, da die Ausführungszeiten der Aktionsanweisungen und die Kommunikationszeiten zwischen den Komponenten von $V_I System(IS)$ als unbekannt vorausgesetzt werden. Das Fairness-Kriterium (2) wird in dem Beweis zu d) in den Fällen 1.3.2.3) und 2.4.2.4) benötigt. Es ist erforderlich, damit bestimmte unerwünschte unendliche Ausführungsschleifen, durch die eine Einflusspropagierung verhindert wird, nicht auftreten können.
Der Beweis von Satz 4.6 macht deutlich, dass die Abarbeitung aller Ausführungsalternativen für die Ausführung Π zwar umfangreich ist, dass allerdings die praktische Durchführbarkeit durch die in Kapitel 3 zur Dynamik gemachten Festlegungen gewährleistet ist. $\square$

Die Kernaussagen des Satzes lassen sich anschaulich wie folgt interpretieren:

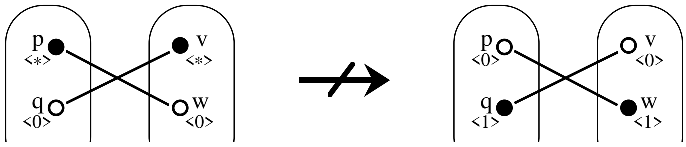
a) beschreibt die *Aufrechterhaltung eines Einflusses* zwischen zwei aktuell eingenommenen Phasen in benachbarten Komponenten. Eine Idee hinter der Erregungsrelation ist, dass sie mögliche Einflüsse beschreibt, die Aktivitäten erzwingen sollen. Die erregende Phase (gemäß Definition 3.8) übt einen Zwang auf die erregte Phase aus. Die Komponente der erregten Phase soll zu einer Phasentransition gezwungen werden. Zwänge werden solange aufrecht erhalten, bis sie bestimmte Auswirkungen erzielen, d.h., ist eine Phase Erreger einer anderen Phase, so kann die *erregende* Phase erst wieder verlassen werden, nachdem die erregte Phase verlassen wurde. Da die Zwangsausübung solange beibehalten wird, bis eine bestimmte Reaktion erfolgt ist, geht die Zwang ausübende Komponente während der Zwangsausübung schließlich in einen stabilen Zustand über.
Graphisch betrachtet ist der folgende Ausschnitt einer Phasentransition, bei der p verlassen wird, nicht möglich.



Die Bezeichnung „$*$" in der Abbildung steht für eine beliebige Phasenqualität ungleich 0.

b) beschreibt die *Auswirkungen eines Einflusses/ eines Zwanges.* Ist eine Phase Erreger einer anderen Phase wie in a), so wird bei der Komponente der *erregten Phase* einer der folgenden drei Fälle eintreten, sofern nicht einer davon schon vorliegt. Erstens: Die aktuelle (erregte) Phase wird verlassen, bevor der Einfluss sich auswirkt. Zweitens: Sie wird instabil, der Einfluss wirkt sich aus. Es werden als nächstes Aktionen gestartet, um dem Zwang nachzugeben, die erregte Phase zu verlassen. Drittens: Sie wird stabil, sofern die erregte Phase selbst Erreger ist. Der Einfluss wird in diesem Fall in seiner Auswirkung blockiert.

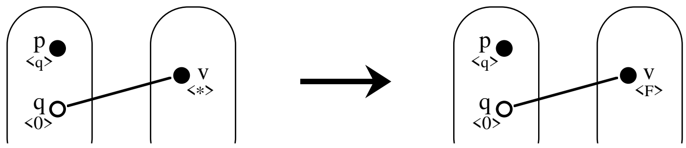
c) verdeutlicht, dass *koinzidente Phasentransitionen bei über Kreuz wechselseitig ausgeschlossenen Nachbarphasen nicht auftreten können.* Wie in Kapitel 1 betont wurde, ist die verteilte Kontrolle ein charakteristisches Merkmal verteilter Systeme. Wechselseitige Synchronisationsmechanismen zwischen Komponenten können deshalb nur über spezielle Kommunikationsprotokolle realisiert werden [43]. Da die technischen Randbedingungen das Kommunikationsverhalten immer nur annähernd genau bestimmen lassen, ist die globale Zeitgleichheit zweier Ereignisse in unterschiedlichen Komponenten nicht realisierbar [21, 86]. Jegliche Hintereinanderausführung der Phasentransitionen wird durch den vorgegebenen kreuzweisen wechselseitigen Ausschluss untersagt.
In der folgenden Graphik sind die nebenläufigen Phasentransitionen $p \rightarrow q$ und $v \rightarrow w$ ausgeschlossen. Der wechselseitige Ausschluss zwischen p und w sowie zwischen q und v wird über die Kopplungsrelation angegeben.



d) beschreibt die *Propagierung eines Einflusses.* Hat eine Komponente das Bestreben, eine Phasentransition durchzuführen, die in Frage kommende Folgephase ist aber aufgrund eines wechselseitigen Ausschlusses mit einer oder mehreren Nachbarphasen nicht frei (gemäß Definition 3.8), dann übt die Komponente, die die Phasentransition durchführen möchte, einen Einfluss auf die Nachbarkomponenten aus, die blockierenden Phasen zu verlassen. Die möglichen Auswirkungen der Einflusspropagierung auf die blockierenden Phasen entsprechen denen der Einflüsse auf die erregten Phasen in b). Das Bestreben zur Phasentransition gilt als beendet, sobald die Ausgangsphase verlassen oder aber Erreger einer Nachbarphase wird.

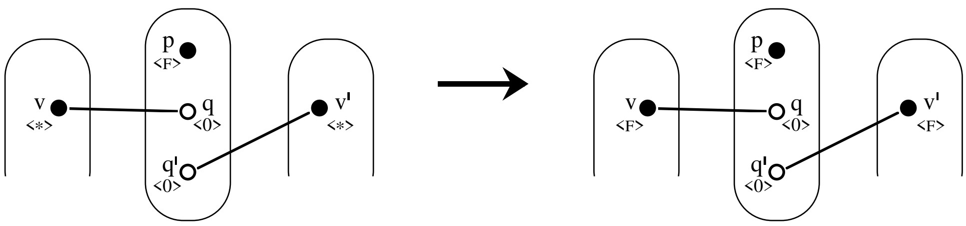
Ergänzung: Für das Bestreben einer Komponente, eine Phasentransition durchzuführen, und daraus resultierende Einflusspropagierungen, gibt es *zwei mögliche Situationen*:

i) Die Komponente hat eine Entscheidung getroffen, in eine Folgephase zu wechseln, aber die Folgephase ist nicht frei. Beispiel:



Hier tritt eine Einflusspropagierung von der Komponente von p zu der Komponente von v auf. Die Auswirkung ist in diesem Beispiel eine Instabilität in v.

ii) Die Komponente unterliegt einem Einfluss, die aktuelle Phase zu verlassen (d.h. die Komponente ist instabil), und es gibt in der Komponente keine freie Phase. Beispiel:



Hier treten Einflusspropagierungen von der Komponente von p zu den Komponenten von v und v' auf. Die Auswirkungen sind in diesem Beispiel Instabilitäten in v und v'.

Der in ii) vorausgesetzte Einfluss, aus der die Instabilität der Komponente (in dem Beispiel die Komponente von p) resultierte und der die propagierten Einflüsse zur Folge hat, kann selbst ein propagierter Einfluss von einer Nachbarkomponente sein. Führt man dieses fort, ergeben sich *Propagierungsketten von Einflüssen*, die sich über das Gesamtsystem erstrecken können. Solche Ketten repräsentieren sich global ausbreitende Auswirkungen von lokalen Gegebenheiten in einem verteilten System. Die Effekte sind naturgemäß nur schwer oder teilweise auch gar nicht vorhersehbar [78]. I–Systeme bieten die Möglichkeit, über die Phasenqualitäten und das bekannte Verhalten gemäß ii), solche globalen Propagierungseffekte im Modell nachvollziehen und den Ausgangspunkt lokalisieren zu können. Das Wissen darüber ist notwendig, wenn man unerwünschte globale Auswirkungen lokaler Ereignisse beheben möchte.

Der Beweis von Satz 4.6.a-4.6.d erfordert die Untersuchung der Aktionen A1-A13 des zugrundeliegenden $\mathrm{V_I}$Systems. Die resultierenden strukturellen Eigenschaften des Verhaltens (als Semantik) eines I–Systems werden deshalb als *elementar* bezeichnet.

Definition 4.7 (Elementare Struktureigenschaft). Wird zum Beweis einer strukturellen Eigenschaft einer Semantik eines I–Systems IS auf die Ausführungen des zugeordneten V_ISystems $V_I System(IS)$ zurückgegriffen, so nennt man diese Eigenschaft *elementar*. □

Den Gegensatz zu den elementaren bilden die *abgeleiteten* Struktureigenschaften. Auf sie wird in Kapitel 10 eingegangen.

Die Aktionen A1-A13 wurden bewusst so konstruiert, dass die elementaren Struktureigenschaften aus Satz 4.6 erfüllt sind. Die oben genannten anschaulichen Interpretationen der Satzaussagen, speziell der Einfluss der Kopplungs- und Erregungsrelation, dienten als Motivation. Durch diesen Ansatz ist es möglich, die Existenz und auch Nicht-Existenz wechselseitiger Einflüsse und deren Auswirkungen in einem verteilten System explizit in einer intuitiv verständlichen Form zu modellieren [89].

4.3 Casetrace-Semantik

Neben den globalen Aktivitätszuständen wurden in Kapitel 3.1.1 Cases als globale Systemzustände eines I–Systems vorgestellt. Diese können zur Bildung einer weiteren Trace-Semantik herangezogen werden. Die *Casetrace-Semantik* eines I–Systems IS umfasst Folgen von Cases, die sich aus den Ausführungen des zugeordneten V_ISystems $V_I Systems(IS)$ ergeben. Die notwendigen Informationen werden aus den z-Globalbelegungen abgeleitet.

Definition 4.8 (Casetrace-Semantik). Sei $IS \in ISystem$ ein I–System. Die *Casetrace-Semantik* $\mathcal{CT}[\![IS]\!]$ von IS ist definiert als:

$$\mathcal{CT}[\![IS]\!] \subseteq Case(IS)^+$$

mit

$$\mathcal{CT}[\![IS]\!] = \bigcup_{c_0 \in Case(IS)} \mathcal{CT}[\![IS]\!](c_0)$$

Die Casetrace-Semantik von IS *bzgl. eines Start-Cases* $c_0 \in Case(IS)$ ist dabei gegeben durch:

$$\mathcal{CT}[\![IS]\!](c_0) =$$
$$\{zc(z^{\Pi,t_0}\langle V_I System(IS)\rangle).zc(z^{\Pi,t_1}\langle V_I System(IS)\rangle).zc(z^{\Pi,t_2}\langle V_I System(IS)\rangle)\ldots \in Case(IS)^+ \mid \Pi \text{ ist eine Ausführung von } V_I System(IS) \text{ und } t_0, t_1, t_2, \ldots \text{ eine max. Folge von Globalzeitpunkten der Ausführung mit: } t_0 \text{ ist Startzeitpunkt, } z^{\Pi,t_0}\langle V_I System(IS)\rangle = cz(c_0), \forall i = 1,2,\ldots : (t_{i-1} < t_i,\ zc(z^{\Pi,t_{i-1}}\langle V_I System(IS)\rangle) \neq zc(z^{\Pi,t_i}\langle V_I System(IS)\rangle), \forall t, t_{i-1} \leq t < t_i : zc(z^{\Pi,t}\langle V_I System(IS)\rangle) = zc(z^{\Pi,t_{i-1}}\langle V_I System(IS)\rangle))\}$$

□

Die Wohldefiniertheit der Casetrace-Semantik basiert auf Korollar 3.16. Dort wird beschrieben, dass $zc(z^{\Pi,t_j}\langle V_I System(IS)\rangle)$, für $j = 0, 1, 2, \ldots$ aus dem Indexbereich der Definition, auch tatsächlich ein Case des I–Systems IS ist. $cz(c_0)$ ist nach Definition 3.6.b immer ein stabiler globaler Aktivitätszustand. Wie schon beim Verhalten eines I–Systems (Definition 4.1) garantiert die Anfangsstabilität einen Ausführungsbeginn, bei dem jede Komponente von $V_I System(IS)$ die Initialisierungsaktion A1 mit wahrer Vorbedingung passieren kann. Die Maximalitätsanforderung für die Folge der Globalzeitpunkte führt zur Erfassung aller während der Ausführung Π auftretenden Cases. Die leere Casetrace-Semantik, also $\mathcal{CT}[\![IS]\!] = \emptyset$, kann auftreten, wenn es keinen Case des I–Systems IS gibt.

An dem folgenden Beispiel soll die Definition der Casetrace-Semantik verdeutlicht werden.

Beispiel 4.9. Bestimmt werden sollen Elemente der Casetrace-Semantik $\mathcal{CT}[\![IS_1]\!]$ des I–Systems IS_1 aus Beispiel 2.2. Betrachtet werden dazu die Ausführungen 1 und 2 von $V_I System(IS_1)$ aus Beispiel 4.2.

Nach Definition 4.8 gilt $\underbrace{zc(z_0)}^{c_0} . \underbrace{zc(z_2)}^{c_1} . \underbrace{zc(z_3)}^{c_2} . \underbrace{zc(z_5)}^{c_3} . \underbrace{zc(z_6)}^{c_4} \in \mathcal{CT}[\![IS_1]\!]$ mit
$c_0c_1c_2c_3c_4 = \{p_2, q_2, v_2\}.\{p_1, q_2, v_2\}.\{p_1, q_3, v_2\}.\{p_2, q_3, v_2\}.\{p_2, q_3, v_1\}$.

Weiterhin gilt $\overbrace{zc(z'_0)}^{c'_0} . \overbrace{zc(z'_2)}^{c'_1} . \overbrace{zc(z'_4)}^{c'_2} \in \mathcal{CT}[\![IS_1]\!]$ mit
$c'_0c'_1c'_2 = \{p_1, q_1, v_2\}.\{p_2, q_1, v_2\}.\{p_2, q_2, v_2\}$.

Die Hintereinanderausführung der beiden Casefolgen liefert $c'_0c'_1c'_2c_1c_2c_3c_4 \in \mathcal{CT}[\![IS_1]\!]$. □

Genau wie beim Verhalten eines I–Systems ist man an den speziellen Charakteristika der Casetrace-Semantik interessiert. Als erste Eigenschaft wird wieder die Nicht-Präfix-Abgeschlossenheit untersucht.

Satz 4.10 (Nicht-Präfix-Abgeschlossenheit der Casetrace-Semantik). Für die Casetrace-Semantik $\mathcal{CT}[\![IS]\!]$ eines I–Systems IS ist Präfix-Abgeschlossenheit nicht garantiert.

Beweis. Es reicht die Angabe eines speziellen I–Systems, bei dem die Präfix-Abgeschlossenheit der Casetrace-Semantik nicht erfüllt ist. Betrachtet wird hierzu das I–System IS_1 aus Beispiel 2.2 und die Cases von IS_1 aus Beispiel 4.9. Es wurde bereits $c'_0c'_1c'_2 \in \mathcal{CT}[\![IS_1]\!]$ gezeigt.

Annahme: $c'_0c'_1 \in \mathcal{CT}[\![IS_1]\!]$

Der formale Beweis ist nicht trivial, da alle in Frage kommenden Phasenqualitäten betrachtet werden müssen, die den Cases zugrunde liegen können. Dies gilt insbesondere für die Überprüfung, ob c'_1 (siehe Abbildung 4.2) als Ende eine Casefolge auftreten kann.

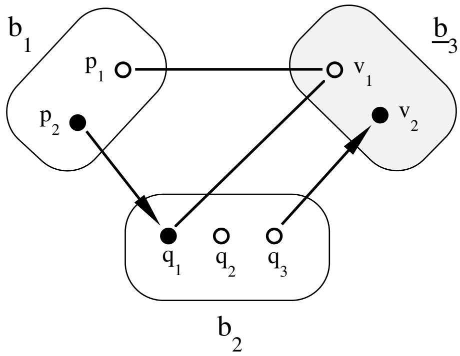


Abbildung 4.2: Case c'_1

Sei nun eine beliebige Ausführung Π von $V_I System(IS_1)$ gegeben, die gemäß Definition 4.8 $c'_0c'_1$ erzeugt, und seien t_0, t_1 die durch die Definition gegebenen Globalzeitpunkte.

$\Rightarrow$ {Definition $\mathcal{CT}[\![\cdot]\!]$}
$\{p_2, q_1\} \in zc(z^{\Pi,t_1}\langle V_I System(IS_1)\rangle)$

$\Rightarrow$ {Definition $zc(\cdot)$}
$z^{\Pi,t_1}\langle V_I System(IS_1)\rangle(p_2) \neq \mathsf{0}$ und $z^{\Pi,t_1}\langle V_I System(IS_1)\rangle(q_1) \neq \mathsf{0}$

$\Rightarrow$ {Definition der z-Globalbelegung}
$z^{\Pi,t_1}\langle V_{b(p_2)}\rangle(p_2) \in \{\mathsf{1}, \mathsf{F}, p_1\}$ und $z^{\Pi,t_1}\langle V_{b(q_1)}\rangle(q_1) \in \{\mathsf{1}, \mathsf{F}, q_2, q_3\}$.

Entsprechend dem Beweis von Satz 4.6, zu a) und b), Fall ii) und iii), existiert ein Zeitpunkt t_2, $t_2 \geq t_1$ mit $z^{\Pi,t_2}\langle V_{b(q_1)}\rangle(q_1) \in \{\mathsf{0}, \mathsf{F}\}$.

Fall 1). $z^{\Pi,t_2}\langle V_{b(q_1)}\rangle(q_1) = 0$.

$\Rightarrow$ Es existiert ein Zeitpunkt t_2, $t_2 > t_1$ mit $q_1 \notin zc(z^{\Pi,t_2}\langle V_I System(IS_1)\rangle)$.

$\Rightarrow$ t_0, t_1 ist keine max. Folge von Globalzeitpunkten gemäß Definition 4.8. Folglich gilt $c_0'c_1' \notin \mathcal{CT}[\![IS_1]\!]$, und damit liegt ein Widerspruch zur Annahme vor.

Fall 2). $z^{\Pi,t_2}\langle V_{b(q_1)}\rangle(q_1) = \mathsf{F}$.

Bei $V_{b(q_1)}$ wird schließlich die Aktion A5 durchgeführt. Wegen der Struktur von IS_1 mit $E(q_1) = \emptyset$ und $K(q_2) = K(q_3) = \emptyset$ gilt, bezogen auf die Aktionsbeschreibung von A5: *not* a), *not* b) und c). Es findet eine Phasentransition $q_1 \rightarrow q_2$ oder $q_1 \rightarrow q_3$ statt.

$\Rightarrow$ Es existiert ein Zeitpunkt t_3, $t_3 > t_2$ mit $z^{\Pi,t_2}\langle V_{b(q_1)}\rangle(q_1) = 0$.

$\Rightarrow$ Es existiert ein Zeitpunkt t_3, $t_3 > t_1$ mit $q_1 \notin zc(z^{\Pi,t_3}\langle V_I System(IS_1)\rangle)$.

$\Rightarrow$ t_0, t_1 ist keine max. Folge von Globalzeitpunkten gemäß Definition 4.8. Folglich gilt $c_0'c_1' \notin \mathcal{CT}[\![IS_1]\!]$, und damit liegt ein Widerspruch zur Annahme vor.

Alle auftretenden Fälle führen zu einem Widerspruch der Annahme. Die Casetrace-Semantik von IS_1 ist somit nicht präfix-abgeschlossen, und mit IS_1 als spezielles I–System gilt Satz 4.10. $\square$

Wie schon in Kapitel 4.2 beschrieben wurde, ist Nicht-Präfix-Abgeschlossenheit eine Grundlage zur Spezifikation und Verifikation von Fortschrittseigenschaften, in diesem Fall auf der Basis der Casetrace-Semantik. Zwei aufeinander folgende Cases repräsentieren dabei eine bestimmte Aktivität in den Bereichen eines I–Systems.

Definition 4.11 (Phasentransition). Sei IS ein I–System, b ein Bereich von IS und $p, q \in b$. Sei $c_0 c_1 \ldots c_{i-1} c_i \ldots \in \mathcal{CT}[\![IS]\!]$ eine beliebige Trace der Casetrace-Semantik mit $c_j \in Case(IS)$, $j = 0, 1, 2, \ldots$ und $i \in I\!N$. Die Teiltrace $c_{i-1}c_i$ repräsentiert folgende Ereignisse:

Bei $p \in \{c_{i-1} \setminus c_i\}$ und $q \in \{c_i \setminus c_{i-1}\}$ sagen wir, dass eine *Phasentransition* von p nach q in b auftritt. $\square$

Es sei bemerkt, dass die Teiltrace $c_{i-1}c_i$ auch mehrere Phasentransitionen in unterschiedlichen Bereichen gleichzeitig repräsentieren kann. Hervorgerufen wird eine Phasentransition durch die Abarbeitung der Aktion A4 oder A5 in einer Komponente des zugrunde liegenden $\mathrm{V_I}$Systems. Im Gegensatz zum Verhalten $\mathcal{V}[\![\cdot]\!]$ ist es bei der Casetrace-Semantik nicht möglich, zwischen freien und erzwungenen Phasentransitionen zu unterscheiden, da die Phasenqualitäten als Markierungshilfe wegfallen.

Betrachtet man alle Casefolgen, die die Casetrace-Semantik eines I–Systems bilden, so lassen sich bestimmte Eigenschaften angeben, die für alle diese Folgen gelten. Es besteht eine Abhängigkeit zur Struktur des zugrunde liegenden I–Systems, insbesondere der Kopplungs- und Erregungsrelation. Bezieht man sich auf Elemente dieser Relationen, lassen sich Aussagen machen über die Zugehörigkeit von bestimmten Phasen zu aufeinander folgenden Cases. Verantwortlich für den charakteristischen Aufbau der Casefolgen ist die Aktivität des zugeordneten $\mathrm{V_I}$Systems, die bestimmt wird durch die Aktionen A1-A13.
Der folgende Satz beschreibt grundlegende Zusammenhänge zwischen der formalen Struktur eines beliebigen I–Systems IS und dessen Casetrace-Semantik $\mathcal{CT}[\![IS]\!]$.

Satz 4.12 (Charakterisierung der Casetrace-Semantik). Sei $IS = (P, B, \underline{B}, K, E)$ ein I–System und $c_0 c_1 c_2 \ldots \in \mathcal{CT}[\![IS]\!]$ ein Element der Casetrace-Semantik mit $c_j \in Case(IS)$, $j = 0, 1, 2, \ldots$. Seien $p, q, v, w \in P$ mit $b(p) = b(q)$ und $b(v) = b(w)$. Dann gelten folgende Aussagen für alle $i = 1, 2, \ldots$:

a) *Stop*
$(p, v) \in E \wedge \{p, v\} \subseteq c_{i-1} \quad \Rightarrow \quad p \in c_i$

b) *Nicht-Koinzidenz*
$(p, w) \in K \wedge (q, v) \in K \wedge \{p, v\} \subseteq c_{i-1} \quad \Rightarrow \quad \{q, w\} \not\subseteq c_i$

Beweis. Es gelten die Bezeichnungen und Voraussetzungen aus dem Satz. Sei eine beliebige Ausführung Π von $V_I System(IS)$ gegeben, die gemäß Definition 4.8 $c_0 c_1 c_2 \dots$ erzeugt. (Unter Bezugnahme auf Π wird im Beweis die Notation 3.12 verwendet.) Seien $t_0, t_1, t_2, \dots$ die durch die Definition gegebenen Globalzeitpunkte und $i \in \{1, 2, \dots\}$ beliebig aber fest, jeweils aus dem zugeordneten Wertebereich.

Im Folgenden macht man sich die Charakterisierung des Verhaltens $\mathcal{V}[\![IS]\!]$ durch Satz 4.6 zunutze. Es muss nur noch formal der Übergang von $\mathcal{CT}[\![IS]\!]$ zu $\mathcal{V}[\![IS]\!]$ durchgeführt werden.

Zu **a**). Sei $(p, v) \in E$. Annahme: $p \notin c_i$.

Nach den Definitionen von $\mathcal{CT}[\![\cdot]\!]$ und $zc(\cdot)$ existieren Zeitpunkte t^1, t^2 mit $t^1 < t^2$, $t_{i-1} \leq t^2 < t_i$, $\forall t \in [t^2, t_i[\; : z^t \langle V_I System(IS) \rangle = z^{t^2} \langle V_I System(IS) \rangle$ sowie $\forall t' \in [t^1, t^2[\; : z^{t'} \langle V_I System(IS) \rangle \neq z^{t^2} \langle V_I System(IS) \rangle$.

Bemerkung: Im Fall $i = 1$ gilt: $t_0 \leq t^1 < t^2 < t_i$, da $z^{t_0} \langle V_I System(IS) \rangle$ stabil ist als Startzustand und von t_0 nach t_1 eine Phasentransition bei $V_{b(p)}$ stattfindet, was mindestens zwei Änderungen von Phasenqualitäten erfordert. Im Fall $i > 1$ gilt immer $t^1 \geq t_0$ wegen $z^{t_0} \langle V_I System(IS) \rangle \neq z^{t_1} \langle V_I System(IS) \rangle$. Die Existenz von t^1 ist demnach garantiert.

Wegen $\{p, v\} \subseteq c_{i-1}$ und $p \notin c_i$ gilt: $z^{t^2} \langle V_I System(IS) \rangle (p) \neq 0$, $z^{t^2} \langle V_I System(IS) \rangle (v) \neq 0$, $z^{t_i} \langle V_I System(IS) \rangle (p) = 0$.

$\Rightarrow$ {Definition $\mathcal{V}[\![\cdot]\!]$}
$\exists ztr_1, ztr_2 \in GZustand(IS)^* : ztr_1.z^{t^2} \langle V_I System(IS) \rangle . z^{t_i} \langle V_I System(IS) \rangle . ztr_2 \in \mathcal{V}[\![IS]\!]$ und es gilt: $z^{t^2} \langle V_I System(IS) \rangle (p) \neq 0$, $z^{t^2} \langle V_I System(IS) \rangle (v) \neq 0$, $z^{t_i} \langle V_I System(IS) \rangle (p) = 0$.

$\Rightarrow$ {Satz 4.6}
Wegen Teil a) des Satzes 4.6 muss auch $z^{t_i} \langle V_I System(IS) \rangle (p) \neq 0$ erfüllt sein. Damit liegt ein Widerspruch vor, und die Annahme ist falsch. Folglich gilt $p \in c_i$.

Zu **b**). Sei $(p, w) \in K$ und $(q, v) \in K$. Annahme: $\{q, w\} \subseteq c_i$.

Seien die Zeitpunkte t^1 und t^2 gegeben wie im Beweisteil zu a). Wegen $\{p, v\} \subseteq c_{i-1}$ und $\{q, w\} \subseteq c_i$ gilt nun: $z^{t^2} \langle V_I System(IS) \rangle (p) \neq 0$, $z^{t^2} \langle V_I System(IS) \rangle (v) \neq 0$, $z^{t_i} \langle V_I System(IS) \rangle (q) \neq 0$, $z^{t_i} \langle V_I System(IS) \rangle (w) \neq 0$.

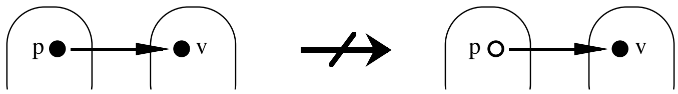
$\Rightarrow$ {Definition $\mathcal{V}[\![\cdot]\!]$}
$\exists ztr_1, ztr_2 \in GZustand(IS)^* : ztr_1.z^{t^2} \langle V_I System(IS) \rangle . z^{t_i} \langle V_I System(IS) \rangle . ztr_2 \in \mathcal{V}[\![IS]\!]$ und es gilt: $z^{t^2} \langle V_I System(IS) \rangle (p) \neq 0$, $z^{t^2} \langle V_I System(IS) \rangle (v) \neq 0$, $z^{t_i} \langle V_I System(IS) \rangle (q) \neq 0$, $z^{t_i} \langle V_I System(IS) \rangle (w) \neq 0$.

$\Rightarrow$ {Satz 4.6}
Wegen Teil c) des Satzes 4.6 muss auch $z^{t_i} \langle V_I System(IS) \rangle (q) = 0$ oder $z^{t_i} \langle V_I System(IS) \rangle (w) = 0$ erfüllt sein. Damit liegt ein Widerspruch vor, und die Annahme ist falsch. Folglich gilt $\{q, w\} \not\subseteq c_i$.
$\square$

Die anschaulichen Interpretationen der Kernaussagen des Satzes 4.12 entsprechen denen von Teilaussagen des Satzes 4.6. Das liegt daran, dass die globalen Aktivitätszustände als Verfeinerungen der Cases aufzufassen sind (gemäß Kapitel 3.1) und das Verhalten somit eine Verfeinerung der Casetrace-Semantik darstellt. Zu Interpretationen beim Verhalten, die die genauen Phasenqualitäten der auftretenden globalen Aktivitätszustände dahingehend vernachlässigen, dass nur von Interesse ist, ob eine Phasenqualität gleich oder ungleich 0 ist, findet sich eine korrespondierende Formulierung bei der Casetrace-Semantik. Zusammenfassend ergibt sich für die beiden Aussagen aus Satz 4.12:

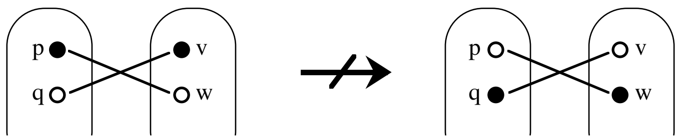
a) beschreibt die *Aufrechterhaltung eines Einflusses* zwischen zwei aktuell eingenommenen Phasen in benachbarten Komponenten. Die dabei zugrunde liegende Idee der Modellierung wirkender Einflüsse bzw. Zwänge, spezifiziert über die Erregungsrelation, wurde bei der Interpretation von Satz 4.6.a bereits vorgestellt. Als Ergebnis kann eine Zwang ausübende (erregende) Phase erst wieder verlassen werden, nachdem sich der Zwang ausgewirkt hat, d.h. die erregte Phase verlassen wurde.

Graphisch betrachtet ist somit der folgende Ausschnitt einer Phasentransition, bei der p verlassen wird, nicht möglich.



Da bei der Casetrace-Semantik nur (Folgen von) Cases betrachtet werden, entfällt in der obigen Graphik die Angabe möglicher Phasenqualitäten.

b) verdeutlicht, dass *koinzidente Phasentransitionen bei über Kreuz wechselseitig ausgeschlossenen Nachbarphasen nicht auftreten können.* Diese Eigenschaft korrespondiert zu Satz 4.6.c und begründet sich daraus, dass die globale Zeitgleichheit zweier Ereignisse in unterschiedlichen Komponenten in einem verteilten System nicht garantiert werden kann. Jegliche Hintereinanderausführung der Phasentransitionen ist durch den vorgegebenen kreuzweisen wechselseitigen Ausschluss ausgeschlossen.
In der folgenden Graphik sind die Phasentransitionen $p \to q$ und $v \to w$ sowohl gleichzeitig, als auch beliebig hintereinander ausgeführt, ausgeschlossen.



Wie schon erwähnt wurde, korrespondieren die Aussagen a) und b) des Satzes 4.12 zu den Aussagen a) und c) des Satzes 4.6. Für die Aussagen b) und d) des Satzes 4.6 existieren hingegen keine entsprechenden Aussagen auf der Basis der Casetrace-Semantik. Der Grund dafür ist, dass die Konzepte von Stabilität/Instabilität von Phasen oder dem Treffen von Entscheidungen nicht auf die Casetrace-Semantik übertragen werden können. Hierzu sind „Zusatzmarkierungen“ bei den Cases notwendig, Phasenqualitäten in der Terminologie der globalen Aktivitätszustände. Die Casetrace-Semantik bietet sich an, wenn ausschließlich kausale Abhängigkeiten zwischen Phasen in unterschiedlichen Komponenten eines verteilten Systems analysiert werden sollen (vgl. [78]). Bei der Modellierung von verteilten Systemen, bei denen die Existenz oder auch die Nicht-Existenz von wechselseitigen Einflüssen zwischen den Komponenten untersucht werden soll (vgl. [88]), ist die Aussagekräftigkeit der Casetrace-Semantik in der Regel zu schwach.

Ist bereits das Verhalten $\mathcal{V}[\![IS]\!]$ eines I–Systems IS bekannt, dann kann daraus auch direkt die Casetrace-Semantik $\mathcal{CT}[\![IS]\!]$ abgeleitet werden. Die *zc-Transformation* dient dabei der Umwandlung von einer Folge von globalen Aktivitätszuständen in eine Folge von aufeinander folgend unterschiedlicher Cases.

Definition 4.13 (zc-Transformation). Sei $IS \in ISystem$ und $ztr := z_0 z_1 z_2 \ldots \in GZustand(IS)^+$. Sei $M := \{i_1, i_2, i_3, \ldots\} \subseteq I\!N$, $i_1 < i_2 < i_3 < \ldots$, genau die Indexmenge, für die gilt: $j \in M$ gdw. $zc(z_j) \neq zc(z_{j-1})$. Die *zc-Transformation* $\lfloor\cdot\rfloor$ wird festgelegt als

$$\lfloor\cdot\rfloor : \qquad GZustand(IS)^+ \longrightarrow Case(IS)^+$$

mit (ztr wie oben):

$$\lfloor ztr \rfloor = zc(z_0) zc(z_{i_1}) zc(z_{i_2}) zc(z_{i_3}) \ldots \qquad \square$$

Satz 4.14 (Alternative Definition der Casetrace-Semantik). Sei IS ein I–System. Es gilt:

a) $\mathcal{CT}[\![IS]\!](c_0) = \{ctr \in Case(IS)^+ \mid \exists ztr \in \mathcal{V}[\![IS]\!](cz(c_0)) : ctr = \lfloor ztr \rfloor\}$

b) $\mathcal{CT}[\![IS]\!] = \{ctr \in Case(IS)^+ \mid \exists ztr \in \mathcal{V}[\![IS]\!] : ctr = \lfloor ztr \rfloor\}$

Beweis. Durch Umformulierungen wird jeweils eine Gleichungskette erzeugt, die das gewünschte Ergebnis liefert. (Unter Bezugnahme auf IS wird die Notation 3.12 verwendet, d.h. $z^{\Pi,t}$ steht für $z^{\Pi,t}\langle V_I System(IS)\rangle$.)

Zu **a)**.

$\{ctr \in Case(IS)^+ \mid \exists ztr \in \mathcal{V}[\![IS]\!](cz(c_0)) : ctr = \lfloor ztr \rfloor\}$

$= \{$Definition von $\mathcal{V}[\![\cdot]\!](\cdot)$, $cz(c_0) \in StabGZustand(IS)$ nach Definition 3.6.b$\}$
$\{\lfloor ztr \rfloor \mid ztr \in \{z^{\Pi,t_0}.z^{\Pi,t_1}.z^{\Pi,t_2}.\ldots \in GZustand(IS)^+ \mid \Pi$ ist eine Ausführung von $V_I System(IS)$ und $t_0, t_1, t_2, \ldots$ eine max. Folge von Globalzeitpunkten der Ausführung mit: t_0 ist Startzeitpunkt, $z^{\Pi,t_0} = cz(c_0)$, $\forall i = 1, 2, \ldots : (t_{i-1} < t_i,\, z^{\Pi,t_{i-1}} \neq z^{\Pi,t_i},\, \forall t, t_{i-1} \leq t < t_i : z^{\Pi,t} = z^{\Pi,t_{i-1}})\}\}$

$= \{$Umformung$\}$
$\{\lfloor z^{\Pi,t_0}.z^{\Pi,t_1}.z^{\Pi,t_2}.\ldots \rfloor \mid \Pi$ ist eine Ausführung von $V_I System(IS)$ und $t_0, t_1, t_2, \ldots$ eine max. Folge von Globalzeitpunkten der Ausführung mit: t_0 ist Startzeitpunkt, $z^{\Pi,t_0} = cz(c_0)$, $\forall i = 1, 2, \ldots : (t_{i-1} < t_i,\, z^{\Pi,t_{i-1}} \neq z^{\Pi,t_i},\, \forall t, t_{i-1} \leq t < t_i : z^{\Pi,t} = z^{\Pi,t_{i-1}})\}$

$= \{$Definition von $\lfloor \cdot \rfloor\}$
$\{zc(z^{\Pi,t_0}).zc(z^{\Pi,t_{l_1}}).zc(z^{\Pi,t_{l_2}}).\ldots \mid \Pi$ ist eine Ausführung von $V_I System(IS)$ und $t_0, t_1, t_2, \ldots$ eine max. Folge von Globalzeitpunkten der Ausführung mit: t_0 ist Startzeitpunkt, $z^{\Pi,t_0} = cz(c_0)$, $\forall i = 1, 2, \ldots : (t_{i-1} < t_i,\, z^{\Pi,t_{i-1}} \neq z^{\Pi,t_i},\, \forall t, t_{i-1} \leq t < t_i : z^{\Pi,t} = z^{\Pi,t_{i-1}})$, $\{l_1, l_2, \ldots\} \subseteq I\!N$, $l_1 < l_2 < \ldots$, $j \in \{l_1, l_2, \ldots\} \Leftrightarrow zc(z^{\Pi,t_{j-1}}) \neq zc(z^{\Pi,t_j})\}$

$= \{$Entsprechend der Definition von $zc(\cdot)$ gilt: $(zc(z^{\Pi,t_{j-1}}) \neq zc(z^{\Pi,t_j})) \Rightarrow (z^{\Pi,t_{j-1}} \neq z^{\Pi,t_j})$ und $(z^{\Pi,t} = z^{\Pi,t_i}) \Rightarrow (zc(z^{\Pi,t}) = zc(z^{\Pi,t_i}))$. Beachte $\{l_1, l_2, \ldots\} \subseteq \{1, 2, \ldots\}$. Setze $t_{l_0} := t_0$.$\}$
$\{zc(z^{\Pi,t_{l_0}}).zc(z^{\Pi,t_{l_1}}).zc(z^{\Pi,t_{l_2}}).\ldots \mid \Pi$ ist eine Ausführung von $V_I System(IS)$ und $t_0, t_1, t_2, \ldots$ eine max. Folge von Globalzeitpunkten der Ausführung mit: t_{l_0} ist Startzeitpunkt, $z^{\Pi,t_{l_0}} = cz(c_0)$, $l_0 = 0$, $\{l_1, l_2, \ldots\} \subseteq I\!N$, $l_1 < l_2 < \ldots$, $\forall i = 1, 2, \ldots : (t_{l_{i-1}} < t_{l_i},\, zc(z^{\Pi,t_{l_{i-1}}}) \neq zc(z^{\Pi,t_{l_i}})$, $\forall t, t_{l_{i-1}} \leq t < t_{l_i} : zc(z^{\Pi,t}) = zc(z^{\Pi,t_{l_{i-1}}}))\}$

$= \{$Umindizierung mit i für l_i, Vereinfachung$\}$
$\{zc(z^{\Pi,t_0}).zc(z^{\Pi,t_1}).zc(z^{\Pi,t_2}).\ldots \mid \Pi$ ist eine Ausführung von $V_I System(IS)$ und $t_0, t_1, t_2, \ldots$ eine max. Folge von Globalzeitpunkten der Ausführung mit: t_0 ist Startzeitpunkt, $z^{\Pi,t_0} = cz(c_0)$, $\forall i = 1, 2, \ldots : (t_{i-1} < t_i,\, zc(z^{\Pi,t_{i-1}}) \neq zc(z^{\Pi,t_i}),\, \forall t, t_{i-1} \leq t < t_i : zc(z^{\Pi,t}) = zc(z^{\Pi,t_{i-1}}))\}$

$= \{$Definition 4.8$\}$
$\mathcal{CT}[\![IS]\!](c_0)$

Zu **b)**.

$\{ctr \in Case(IS)^+ \mid \exists ztr \in \mathcal{V}[\![IS]\!] : ctr = \lfloor ztr \rfloor\}$

$= \{$Definition von $\mathcal{V}[\![IS]\!]\}$
$$\{ctr \in Case(IS)^+ \mid \exists ztr \in \bigcup_{z_0 \in StabGZustand(IS)} \mathcal{V}[\![IS]\!](z_0) : ctr = \lfloor ztr \rfloor\}$$

$= \{$Aus den Definitionen von $cz(\cdot)$ und $StabGZustand(\cdot)$ folgt direkt: $\{cz(c_0) \mid c_0 \in Case(IS)\} = StabGZustand(IS)$.$\}$
$$\{ctr \in Case(IS)^+ \mid \exists ztr \in \bigcup_{c_0 \in Case(IS)} \mathcal{V}[\![IS]\!](cz(c_0)) : ctr = \lfloor ztr \rfloor\}$$

$= \{$Umformung$\}$
$$\bigcup_{c_0 \in Case(IS)} \{ctr \in Case(IS)^+ \mid \exists ztr \in \mathcal{V}[\![IS]\!](cz(c_0)) : ctr = \lfloor ztr \rfloor\}$$

$= \{$Teil a) des Satzes$\}$
$$\bigcup_{c_0 \in Case(IS)} \mathcal{CT}[\![IS]\!](c_0)$$

$= \{$Definition 4.8$\}$
$\mathcal{CT}[\![IS]\!]$ □

Die Bestimmung der Casetrace-Semantik eines I–Systems mit Hilfe dessen Verhaltens, entsprechend Satz 4.14, erfolgt auf der formalen Ebene. Dies ist der Unterschied zu Definition 4.8.

Dort wird auf die Ausführungen des zugeordneten $\mathrm{V_I}$Systems und damit auf die algorithmische Ebene zurückgegriffen. Mit Hilfe von Satz 4.14 lassen sich vorliegende Aussagen und Beweise über semantische Eigenschaften des Verhaltens auf die Casetrace-Semantik übertragen, ohne erneut die Aktionen des $\mathrm{V_I}$Systems untersuchen zu müssen. Zudem ist der Satz hilfreich beim Vergleich verschiedener Semantiken für I–Systeme. Beides wird in den nächsten Kapiteln deutlich werden. Die Definition 4.8 der Casetrace-Semantik bietet sich an, wenn auf bestimmte Globalzeitpunkte der Ausführungen des $\mathrm{V_I}$Systems Bezug genommen werden muss. Die Zeitpunkte des Auftretens neuer Cases, abgeleitet aus den z-Globalbelegungen, dienen als Ausgangspunkt bei der Bestimmung möglicher weiterer Aktivitäten innerhalb des $\mathrm{V_I}$Systems, was bei Beweisen semantischer Eigenschaften Anwendung findet (z.B. beim Beweis von Satz 4.10).

Bemerkung 4.15. Ist die Casetrace-Semantik $\mathcal{CT}[\![IS]\!]$ eines I–Systems IS bekannt, lässt sich daraus *nicht* das Verhalten $\mathcal{V}[\![IS]\!]$ berechnen.

Dieses gilt offensichtlich, da die globalen Aktivitätszustände eines I–Systems (d.h. die Elemente der Traces des Verhaltens) aufgrund der Phasenqualitäten mehr Informationen beinhalten als die Cases (die Elemente der Traces der Casetrace-Semantik).

4.4 Erweiterte Casetrace-Semantik

Wie schon bei der Casetrace-Semantik (Abschnitt 4.3), sind Cases als globale Systemzustände eines I–Systems das zentrale Strukturelement für die Konstruktion der *Erweiterten Casetrace-Semantik*. Über Folgen von Cases, die sich aus den Ausführungen des zugeordneten $\mathrm{V_I}$Systems $V_I System(IS)$ ergeben, definieren sich wiederum die Aktivitäten in den Bereichen eines I–Systems IS in Form von Phasentransitionen.
Nun wurde in den vorherigen Kapiteln bereits erwähnt, dass man bei den Ereignissen in einem verteilten System unterscheiden kann zwischen Ereignissen, die eintreten können (oder auch nicht), und Ereignissen, die aufgrund eines zwingenden externen Einflusses eintreten werden. Ist es bei einer Analyse eines I–Systems erforderlich, die Unterscheidung der Ereignisse zu berücksichtigen (z.B. um Auswirkungen von Einflüssen untersuchen zu können), dann unterliegt man der Notwendigkeit, Phasentransitionen über die Semantik klassifizieren zu können. Die „normale“ Casetrace-Semantik erlaubt diesbezüglich keine Einteilung. Um nun den Typ einer Phasentransition innerhalb der Semantik zu erfassen, wird zwischen zwei aufeinander folgenden Cases eine Zusatzinformation eingefügt, die erkennen lässt, durch welche Aktionen beim zugeordneten $\mathrm{V_I}$System eine Phasentransition bewirkt wird, welche sich auf der formalen Ebene als Casewechsel bemerkbar macht. Da für Phasentransitionen beim $\mathrm{V_I}$System ausschließlich die Aktionen A4 (Phasentransition wegen Entscheidung) oder A5 (Phasentransition wegen Erregung) verantwortlich sind, reicht es, sich auf einen Fall zu beschränken und o.B.d.A. die Bereiche b anzugeben, deren zugeordnete Komponenten V_b die Aktion A4 ausführen. Die A5-Fälle ergeben sich dann automatisch als Komplement.

Definition 4.16 (Erweiterte Casetrace-Semantik). Für ein I–System $IS \in ISystem$ ist die *Erweiterte Casetrace-Semantik* $\mathcal{ECT}[\![IS]\!]$ definiert als:

$$\mathcal{ECT}[\![IS]\!] \subseteq Case(IS).\,(\mathcal{P}(B).Case(IS))^*$$

mit

$$\mathcal{ECT}[\![IS]\!] = \bigcup_{c_0 \in Case(IS)} \mathcal{ECT}[\![IS]\!](c_0)$$

Die Erweiterte Casetrace-Semantik *bzgl. eines Start-Cases* $c_0 \in Case(IS)$ ist dabei gegeben durch:

$\mathcal{ECT}[\![IS]\!](c_0) =$

$\{zc(z^{\Pi,t_0}\langle V_I System(IS)\rangle).\delta_1.zc(z^{\Pi,t_1}\langle V_I System(IS)\rangle).\delta_2.zc(z^{\Pi,t_2}\langle V_I System(IS)\rangle)\ldots \in Case(IS).(\mathcal{P}(B).Case(IS))^* \mid \Pi$ ist eine Ausführung von $V_I System(IS)$ und $t_0, t_1, t_2, \ldots$ eine max. Folge von Globalzeitpunkten der Ausführung mit: t_0 ist Startzeitpunkt, $z^{\Pi,t_0}\langle V_I System(IS)\rangle = cz(c_0)$, $\forall i = 1,2,\ldots : (t_{i-1} < t_i$, $zc(z^{\Pi,t_{i-1}}\langle V_I System(IS)\rangle) \neq zc(z^{\Pi,t_i}\langle V_I System(IS)\rangle)$, $\forall t, t_{i-1} \leq t < t_i$: $zc(z^{\Pi,t}\langle V_I System(IS)\rangle) = zc(z^{\Pi,t_{i-1}}\langle V_I System(IS)\rangle)$, $\delta_i = \{b(p) \in AB(IS) \mid p \in zc(z^{\Pi,t_i}\langle V_I System(IS)\rangle) \setminus zc(z^{\Pi,t_{i-1}}\langle V_I System(IS)\rangle)$ und $V_{b(p)}$ befindet sich zum Zeitpunkt t_i in Aktion A4.$\}$ $)\}$

□

Die Wohldefiniertheit der Erweiterten Casetrace-Semantik basiert auf der Wohldefiniertheit der Casetrace-Semantik. Definition 4.16 ist eine Erweiterung von Definition 4.8. Auf die Wohldefiniertheit von $zc(z^{\Pi,t_j}\langle V_I System(IS)\rangle)$ (für $j = 0,1,2,\ldots$) als Case von IS und damit auch auf die Stabilität von $cz(c_0)$ hat die Erweiterung keinen Einfluss. Hinzu kommen lediglich die Festlegungen für δ_i. Die Mengensubtraktion $zc(z^{\Pi,t_i}\langle V_I System(IS)\rangle) \setminus zc(z^{\Pi,t_{i-1}}\langle V_I System(IS)\rangle)$ aus der obigen Definition, als eine Subtraktion von unterschiedlichen Cases, hat eine nichtleere Menge von Phasen p als Ergebnis. $b(p)$ ist der Bereich von p und δ_i folglich wie angegeben eine (möglicherweise leere) Menge von Bereichen. $V_{b(p)}$ bezeichnet (gemäß Definition 3.9) die dem Bereich $b(p)$ zugeordnete Komponente im zugeordneten $\mathrm{V_I}$System. Das Verhalten der Komponenten eines $\mathrm{V_I}$Systems ist durch die Aktionen A1-A13 festgelegt. Die Formulierung „$V_{b(p)}$ befindet sich zum Zeitpunkt t_i in Aktion A4.“ ist somit erlaubt. Die leere Erweiterte Casetrace-Semantik, also $\mathcal{ECT}[\![IS]\!] = \emptyset$, kann auftreten, wenn es keinen Case des I–Systems IS gibt.

Das folgende Beispiel beschreibt eine Anwendung der Definition der Erweiterten Casetrace-Semantik.

Beispiel 4.17. Bestimmt werden soll ein Element der Erweiterten Casetrace-Semantik $\mathcal{ECT}[\![IS_1]\!]$ des I–Systems IS_1 aus Beispiel 2.2. Betrachtet wird hierzu eine weitere Ausführung von $V_I System(IS_1)$, in der Notation analog zu Beispiel 4.2.

Ausführung 3:

$$\overbrace{[p_1{<}1{>}, q_2{<}1{>}, v_2{<}1{>}]}^{z_0} \xrightarrow{b_1.\mathrm{A3}, b_2.\mathrm{A3}} \overbrace{[p_1{<}p_2{>}, q_2{<}q_3{>}, v_2{<}1{>}]}^{z_1} \xrightarrow{b_1.\mathrm{A4}, b_2.\mathrm{A4}}$$

$$\overbrace{[p_2{<}1{>}, q_3{<}1{>}, v_2{<}1{>}]}^{z_2} \xrightarrow{\underline{b_3}.\mathrm{A10}} \overbrace{[p_2{<}1{>}, q_3{<}1{>}, v_2{<}\mathsf{F}{>}]}^{z_3} \xrightarrow{\underline{b_3}.\mathrm{A5}} \overbrace{[p_2{<}1{>}, q_3{<}1{>}, v_1{<}1{>}]}^{z_4}$$

$$\xrightarrow{b_2.\mathrm{A3}} \overbrace{[p_2{<}1{>}, q_3{<}q_2{>}, v_1{<}1{>}]}^{z_5} \xrightarrow{b_2.\mathrm{A4}} \overbrace{[p_2{<}1{>}, q_2{<}1{>}, v_1{<}1{>}]}^{z_6} \xrightarrow{b_1.\mathrm{A3}}$$

$$\overbrace{[p_2{<}p_1{>}, q_2{<}1{>}, v_1{<}1{>}]}^{z_7} \xrightarrow{b_2.\mathrm{A3}, \underline{b_3}.\mathrm{A11}} \overbrace{[p_2{<}p_1{>}, q_2{<}q_3{>}, v_1{<}\mathsf{F}{>}]}^{z_8} \xrightarrow{b_2.\mathrm{A4}, \underline{b_3}.\mathrm{A5}}$$

$$\overbrace{[p_2{<}p_1{>}, q_3{<}1{>}, v_2{<}1{>}]}^{z_9} \xrightarrow{b_1.\mathrm{A4}, \underline{b_3}.\mathrm{A10}} \overbrace{[p_1{<}1{>}, q_3{<}1{>}, v_2{<}\mathsf{F}{>}]}^{z_{10}} \xrightarrow{b_1.\mathrm{A11}} \overbrace{[p_1{<}\mathsf{F}{>}, q_3{<}1{>}, v_2{<}\mathsf{F}{>}]}^{z_{11}}$$

$$\xrightarrow{b_1.\mathrm{A5}} \overbrace{[p_2{<}1{>}, q_3{<}1{>}, v_2{<}\mathsf{F}{>}]}^{z_{12}} \xrightarrow{\underline{b_3}.\mathrm{A5}} \overbrace{[p_2{<}1{>}, q_3{<}1{>}, v_1{<}1{>}]}^{z_{13}}$$

Betrachte die mittels $zc(\cdot)$ abgeleiteten Cases sowie die Aktionen bei den Komponenten von $V_I System(IS_1)$, die zu Caseübergängen führen (angegeben als Pfeilbeschriftungen):

$$\overbrace{\{p_1, q_2, v_2\}}^{c_0} = zc(z_0) = zc(z_1) \neq zc(z_2) = \overbrace{\{p_2, q_3, v_2\}}^{c_1}, \qquad c_1 \setminus c_0 = \{p_2, q_3\}$$

$$\left.\begin{array}{l} V_{b_1} \text{ befindet sich zum Zeitpunkt des Casewechsels (von } c_0 \text{ nach } c_1\text{) in Aktion A4.} \\ V_{b_2} \text{ befindet sich zum Zeitpunkt des Casewechsels (von } c_0 \text{ nach } c_1\text{) in Aktion A4.} \\ \text{Die Struktur von } IS_1 \text{ liefert: } b_1 = b(p_2),\ b_2 = b(q_3),\ \{b_1, b_2\} \subseteq AB(IS_1) \end{array}\right\} \delta_1 := \{b_1, b_2\}$$

$$zc(z_2) = zc(z_3) \neq zc(z_4) = \overbrace{\{p_2, q_3, v_1\}}^{c_2}, \qquad c_2 \setminus c_1 = \{v_1\}$$
$$\left.\begin{array}{l}V_{\underline{b}_3} \text{ befindet sich zum Zeitpunkt des Casewechsels (von } c_1 \text{ nach } c_2\text{) in Aktion A5.}\\ \text{Die Struktur von } IS_1 \text{ liefert: } \underline{b}_3 = b(v_1),\ \underline{b}_3 \notin AB(IS_1)\end{array}\right\} \delta_2 := \{\}$$

$$zc(z_4) = zc(z_5) \neq zc(z_6) = \overbrace{\{p_2, q_2, v_1\}}^{c_3}, \qquad c_3 \setminus c_2 = \{q_2\}$$
$$\left.\begin{array}{l}V_{b_2} \text{ befindet sich zum Zeitpunkt des Casewechsels (von } c_2 \text{ nach } c_3\text{) in Aktion A4.}\\ \text{Die Struktur von } IS_1 \text{ liefert: } b_2 = b(q_2),\ b_2 \in AB(IS_1)\end{array}\right\} \delta_3 := \{b_2\}$$

$$zc(z_6) = zc(z_7) = zc(z_8) \neq zc(z_9) = \overbrace{\{p_2, q_3, v_2\}}^{c_4}, \qquad c_4 \setminus c_3 = \{q_3, v_2\}$$
$$\left.\begin{array}{l}V_{b_2} \text{ befindet sich zum Zeitpunkt des Casewechsels (von } c_3 \text{ nach } c_4\text{) in Aktion A4.}\\ V_{\underline{b}_3} \text{ befindet sich zum Zeitpunkt des Casewechsels (von } c_3 \text{ nach } c_4\text{) in Aktion A5.}\\ \text{Die Struktur von } IS_1 \text{ liefert: } b_2 = b(q_3),\ \underline{b}_3 = b(v_2),\ b_2 \in AB(IS_1),\ \underline{b}_3 \notin AB(IS_1)\end{array}\right\} \delta_4 := \{b_2\}$$

$$zc(z_9) \neq zc(z_{10}) = \overbrace{\{p_1, q_3, v_2\}}^{c_5}, \qquad c_5 \setminus c_4 = \{p_1\}$$
$$\left.\begin{array}{l}V_{b_1} \text{ befindet sich zum Zeitpunkt des Casewechsels (von } c_4 \text{ nach } c_5\text{) in Aktion A4.}\\ \text{Die Struktur von } IS_1 \text{ liefert: } b_1 = b(p_1),\ b_1 \in AB(IS_1)\end{array}\right\} \delta_5 := \{b_1\}$$

$$zc(z_{10}) = zc(z_{11}) \neq zc(z_{12}) = \overbrace{\{p_2, q_3, v_2\}}^{c_6}, \qquad c_6 \setminus c_5 = \{p_2\}$$
$$\left.\begin{array}{l}V_{b_1} \text{ befindet sich zum Zeitpunkt des Casewechsels (von } c_5 \text{ nach } c_6\text{) in Aktion A5.}\\ \text{Die Struktur von } IS_1 \text{ liefert: } b_1 = b(p_2),\ b_1 \in AB(IS_1)\end{array}\right\} \delta_6 := \{\}$$

$$zc(z_{12}) \neq zc(z_{13}) = \overbrace{\{p_2, q_3, v_1\}}^{c_7}, \qquad c_7 \setminus c_6 = \{v_1\}$$
$$\left.\begin{array}{l}V_{\underline{b}_3} \text{ befindet sich zum Zeitpunkt des Casewechsels (von } c_6 \text{ nach } c_7\text{) in Aktion A5.}\\ \text{Die Struktur von } IS_1 \text{ liefert: } \underline{b}_3 = b(v_1),\ \underline{b}_3 \notin AB(IS_1)\end{array}\right\} \delta_7 := \{\}$$

Nach Definition 4.16 gilt: $c_0\delta_1c_1\delta_2c_2\delta_3c_3\delta_4c_4\delta_5c_5\delta_6c_6\delta_7c_7 \in \mathcal{ECT}[\![IS_1]\!]$ □

Wie schon bei dem Verhalten und der Casetrace-Semantik eines I–Systems, repräsentieren aufeinander folgende Elemente in einer Trace der Erweiterten Casetrace-Semantik eine bestimmte Aktivität in den Bereichen des I–Systems. Analog zur Casetrace-Semantik lassen sich über zwei aufeinander folgende Cases Phasentransitionen definieren. Die bei der Erweiterten Casetrace-Semantik zusätzlich zur Verfügung stehenden dazwischenliegenden Bereichsmengen δ_i werden benutzt, um die Ursache der Phasentransitionen genauer zu spezifizieren.

Definition 4.18 (Freie/Erzwungene Phasentransition). Sei IS ein I–System, b ein Bereich von IS und $p, q \in b$. Sei $c_0\delta_1c_1 \ldots \delta_{i-1}c_{i-1}\delta_ic_i \ldots \in \mathcal{ECT}[\![IS]\!]$ eine beliebige Trace der Erweiterten Casetrace-Semantik mit $c_0, c_j \in Case(IS)$, $\delta_j \subseteq B$, $j = 1, 2, \ldots$ und $i \in \mathbb{N}$. Die Teiltrace $c_{i-1}\delta_ic_i$ repräsentiert folgende Ereignisse:

a) Bei $p \in \{c_{i-1} \setminus c_i\} \wedge q \in \{c_i \setminus c_{i-1}\} \wedge b \in \delta_i$ sagen wir, dass eine *freie Phasentransition* von p nach q in b auftritt.

b) Bei $p \in \{c_{i-1} \setminus c_i\} \wedge q \in \{c_i \setminus c_{i-1}\} \wedge b \notin \delta_i$ sagen wir, dass eine *erzwungene Phasentransition* von p nach q in b auftritt. □

Für das Beispiel 4.17 mit der Trace $c_0\delta_1c_1 \ldots \delta_7c_7$ der Erweiterten Casetrace-Semantik $\mathcal{ECT}[\![IS_1]\!]$ des I–Systems IS_1 bedeutet Definition 4.18, dass die Teiltrace $c_0\delta_1c_1$ das Auftreten von zwei freien Phasentransitionen, nämlich von p_1 nach p_2 in b_1 und von q_2 nach q_3 in b_2, repräsentiert. Die Teiltrace $c_1\delta_2c_2$ repräsentiert eine erzwungene Phasentransition von v_2 nach v_1 in $\underline{b}_3$ und $c_2\delta_3c_3$ eine freie Phasentransition von q_3 nach q_2 in b_2. Eine freie Phasentransition von q_2 nach q_3 in b_2 und gleichzeitig eine erzwungene Phasentransition von v_1 nach v_2 in $\underline{b}_3$ wird durch $c_3\delta_4c_4$ beschrieben. $c_4\delta_5c_5$ repräsentiert das Auftreten einer freien Phasentransition von p_2 nach p_1 in b_1, $c_5\delta_6c_6$ das Auftreten einer erzwungenen Phasentransition von p_1 nach p_2 in b_1, und $c_6\delta_7c_7$ das Auftreten einer erzwungenen Phasentransition von v_2 nach v_1 in $\underline{b}_3$.

Im Gegensatz zur Casetrace-Semantik $\mathcal{CT}[\![\cdot]\!]$ erlaubt es die Erweiterte Casetrace-Semantik $\mathcal{ECT}[\![\cdot]\!]$ wie gezeigt, zwischen freien und erzwungenen Phasentransitionen zu unterscheiden. Im Vergleich mit dem Verhalten $\mathcal{V}[\![\cdot]\!]$ ist es auf ihrer Basis allerdings nur möglich, Auswirkungen von Einflüssen auszudrücken, die sich als Phasentransitionen in den Bereichen widerspiegeln. Resultierende Instabilitäten, die nicht notwendigerweise auch zu Phasentransitionen führen müssen, und die beim Verhalten über die Phasenqualitäten (speziell Phasenqualität F) erfasst werden, können durch die Traces der Erweiterten Casetrace-Semantik nicht ausgedrückt werden. Je nach Anwendungsszenario und sich daraus ergebenden Analyseanforderungen bleibt es dem Benutzer überlassen, die geeignete Semantik zu wählen. In dieser Arbeit werden alle drei bisher vorgestellten Semantiken zum Einsatz kommen.

Ist bereits das Verhalten $\mathcal{V}[\![IS]\!]$ eines I–Systems IS bekannt, kann daraus auch direkt die Erweiterte Casetrace-Semantik $\mathcal{ECT}[\![IS]\!]$ abgeleitet werden, ohne auf die Ausführungen des zugeordneten $\mathrm{V_I}$Systems zurückgreifen zu müssen. Die *Erweiterte zc-Transformation* dient dabei der Umwandlung von einer Folge von globalen Aktivitätszuständen in eine Folge von aufeinander folgend unterschiedlicher Cases, jeweils verbunden durch eine Bereichsmenge. Aus der Erweiterten Casetrace-Semantik kann die „normale“ Casetrace-Semantik $\mathcal{CT}[\![IS]\!]$ einfach durch Streichen der Bereichsmengen in den Traces gewonnen werden. Der nächste Satz fasst die entscheidenden semantischen Abhängigkeiten zusammen.

Definition 4.19 (Erweiterte zc-Transformation). Sei $IS \in ISystem$ und $ztr := z_0 z_1 z_2 \ldots \in GZustand(IS)^+$. Sei $M := \{i_1, i_2, i_3, \ldots\} \subseteq I\!N$, $i_1 < i_2 < i_3 < \ldots$, genau die Indexmenge, für die gilt: $j \in M$ gdw. $zc(z_j) \neq zc(z_{j-1})$. Setze $i_0 := 0$. Die *Erweiterte zc-Transformation* $\lfloor\cdot\rfloor^{\mathrm{e}}$ wird festgelegt als:

$$\lfloor\cdot\rfloor^{\mathrm{e}} : \quad GZustand(IS)^+ \longrightarrow Case(IS)^+.\,(\mathcal{P}(B).Case(IS))^*$$

mit (ztr wie oben):

$$\lfloor ztr\rfloor^{\mathrm{e}} = zc(z_{i_0})\delta_{i_1} zc(z_{i_1})\delta_{i_2} zc(z_{i_2})\delta_{i_3} zc(z_{i_3})\ldots$$

wobei

$$\delta_{i_v} = \big\{b \in B \mid \forall p \in b : (z_{i_{v-1}}(p) \neq \mathsf{0} \wedge z_{i_v}(p) = \mathsf{0}) \Rightarrow z_{i_v-1}(p) \neq \mathsf{F}\big\}, \text{ für } v = 1,2,3,\ldots \qquad \Box$$

Satz 4.20 (Verhalten, Casetrace-, Erweiterte Casetrace-Semantik). Sei IS ein I–System. Es gilt:

a) $\mathcal{ECT}[\![IS]\!](c_0) = \{ectr \in Case(IS)^+.\,(\mathcal{P}(B).Case(IS))^* \mid \exists ztr \in \mathcal{V}[\![IS]\!](cz(c_0)) : ectr = \lfloor ztr\rfloor^{\mathrm{e}}\}$

b) $\mathcal{ECT}[\![IS]\!] = \{ectr \in Case(IS)^+.\,(\mathcal{P}(B).Case(IS))^* \mid \exists ztr \in \mathcal{V}[\![IS]\!] : ectr = \lfloor ztr\rfloor^{\mathrm{e}}\}$

c) $\mathcal{CT}[\![IS]\!](c_0) = \{c_0c_1c_2\ldots \in Case(IS)^+ \mid \exists \delta_1,\delta_2,\ldots \in \mathcal{P}(B) : c_0\delta_1c_1\delta_2c_2\ldots \in \mathcal{ECT}[\![IS]\!](c_0)$ mit $c_0,c_1,c_2,\ldots \in Case(IS)\}$

d) $\mathcal{CT}[\![IS]\!] = \{c_0c_1c_2\ldots \in Case(IS)^+ \mid \exists \delta_1,\delta_2,\ldots \in \mathcal{P}(B) : c_0\delta_1c_1\delta_2c_2\ldots \in \mathcal{ECT}[\![IS]\!]$ mit $c_0,c_1,c_2,\ldots \in Case(IS)\}$

Beweis. Durch Umformulierungen wird jeweils eine Gleichungskette erzeugt, die das gewünschte Ergebnis liefert. (Unter Bezugnahme auf IS wird die Notation 3.12 verwendet, d.h. $z^{\Pi,t}$ steht für $z^{\Pi,t}\langle V_I System(IS)\rangle$.)

Zu **a)**.

$\{ectr \in Case(IS)^+.\,(\mathcal{P}(B).Case(IS))^* \mid \exists ztr \in \mathcal{V}[\![IS]\!](cz(c_0)) : ectr = \lfloor ztr\rfloor^{\mathrm{e}}\}$

$=$ {Definition von $\mathcal{V}[\![\cdot]\!](\cdot)$, $cz(c_0) \in StabGZustand(IS)$ nach Definition 3.6.b}
$\{\lfloor ztr\rfloor^{\mathrm{e}} \mid ztr \in \{z^{\Pi,t_0}.z^{\Pi,t_1}.z^{\Pi,t_2}\ldots \in GZustand(IS)^+ \mid \Pi$ ist eine Ausführung von $V_I System(IS)$ und $t_0,t_1,t_2,\ldots$ eine max. Folge von Globalzeitpunkten der Ausführung mit: t_0 ist Startzeitpunkt, $z^{\Pi,t_0} = cz(c_0)$, $\forall i = 1,2,\ldots : (t_{i-1} < t_i,\ z^{\Pi,t_{i-1}} \neq z^{\Pi,t_i},\ \forall t, t_{i-1} \leq t < t_i : z^{\Pi,t} = z^{\Pi,t_{i-1}})\}\}$

$=$ {Umformung}
$\{\lfloor z^{\Pi,t_0}.z^{\Pi,t_1}.z^{\Pi,t_2}\ldots\rfloor^{\mathrm{e}} \mid \Pi$ ist eine Ausführung von $V_I System(IS)$ und $t_0,t_1,t_2,\ldots$ eine max. Folge von Globalzeitpunkten der Ausführung mit: t_0 ist Startzeitpunkt, $z^{\Pi,t_0} = cz(c_0)$, $\forall i = 1,2,\ldots$ $: (t_{i-1} < t_i,\ z^{\Pi,t_{i-1}} \neq z^{\Pi,t_i},\ \forall t, t_{i-1} \leq t < t_i : z^{\Pi,t} = z^{\Pi,t_{i-1}})\}$

$= \{$Definition von $\lfloor\cdot\rfloor^{e}\}$
$\{zc(z^{\Pi,t_{l_0}}).\delta_{l_1}.zc(z^{\Pi,t_{l_1}}).\delta_{l_2}.zc(z^{\Pi,t_{l_2}})\dots \mid \Pi$ ist eine Ausführung von $V_I System(IS)$ und $t_0, t_1, t_2, \dots$ eine max. Folge von Globalzeitpunkten der Ausführung mit: t_0 ist Startzeitpunkt, $z^{\Pi,t_0} = cz(c_0)$, $\forall i = 1,2,\dots : (t_{i-1} < t_i,\ z^{\Pi,t_{i-1}} \neq z^{\Pi,t_i},\ \forall t, t_{i-1} \leq t < t_i : z^{\Pi,t} = z^{\Pi,t_{i-1}})$, $l_0 = 0$, $\{l_1, l_2, \dots\} \subseteq \mathbb{N}$, $l_1 < l_2 < \dots$, $j \in \{l_1, l_2, \dots\} \Leftrightarrow zc(z^{\Pi,t_{j-1}}) \neq zc(z^{\Pi,t_j})$, $\delta_{l_v} = \{b \in B \mid \forall p \in b : (z^{\Pi,t_{l_v-1}}(p) \neq 0 \wedge z^{\Pi,t_{l_v}}(p) = 0) \Rightarrow z^{\Pi,t_{l_v-1}}(p) \neq \mathsf{F}\}$, $v = 1,2,\dots\}$

$= \{$Umschreiben von δ_{l_v}. Aufgrund der Definition von $zc(\cdot)$ und der Wahl der $l_0, l_1, l_2, \dots$ gilt: $(z^{\Pi,t_{l_v-1}}(p) \neq 0 \wedge z^{\Pi,t_{l_v}}(p) = 0) \Leftrightarrow (zc(z^{\Pi,t_{l_v-1}}) \neq zc(z^{\Pi,t_{l_v}})$ und $p \in zc(z^{\Pi,t_{l_v}}) \setminus zc(z^{\Pi,t_{l_v-1}}))$ $\Leftrightarrow$ (Bei $V_{b(p)}$ findet zum Zeitpunkt t_{l_v} eine Phasentransition $p \rightarrow \cdot$ statt.)
Eine Phasentransition erfordert ein Verhalten der Komponente $V_{b(p)}$ entsprechend Aktion A4 oder A5, unter Beachtung der jeweiligen Vorbedingungen. Wegen der Maximalitätsforderung der Folge der Globalzeitpunkte folgt: $(z^{\Pi,t_{l_v-1}}(p) \neq \mathsf{F}) \Rightarrow (V_{b(p)}$ befindet sich zum Zeitpunkt t_{l_v} in Aktion A4.) $\}$
$\{ \cdots \mid \cdots, \delta_{l_v} = \{b \in B \mid \exists p \in b : p \in zc(z^{\Pi,t_{l_v}}) \setminus zc(z^{\Pi,t_{l_v-1}})$ und V_b befindet sich zum Zeitpunkt t_{l_v} in Aktion A4.$\}$, $v = 1,2,\dots\}$

$= \{$Entsprechend der Definition von $zc(\cdot)$ gilt: $(zc(z^{\Pi,t_{j-1}}) \neq zc(z^{\Pi,t_j})) \Rightarrow (z^{\Pi,t_{j-1}} \neq z^{\Pi,t_j})$ und $(z^{\Pi,t} = z^{\Pi,t_i}) \Rightarrow (zc(z^{\Pi,t}) = zc(z^{\Pi,t_i}))$. Beachte $\{l_1, l_2, \dots\} \subseteq \{1, 2, \dots\}$.$\}$
$\{zc(z^{\Pi,t_{l_0}}).\delta_{l_1}.zc(z^{\Pi,t_{l_1}}).\delta_{l_2}.zc(z^{\Pi,t_{l_2}})\dots \mid \Pi$ ist eine Ausführung von $V_I System(IS)$ und $t_0, t_1, t_2, \dots$ eine max. Folge von Globalzeitpunkten der Ausführung mit: t_{l_0} ist Startzeitpunkt, $z^{\Pi,t_{l_0}} = cz(c_0)$, $l_0 = 0$, $\{l_1, l_2, \dots\} \subseteq \mathbb{N}$, $l_1 < l_2 < \dots$, $\forall i = 1,2,\dots : (t_{l_{i-1}} < t_{l_i},\ zc(z^{\Pi,t_{l_{i-1}}}) \neq zc(z^{\Pi,t_{l_i}}),\ \forall t, t_{l_{i-1}} \leq t < t_{l_i} : zc(z^{\Pi,t}) = zc(z^{\Pi,t_{l_{i-1}}}))$, $\delta_{l_v} = \cdots\}$

$= \{$Umindizierung mit i für l_i und v für l_v, Vereinfachung. Aktion A4 kann nur bei autonomen Komponenten V_b mit $b \in AB(IS)$ zur Anwendung kommen.$\}$
$\{zc(z^{\Pi,t_0}).\delta_1.zc(z^{\Pi,t_1}).\delta_2.zc(z^{\Pi,t_2})\dots \mid \Pi$ ist eine Ausführung von $V_I System(IS)$ und $t_0, t_1, t_2, \dots$ eine max. Folge von Globalzeitpunkten der Ausführung mit: t_0 ist Startzeitpunkt, $z^{\Pi,t_0} = cz(c_0)$, $\forall i = 1,2,\dots : (t_{i-1} < t_i,\ zc(z^{\Pi,t_{i-1}}) \neq zc(z^{\Pi,t_i}),\ \forall t, t_{i-1} \leq t < t_i : zc(z^{\Pi,t}) = zc(z^{\Pi,t_{i-1}}))$, $\delta_v = \{b(p) \in AB(IS) \mid p \in zc(z^{\Pi,t_v}) \setminus zc(z^{\Pi,t_{v-1}})$ und $V_{b(p)}$ befindet sich zum Zeitpunkt t_v in Aktion A4.$\}$, $v = 1,2,\dots\}$

$= \{$Umindizierung mit i für v, eine Laufvariable; Definition 4.16$\}$
$\mathcal{ECT}[\![IS]\!](c_0)$

Zu **b)**.

$\{ectr \in Case(IS)^+.(\mathcal{P}(B).Case(IS))^* \mid \exists ztr \in \mathcal{V}[\![IS]\!] : ectr = \lfloor ztr \rfloor^{e}\}$

$= \{$Definition von $\mathcal{V}[\![IS]\!]\}$
$\{ectr \in Case(IS)^+.(\mathcal{P}(B).Case(IS))^* \mid \exists ztr \in \bigcup_{z_0 \in StabGZustand(IS)} \mathcal{V}[\![IS]\!](z_0) : ectr = \lfloor ztr \rfloor^{e}\}$

$= \{$Aus den Definitionen von $cz(\cdot)$ und $StabGZustand(\cdot)$ folgt direkt: $\{cz(c_0) \mid c_0 \in Case(IS)\} = StabGZustand(IS)$.$\}$
$\{ectr \in Case(IS)^+.(\mathcal{P}(B).Case(IS))^* \mid \exists ztr \in \bigcup_{c_0 \in Case(IS)} \mathcal{V}[\![IS]\!](cz(c_0)) : ectr = \lfloor ztr \rfloor^{e}\}$

$= \{$Umformung$\}$
$\bigcup_{c_0 \in Case(IS)} \{ectr \in Case(IS)^+.(\mathcal{P}(B).Case(IS))^* \mid \exists ztr \in \mathcal{V}[\![IS]\!](cz(c_0)) : ectr = \lfloor ztr \rfloor^{e}\}$

$= \{$Teil a) des Satzes$\}$
$\bigcup_{c_0 \in Case(IS)} \mathcal{ECT}[\![IS]\!](c_0)$

$= \{$Definition 4.16$\}$
$\mathcal{ECT}[\![IS]\!]$

Zu **c)**.

$\{c_0 c_1 c_2 \dots \in Case(IS)^+ \mid \exists \delta_1, \delta_2, \delta_3, \dots \in \mathcal{P}(B) : c_0 \delta_1 c_1 \delta_2 c_2 \delta_3 \dots \in \mathcal{ECT}[\![IS]\!](c_0)$ mit $c_0, c_1, c_2, \dots \in Case(IS)\}$

$= \{$Definition $\mathcal{ECT}[\![\cdot]\!]\}$
$\{c_0c_1c_2\ldots \in Case(IS)^+ \mid \exists\delta_1,\delta_2,\delta_3,\ldots \in \mathcal{P}(B) : c_0\delta_1c_1\delta_2c_2\delta_3\ldots \in \{zc(z^{\Pi,t_0}).\delta'_1.zc(z^{\Pi,t_1}).\delta'_2.zc(z^{\Pi,t_2}).\ldots \in Case(IS)^+.(\mathcal{P}(B).Case(IS))^* \mid \Pi$ ist eine Ausführung von $V_ISystem(IS)$ und $t_0,t_1,t_2,\ldots$ eine max. Folge von Globalzeitpunkten der Ausführung mit: t_0 ist Startzeitpunkt, $z^{\Pi,t_0} = cz(c_0)$, $\forall i = 1,2,\ldots : (t_{i-1} < t_i,\ zc(z^{\Pi,t_{i-1}}) \neq zc(z^{\Pi,t_i})$, $\forall t, t_{i-1} \leq t < t_i : zc(z^{\Pi,t}) = zc(z^{\Pi,t_{i-1}}))$, $\delta'_i = \{b(p) \in AB(IS) \mid p \in zc(z^{\Pi,t_i}) \setminus zc(z^{\Pi,t_{i-1}})$ und $V_{b(p)}$ befindet sich zum Zeitpunkt t_i in Aktion A4.$\}\}$ mit $c_0,c_1,c_2,\ldots \in Case(IS)\}$

$= \{$Umformung$\}$
$\{zc(z^{\Pi,t_0}).zc(z^{\Pi,t_1}).zc(z^{\Pi,t_2})\ldots \in Case(IS)^+ \mid \exists\delta_1,\delta_2,\delta_3,\ldots \in \mathcal{P}(B) : \Pi$ ist eine Ausführung von $V_ISystem(IS)$ und $t_0,t_1,t_2,\ldots$ eine max. Folge von Globalzeitpunkten der Ausführung mit: t_0 ist Startzeitpunkt, $z^{\Pi,t_0} = cz(c_0)$, $\forall i = 1,2,\ldots : (t_{i-1} < t_i,\ zc(z^{\Pi,t_{i-1}}) \neq zc(z^{\Pi,t_i})$, $\forall t, t_{i-1} \leq t < t_i : zc(z^{\Pi,t}) = zc(z^{\Pi,t_{i-1}}))$, $\delta_i = \{b(p) \in AB(IS) \mid p \in zc(z^{\Pi,t_i}) \setminus zc(z^{\Pi,t_{i-1}})$ und $V_{b(p)}$ befindet sich zum Zeitpunkt t_i in Aktion A4.$\}\}$

$= \{$Vereinfachung. Die Existenz der δ_i ist gewährleistet, sie spielen für die Bildung der Menge keine Rolle.$\}$
$\{zc(z^{\Pi,t_0}).zc(z^{\Pi,t_1}).zc(z^{\Pi,t_2})\ldots \in Case(IS)^+ \mid \Pi$ ist eine Ausführung von $V_ISystem(IS)$ und $t_0,t_1,t_2,\ldots$ eine max. Folge von Globalzeitpunkten der Ausführung mit: t_0 ist Startzeitpunkt, $z^{\Pi,t_0} = cz(c_0)$, $\forall i = 1,2,\ldots : (t_{i-1} < t_i,\ zc(z^{\Pi,t_{i-1}}) \neq zc(z^{\Pi,t_i})$, $\forall t, t_{i-1} \leq t < t_i : zc(z^{\Pi,t}) = zc(z^{\Pi,t_{i-1}}))\}$

$= \{$Definition $\mathcal{CT}[\![\cdot]\!]\}$
$\mathcal{CT}[\![IS]\!](c_0)$

Zu **d)**.

$\{c_0c_1c_2\ldots \in Case(IS)^+ \mid \exists\delta_1,\delta_2,\ldots \in \mathcal{P}(B) : c_0\delta_1c_1\delta_2c_2\ldots \in \mathcal{ECT}[\![IS]\!]$ mit $c_0,c_1,c_2,\ldots \in Case(IS)\}$

$= \{$Definition $\mathcal{ECT}[\![\cdot]\!]\}$
$\{c_0c_1c_2\ldots \in Case(IS)^+ \mid \exists\delta_1,\delta_2,\ldots \in \mathcal{P}(B) : c_0\delta_1c_1\delta_2c_2\ldots \in \bigcup_{c_0 \in Case(IS)} \mathcal{ECT}[\![IS]\!](c_0)$ mit $c_0,c_1,c_2,\ldots \in Case(IS)\}$

$= \{$Umformung$\}$
$\bigcup_{c_0 \in Case(IS)} \{c_0c_1c_2\ldots \in Case(IS)^+ \mid \exists\delta_1,\delta_2,\ldots \in \mathcal{P}(B) : c_0\delta_1c_1\delta_2c_2\ldots \in \mathcal{ECT}[\![IS]\!](c_0)$ mit $c_0,c_1,c_2,\ldots \in Case(IS)\}$

$= \{$Teil c) des Satzes$\}$
$\bigcup_{c_0 \in Case(IS)} \mathcal{CT}[\![IS]\!](c_0)\}$

$= \{$Definition $\mathcal{CT}[\![\cdot]\!]\}$
$\mathcal{CT}[\![IS]\!]$ □

Die durch Satz 4.20 aufgezeigten semantischen Abhängigkeiten beinhalten keinen Bezug zu einem $\mathrm{V_I}$System. Die Berechnungen finden ausschließlich innerhalb der formalen Ebene statt. Ist $\mathcal{V}[\![IS]\!]$ bekannt, kann $\mathcal{ECT}[\![IS]\!]$ berechnet; ist $\mathcal{ECT}[\![IS]\!]$ bekannt, kann $\mathcal{CT}[\![IS]\!]$ berechnet werden, ohne auf die Axiome oder die algorithmisch spezifizierten Aktionen, die die Ausführungen von $V_ISystem(IS)$ steuern, zurückgreifen zu müssen. Die Erweiterte Casetrace-Semantik nimmt folglich bezüglich ihrer Ausdruckskraft eine Zwischenstellung zwischen dem Verhalten und der Casetrace-Semantik ein. Über die Berechnungsschritte lassen sich in vielen Fällen vorliegende Aussagen und Beweise über semantische Eigenschaften des Verhaltens auf die Erweiterte Casetrace-Semantik, und über semantische Eigenschaften der Erweiterten Casetrace-Semantik auf die Casetrace-Semantik übertragen, ohne erneut die Aktionen eines $\mathrm{V_I}$Systems untersuchen zu müssen. Beweise werden dadurch in der Regel vereinfacht, da die Protokollierung der verschiedenen nebenläufigen Abläufe in den einzelnen Komponenten des $\mathrm{V_I}$Systems entfällt und nur noch die formalen Schnittstellen zwischen den verwendeten Semantiken hergestellt werden müssen. Beispiele hierzu bieten die folgenden Sätze 4.21 und 4.22.

Die Nicht-Präfix-Abgeschlossenheit ist eine wichtige Eigenschaft der Casetrace-Semantik (Satz 4.10) und mögliche Grundlage zur Spezifikation und Verifikation von Fortschrittseigenschaften. Da genau jede Casefolge der Casetrace-Semantik auch in der Erweiterten Casetrace-Semantik vorkommt, nur erweitert um Zwischenelemente in Form von Bereichsmengen, ist es nahe liegend, dass Nicht-Präfix-Abgeschlossenheit auch für die Erweiterte Casetrace-Semantik gilt. Der folgende Satz beinhaltet diese Aussage.

Satz 4.21 (Nicht-Präfix-Abgeschlossenheit der Erweiterten Casetrace-Semantik). Für die Erweiterte Casetrace-Semantik $\mathcal{ECT}[\![IS]\!]$ eines I–Systems IS ist Präfix-Abgeschlossenheit nicht garantiert.

Beweis. Um ein Beispiel für Nicht-Präfix-Abgeschlossenheit zu geben, wird Bezug genommen auf den Beweis von Satz 4.10 und das dort angeführte I–System sowie die dort verwendeten Bezeichnungen. Es wird ausgegangen von $c'_0c'_1c'_2 \in \mathcal{CT}[\![IS]\!]$.

Nach Satz 4.20.d existieren $\delta_1, \delta_2 \subseteq B$, so dass $c'_0\delta_1c'_1\delta_2c'_2 \in \mathcal{ECT}[\![IS]\!]$ gilt.

Im Beweis zu Satz 4.10 wurde $c'_0c'_1 \notin \mathcal{CT}[\![IS]\!]$ gezeigt, folglich gilt, erneut wegen Satz 4.20.d, $c'_0\delta c'_1 \notin \mathcal{ECT}[\![IS]\!]$ für jedes beliebige $\delta \subseteq B$, also auch δ_1. □

Analysiert man die Elemente (Traces) der Erweiterten Casetrace-Semantik eines I–Systems, so ergeben sich strukturelle Eigenschaften, die für alle diese Elemente gelten und insbesondere den Einfluss der Kopplungs- und Erregungsrelation auf diese Semantik des I–Systems widerspiegeln. Verantwortlich für den charakteristischen Aufbau der Traces ist die Aktivität des zugeordneten $\mathrm{V_I}$Systems, festgelegt durch die Axiome VA1-VA3, sowie die Aktionen A1-A13. Da die Erweiterte Casetrace-Semantik in Übereinstimmung mit der Casetrace-Semantik alle aus den Ausführungen des $\mathrm{V_I}$Systems ableitbaren Casefolgen beinhaltet, können die Aussagen a) und b) aus Satz 4.12 direkt übernommen werden. Durch die Hinzunahme von Bereichsmengen zur Präzisierung der Caseübergänge kommt zusätzlich die besondere Rolle der trägen bzw. autonomen Bereiche zum Tragen.
Der folgende Satz beschreibt grundlegende Zusammenhänge zwischen der formalen Struktur eines beliebigen I–Systems IS und dessen Erweiterter Casetrace-Semantik $\mathcal{ECT}[\![IS]\!]$.

Satz 4.22 (Charakterisierung der Erweiterten Casetrace-Semantik). Sei $IS = (P, B, \underline{B}, K, E)$ ein I–System und $c_0\delta_1c_1\delta_2c_2\ldots \in \mathcal{ECT}[\![IS]\!]$ ein Element der Erweiterten Casetrace-Semantik mit $c_0, c_j \in Case(IS)$, $\delta_j \subseteq B$, $j = 1, 2, \ldots$. Seien $p, q, v, w \in P$ mit $b(p) = b(q)$ und $b(v) = b(w)$. Dann gelten folgende Aussagen für alle $i = 1, 2, \ldots$:

a) *Stop*
$(p, v) \in E \wedge \{p, v\} \subseteq c_{i-1} \quad \Rightarrow \quad p \in c_i$

b) *Nicht-Koinzidenz*
$(p, w) \in K \wedge (q, v) \in K \wedge \{p, v\} \subseteq c_{i-1} \quad \Rightarrow \quad \{q, w\} \not\subseteq c_i$

c) *Trägheit*
$b(p) \in \underline{B} \quad \Rightarrow \quad b(p) \notin \delta_i$

Beweis. Bei a) und b) wird auf die Eigenschaften der Casetrace-Semantik zurückgegriffen. c) erfordert eine genauere Betrachtung der Festlegungen für δ_i. Es gelten im Folgenden die Bezeichnungen und Voraussetzungen aus dem Satz.
Zu **a)** und **b)**.

$c_0\delta_1c_1\delta_2c_2\ldots \in \mathcal{ECT}[\![IS]\!]$

$\Rightarrow$ {Satz 4.20.d}
$c_0c_1c_2\ldots \in \mathcal{CT}[\![IS]\!]$

$\Rightarrow$ {Satz 4.12}
Es gelten die Aussagen a) und b).

Zu **c)**.

Gemäß den Festlegungen für δ_i in Definition 4.16 gilt $\delta_i \subseteq AB(IS)$. $AB(IS)$, die Menge der autonomen Bereiche von IS, ist definiert als $B \setminus \underline{B}$ (Definition 2.1). Es folgt also direkt $(b(p) \in \underline{B}) \Rightarrow (b(p) \notin \delta_i)$. □

Wie schon in der Hinleitung zu Satz 4.22 erwähnt wurde, sind die Aussagen a) und b) des Satzes aus Satz 4.12 übernommen. Gleiches gilt jetzt für die anschaulichen Interpretationen dieser Aussagen, die im Folgenden kurz wiederholt werden. Aussage c) bezieht sich auf die Bereichsmengen in den Traces der Erweiterten Casetrace-Semantik. Da solche Elemente zur Präzisierung der Caseübergänge bzw. der auftretenden Phasentransitionen bei der Casetrace-Semantik nicht existieren, gibt es zu c) kein Gegenstück im Rahmen der Casetrace-Semantik.

a) entspricht in seiner Interpretation Satz 4.12.a). Es wird die *Aufrechterhaltung eines Einflusses* zwischen zwei aktuell eingenommenen Phasen in benachbarten Komponenten beschrieben. Eine erregende Phase kann erst wieder verlassen werden, nachdem sich der Einfluss in Form einer Phasentransition ausgewirkt hat, d.h. die erregte Phase verlassen wurde.

b) entspricht in seiner Interpretation Satz 4.12.b). *Koinzidente Phasentransitionen bei über Kreuz wechselseitig ausgeschlossenen Nachbarphasen können nicht auftreten*, da eine globale Zeitgleichheit zweier Ereignisse in unterschiedlichen Komponenten in einem verteilten System nicht gewährleistet werden kann. Jegliche Hintereinanderausführung der Phasentransitionen ist durch den vorgegebenen kreuzweisen wechselseitigen Ausschluss ausgeschlossen.

c) beschreibt eine Eigenschaft, die in Sätzen zur Casetrace-Semantik nicht erfasst werden kann. Es wird ausgedrückt, dass *in einem trägen Bereich jede auftretende Phasentransition erzwungen* ist. Die trägen Bereiche repräsentieren reaktive Komponenten eines verteilten Systems (vgl. Kapitel 2.1), bei denen jegliche Aktivität durch externe Einflüsse angestoßen werden muss. Lokale Kontrollentscheidungen, eine beliebige Aktion spontan auszuführen, sind ausgeschlossen.

Bemerkung 4.23. Es gelte ebenfalls Teil c) des Satzes 4.22, wenn in Definition 4.16 bei δ_i nicht explizit $b(p) \in AB(IS)$ gefordert würde, sondern nur $b(p) \in B$. Das liegt an dem speziellen Verhalten von $V_I System(IS)$. Es gilt nämlich unter den Voraussetzungen der Definition 4.16 die Implikationskette:

$b \in \delta_i$

$\Rightarrow$ {Festlegung von δ_i}
V_b befindet sich zum Zeitpunkt t_i in Aktion A4.

$\Rightarrow$ {Vorbedingung der Aktion A4}
Es existieren $p, q \in b$, so dass gilt: $z^{\Pi,t_i}\langle V_b\rangle(p) = q$.

$\Rightarrow$ {Zum Start-Globalzeitpunkt t_0 gilt als (Stabilitäts-) Voraussetzung $z^{\Pi,t_0}\langle V_b\rangle(p) = 1$.
Zuweisungen der Form $\_z(p) := q$ gibt es nur in Aktion A3.}
Es existiert ein Globalzeitpunkt $t_x < t_i$ und eine Phase $p \in b$ mit: $z^{\Pi,t_x}\langle V_b\rangle(p) = 1$ und V_b befindet sich zum Zeitpunkt t_x in Aktion A3.

$\Rightarrow$ {Vorbedingung von A3}
b ist autonom, d.h. $b \in AB(IS)$.

$\Rightarrow$ {$AB(IS)$ ist definiert als $B \setminus \underline{B}$.}
$b \notin \underline{B}$.

Das bei der Definition der Erweiterten Casetrace-Semantik angewendete Prinzip, Zustands- oder Ereignistraces bei der semantischen Beschreibung eines formalen Modells durch Zusatzbezeichnungen an oder zwischen den einzelnen Traceelementen zu ergänzen, um das Informationspotential zur Systembeschreibung zu erhöhen, findet sich auch bei anderen formalen Modellen wieder. Als Beispiele seien die Integration von Zeit-Spezifikationen in [41], Nachrichtenflüsse in [64] oder Ausführungsalternativen in [70] genannt.

Zur Anwendung kommt die Erweiterte Casetrace-Semantik in dieser Arbeit unter anderem bei der Modellierung lokaler Ereignisstrukturen in einer Komponente eines verteilten Systems (siehe Kapitel 9.7). Hierbei wird besonderer Wert auf die Unterscheidung zwischen freien und erzwungenen Phasentransitionen gelegt. Eine vergleichbare Zweiteilung von Ereignissen findet sich sonst nur bei Reisig [75] in Form von „progressive“ und „quiescent“ Transitionen in so genannten System Nets. Reisigs Ansatz zielt allerdings nicht auf die explizite Darstellung der Ursache von Zwängen (siehe auch Kapitel 1.3).

Kapitel 5

Interleaving

In einem verteilten System kann man sich die Aktivität einer einzelnen Komponente als eine Ausführungsfolge der atomaren Aktionen der Komponente vorstellen (vgl. [4]). Im Rahmen der I–Systeme spiegeln sich diese in den lokalen Variablenzuweisungen in den Komponenten des zugeordneten V_ISystems wider (siehe Axiom VA3). Atomare Aktionen sind semantisch als Ausführungseinheiten zu verstehen. Die Ausführung verschiedener Komponenten schreitet asynchron voran, d.h. es kann keine Annahme über die relative Geschwindigkeit getroffen werden, mit der die einzelnen Komponenten ihre Aktionen ausführen. Die Ausführung von atomaren Aktionen verschiedener Komponenten eines verteilten Systems kann man sich als globalzeitlich überlappend vorstellen, sofern sie völlig unabhängig voneinander sind. Aufgrund dieser Einschränkung ist es möglich, den Effekt der überlappenden Ausführung von atomaren Aktionen verschiedener Komponenten durch eine nicht-überlappende Vermischung oder *Interleaving* der sequentiellen Ausführung der einzelnen Komponenten in beliebiger Reihenfolge zu modellieren.

Bei der formalen Arbeit mit parallelen, nebenläufigen und verteilten Programmen und Systemen werden immer wieder Interleaving-Ansätze mit betrachtet. So unterscheidet [10] drei Typen von operationellen Semantiken eines parallelen Programms: die parallelen, sequentiellen und kausalen Semantiken. Die sequentiellen Semantiken setzen sich dabei aus den Interleavings, d.h. den erlaubten sequentiellen Ausführungen des Programms, zusammen. Die Verifikation paralleler und verteilter Programme, zu deren Zweck die Ausführung eines Programms durch das Verschachteln der Ausführungen der Komponenten modelliert wird, beschreibt [4]. Neben so genannten Concurrent Runs werden in [75] auch Interleaved Runs zur semantischen Beschreibung von System Nets, einer Variante von Petri-Netzen, eingeführt. Interleaving Semantiken werden traditionell auch eingesetzt beim Model Checking bei der Verifikation, ob ein nebenläufiges Programm eine temporal-logische Spezifikation erfüllt. Nebenläufigkeiten werden in diesem Fall durch Interleavings modelliert [34, 53]. Kompositionalität ist eines der Schlüsselkonzepte in [26], um die Komplexität bei der Analyse nebenläufiger Prozesse in den Griff zu bekommen. Dies geschieht dort unter anderem auf der Basis des Interleaving Modells.

Die herausgestellte Bedeutung des Interleavings im Bereich der formalen Behandlung nebenläufiger und verteilter Prozesse, Programme und Systeme motiviert die Untersuchung, wie Interleaving Ansätze im Modell der I–Systeme, vornehmlich bezogen auf die einzelnen Trace-Semantiken, einzuordnen sind. Der restliche Teil dieses Kapitels liefert diesbezüglich erste Ergebnisse.

5.1 Interleaving Verhalten

Der folgende Satz sagt aus, dass es zu einer Trace des Verhaltens $\mathcal{V}[\![IS]\!]$ eines I–Systems IS, in denen zwei aufeinander folgende globale Aktivitätszustände das aus globaler Sicht gleichzeitige Auftreten mindestens zweier Änderungen von Phasenqualitäten in unterschiedlichen Bereichen beschreiben, ebenfalls Traces im Verhalten $\mathcal{V}[\![IS]\!]$ gibt, in denen die beiden Änderungen nacheinander in beliebiger Reihenfolge auftreten, bei gleicher Resttrace. Die Umkehrrichtung gilt zudem.

Satz 5.1 (Interleaving-Eigenschaft des Verhaltens). Sei IS ein I–System mit Phasenmenge P und Bereichsmenge B. Seien $ztr_1, ztr_2 \in GZustand(IS)^*$, $z_x, z_{xy}, z'_{xy}, z_y \in GZustand(IS)$ und $b_1, b_2 \in B$ mit $b_1 \neq b_2$. Unter den Voraussetzungen $z_x|_{b_1} \neq z_{xy}|_{b_1} = z_y|_{b_1}$, $z_{xy}|_{P\setminus b_1} = z_x|_{P\setminus b_1}$, $z_x|_{b_2} \neq z'_{xy}|_{b_2} = z_y|_{b_2}$, $z'_{xy}|_{P\setminus b_2} = z_x|_{P\setminus b_2}$ gilt:

$$ztr_1.z_x.z_y.ztr_2 \in \mathcal{V}[\![IS]\!] \quad \Leftrightarrow \quad ztr_1.z_x.z_{xy}.z_y.ztr_2 \in \mathcal{V}[\![IS]\!] \wedge ztr_1.z_x.z'_{xy}.z_y.ztr_2 \in \mathcal{V}[\![IS]\!]$$

Beweis.

„⇒“.
Es reicht zu zeigen: $ztr_1.z_x.z_y.ztr_2 \in \mathcal{V}[\![IS]\!] \Rightarrow ztr_1.z_x.z_{xy}.z_y.ztr_2 \in \mathcal{V}[\![IS]\!]$. Der Rest gilt dann symmetrisch.

Sei nun $ztr_1.z_x.z_y.ztr_2 \in \mathcal{V}[\![IS]\!]$, $b_1, b_2 \in B$ und es gelten die Voraussetzungen aus dem Satz.

Gemäß der Definition von $\mathcal{V}[\![\cdot]\!]$ (Definition 4.1) existiert eine Ausführung Π_1 von $V_I System(IS)$ sowie Globalzeitpunkte t_x, t_y mit $z^{\Pi_1,t_x}\langle V_I System(IS)\rangle = z_x \neq z_y = z^{\Pi_1,t_y}\langle V_I System(IS)\rangle$ und $z^{\Pi_1,t}\langle V_I System(IS)\rangle = z_x$ für alle $t \in [t_x, t_y[$.
Gemäß der Definition der z-Globalbelegung (Definition 3.11) existieren Phasen $p \in b_1$ und $v \in b_2$ mit $z^{\Pi_1,t_x}\langle V_{b_1}\rangle(p) \neq z^{\Pi_1,t_y}\langle V_{b_1}\rangle(p)$ und $z^{\Pi_1,t_x}\langle V_{b_2}\rangle(v) \neq z^{\Pi_1,t_y}\langle V_{b_2}\rangle(v)$ und $z^{\Pi_1,t}\langle V_{b_1}\rangle(p) = z^{\Pi_1,t_x}\langle V_{b_1}\rangle(p)$, $z^{\Pi_1,t}\langle V_{b_2}\rangle(v) = z^{\Pi_1,t_x}\langle V_{b_2}\rangle(v)$ für alle $t \in [t_x, t_y[$.

Es finden Änderungen der Phasenqualitäten von p und v statt, geregelt durch die Axiome und Aktionen für $V_I System(IS)$. Zum Zeitpunkt t_y befindet sich V_{b_1} bei einer der Aktionen A1 - A13 und führt eine Zuweisung der Form $\_z(p) := \cdot$ aus, die eine Änderung der $\_z(p)$-Variablen zur Folge hat. Gleiches gilt für V_{b_2} mit v statt p.
Entsprechend Axiom VA3 sind die Ausführungszeiten der Aktionen unbekannt. Daher ist es möglich, dass die Zuweisung bei V_{b_1} schon vor t_y ausgeführt werden kann zu einem Zeitpunkt t_{xy}, $t_x < t_{xy} < t_y$, bei gleichem Verhalten danach und bei den anderen Komponenten. Solch ein Zeitpunkt existiert (da ein kontinuierliches Zeitmodell zugrunde liegt) immer, z.B. $t_{xy} := (t_x + t_y)/2$.

Somit gibt es eine Ausführung Π_2 von $V_I System(IS)$, die Π_1 entspricht bis einschließlich t_x und ab einschließlich t_y, und für die weiterhin gilt: $z^{\Pi_2,t}\langle V_{b_1}\rangle(p) = z^{\Pi_2,t_x}\langle V_{b_1}\rangle(p)$, $z^{\Pi_2,t'}\langle V_{b_1}\rangle(p) = z^{\Pi_2,t_y}\langle V_{b_1}\rangle(p)$, $z^{\Pi_2,t''}\langle V_b\rangle(p') = z^{\Pi_2,t_x}\langle V_b\rangle(p')$ für alle $t \in [t_x, t_{xy}[$, $t' \in [t_{xy}, t_y[$, $t'' \in [t_x, t_y[$, $b \in B \setminus \{b_1\}$, $p' \in b$.
Wegen der Existenz von Π_2 folgt, gemäß der Definition von $\mathcal{V}[\![\cdot]\!]$: $ztr_1.z_x.z_{xy}.z_y.ztr_2 \in \mathcal{V}[\![IS]\!]$.

„⇐“.
Sei $ztr_1.z_x.z_{xy}.z_y.ztr_2 \in \mathcal{V}[\![IS]\!]$, $b_1, b_2 \in B$, und es gelten die Voraussetzungen aus dem Satz.

Gemäß der Definition von $\mathcal{V}[\![\cdot]\!]$ existiert eine Ausführung Π_1 von $V_I System(IS)$ sowie Globalzeitpunkte t_x, t_{xy}, t_y mit $z^{\Pi_1,t_x}\langle V_I System(IS)\rangle = z_x \neq z_{xy} = z^{\Pi_1,t_{xy}}\langle V_I System(IS)\rangle \neq z^{\Pi_1,t_y}\langle V_I System(IS)\rangle = z_y$ und $z^{\Pi_1,t}\langle V_I System(IS)\rangle = z_x$ für alle $t \in [t_x, t_{xy}[$ sowie $z^{\Pi_1,t'}\langle V_I System(IS)\rangle = z_{xy}$ für alle $t' \in [t_{xy}, t_y[$.

Gemäß der Definition der z-Globalbelegung und unter Beachtung der durch die Voraussetzungen gegebenen Abhängigkeiten der einzelnen Aktivitätszustände existieren Phasen $p \in b_1$, $v \in b_2$ mit: $z^{\Pi_1,t}\langle V_{b_1}\rangle(p) = z^{\Pi_1,t_x}\langle V_{b_1}\rangle(p) \neq z^{\Pi_1,t'}\langle V_{b_1}\rangle(p) = z^{\Pi_1,t_y}\langle V_{b_1}\rangle(p)$, $z^{\Pi_1,t_x}\langle V_{b_2}\rangle(v) \neq z^{\Pi_1,t_y}\langle V_{b_2}\rangle(v)$, $z^{\Pi_1,t''}\langle V_b\rangle(p') = z^{\Pi_1,t_x}\langle V_b\rangle(p')$ für alle $t \in [t_x, t_{xy}[$, $t' \in [t_{xy}, t_y[$, $t'' \in [t_x, t_y[$, $b \in B \setminus \{b_1\}$, $p' \in b$.

Zum Globalzeitpunkt t_{xy} befindet sich V_{b_1} bei einer Aktion A_1 aus A1 - A13 und führt eine die Variablenbelegung verändernde Zuweisung der Form $\_z(p) := \alpha$ aus, $\alpha \in \{0, 1, \mathsf{F}\} \cup b_1 \setminus \{p\}$. Zum Zeitpunkt t_y befindet sich V_{b_2} bei einer Aktion A_2 und führt eine die Variablenbelegung verändernde Zuweisung der Form $\_z(v) := \beta$ aus, $\beta \in \{0, 1, \mathsf{F}\} \cup b_2 \setminus \{v\}$.

Sei weiterhin $ztr_1.z_x.z'_{xy}.z_y.ztr_2 \in \mathcal{V}[\![IS]\!]$.

Analog zu Π_1 existiert eine Ausführung Π_2, die Π_1 entspricht bis einschließlich t_x und ab einschließlich t_y. Es gelten die gleichen vorangegangenen Betrachtungen mit t'_{xy} statt t_{xy}, z'_{xy} statt z_{xy}, b_1 und b_2 vertauscht, $p \in b_1$, $v \in b_2$.

Zum Globalzeitpunkt t'_{xy} der Ausführung Π_2 befindet sich V_{b_2} bei einer Aktion A_3 und führt die Zuweisung $\_z(v) := \beta$ aus. Zum Zeitpunkt t_y befindet sich V_{b_1} bei einer Aktion A_4 und führt die Zuweisung der Form $\_z(p) := \alpha$ aus.

Annahme: $ztr_1.z_x.z_y.ztr_2 \notin \mathcal{V}[\![IS]\!]$.

Dann existiert *keine* Ausführung Π_3, die Π_1 entspricht bis einschließlich t_x und ab einschließlich t_y und ohne zwischenzeitliche Veränderungen, d.h.: $z^{\Pi_3,t}\langle V_{b_1}\rangle(p) = z^{\Pi_3,t_x}\langle V_{b_1}\rangle(p)$, $z^{\Pi_3,t}\langle V_{b_2}\rangle(v) = z^{\Pi_3,t_x}\langle V_{b_2}\rangle(v)$, $z^{\Pi_3,t_x}\langle V_{b_1}\rangle(p) \neq z^{\Pi_3,t_y}\langle V_{b_1}\rangle(p)$, $z^{\Pi_3,t_x}\langle V_{b_2}\rangle(v) \neq z^{\Pi_3,t_y}\langle V_{b_2}\rangle(v)$, für alle $t \in [t_x, t_y[$.

Folglich kann *nicht* auftreten, dass sich V_{b_1} zum Zeitpunkt t_y der Ausführung Π_3 bei einer der Aktionen A_1 oder A_3 befindet, die Zuweisung $\_z(p) := \alpha$ ausführt, und gleichzeitig befindet sich V_{b_2} bei einer der Aktionen A_2 oder A_4 und führt die Zuweisung $\_z(v) := \beta$ aus.

Die Nicht-Existenz von Π_3 bedeutet für Π_1, dass die Durchführung von A_1 bei V_{b_1} erst die Voraussetzungen (durch Nachrichtentransfer und entsprechende Reaktionen gemäß der Aktionen A1 - A13) für die Durchführung von A_2 bei V_{b_2} schafft, und dass dieser Einfluss auch notwendig ist, um die Zuweisung $\_z(v) := \beta$ bei V_{b_2} im Zeitraum $]t_x, t_y]$ durchzuführen.

Damit liegt ein Widerspruch zur Existenz von Π_2 vor, bei der die Ausführung von $\_z(v) := \beta$ bei V_{b_2} vor $\_z(p) := \alpha$ bei V_{b_1} erfolgt, im Zeitraum $]t_x, t_y]$. Die Annahme ist somit falsch und der Satz bewiesen. □

Die ⇒-Richtung des Satzes lässt sich so interpretieren, dass globalzeitlich gleichzeitige Zustandswechsel bei zwei unterschiedlichen Komponenten eines verteilten Systems globalzeitlich gesehen auch in beliebiger Reihenfolge auftreten können, bei sonst gleichem restlichen Systemverhalten. Die Hintereinanderanwendung dieser Satzrichtung ermöglicht die Auflösung von Nebenläufigkeit in beliebig vielen Komponenten. Die ⇐-Richtung erlaubt es, mögliches nebenläufiges Verhalten aus sequentiellem abzuleiten. Der Beweis beruht im Wesentlichen auf der Ungewissheit lokaler Ausführungszeiten (Verhaltensaxiom VA3) im zugeordneten $\mathrm{V_I}$System. Eine bezüglich der Globalzeit auftretende Gleichzeitigkeit von Aktionsausführungen in unterschiedlichen Komponenten kann in dem verteilten System nicht garantiert werden.

Beim *Interleaving Verhalten* eines I–Systems werden gegenüber dem „normalen“ Verhalten nur Traces berücksichtigt, in denen aufeinander folgende globale Aktivitätszustände den Wechsel von Phasenqualitäten in jeweils genau einem Bereich beschreiben.

Definition 5.2 (Interleaving Verhalten eines I–Systems). Für ein I–System IS ist das *Interleaving Verhalten* $\mathcal{V}^{\mathrm{i}}[\![IS]\!]$ definiert als:

$$\mathcal{V}^{\mathrm{i}}[\![IS]\!] \subseteq \mathcal{V}[\![IS]\!] \subseteq GZustand(IS)^+$$

mit

$$\mathcal{V}^{\mathrm{i}}[\![IS]\!] = \{z_0 z_1 z_2 \ldots \in \mathcal{V}[\![IS]\!] \mid \forall i = 1, 2, \ldots : \exists!\, b \in B \text{ mit } z_i|_b \neq z_{i-1}|_b\}.$$

Das Interleaving Verhalten von IS *bzgl. eines stabilen globalen Start-Aktivitätszustandes* $z_0 \in StabGZustand(IS)$ ist gegeben durch:

$$\mathcal{V}^{\mathrm{i}}[\![IS]\!](z_0) = \mathcal{V}^{\mathrm{i}}[\![IS]\!] \cap \mathcal{V}[\![IS]\!](z_0).$$ □

Das Interleaving Verhalten $\mathcal{V}^{\mathrm{i}}[\![IS]\!]$ wird in obiger Definition aus dem Verhalten $\mathcal{V}[\![IS]\!]$ abgeleitet. Somit bewegt man sich innerhalb der formalen Ebene und greift nicht auf die algorithmische Ebene, d.h. auf die Ausführungen des IS zugeordneten $\mathrm{V_I}$Systems zurück. Es ist allerdings kein Problem, auf diese indirekte Berechnung zu verzichten und durch leichte Modifikation von Definition 4.1 sich direkt auf die Ausführungen zu beziehen. Alle Ausführungen, in denen gleichzeitig bei unterschiedlichen Komponenten Veränderungen von Phasenqualitäten auftreten, werden dann außer Acht gelassen. Auf eine genauere Darstellung soll an dieser Stelle verzichtet werden, da zum einen die Modifikation trivial ist und anderseits die Beziehungen der einzelnen Semantiken zueinander im Vordergrund stehen.
Das folgende Beispiel verdeutlicht die Berechnung von Teilen des Interleaving Verhaltens eines I–Systems unter Rückgriff auf dessen Verhalten.

Beispiel 5.3. Bestimmt werden sollen Elemente des Interleaving Verhaltens $\mathcal{V}^{\mathrm{i}}[\![IS_1]\!]$ des I–Systems IS_1 aus Beispiel 2.2. Verwendet werden hierzu die Notationen und Ergebnisse aus Beispiel 4.2.

Die Ausführung 1 lieferte: $z_0 z_1 z_2 z_3 z_4 z_5 z_6 \in \mathcal{V}[\![IS_1]\!]$.
Wegen $z_i|_{b_1} \neq z_{i-1}|_{b_1} \wedge z_i|_{b_2} \neq z_{i-1}|_{b_2}$ für $i \in \{1,3\}$ gilt: $z_0 z_1 z_2 z_3 z_4 z_5 z_6 \notin \mathcal{V}^{\mathrm{i}}[\![IS_1]\!]$.

Die wiederholte Anwendung von Satz 5.1.$\Rightarrow$ liefert:
$\overbrace{z_0 z_{01} z_1 z_2 z_{23} z_3 z_4 z_5 z_6}^{ztr_a} \in \mathcal{V}[\![IS_1]\!]$, $\overbrace{z_0 z'_{01} z_1 z_2 z_{23} z_3 z_4 z_5 z_6}^{ztr_b} \in \mathcal{V}[\![IS_1]\!]$, $\overbrace{z_0 z_{01} z_1 z_2 z'_{23} z_3 z_4 z_5 z_6}^{ztr_c} \in \mathcal{V}[\![IS_1]\!]$,
$\overbrace{z_0 z'_{01} z_1 z_2 z'_{23} z_3 z_4 z_5 z_6}^{ztr_d} \in \mathcal{V}[\![IS_1]\!]$,
mit $z_{01} = [p_2<p_1>, q_2<1>, v_2<1>]$, $z'_{01} = [p_2<1>, q_2<q_3>, v_2<1>]$, $z_{23} = [p_1<p_2>, q_2<q_3>, v_2<1>]$, $z'_{23} = [p_1<1>, q_3<1>, v_2<1>]$.

Für alle vier Traces gilt, dass sich aufeinander folgende globale Aktivitätszustände bei Einschränkung des Definitionsbereichs auf die einzelnen Bereiche in genau einem Fall unterscheiden. Entsprechend Definition 5.2 gilt deshalb: $ztr_a, ztr_b, ztr_c, ztr_d \in \mathcal{V}^{\mathrm{i}}[\![IS_1]\!]$.

Bei Ausführung 2 findet bei jedem Übergang zwischen zwei globalen Aktivitätszuständen in genau einer Komponente von $V_I System(IS_1)$ eine Veränderung der Phasenqualitäten statt, veranschaulicht dadurch, dass über jedem Pfeil genau eine Aktion aufgeführt ist. Es folgt daraus direkt für die resultierende Trace von globalen Aktivitätszuständen: $z'_0 z'_1 z'_2 z'_3 z'_4 \in \mathcal{V}^{\mathrm{i}}[\![IS_1]\!]$. □

Definition 5.2 gibt explizit an, wie das Interleaving Verhalten $\mathcal{V}^{\mathrm{i}}[\![IS]\!]$ eines I–Systems IS aus dessen Verhalten $\mathcal{V}[\![IS]\!]$ abgeleitet werden kann. Es stellt sich als nächstes die Frage, ob sich auch die Rückrichtung beweisen lässt, d.h. dass $\mathcal{V}[\![IS]\!]$ aus $\mathcal{V}^{\mathrm{i}}[\![IS]\!]$ berechnet werden kann. Der folgende Satz bejaht dieses.

Satz 5.4 (Rekonstruktion des Verhaltens). Sei IS ein I–System mit Phasenmenge P und Bereichsmenge B. Die Menge V sei durch folgende drei Kriterien festgelegt:

(1) $\mathcal{V}^{\mathrm{i}}[\![IS]\!] \subseteq V$

(2) Wenn $ztr_1.z_x.z_{xy}.z_y.ztr_2 \in V$ und $ztr_1.z_x.z'_{xy}.z_y.ztr_2 \in V$ mit $ztr_1, ztr_2 \in GZustand(IS)^*$, $z_x, z_{xy}, z'_{xy}, z_y \in GZustand(IS)$, und wenn $b_1, b_2 \in B$, $b_1 \neq b_2$, existieren mit: $z_x|_{b_1} \neq z_{xy}|_{b_1} = z_y|_{b_1}$, $z_{xy}|_{P\setminus b_1} = z_x|_{P\setminus b_1}$, $z_x|_{b_2} \neq z'_{xy}|_{b_2} = z_y|_{b_2}$, $z'_{xy}|_{P\setminus b_2} = z_x|_{P\setminus b_2}$, dann ist auch $ztr_1.z_x.z_y.ztr_2 \in V$.

(3) V ist minimal.

Dann gilt $V = \mathcal{V}[\![IS]\!]$.

Beweis.

Notation 5.5. Für eine Folge von globalen Aktivitätszuständen $z_0.z_1.z_2\ldots$ setze $\#_{\|}(z_0.z_1.z_2\ldots) := \sum_{i=1,2,\ldots} \#_{\|}(z_{i-1}.z_i)$, mit
$\#_{\|}(z_{i-1}.z_i) := |\{b \in B \mid \exists b' \in B, b' \neq b : z_{i-1}|_b \neq z_i|_b \wedge z_{i-1}|_{b'} \neq z_i|_{b'}\}|$.

„$\subseteq$".
Sei $ztr \in V$. Zu zeigen: $ztr \in \mathcal{V}[\![IS]\!]$.
Der Beweis wird durchgeführt mittels Induktion über $\#_{\|}(ztr)$.

IA.1: $\#_{\|}(ztr) = 0$.

Aus der Minimalitätsbedingung für V folgt: $ztr \in \mathcal{V}^{\mathrm{i}}[\![IS]\!]$ (Kriterium 1). $\mathcal{V}^{\mathrm{i}}[\![IS]\!]$ ist eine Teilmenge von $\mathcal{V}[\![IS]\!]$ nach Definition 5.2.

IA.2: $\#_{\|}(ztr) = 1$.

Entsprechend der Festlegungen für $\#_{\|}(\cdot)$ (Notation 5.5) kann dieser Fall nicht eintreten. (Die Existenz von b bedingt die Existenz von b'.)

IS: Sei $\#_{\|}(ztr) \geq 2$.

Es gilt $ztr \notin \mathcal{V}^{\mathrm{i}}[\![IS]\!]$. Wegen der Minimalitätsbedingung für V ist Kriterium 2 anwendbar und es existieren somit zwei Folgen von Aktivitätszuständen $ztr', ztr'' \in V$, aus denen ztr berechnet werden kann. Es existieren $ztr_1, ztr_2 \in GZustand(IS)^*$, $z_x, z_{xy}, z'_{xy}, z_y \in GZustand(IS)$, $b_1, b_2 \in B$ mit $b_1 \neq b_2$, $z_x|_{b_1} \neq z_{xy}|_{b_1} = z_y|_{b_1}$, $z_{xy}|_{P\setminus b_1} = z_x|_{P\setminus b_1}$, $z_x|_{b_2} \neq z'_{xy}|_{b_2} = z_y|_{b_2}$, $z'_{xy}|_{P\setminus b_2} = z_x|_{P\setminus b_2}$ und $ztr = ztr_1.z_x.z_y.ztr_2$, $ztr' = ztr_1.z_x.z_{xy}.z_y.ztr_2$, $ztr'' = ztr_1.z_x.z'_{xy}.z_y.ztr_2$.

Betrachte $\#_{\|}(\cdot)$ der einzelnen (Teil-)Folgen:

$\#_{\|}(ztr) = \#_{\|}(ztr_1.z_x) + \#_{\|}(z_x.z_y) + \#_{\|}(z_y.ztr_2)$ $(*_1)$
$\#_{\|}(ztr') = \#_{\|}(ztr_1.z_x) + \#_{\|}(z_x.z_{xy}) + \#_{\|}(z_{xy}.z_y) + \#_{\|}(z_y.ztr_2)$ $(*_2)$
$\#_{\|}(ztr'') = \#_{\|}(ztr_1.z_x) + \#_{\|}(z_x.z'_{xy}) + \#_{\|}(z'_{xy}.z_y) + \#_{\|}(z_y.ztr_2)$ $(*_3)$
Wegen $z_x|_{b_1} \neq z_{xy}|_{b_1} \wedge z_x|_{P\setminus b_1} = z_{xy}|_{P\setminus b_1}$ gilt: $\#_{\|}(z_x.z'_{xy}) = 0$. $(*_4)$
Wegen $z_x|_{b_2} \neq z'_{xy}|_{b_2} \wedge z_x|_{P\setminus b_2} = z'_{xy}|_{P\setminus b_2}$ gilt: $\#_{\|}(z_x.z'_{xy}) = 0$. $(*_5)$
Wegen $z_x|_{b_1} \neq z_y|_{b_1} \wedge z_x|_{b_2} \neq z_y|_{b_2} \wedge z_{xy}|_{b_1} = z_y|_{b_1} \wedge z_{xy}|_{P\setminus b_1} = z_x|_{P\setminus b_1}$ gilt:
$\#_{\|}(z_{xy}.z_y) < \#_{\|}(z_x.z_y)$. $(*_6)$
Wegen $z_x|_{b_1} \neq z_y|_{b_1} \wedge z_x|_{b_2} \neq z_y|_{b_2} \wedge z'_{xy}|_{b_2} = z_y|_{b_2} \wedge z'_{xy}|_{P\setminus b_2} = z_x|_{P\setminus b_2}$ gilt:
$\#_{\|}(z_{xy}.z_y) < \#_{\|}(z_x.z_y)$. $(*_7)$

Aus $(*_1)$,$(*_2)$,$(*_4)$,$(*_6)$ folgt $\#_{\|}(ztr') < \#_{\|}(ztr)$, und aus $(*_1)$,$(*_3)$,$(*_5)$,$(*_7)$ folgt $\#_{\|}(ztr'') < \#_{\|}(ztr)$.

Die Induktionsvoraussetzung ist anwendbar, sie liefert: $ztr' \in \mathcal{V}[\![IS]\!]$ und $ztr'' \in \mathcal{V}[\![IS]\!]$.

Die Struktur von ztr, ztr', ztr'' erlaubt die Anwendung von Satz 5.1.$\Leftarrow$, mit dem Ergebnis: $ztr \in \mathcal{V}[\![IS]\!]$.

„$\supseteq$“:
Sei $ztr \in \mathcal{V}[\![IS]\!]$. Zu zeigen: $ztr \in V$.
Der Beweis wird durchgeführt mittels Induktion über $\#_{\|}(ztr)$.

IA.1: $\#_{\|}(ztr) = 0$.

Mit Definition 5.2 folgt: $ztr \in \mathcal{V}^{\mathrm{i}}[\![IS]\!]$. Kriterium 1 des Satzes liefert direkt $\mathcal{V}^{\mathrm{i}}[\![IS]\!] \subseteq V$, also $ztr \in V$.

IA.2: $\#_{\|}(ztr) = 1$.

Dieser Fall kann gemäß der Festlegungen für $\#_{\|}(\cdot)$ nicht eintreten.

IS: Sei $\#_{\|}(ztr) \geq 2$.

Entsprechend der Definition von $\#_{\|}(\cdot)$ existieren $ztr_1, ztr_2 \in GZustand(IS)^*$, $z_x, z_y \in GZustand(IS)$, $b_1, b_2 \in B$ mit $b_1 \neq b_2$, so dass gilt: $ztr = ztr_1.z_x.z_y.ztr_2$ und $z_x|_{b_1} \neq z_y|_{b_1}$, $z_x|_{b_2} \neq z_y|_{b_2}$.

Konstruiere $z_{xy}, z'_{xy} \in GZustand(IS)$ derart, dass folgende Übereinstimmungen erfüllt sind: $z_{xy}|_{b_1} = z_y|_{b_1}$, $z_{xy}|_{P\setminus b_1} = z_x|_{P\setminus b_1}$, $z'_{xy}|_{b_2} = z_y|_{b_2}$, $z'_{xy}|_{P\setminus b_2} = z_x|_{P\setminus b_2}$. Die Konstruktion ist möglich und eindeutig.

Nach Satz 5.1 gilt: $\overbrace{ztr_1.z_x.z_{xy}.z_y.ztr_2}^{ztr'} \in \mathcal{V}[\![IS]\!]$ und $\overbrace{ztr_1.z_x.z'_{xy}.z_y.ztr_2}^{ztr''} \in \mathcal{V}[\![IS]\!]$. Gleichzeitig werden durch die Art der Konstruktion $(*_1)$-$(*_7)$ gewährleistet und damit $\#_{\|}(ztr') < \#_{\|}(ztr)$ und $\#_{\|}(ztr'') < \#_{\|}(ztr)$, analog zum ersten Teil. Die Induktionsvoraussetzung ist anwendbar, es folgt: $ztr' \in V$ und $ztr'' \in V$.

Ausgehend von ztr' und ztr'', deren Struktur erfüllt die geforderten Voraussetzungen, liefert Kriterium 2: $ztr_1.z_x.z_y.ztr_2 = ztr \in V$. $\square$

Der Beweis von Satz 5.4 beruht im Wesentlichen auf der Anwendung von Satz 5.1. Dabei kommt zum Tragen, dass sich die $\Leftarrow$-Richtung von Satz 5.1 direkt in Kriterium 2 des Satzes 5.4 wiederfindet. Da Satz 5.1 nur den Fall von gleichzeitigen Phasenqualitätswechseln für zwei Bereiche an einer Stelle der Trace behandelt, bedarf es zur Bildung beliebiger Parallelität bei beliebig vielen

Komponenten der iterierten Anwendung des Satzes. Beweistechnisch wird ein Induktionsverfahren angewendet, um diese Satzwiederholung erfassen zu können. Satz 5.4 ist konstruktiv, d.h. es kommt nicht nur zum Ausdruck, dass das Verhalten aus dem Interleaving Verhalten berechnet werden kann, sondern es wird explizit die Berechnungsvorschrift mit angegeben.

Aus Satz 5.4 folgt, dass das Interleaving Verhalten $\mathcal{V}^{\mathrm{i}}[\![IS]\!]$ eines I–Systems IS das gleiche oder ein größeres semantisches Informationspotential besitzt wie das „normale“ Verhalten $\mathcal{V}[\![IS]\!]$. Die Gleichheit der Ausdruckskraft ergibt sich schließlich aus der Kombination von Satz 5.4 mit Definition 5.2, in der $\mathcal{V}^{\mathrm{i}}[\![IS]\!]$ als Teilmenge von $\mathcal{V}[\![IS]\!]$ spezifiziert wird. Beachtet werden muss allerdings, dass es der Kenntnis des vollständigen Interleaving Verhaltens bedarf, um das Verhalten rekonstruieren zu können. Die Möglichkeit paralleler Phasenqualitätswechsel kann nur nachvollzogen werden, wenn bekannt ist, ob alle Permutationen der sequentiellen Ausführungen vorliegen.

5.2 Erweiterte Interleaving Casetrace-Semantik

Entsprechend der Betrachtung des Interleavings beim Verhalten $\mathcal{V}[\![IS]\!]$ eines I–Systems IS soll dieses jetzt für dessen Erweiterte Casetrace-Semantik $\mathcal{ECT}[\![IS]\!]$ erfolgen. Als erster Schritt werden hierzu, analog zum Vorgehen beim Verhalten, die strukturellen Eigenschaften der Erweiterten Casetrace-Semantik in Bezug auf Nebenläufigkeit und Sequentialität formuliert und bewiesen.
Der folgende Satz beschreibt, dass es zu einer Trace von $\mathcal{ECT}[\![IS]\!]$, in denen zwei aufeinanderfolgende und durch eine Bereichsmenge verbundene Cases das aus globaler Sicht gleichzeitige Auftreten mindestens zweier (freier oder erzwungener) Phasentransitionen in unterschiedlichen Bereichen repräsentieren, ebenfalls Traces in $\mathcal{ECT}[\![IS]\!]$ gibt, in denen die beiden Phasentransitionen nacheinander in beliebiger Reihenfolge auftreten, bei gleicher Resttrace. Die Umkehrrichtung gilt auch hier.

Satz 5.6 (Interleaving-Eigenschaft der Erweiterten Casetrace-Semantik). Sei IS ein I–System mit Phasenmenge P und Bereichsmenge B. Seien $ectr_1 \in (Case(IS).\mathcal{P}(B))^*$, $ectr_2 \in (\mathcal{P}(B).Case(IS))^*$, $c_x, c_{xy}, c'_{xy}, c_y \in Case(IS)$, $\delta, \delta_1, \delta_2, \delta'_1, \delta'_2 \subseteq B$ und $p_1, p_2 \in P$ mit $p_1 \neq p_2$. Unter den Voraussetzungen $\{p_1, p_2\} \subseteq c_y \setminus c_x$, $\{p_1\} = c_{xy} \setminus c_x$, $\{p_2\} = c'_{xy} \setminus c_x$, $\delta_1 = \delta \cap \{b(p_1)\}$, $\delta_2 = \delta \setminus \{b(p_1)\}$, $\delta'_1 = \delta \cap \{b(p_2)\}$, $\delta'_2 = \delta \setminus \{b(p_2)\}$, gilt:

$$\begin{aligned} ectr_1.c_x.\delta.c_y.ectr_2 \in \mathcal{ECT}[\![IS]\!] \quad \Leftrightarrow \quad & ectr_1.c_x.\delta_1.c_{xy}.\delta_2.c_y.ectr_2 \in \mathcal{ECT}[\![IS]\!] \wedge \\ & ectr_1.c_x.\delta'_1.c'_{xy}.\delta'_2.c_y.ectr_2 \in \mathcal{ECT}[\![IS]\!] \end{aligned}$$

Beweis.

„⇒“.
Sei $ectr_1.c_x.\delta.c_y.ectr_2 \in \mathcal{ECT}[\![IS]\!]$ und es gelten die Voraussetzungen aus dem Satz.

Nach Satz 4.20.b und Definition 4.19 existieren $ztr_1, ztr_2 \in GZustand(IS)^*$, $z_x, z_y \in GZustand(IS)$, so dass gilt:
$ztr_1.z_x.z_y.ztr_2 \in \mathcal{V}[\![IS]\!]$ und $\lfloor ztr_1.z_x.z_y.ztr_2 \rfloor^{\mathrm{e}} = ectr_1.c_x.\delta.c_y.ectr_2$, $c_x = zc(z_x)$, $c_y = zc(z_y)$ sowie $\delta = \{b \in B \mid \forall p \in b : (z_x(p) \neq 0 \wedge z_y(p) = 0) \Rightarrow z_x(p) \neq \mathsf{F}\}$.

Aus der Voraussetzung $\{p_1, p_2\} \subseteq c_y \setminus c_x$ folgt: $c_x \cap b(p_1) \neq c_y \cap b(p_1)$, $c_x \cap b(p_2) \neq c_y \cap b(p_2)$, $b(p_1) \neq b(p_2)$.

Betrachtet man die Definition von $zc(\cdot)$, es findet dort ein Übergang von Mengen nach Funktionen statt, ergibt sich: $z_x|_{b(p_1)} \neq z_y|_{b(p_1)}$, $z_x|_{b(p_2)} \neq z_y|_{b(p_2)}$, bei weiterhin $b(p_1) \neq b(p_2)$. (*)

Satz 5.1.⇒ findet Anwendung mit dem Ergebnis: $ztr_1.z_x.z_{xy}.z_y.ztr_2 \in \mathcal{V}[\![IS]\!]$ und $ztr_1.z_x.z'_{xy}.z_y.ztr_2 \in \mathcal{V}[\![IS]\!]$, wobei z_{xy} und z'_{xy} festgelegt sind durch $z_{xy}|_{b(p_1)} = z_y|_{b(p_1)}$, $z_{xy}|_{P \setminus b(p_1)} = z_x|_{P \setminus b(p_1)}$, $z'_{xy}|_{b(p_2)} = z_y|_{b(p_2)}$, $z'_{xy}|_{P \setminus b(p_2)} = z_x|_{P \setminus b(p_2)}$. (**)

Mit Satz 4.20.b wird wieder der Übergang zur Casetrace-Semantik erreicht. Man erhält:
$\lfloor ztr_1.z_x.z_{xy}.z_y.ztr_2 \rfloor^{\mathrm{e}} = ectr_1.c_x.\bar{\delta_1}.zc(z_{xy}).\bar{\delta_2}.c_y.ectr_2 \in \mathcal{ECT}[\![IS]\!]$ sowie $\lfloor ztr_1.z_x.z'_{xy}.z_y.ztr_2 \rfloor^{\mathrm{e}} = ectr_1.c_x.\bar{\delta'_1}.zc(z'_{xy}).\bar{\delta'_2}.c_y.ectr_2 \in \mathcal{ECT}[\![IS]\!]$ mit $\bar{\delta_1} = \{b \in B \mid \forall p \in b : (z_x(p) \neq 0 \wedge z_{xy}(p) =$

$0) \Rightarrow z_x(p) \neq \mathsf{F}\}$, $\bar{\delta_2} = \{b \in B \mid \forall p \in b : (z_{xy}(p) \neq 0 \wedge z_y(p) = 0) \Rightarrow z_{xy}(p) \neq \mathsf{F}\}$, $\bar{\delta_1'} = \{b \in B \mid \forall p \in b : (z_x(p) \neq 0 \wedge z'_{xy}(p) = 0) \Rightarrow z_x(p) \neq \mathsf{F}\}$, $\bar{\delta_2'} = \{b \in B \mid \forall p \in b : (z'_{xy}(p) \neq 0 \wedge z_y(p) = 0) \Rightarrow z'_{xy}(p) \neq \mathsf{F}\}$.

Die Übertragung der Einschränkungen für z_{xy} und z'_{xy} auf die Phasenmengen liefert: $zc(z_{xy}) \cap b(p_1) = c_y \cap b(p_1)$, $zc(z_{xy}) \cap P \setminus b(p_1) = c_x \cap P \setminus b(p_1)$, $zc(z'_{xy}) \cap b(p_2) = c_y \cap b(p_2)$, $zc(z'_{xy}) \cap P \setminus b(p_2) = c_x \cap P \setminus b(p_2)$.

Wegen $(*)$ und $(**)$ folgt zusätzlich: $\bar{\delta_1} = \{b(p_1)\}$ falls $b(p_1) \in \delta$ und $\bar{\delta_1} = \{\}$ sonst, $\bar{\delta_2} = \{b \in B \mid b \in \delta \setminus \{b(p_1)\}\}$, $\bar{\delta_1'} = \{b(p_2)\}$ falls $b(p_2) \in \delta$ und $\bar{\delta_1'} = \{\}$ sonst, $\bar{\delta_2'} = \{b \in B \mid b \in \delta \setminus \{b(p_2)\}\}$.

Wegen $\{p_1, p_2\} \subseteq c_y \setminus c_x$ folgt nun: $\{p_1\} = zc(z_{xy}) \setminus c_x$ und $\{p_2\} = zc(z'_{xy}) \setminus c_x$.
Das Umschreiben der Deltas führt zu $\bar{\delta_1} = \delta \cap \{b(p_1)\}$, $\bar{\delta_2} = \delta \setminus \{b(p_1)\}$, $\bar{\delta_1'} = \delta \cap \{b(p_2)\}$, $\bar{\delta_2'} = \delta \setminus \{b(p_2)\}$.

Unter den Voraussetzungen des Satzes ergeben sich folgende Gleichheiten: $zc(z_{xy}) = c_{xy}$, $zc(z'_{xy}) = c'_{xy}$, $\bar{\delta_1} = \delta_1$, $\bar{\delta_2} = \delta_2$, $\bar{\delta_1'} = \delta'_1$, $\bar{\delta_2'} = \delta'_2$. Die $\Rightarrow$-Richtung ist somit gezeigt.

„$\Leftarrow$“.
Seien $ctr_1.c_x.\delta_1.c_{xy}.\delta_2.c_y.ctr_2 \in \mathcal{ECT}[\![IS]\!]$ und $ctr_1.c_x.\delta'_1.c'_{xy}.\delta'_2.c_y.ctr_2 \in \mathcal{ECT}[\![IS]\!]$ und es gelten die Voraussetzungen aus dem Satz.

Die Definition der Erweiterten Casetrace-Semantik legt die einzelnen Deltas wie folgt fest: $\delta_1 = \{b \in B \mid \forall p \in b : (z_x(p) \neq 0 \wedge z_{xy}(p) = 0) \Rightarrow z_x(p) \neq \mathsf{F}\}$, $\delta_2 = \{b \in B \mid \forall p \in b : (z_{xy}(p) \neq 0 \wedge z_y(p) = 0) \Rightarrow z_{xy}(p) \neq \mathsf{F}\}$, $\delta'_1 = \{b \in B \mid \forall p \in b : (z_x(p) \neq 0 \wedge z'_{xy}(p) = 0) \Rightarrow z_x(p) \neq \mathsf{F}\}$, $\delta'_2 = \{b \in B \mid \forall p \in b : (z'_{xy}(p) \neq 0 \wedge z_y(p) = 0) \Rightarrow z'_{xy}(p) \neq \mathsf{F}\}$. $(***)$

Nach Satz 4.20.b existieren $ztr_1, ztr_2 \in GZustand(IS)^*$, $z_x, z_{xy}, z'_{xy}, z_y \in GZustand(IS)$, so dass gilt: $ztr_1.z_x.z_{xy}.z_y.ztr_2 \in \mathcal{V}[\![IS]\!]$ und $ztr_1.z_x.z'_{xy}.z_y.ztr_2 \in \mathcal{V}[\![IS]\!]$ mit $\lfloor ztr_1.z_x.z_{xy}.z_y.ztr_2 \rfloor^{\mathrm{e}} = ctr_1.c_x.\delta_1.c_{xy}.\delta_2.c_y.ctr_2$, $\lfloor ztr_1.z_x.z'_{xy}.z_y.ztr_2 \rfloor^{\mathrm{e}} = ctr_1.c_x.\delta'_1.c'_{xy}.\delta'_2.c_y.ctr_2$ und $c_x = zc(z_x)$, $c_{xy} = zc(z_{xy})$, $c'_{xy} = zc(z'_{xy})$, $c_y = zc(z_y)$.

Aus der Voraussetzung $\{p_1, p_2\} \subseteq c_y \setminus c_x$ folgt: $c_x \cap b(p_1) \neq c_y \cap b(p_1)$, $c_x \cap b(p_2) \neq c_y \cap b(p_2)$, $b(p_1) \neq b(p_2)$. Aus $\{p_1, p_2\} \subseteq c_y \setminus c_x \wedge \{p_1\} = c_{xy} \setminus c_x$ folgt: $c_{xy} \cap b(p_1) = c_y \cap b(p_1)$, $c_{xy} \cap P \setminus b(p_1) = c_x \cap P \setminus b(p_1)$. Aus $\{p_1, p_2\} \subseteq c_y \setminus c_x \wedge \{p_2\} = c'_{xy} \setminus c_x$ folgt: $c'_{xy} \cap b(p_2) = c_y \cap b(p_2)$, $c'_{xy} \cap P \setminus b(p_2) = c_x \cap P \setminus b(p_2)$.

Die Definition von $zc(\cdot)$, der dort auftretende Übergang von Mengen nach Funktionen, bedingt die Gültigkeit der folgenden Einschränkungen: $z_x|_{b(p_1)} \neq z_y|_{b(p_1)}$, $z_x|_{b(p_2)} \neq z_y|_{b(p_2)}$, $z_{xy}|_{b(p_1)} = z_y|_{b(p_1)}$, $z_{xy}|_{P \setminus b(p_1)} = z_x|_{P \setminus b(p_1)}$, $z'_{xy}|_{b(p_2)} = z_y|_{b(p_2)}$, $z'_{xy}|_{P \setminus b(p_2)} = z_x|_{P \setminus b(p_2)}$, bei weiterhin $b(p_1) \neq b(p_2)$. $(****)$

Satz 5.1.$\Leftarrow$ findet Anwendung mit dem Ergebnis: $ztr_1.z_x.z_y.ztr_2 \in \mathcal{V}[\![IS]\!]$.

Satz 4.20.b stellt den Übergang zurück zur Erweiterten Casetrace-Semantik her. Man erhält: $\lfloor ztr_1.z_x.z_y.ztr_2 \rfloor^{\mathrm{e}} = ctr_1.c_x.\bar{\delta}.c_y.ctr_2 \in \mathcal{ECT}[\![IS]\!]$ mit $\bar{\delta} = \{b \in B \mid \forall p \in b : (z_x(p) \neq 0 \wedge z_y(p) = 0) \Rightarrow z_x(p) \neq \mathsf{F}\}$.

Unter Berücksichtigung der Gleichheiten von $(***)$ und unter Hinzuziehung der Einschränkungen von $(****)$ folgt: $\bar{\delta} \cap \{b(p_1)\} = \delta_1$, $\bar{\delta} \setminus \{b(p_1)\} = \delta_2$, $\bar{\delta} \cap \{b(p_2)\} = \delta'_1$, $\bar{\delta} \setminus \{b(p_2)\} = \delta'_2$.

Die Voraussetzungen des Satzes liefern unter anderem: $\delta_1 = \delta \cap \{b(p_1)\}$, $\delta_2 = \delta \setminus \{b(p_1)\}$, $\delta'_1 = \delta \cap \{b(p_2)\}$, $\delta'_2 = \delta \setminus \{b(p_2)\}$.

Somit gilt zum einen $\bar{\delta} \cap \{b(p_1)\} = \delta \cap \{b(p_1)\}$ gleichzeitig mit $\bar{\delta} \setminus \{b(p_1)\} = \delta \setminus \{b(p_1)\}$ und zum anderen $\bar{\delta} \cap \{b(p_2)\} = \delta \cap \{b(p_2)\}$ gleichzeitig mit $\bar{\delta} \setminus \{b(p_2)\} = \delta \setminus \{b(p_2)\}$. In beiden Fällen folgt direkt $\bar{\delta} = \delta$. Damit gilt auch die $\Leftarrow$-Richtung des Satzes. $\square$

Der Beweis stützt sich auf die in Abschnitt 4.4 festgestellten semantischen Beziehungen zwischen der Erweiterten Casetrace-Semantik und dem Verhalten eines I–Systems. Diese Beziehungen ermöglichen einen Übergang bei den Beweisschritten von der Erweiterten Casetrace-Semantik zum Verhalten, um dann Satz 5.1 anzuwenden. Die Ergebnisse können dann wieder auf die Erweiterte Casetrace-Semantik zurückübertragen werden.

Die $\Rightarrow$-Richtung des Satzes 5.6 drückt aus, dass globalzeitlich gleichzeitige Phasentransitionen bei zwei unterschiedlichen Komponenten eines verteilten Systems globalzeitlich gesehen auch in beliebiger Reihenfolge auftreten können, bei sonst gleichem restlichen Systemverhalten. Die Hintereinanderanwendung dieser Satzrichtung ermöglicht die Auflösung von Nebenläufigkeit in beliebig vielen Komponenten. Die $\Leftarrow$-Richtung erlaubt es, mögliche nebenläufige Phasentransitionen aus sequentiellen abzuleiten. Das besondere Merkmal der Erweiterten Casetrace-Semantik sind die Bereichsmengen innerhalb der einzelnen Traces, die der Klassifizierung (als frei oder erzwungen) der auftretenden Phasentransitionen dienen. Es ist notwendig, dass diese Bereichsmengen innerhalb der durch die beiden Satzrichtungen angegebenen Konstruktionen korrekt rekonstruiert werden können. Für beide Richtungen ist dies der Fall, wie der Beweis zeigt.

Bei der *Erweiterten Interleaving Casetrace-Semantik* eines I–Systems werden gegenüber der „normalen“ Erweiterten Casetrace-Semantik nur Traces berücksichtigt, in denen aufeinander folgende Cases, die definitionsgemäß durch eine dazwischenliegende Bereichsmenge verbunden sind, genau eine Phasentransition in jeweils genau einem Bereich repräsentieren.

Definition 5.7 (Erweiterte Interleaving Casetrace-Semantik eines I–Systems). Für ein I–System IS ist die *Erweiterte Interleaving Casetrace-Semantik* $\mathcal{ECT}^{\mathrm{i}}[\![IS]\!]$ definiert als:

$$\mathcal{ECT}^{\mathrm{i}}[\![IS]\!] \subseteq \mathcal{ECT}[\![IS]\!] \subseteq Case(IS).\,(\mathcal{P}(B).Case(IS))^*$$

mit

$$\mathcal{ECT}^{\mathrm{i}}[\![IS]\!] = \{c_0\delta_1c_1\delta_2c_2\ldots \in \mathcal{ECT}[\![IS]\!] \mid \forall i = 1,2,\ldots : |c_i \setminus c_{i-1}| = 1\}.$$

Die Erweiterte Interleaving Casetrace-Semantik von IS *bzgl. eines Start-Cases* $c_0 \in Case(IS)$ ist gegeben durch:

$$\mathcal{ECT}^{\mathrm{i}}[\![IS]\!](c_0) = \mathcal{ECT}^{\mathrm{i}}[\![IS]\!] \cap \mathcal{ECT}[\![IS]\!](c_0). \qquad \square$$

Die Erweiterte Interleaving Casetrace-Semantik $\mathcal{ECT}^{\mathrm{i}}[\![IS]\!]$ wird aus der Erweiterten Casetrace-Semantik $\mathcal{ECT}[\![IS]\!]$ abgeleitet. Wie schon bei der Definition des Interleaving Verhaltens bewegt man sich bei der Ableitung innerhalb der formalen Ebene und greift nicht auf die algorithmische Ebene und auf die Ausführungen des IS zugeordneten $\mathrm{V_I}$Systems zurück. Möchte man diesen Bezug allerdings herstellen, reicht eine kleine Modifikation von Definition 4.16: Alle Ausführungen, in denen gleichzeitig bei unterschiedlichen Komponenten Phasentransitionen auftreten, werden dann nicht berücksichtigt. Auf die genaue formale Präzisierung soll hier verzichtet werden, da diese trivial ist und die Einführung einer alternativen Definition zu keinen weiteren Erkenntnissen bei den Beziehungen zwischen den einzelnen Semantiken führt.
Das folgende Beispiel verdeutlicht die Berechnung von Teilen der Erweiterten Interleaving Casetrace-Semantik eines I–Systems, unter Rückgriff auf dessen Erweiterte Casetrace-Semantik.

Beispiel 5.8. Bestimmt werden sollen Elemente der Erweiterten Interleaving Casetrace-Semantik $\mathcal{ECT}^{\mathrm{i}}[\![IS_1]\!]$ des I–Systems IS_1 aus Beispiel 2.2. Verwendet werden hierzu die Notationen und Ergebnisse aus Beispiel 4.17.

Die Ausführung 3 lieferte: $\overbrace{c_0.\delta_1.c_1.\delta_2.c_2.\delta_3.c_3.\delta_4.c_4.\delta_5.c_5.\delta_6.c_6.\delta_7.c_7}^{ectr} \in \mathcal{ECT}[\![IS_1]\!]$.
Wegen $|c_i \setminus c_{i-1}| = 2 \neq 1$ für $i \in \{1,4\}$ gilt: $ectr \notin \mathcal{ECT}^{\mathrm{i}}[\![IS_1]\!]$.

Die wiederholte Anwendung von Satz 5.6.$\Rightarrow$ liefert:

$\overbrace{c_0.\delta_a.c_{01}.\delta_b.c_1.\delta_2.c_2.\delta_3.c_3.\delta_c.c_{34}.\delta_d.c_4.\delta_5.c_5.\delta_6.c_6.\delta_7.c_7}^{ectr_a} \in \mathcal{ECT}[\![IS_1]\!]$,

$\overbrace{c_0.\delta'_a.c'_{01}.\delta'_b.c_1.\delta_2.c_2.\delta_3.c_3.\delta_c.c_{34}.\delta_d.c_4.\delta_5.c_5.\delta_6.c_6.\delta_7.c_7}^{ectr_b} \in \mathcal{ECT}[\![IS_1]\!]$,

$\overbrace{c_0.\delta_a.c_{01}.\delta_b.c_1.\delta_2.c_2.\delta_3.c_3.\delta'_c.c'_{34}.\delta'_d.c_4.\delta_5.c_5.\delta_6.c_6.\delta_7.c_7}^{ectr_c} \in \mathcal{ECT}[\![IS_1]\!]$,

$\overbrace{c_0.\delta'_a.c'_{01}.\delta'_b.c_1.\delta_2.c_2.\delta_3.c_3.\delta'_c.c'_{34}.\delta'_d.c_4.\delta_5.c_5.\delta_6.c_6.\delta_7.c_7}^{ectr_d} \in \mathcal{ECT}[\![IS_1]\!]$,
mit:

$\delta_a = \{b_1\}$, $c_{01} = \{p_2, q_2, v_2\}$, $\delta_b = \{b_2\}$, $\delta'_a = \{b_2\}$, $c'_{01} = \{p_1, q_3, v_2\}$, $\delta'_b = \{b_1\}$,
$\delta_c = \{b_2\}$, $c_{34} = \{p_2, q_3, v_1\}$, $\delta_d = \{\}$, $\delta'_c = \{\}$, $c'_{34} = \{p_2, q_2, v_2\}$, $\delta'_d = \{b_2\}$.

Für alle vier Traces gilt, dass die Mengensubtraktion aufeinander folgender Cases (nur getrennt durch eine Bereichsmenge) eine einelementige Menge liefert. Entsprechend Definition 5.7 gilt deshalb: $ectr_a, ectr_b, ectr_c, ectr_d \in \mathcal{ECT}^{\mathrm{i}}[\![IS_1]\!]$. □

Definition 5.7 gibt explizit an, wie die Erweiterte Interleaving Casetrace-Semantik $\mathcal{ECT}^{\mathrm{i}}[\![IS]\!]$ eines I–Systems IS aus dessen Erweiterter Casetrace-Semantik $\mathcal{ECT}[\![IS]\!]$ abgeleitet werden kann. Es stellt sich als nächstes die Frage, ob sich auch die Rückrichtung beweisen lässt, d.h. dass $\mathcal{ECT}[\![IS]\!]$ aus $\mathcal{ECT}^{\mathrm{i}}[\![IS]\!]$ berechnet werden kann. Der folgende Satz bejaht dieses.

Satz 5.9 (Rekonstruktion der Erweiterten Casetrace-Semantik). Sei IS ein I–System mit Phasenmenge P und Bereichsmenge B. Die Menge ECT sei durch folgende drei Kriterien festgelegt:

(1) $\mathcal{ECT}^{\mathrm{i}}[\![IS]\!] \subseteq ECT$

(2) Wenn $ectr_1.c_x.\delta_1.c_{xy}.\delta_2.c_y.ectr_2 \in ECT$ und $ectr_1.c_x.\delta'_1.c'_{xy}.\delta'_2.c_y.ectr_2 \in ECT$ mit $ectr_1 \in (Case(IS).\mathcal{P}(B))^*$, $ectr_2 \in (\mathcal{P}(B).Case(IS))^*$, $c_x, c_{xy}, c'_{xy}, c_y \in Case(IS)$, $\delta_1, \delta_2, \delta'_1, \delta'_2 \subseteq B$, und wenn $p_1, p_2 \in P$, $\delta \subseteq B$ existieren mit: $p_1 \neq p_2$, $\{p_1, p_2\} \subseteq c_y \setminus c_x$, $\{p_1\} = c_{xy} \setminus c_x$, $\{p_2\} = c'_{xy} \setminus c_x$, $\delta_1 = \delta \cap \{b(p_1)\}$, $\delta_2 = \delta \setminus \{b(p_1)\}$, $\delta'_1 = \delta \cap \{b(p_2)\}$, $\delta'_2 = \delta \setminus \{b(p_2)\}$, dann ist auch $ectr_1.c_x.\delta.c_y.ectr_2 \in ECT$.

(3) ECT ist minimal.

Dann gilt $ECT = \mathcal{ECT}[\![IS]\!]$.

Beweis.

Notation 5.10. Für eine alternierende Folge von Cases und Bereichsmengen $c_0.\delta_1.c_1.\delta_2.c_2\ldots$ setze $\#_{\|}(c_0.\delta_1.c_1.\delta_2.c_2\ldots) := \sum\limits_{i=1,2,\ldots} \#_{\|}(c_{i-1}.c_i)$, mit
$\#_{\|}(c_{i-1}.c_i) := |\{p \in P \mid \exists p' \in P, p' \neq p : \{p, p'\} \subseteq c_i \setminus c_{i-1}\}|$.

„$\subseteq$“.
Sei $ectr \in ECT$. Zu zeigen: $ectr \in \mathcal{ECT}[\![IS]\!]$.
Der Beweis wird durchgeführt mittels Induktion über $\#_{\|}(ectr)$.

IA.1: $\#_{\|}(ectr) = 0$.

Aus der Minimalitätsbedingung für ECT folgt: $ectr \in \mathcal{ECT}^{\mathrm{i}}[\![IS]\!]$ (Kriterium 1). $\mathcal{ECT}^{\mathrm{i}}[\![IS]\!]$ ist eine Teilmenge von $\mathcal{ECT}[\![IS]\!]$ nach Definition 5.7.

IA.2: $\#_{\|}(ectr) = 1$.

Entsprechend der Festlegungen für $\#_{\|}(\cdot)$ (Notation 5.10) kann dieser Fall nicht eintreten. (Die Existenz von p bedingt die Existenz von p'.)

IS: Sei $\#_{\|}(ectr) \geq 2$.

Es gilt $ectr \notin \mathcal{ECT}^{\mathrm{i}}[\![IS]\!]$. Wegen der Minimalitätsbedingung für ECT ist Kriterium 2 anwendbar und es existieren somit zwei alternierende Folgen von Cases und Bereichsmengen $ectr', ectr'' \in ECT$, aus denen $ectr$ berechnet werden kann. Es existieren $ectr_1 \in (Case(IS).\mathcal{P}(B))^*$, $ectr_2 \in (\mathcal{P}(B).Case(IS))^*$, $c_x, c_{xy}, c'_{xy}, c_y \in Case(IS)$, $\delta, \delta_1, \delta_2, \delta'_1, \delta'_2 \subseteq B$, $p_1, p_2 \in P$ mit $p_1 \neq p_2$, $\{p_1, p_2\} \subseteq c_y \setminus c_x$, $\{p_1\} = c_{xy} \setminus c_x$, $\{p_2\} = c'_{xy} \setminus c_x$, $\delta_1 = \delta \cap \{b(p_1)\}$, $\delta_2 = \delta \setminus \{b(p_1)\}$, $\delta'_1 = \delta \cap \{b(p_2)\}$, $\delta'_2 = \delta \setminus \{b(p_2)\}$ und $ectr = ectr_1.c_x.\delta.c_y.ectr_2$, $ectr' = ectr_1.c_x.\delta_1.c_{xy}.\delta_2.c_y.ectr_2$, $ectr'' = ectr_1.c_x.\delta'_1.c'_{xy}.\delta'_2.c_y.ectr_2$.

Betrachte $\#_{\|}(\cdot)$ der einzelnen (Teil-)Folgen:

$$\#_{\|}(ectr) = \#_{\|}(ectr_1.c_x) + \#_{\|}(c_x.\delta.c_y) + \#_{\|}(c_y.ectr_2) \qquad (*_1)$$
$$\#_{\|}(ectr') = \#_{\|}(ectr_1.c_x) + \#_{\|}(c_x.\delta_1.c_{xy}) + \#_{\|}(c_{xy}.\delta_2.c_y) + \#_{\|}(c_y.ectr_2) \qquad (*_2)$$

$\#_{\|}(ectr'') = \#_{\|}(ectr_1.c_x) + \#_{\|}(c_x.\delta_1'.c_{xy}') + \#_{\|}(c_{xy}'.\delta_2'.c_y) + \#_{\|}(c_y.ectr_2)$ $(*_3)$
Wegen $c_{xy} \setminus c_x = \{p_1\}$ gilt: $\#_{\|}(c_x.\delta_1.c_{xy}) = 0.$ $(*_4)$
Wegen $c_{xy}' \setminus c_x = \{p_2\}$ gilt: $\#_{\|}(c_x.\delta_1'.c_{xy}') = 0.$ $(*_5)$
Wegen $(\{p_1, p_2\} \subseteq c_y \setminus c_x \wedge \{p_1\} = c_{xy} \setminus c_x) \Rightarrow (p_1 \notin c_y \setminus c_{xy})$ gilt: $\#_{\|}(c_{xy}.\delta_2.c_y) < \#_{\|}(c_x.\delta.c_y)$.$(*_6)$
Wegen $(\{p_1, p_2\} \subseteq c_y \setminus c_x \wedge \{p_2\} = c_{xy}' \setminus c_x) \Rightarrow (p_2 \notin c_y \setminus c_{xy}')$ gilt: $\#_{\|}(c_{xy}'.\delta_2'.c_y) < \#_{\|}(c_x.\delta.c_y)$.$(*_7)$

Aus $(*_1)$,$(*_2)$,$(*_4)$,$(*_6)$ folgt $\#_{\|}(ectr') < \#_{\|}(ectr)$, und aus $(*_1)$,$(*_3)$,$(*_5)$,$(*_7)$ folgt $\#_{\|}(ectr'') < \#_{\|}(ectr)$.

Die Induktionsvoraussetzung ist anwendbar, sie liefert: $ectr' \in \mathcal{ECT}[\![IS]\!]$ und $ectr'' \in \mathcal{ECT}[\![IS]\!]$.

Die Struktur von $ectr$, $ectr'$, $ectr''$ erlaubt die Anwendung von Satz 5.6.$\Leftarrow$, mit dem Ergebnis: $ectr \in \mathcal{ECT}[\![IS]\!]$.

„$\supseteq$“:
Sei $ectr \in \mathcal{ECT}[\![IS]\!]$. Zu zeigen: $ectr \in ECT$.
Der Beweis wird durchgeführt mittels Induktion über $\#_{\|}(ectr)$.

IA.1: $\#_{\|}(ectr) = 0$.

Mit Definition 5.7 folgt: $ectr \in \mathcal{ECT}^{\mathrm{i}}[\![IS]\!]$. Kriterium 1 des Satzes liefert direkt $\mathcal{ECT}^{\mathrm{i}}[\![IS]\!] \subseteq ECT$, also $ectr \in ECT$.

IA.2: $\#_{\|}(ectr) = 1$.

Dieser Fall kann gemäß der Festlegungen für $\#_{\|}(\cdot)$ nicht eintreten.

IS: Sei $\#_{\|}(ectr) \geq 2$.

Entsprechend der Definition von $\#_{\|}(\cdot)$ existieren $ectr_1 \in (Case(IS).\mathcal{P}(B))^*$, $ectr_2 \in (\mathcal{P}(B).Case(IS))^*$, $c_x, c_y \in Case(IS)$, $\delta \subseteq B$, $p_1, p_2 \in P$ mit $p_1 \neq p_2$, so dass gilt: $ectr = ectr_1.z_x.\delta.z_y.ectr_2$ und $\{p_1, p_2\} \subseteq c_y \setminus c_x$.

Konstruiere $c_{xy}, c_{xy}' \in Case(IS)$ und $\delta_1, \delta_2, \delta_1', \delta_2' \subseteq B$ derart, dass folgende Übereinstimmungen erfüllt sind: $\{p_1\} = c_{xy} \setminus c_x$, $\{p_2\} = c_{xy}' \setminus c_x$, $\delta_1 = \delta \cap \{b(p_1)\}$, $\delta_2 = \delta \setminus \{b(p_1)\}$, $\delta_1' = \delta \cap \{b(p_2)\}$, $\delta_2' = \delta \setminus \{b(p_2)\}$. Die Konstruktion ist möglich und eindeutig.

Nach Satz 5.6 gilt: $\overbrace{ectr_1.c_x.\delta_1.c_{xy}.\delta_2.c_y.ectr_2}^{ectr'} \in \mathcal{ECT}[\![IS]\!]$ und $\overbrace{ectr_1.c_x.\delta_1'.c_{xy}'.\delta_2'.c_y.ectr_2}^{ectr''} \in \mathcal{ECT}[\![IS]\!]$. Gleichzeitig werden durch die Art der Konstruktion $(*_1)$-$(*_7)$ gewährleistet und damit $\#_{\|}(ectr') < \#_{\|}(ectr)$ und $\#_{\|}(ectr'') < \#_{\|}(ectr)$, analog zum ersten Teil. Die Induktionsvoraussetzung ist anwendbar, es folgt: $ectr' \in ECT$ und $ectr'' \in ECT$.

Ausgehend von $ectr'$ und $ectr''$, deren Struktur erfüllt die geforderten Voraussetzungen, liefert Kriterium 2: $ectr_1.c_x.\delta.c_y.ectr_2 = ectr \in ECT$. $\square$

Der Beweis von Satz 5.9 beruht auf der Anwendung von Satz 5.6. Dabei kommt zum Tragen, dass sich die $\Leftarrow$-Richtung von Satz 5.6 direkt in Kriterium 2 des Satzes 5.9 wiederfindet. Da Satz 5.6 nur den Fall von gleichzeitigen Phasentransitionen in zwei Bereichen an einer Stelle der Trace behandelt, bedarf es zur Bildung beliebiger Parallelität bei beliebig vielen Komponenten der iterierten Anwendung des Satzes. Beweistechnisch wird ein Induktionsverfahren angewendet, um diese Satzwiederholung erfassen zu können. Satz 5.9 ist konstruktiv, d.h. es kommt nicht nur zum Ausdruck, dass die Erweiterte Casetrace-Semantik aus der Erweiterten Interleaving Casetrace-Semantik berechnet werden kann, sondern es wird explizit die Berechnungsvorschrift mit angegeben.

Aus Satz 5.9 folgt, dass die Erweiterte Interleaving Casetrace-Semantik $\mathcal{ECT}^{\mathrm{i}}[\![IS]\!]$ eines I–Systems IS das gleiche oder ein größeres semantisches Informationspotential besitzt wie die Erweiterte Casetrace-Semantik $\mathcal{ECT}[\![IS]\!]$. Die Gleichheit der Ausdrucksstärke ergibt sich aus der Kombination von Satz 5.9 mit Definition 5.7, in der $\mathcal{ECT}^{\mathrm{i}}[\![IS]\!]$ als Teilmenge von $\mathcal{ECT}[\![IS]\!]$ spezifiziert wird. Beachtet werden muss allerdings, dass es der Kenntnis der vollständigen Erweiterten Interleaving Casetrace-Semantik bedarf, um die Erweiterte Casetrace-Semantik rekonstruieren zu können. Die

Möglichkeit gleichzeitiger Phasentransitionen kann nur nachvollzogen werden, wenn bekannt ist, ob alle Permutationen der sequentiellen Ausführungen vorliegen.

Bemerkung 5.11. Bewegt man sich nur innerhalb der Erweiterten Interleaving Casetrace-Semantik, ergibt sich eine gewisse Redundanz bei den vermittelten Informationen durch die Bereichsmengen δ_i. Die δ_i brauchen in diesem Fall nämlich nur festzuhalten, ob ein freier oder ein erzwungener Caseübergang erfolgt. Die Aufzählung einzelner Bereiche ist nicht notwendig, da durch die Interleaving-Eigenschaft immer nur ein Bereich in Frage kommt, der durch die beiden umgebenden Cases eindeutig bestimmbar ist. Genauer: Mit $IS \in ISystem$, $ectr_1 \in (Case(IS).\mathcal{P}(B))^*$, $ectr_2 \in (\mathcal{P}(B).Case(IS))^*$, $c_x, c_y \in Case(IS)$, $\delta \subseteq B$ gilt:

$ectr_1.c_x.\delta.c_y.ectr_2 \in \mathcal{ECT}^{\mathrm{i}}[\![IS]\!]$
$\Rightarrow$ {Definition 5.7} $|c_y \setminus c_x| = 1$
$\Rightarrow$ {Spezifikation der δ_i gemäß Definition 4.16} $\delta = \emptyset \vee \delta = \{b(p)\}$ mit $\{p\} = c_y \setminus c_x$.

Um die Einheitlichkeit bei den Notationen der Erweiterten Casetrace-Semantik und der Erweiterten Interleaving Casetrace-Semantik aufrechtzuerhalten, wird diese Redundanz in Kauf genommen, anstatt zusätzliche Notationen einzuführen, die auf die Mengenaufzählung verzichten.

5.3 Interleaving Casetrace-Semantik

Nach Betrachtung des Interleavings beim Verhalten $\mathcal{V}[\![IS]\!]$ und bei der Erweiterten Casetrace-Semantik $\mathcal{ECT}[\![IS]\!]$ eines I–Systems IS in den vorangegangenen Abschnitten, soll dieses abschließend noch für die Casetrace-Semantik $\mathcal{CT}[\![IS]\!]$, als dritte der bisher eingeführten Semantiken von I–Systemen, erfolgen. Da die Casetrace-Semantik eine Verallgemeinerung der Erweiterten Casetrace-Semantik darstellt, indem die zusätzlichen Bereichsmengen zur Klassifizierung der Phasentransitionen (als frei oder erregt) wegfallen, und da Interleaving nur über die auftretenden Cases definiert wird, können die vorherigen Sätze und Beweise aus Abschnitt 5.2 im Wesentlichen übernommen werden, vereinfacht dadurch, dass keine Korrektheit der Bereichsmengen mitbetrachtet und überprüft werden muss. Als erstes werden die strukturellen Eigenschaften der Casetrace-Semantik in Bezug auf Nebenläufigkeit und Sequentialität formuliert und bewiesen.
Der folgende Satz sagt aus, dass es zu einer Trace von $\mathcal{CT}[\![IS]\!]$, in denen zwei aufeinander folgende Cases das aus globaler Sicht gleichzeitige Auftreten mindestens zweier Phasentransitionen in unterschiedlichen Bereichen repräsentieren, ebenfalls Traces in $\mathcal{CT}[\![IS]\!]$ gibt, in denen die beiden Phasentransitionen nacheinander in beliebiger Reihenfolge auftreten, bei gleicher Resttrace. Die Umkehrrichtung gilt ebenfalls.

Satz 5.12 (Interleaving-Eigenschaft der Casetrace-Semantik). Sei IS ein I–System mit Phasenmenge P und Bereichsmenge B. Seien $ctr_1, ctr_2 \in Case(IS)^*$, $c_x, c_{xy}, c'_{xy}, c_y \in Case(IS)$ und $p_1, p_2 \in P$ mit $p_1 \neq p_2$. Unter den Voraussetzungen $\{p_1, p_2\} \subseteq c_y \setminus c_x$, $\{p_1\} = c_{xy} \setminus c_x$, $\{p_2\} = c'_{xy} \setminus c_x$ gilt:

$$ctr_1.c_x.c_y.ctr_2 \in \mathcal{CT}[\![IS]\!] \quad \Leftrightarrow \quad ctr_1.c_x.c_{xy}.c_y.ctr_2 \in \mathcal{CT}[\![IS]\!] \wedge ctr_1.c_x.c'_{xy}.c_y.ctr_2 \in \mathcal{CT}[\![IS]\!]$$

Beweis.

„$\Rightarrow$“.
Sei $ctr_1.c_x.c_y.ctr_2 \in \mathcal{CT}[\![IS]\!]$, und es gelten die Voraussetzungen aus dem Satz.

Nach Satz 4.20.d existieren $c_i \in Case(IS)$ und $\delta, \bar{\delta}_i \subseteq B$, $i = 1, 2, \ldots$, so dass gilt:
$\overbrace{c_1.\bar{\delta}_1.c_2.\bar{\delta}_2 \ldots c_k.\bar{\delta}_k}^{ectr_1}.c_x.\delta.c_y.\overbrace{\bar{\delta}_{k+1}.c_{k+1}.\bar{\delta}_{k+2}.c_{k+2}\ldots}^{ectr_2} \in \mathcal{ECT}[\![IS]\!]$, mit $k \in I\!N_0$ und $c_1 c_2 \ldots c_k = ctr_1$ sowie $c_{k+1} c_{k+2} \ldots = ctr_2$.

Satz 5.6.$\Rightarrow$ garantiert die Existenz von $\bar{c}_{xy}, \bar{c}'_{xy} \in Case(IS)$, $\delta_1, \delta_2, \delta'_1, \delta'_2 \subseteq B$ mit:
$ectr_1.c_x.\delta_1.\bar{c}_{xy}.\delta_2.c_y.ectr_2 \in \mathcal{ECT}[\![IS]\!]$ und $ectr_1.c_x.\delta'_1.\bar{c}'_{xy}.\delta'_2.c_y.ectr_2 \in \mathcal{ECT}[\![IS]\!]$,
wobei gilt: $\{p_1\} = \bar{c}_{xy} \setminus c_x$ und $\{p_2\} = \bar{c}'_{xy} \setminus c_x$. $(*)$
Die Einschränkungen für die Deltas können vernachlässigt werden, da sie beim Übergang zur Casetrace-Semantik keine Rolle spielen.

Über Satz 4.20.d erfolgt der Übergang zur Casetrace-Semantik, mit dem Ergebnis:
$ctr_1.c_x.\bar{c}_{xy}.c_y.ctr_2 \in \mathcal{CT}[\![IS]\!]$ und $ctr_1.c_x.\bar{c}'_{xy}.c_y.ctr_2 \in \mathcal{CT}[\![IS]\!]$.

Aus den Voraussetzungen $\{p_1, p_2\} \subseteq c_y \setminus c_x$, $\{p_1\} = c_{xy} \setminus c_x$ und $\{p_2\} = c'_{xy} \setminus c_x$ des Satzes, in Verbindung mit $(*)$, folgt: $\bar{c}_{xy} = c_{xy}$ und $\bar{c}'_{xy} = c'_{xy}$. Damit ist die $\Rightarrow$-Richtung des Satzes gezeigt.

„$\Leftarrow$".
Seien $ctr_1.c_x.c_{xy}.c_y.ctr_2 \in \mathcal{CT}[\![IS]\!]$ und $ctr_1.c_x.c'_{xy}.c_y.ctr_2 \in \mathcal{CT}[\![IS]\!]$, und es gelten die Voraussetzungen aus dem Satz.

Nach Satz 4.20.d existieren $c_i \in Case(IS)$ und $\delta_1, \delta_2, \delta'_1, \delta'_2, \bar{\delta}_i \subseteq B$, $i = 1, 2, \ldots$, so dass gilt:

$$\overbrace{c_1.\bar{\delta}_1.c_2.\bar{\delta}_2 \ldots c_k.\bar{\delta}_k}^{ectr_1}.c_x.\delta_1.c_{xy}.\delta_2.c_y.\overbrace{\bar{\delta}_{k+1}.c_{k+1}.\bar{\delta}_{k+2}.c_{k+2}.\ldots}^{ectr_2} \in \mathcal{ECT}[\![IS]\!],$$
$$\underbrace{c_1.\bar{\delta}_1.c_2.\bar{\delta}_2 \ldots c_k.\bar{\delta}_k}_{ectr_1}.c_x.\delta'_1.c'_{xy}.\delta'_2.c_y.\underbrace{\bar{\delta}_{k+1}.c_{k+1}.\bar{\delta}_{k+2}.c_{k+2}.\ldots}_{ectr_2} \in \mathcal{ECT}[\![IS]\!],$$

mit $k \in I\!N_0$ und $c_1 c_2 \ldots c_k = ctr_1$ sowie $c_{k+1} c_{k+2} \ldots = ctr_2$.

Satz 5.6.$\Leftarrow$ garantiert, unter Einbeziehung der Voraussetzungen, die Existenz von $\delta \subseteq B$ mit: $ectr_1.c_x.\delta.c_y.ectr_2 \in \mathcal{ECT}[\![IS]\!]$.
Die Abhängigkeiten der einzelnen Deltas können wiederum vernachlässigt werden.

Über Satz 4.20.d erfolgt der Übergang zur Casetrace-Semantik, mit dem Ergebnis:
$ctr_1.c_x.c_y.ctr_2 \in \mathcal{CT}[\![IS]\!]$, womit auch die $\Leftarrow$-Richtung des Satzes gezeigt ist. □

Im Beweis wurden die in Kapitel 4.4 beschriebenen semantischen Beziehungen zwischen der Casetrace- und der Erweiterten Casetrace-Semantik eines I–Systems ausgenutzt, um im Kontext der Erweiterten Casetrace-Semantik Satz 5.6 anzuwenden. Die Ergebnisse können dann auf die Casetrace-Semantik übertragen werden, indem die zusätzlichen Bereichsmengen ausgelassen werden.
Die Interpretation des Satzes 5.12 entspricht der von Satz 5.6: Die $\Rightarrow$-Richtung drückt aus, dass globalzeitlich gleichzeitige Phasentransitionen bei zwei unterschiedlichen Komponenten eines verteilten Systems auch in beliebiger Reihenfolge auftreten können, bei sonst gleichem restlichen Systemverhalten. Die Hintereinanderanwendung dieser Satzrichtung ermöglicht die Auflösung von Nebenläufigkeit in beliebig vielen Komponenten. Die $\Leftarrow$-Richtung erlaubt es, mögliche nebenläufige Phasentransitionen aus sequentiellen abzuleiten.

Bei der *Interleaving Casetrace-Semantik* eines I–Systems werden gegenüber der „normalen" Casetrace-Semantik nur Traces berücksichtigt, in denen aufeinander folgende Cases genau eine Phasentransition in jeweils genau einem Bereich repräsentieren.

Definition 5.13 (Interleaving Casetrace-Semantik eines I–Systems). Für ein I–System IS ist die *Interleaving Casetrace-Semantik* $\mathcal{CT}^{\mathrm{i}}[\![IS]\!]$ definiert als:

$$\mathcal{CT}^{\mathrm{i}}[\![IS]\!] \subseteq \mathcal{CT}[\![IS]\!] \subseteq Case(IS)^+$$

mit

$$\mathcal{CT}^{\mathrm{i}}[\![IS]\!] = \{c_0 c_1 c_2 \ldots \in \mathcal{CT}[\![IS]\!] \mid \forall i = 1, 2, \ldots : |c_i \setminus c_{i-1}| = 1\}.$$

Die Interleaving Casetrace-Semantik von IS *bzgl. eines Start-Cases* $c_0 \in Case(IS)$ ist gegeben durch:

$$\mathcal{CT}^{\mathrm{i}}[\![IS]\!](c_0) = \mathcal{CT}^{\mathrm{i}}[\![IS]\!] \cap \mathcal{CT}[\![IS]\!](c_0).$$ □

Die Interleaving Casetrace-Semantik $\mathcal{CT}^{\mathrm{i}}[\![IS]\!]$ wird in Definition 5.13 direkt aus der Casetrace-Semantik $\mathcal{CT}[\![IS]\!]$ abgeleitet, auf der formalen Ebene, d.h. ohne Einbeziehung des IS zugeordneten $\mathrm{V_I}$Systems $V_I System(IS)$. Analog zur Erweiterten Interleaving Casetrace-Semantik kann eine Definition der Interleaving Casetrace-Semantik auch mit Bezug auf die algorithmische Ebene, d.h. unter Rückgriff auf die Ausführungen von $V_I System(IS)$, angegeben werden. Dazu ist nur eine kleine Modifikation von Definition 4.8 notwendig. Ausführungen, in denen parallel bei unterschiedlichen Komponenten Phasentransitionen auftreten, werden nicht beachtet. Auf die genaue formale Präzisierung wird an dieser Stelle ebenfalls verzichtet.

Das folgende Beispiel verdeutlicht die Berechnung von Teilen der Interleaving Casetrace-Semantik eines I–Systems, unter Rückgriff auf dessen Casetrace-Semantik und Erweiterte Casetrace-Semantik.

Beispiel 5.14. Bestimmt werden sollen Elemente der Interleaving Casetrace-Semantik $\mathcal{CT}^{\mathrm{i}}[\![IS_1]\!]$ des I–Systems IS_1 aus Beispiel 2.2. Ausgangspunkt dazu sind die Notationen und Ergebnisse aus Beispiel 4.17.

Die Ausführung 3 lieferte: $c_0.\delta_1.c_1.\delta_2.c_2.\delta_3.c_3.\delta_4.c_4.\delta_5.c_5.\delta_6.c_6.\delta_7.c_7 \in \mathcal{ECT}[\![IS_1]\!]$.
Unter Vernachlässigung der Bereichsmengen, entsprechend Satz 4.20.d, ergibt sich:
$\overbrace{c_0.c_1.c_2.c_3.c_4.c_5.c_6.c_7}^{ctr} \in \mathcal{CT}[\![IS_1]\!]$.
Wegen $|c_i \setminus c_{i-1}| = 2 \neq 1$ für $i \in \{1,4\}$ gilt: $ctr \notin \mathcal{CT}^{\mathrm{i}}[\![IS_1]\!]$.

Die wiederholte Anwendung von Satz 5.12.⇒ liefert:
$\overbrace{c_0.c_{01}.c_1.c_2.c_3.c_{34}.c_4.c_5.c_6.c_7}^{ctr_a} \in \mathcal{CT}[\![IS_1]\!]$, $\overbrace{c_0.c'_{01}.c_1.c_2.c_3.c_{34}.c_4.c_5.c_6.c_7}^{ctr_b} \in \mathcal{CT}[\![IS_1]\!]$,
$\overbrace{c_0.c_{01}.c_1.c_2.c_3.c'_{34}.c_4.c_5.c_6.c_7}^{ctr_c} \in \mathcal{CT}[\![IS_1]\!]$, $\overbrace{c_0.c'_{01}.c_1.c_2.c_3.c'_{34}.c_4.c_5.c_6.c_7}^{ctr_d} \in \mathcal{CT}[\![IS_1]\!]$,
mit:
$c_{01} = \{p_2, q_2, v_2\}$, $c'_{01} = \{p_1, q_3, v_2\}$, $c_{34} = \{p_2, q_3, v_1\}$, $c'_{34} = \{p_2, q_2, v_2\}$.

Für alle vier Traces gilt, dass die Mengensubtraktion aufeinander folgender Cases eine einelementige Menge liefert. Entsprechend Definition 5.13 gilt deshalb: $ctr_a, ctr_b, ctr_c, ctr_d \in \mathcal{CT}^{\mathrm{i}}[\![IS_1]\!]$.□

Definition 5.13 gibt explizit an, wie die Interleaving Casetrace-Semantik $\mathcal{CT}^{\mathrm{i}}[\![IS]\!]$ eines I–Systems IS aus dessen Casetrace-Semantik $\mathcal{CT}[\![IS]\!]$ abgeleitet werden kann. Der folgende Satz zeigt, dass auch die Rückrichtung gilt, d.h. dass $\mathcal{CT}[\![IS]\!]$ aus $\mathcal{CT}^{\mathrm{i}}[\![IS]\!]$ berechnet werden kann.

Satz 5.15 (Rekonstruktion der Casetrace-Semantik). Sei IS ein I–System mit Phasenmenge P und Bereichsmenge B. Die Menge CT sei durch folgende drei Kriterien festgelegt:

(1) $\mathcal{CT}^{\mathrm{i}}[\![IS]\!] \subseteq CT$

(2) Wenn $ctr_1.c_x.c_{xy}.c_y.ctr_2 \in CT$ und $ctr_1.c_x.c'_{xy}.c_y.ctr_2 \in CT$ mit $ctr_1, ctr_2 \in Case(IS)^*$, $c_x, c_{xy}, c'_{xy}, c_y \in Case(IS)$, und wenn $p_1, p_2 \in P$ existieren mit: $p_1 \neq p_2$, $\{p_1, p_2\} \subseteq c_y \setminus c_x$, $\{p_1\} = c_{xy} \setminus c_x$, $\{p_2\} = c'_{xy} \setminus c_x$, dann ist auch $ctr_1.z_x.z_y.ctr_2 \in CT$.

(3) CT ist minimal.

Dann gilt $CT = \mathcal{CT}[\![IS]\!]$.

Beweis.

Notation 5.16. Für eine Folge von Cases $c_0.c_1.c_2\ldots$ setze $\#_{\|}(c_0.c_1.c_2\ldots) := \sum\limits_{i=1,2,\ldots} \#_{\|}(c_{i-1}.c_i)$, mit $\#_{\|}(c_{i-1}.c_i) := |\{p \in P \mid \exists p' \in P, p' \neq p : \{p,p'\} \subseteq c_i \setminus c_{i-1}\}|$.

„⊆“.
Sei $ctr \in CT$. Zu zeigen: $ctr \in \mathcal{CT}[\![IS]\!]$.
Der Beweis wird durchgeführt mittels Induktion über $\#_{\|}(ctr)$.

IA.1: $\#_{\|}(ctr) = 0$.

Aus der Minimalitätsbedingung für CT folgt: $ctr \in \mathcal{CT}^{\mathrm{i}}[\![IS]\!]$ (Kriterium 1). $\mathcal{CT}^{\mathrm{i}}[\![IS]\!]$ ist eine Teilmenge von $\mathcal{CT}[\![IS]\!]$ nach Definition 5.13.

IA.2: $\#_{\|}(ctr) = 1$.

Entsprechend der Festlegungen für $\#_{\|}(\cdot)$ (Notation 5.16) kann dieser Fall nicht eintreten. (Die Existenz von p bedingt die Existenz von p'.)

IS: Sei $\#_{\|}(ctr) \geq 2$.

Es gilt $ctr \notin \mathcal{CT}^{\mathrm{i}}[\![IS]\!]$. Wegen der Minimalitätsbedingung für CT ist Kriterium 2 anwendbar und es existieren somit zwei Folgen von Cases $ctr', ctr'' \in CT$, aus denen ctr berechnet werden kann. Es existieren $ctr_1, ctr_2 \in Case(IS)^*$, $c_x, c_{xy}, c'_{xy}, c_y \in Case(IS)$, $p_1, p_2 \in P$ mit $p_1 \neq p_2$, $\{p_1, p_2\} \subseteq c_y \backslash c_x$, $\{p_1\} = c_{xy} \backslash c_x$, $\{p_2\} = c'_{xy} \backslash c_x$ und $ctr = ctr_1.c_x.c_y.ctr_2$, $ctr' = ctr_1.c_x.c_{xy}.c_y.ctr_2$, $ctr'' = ctr_1.c_x.c'_{xy}.c_y.ctr_2$.

Betrachte $\#_{\|}(\cdot)$ der einzelnen (Teil-)Folgen:

$\#_{\|}(ctr) = \#_{\|}(ctr_1.c_x) + \#_{\|}(c_x.c_y) + \#_{\|}(c_y.ctr_2)$ $(*_1)$
$\#_{\|}(ctr') = \#_{\|}(ctr_1.c_x) + \#_{\|}(c_x.c_{xy}) + \#_{\|}(c_{xy}.c_y) + \#_{\|}(c_y.ctr_2)$ $(*_2)$
$\#_{\|}(ctr'') = \#_{\|}(ctr_1.c_x) + \#_{\|}(c_x.c'_{xy}) + \#_{\|}(c'_{xy}.c_y) + \#_{\|}(c_y.ctr_2)$ $(*_3)$
Wegen $c_{xy} \setminus c_x = \{p_1\}$ gilt: $\#_{\|}(c_x.c_{xy}) = 0$. $(*_4)$
Wegen $c'_{xy} \setminus c_x = \{p_2\}$ gilt: $\#_{\|}(c_x.c'_{xy}) = 0$. $(*_5)$
Wegen $(\{p_1, p_2\} \subseteq c_y \setminus c_x \wedge \{p_1\} = c_{xy} \setminus c_x) \Rightarrow (p_1 \notin c_y \setminus c_{xy})$ gilt: $\#_{\|}(c_{xy}.c_y) < \#_{\|}(c_x.c_y)$. $(*_6)$
Wegen $(\{p_1, p_2\} \subseteq c_y \setminus c_x \wedge \{p_2\} = c'_{xy} \setminus c_x) \Rightarrow (p_2 \notin c_y \setminus c'_{xy})$ gilt: $\#_{\|}(c'_{xy}.c_y) < \#_{\|}(c_x.c_y)$. $(*_7)$

Aus $(*_1)$,$(*_2)$,$(*_4)$,$(*_6)$ folgt $\#_{\|}(ctr') < \#_{\|}(ctr)$, und aus $(*_1)$,$(*_3)$,$(*_5)$,$(*_7)$ folgt $\#_{\|}(ctr'') < \#_{\|}(ctr)$.

Die Induktionsvoraussetzung ist anwendbar, sie liefert: $ctr' \in \mathcal{CT}[\![IS]\!]$ und $ctr'' \in \mathcal{CT}[\![IS]\!]$.

Die Struktur von ctr, ctr', ctr'' erlaubt die Anwendung von Satz 5.12.$\Leftarrow$, mit dem Ergebnis: $ctr \in \mathcal{CT}[\![IS]\!]$.

„$\supseteq$“:
Sei $ctr \in \mathcal{CT}[\![IS]\!]$. Zu zeigen: $ctr \in CT$.
Der Beweis wird durchgeführt mittels Induktion über $\#_{\|}(ctr)$.

IA.1: $\#_{\|}(ctr) = 0$.

Mit Definition 5.13 folgt: $ctr \in \mathcal{CT}^{\mathrm{i}}[\![IS]\!]$. Kriterium 1 des Satzes liefert direkt $\mathcal{CT}^{\mathrm{i}}[\![IS]\!] \subseteq CT$, also $ctr \in CT$.

IA.2: $\#_{\|}(ctr) = 1$.

Dieser Fall kann gemäß der Festlegungen für $\#_{\|}(\cdot)$ nicht eintreten.

IS: Sei $\#_{\|}(ctr) \geq 2$.

Entsprechend der Definition von $\#_{\|}(\cdot)$ existieren $ctr_1, ctr_2 \in Case(IS)^*$, $c_x, c_y \in Case(IS)$, $p_1, p_2 \in P$ mit $p_1 \neq p_2$, so dass gilt: $ctr = ctr_1.z_x.z_y.ctr_2$ und $\{p_1, p_2\} \subseteq c_y \setminus c_x$.

Konstruiere $c_{xy}, c'_{xy} \in Case(IS)$ derart, dass folgende Übereinstimmungen erfüllt sind: $\{p_1\} = c_{xy} \setminus c_x$, $\{p_2\} = c'_{xy} \setminus c_x$. Die Konstruktion ist möglich und eindeutig.

Nach Satz 5.12 gilt: $\overbrace{ctr_1.c_x.c_{xy}.c_y.ctr_2}^{ctr'} \in \mathcal{CT}[\![IS]\!]$ und $\overbrace{ctr_1.c_x.c'_{xy}.c_y.ctr_2}^{ctr''} \in \mathcal{CT}[\![IS]\!]$. Gleichzeitig werden durch die Art der Konstruktion $(*_1)$-$(*_7)$ gewährleistet und damit $\#_{\|}(ctr') < \#_{\|}(ctr)$ und $\#_{\|}(ctr'') < \#_{\|}(ctr)$, analog zum ersten Teil. Die Induktionsvoraussetzung ist anwendbar, es folgt: $ctr' \in CT$ und $ctr'' \in CT$.

Ausgehend von ctr' und ctr'', deren Struktur erfüllt die geforderten Voraussetzungen, liefert Kriterium 2: $ctr_1.c_x.c_y.ctr_2 = ctr \in CT$. □

Der Beweis von Satz 5.15 verläuft analog zum Beweis von Satz 5.9, unter Rückgriff auf Satz 5.12. Die Bereichsmengen (Deltas) können dabei vernachlässigt werden, da sie bei der Casetrace-Semantik, im Gegensatz zur Erweiterten Casetrace-Semantik, nicht vorkommen. Satz 5.15 ist ebenfalls konstruktiv, d.h. es wird direkt eine Berechnungsvorschrift mit angegeben, wie die Casetrace-Semantik aus der Interleaving Casetrace-Semantik gewonnen werden kann.

Aus der Kombination von Satz 5.15 und Definition 5.13 folgt die Gleichheit in der Ausdruckskraft von der Interleaving Casetrace-Semantik $\mathcal{CT}^{\mathrm{i}}[\![IS]\!]$ und der Casetrace-Semantik $\mathcal{CT}[\![IS]\!]$ eines

I–Systems IS, weil jeweils eine Semantik der beiden aus der anderen berechnet werden kann. Da die Möglichkeit des Auftretens mehrerer zeitgleicher Phasentransitionen nur nachvollzogen werden kann, wenn alle Permutationen der sequentiellen Ausführungen vorliegen, bedarf es der Kenntnis der vollständigen Interleaving Casetrace-Semantik, um die Casetrace-Semantik rekonstruieren zu können.

In diesem Kapitel wurden von den einzelnen bisher vorgestellten Semantiken eines I–Systems, d.h. vom Verhalten, von der Casetrace-Semantik und von der Erweiterten Casetrace-Semantik, Teilmengen betrachtet, in denen nur Interleavings vorkommen und nebenläufige Ereignisse durch bestimmte Permutationen von Interleavings modelliert werden können, sofern diese bekannt sind. Im nächsten Kapitel werden die Interleaving-Varianten der einzelnen Semantiken eines I–Systems benutzt, um endliche Zustandsgraphen als kompakte anschauliche Darstellungen der Semantiken zu definieren.

Kapitel 6

Zustandsgraphen

Graphische Modelle sind wichtige Werkzeuge sowohl zur Darstellung von verteilten Strukturen als auch zur Darstellung von kausalen Abhängigkeiten. Durch Graphen mit ihren bildlichen Symbolen können (bis zu einem bestimmten Grad) auch komplexe Abhängigkeiten zwischen Objekten anschaulich dargestellt werden. Sie bieten sich deshalb an zum Einsatz in Tools für formale Modelle, um das Verständnis und die Nachvollziehbarkeit modellspezifischer Aussagen zu erleichtern. I–Systeme besitzen bereits für ihre formale Struktur und die Elemente der Dynamik graphische Darstellungen, welche in den Kapiteln 2 und 3 vorgestellt wurden.
In diesem Kapitel sollen nun graphische Darstellungsformen für die Semantiken eines I–Systems betrachtet werden. Hierzu werden unterschiedliche *Zustandsgraphen* eingeführt. Jeder Knoten eines (gerichteten) Zustandsgraphen entspricht einem diskreten Systemzustand. Die einzelnen Knoten werden durch Kanten miteinander verbunden, welche Zustandsübergänge repräsentieren (vgl. [55]). Im Bereich der Petri-Netze werden die so konstruierten Graphen als „Erreichbarkeitsgraphen" [80] oder „occurrence graphs" [50] bezeichnet. Bei I–Systemen entsprechen die Cases und die globalen Aktivitätszustände den diskreten Systemzuständen. Die Zustandsübergänge (Kanten) werden durch die Ausführungen der zugeordneten V_ISysteme festgelegt und lassen sich aus den Interleaving-Varianten der Trace-Semantiken ablesen. Auf diese Weise lassen sich *Verhaltensgraphen*, *Casegraphen* und *Erweiterte Casegraphen* für I–Systeme definieren. In diesem Kapitel werden die einzelnen Definitionen vorgestellt. Aus der Endlichkeit der Mengen der Cases und der globalen Aktivitätszustände eines I–Systems folgt die Endlichkeit der Zustandsgraphen.

Im Rahmen der Spezifikation von Semantiken für I–Systeme werden in dieser Arbeit zwei Darstellungsformen zur Repräsentation der Ausführungen des zugeordneten V_ISystems betrachtet: Pfade in endlichen Zustandsgraphen und Traces als Elemente von Trace-Semantiken. In diesem Kapitel wird untersucht, ob beide Darstellungsformen als gleichwertig anzusehen sind im Hinblick auf die Modellierung des Systemverhaltens eines verteilten Systems.

6.1 Verhaltensgraph

Beim *Verhaltensgraphen* eines I–Systems wird die Knotenmenge durch die relevanten globalen Aktivitätszustände des I–Systems gebildet. Das sind die globalen Aktivitätszustände, für die es jeweils mindestens eine Ausführung des zugeordneten V_ISystems gibt, in der sie als z-Globalbelegung auftreten (Definition 3.15). Wegen Satz 5.1 ändert sich die Menge der relevanten globalen Aktivitätszustände beim Übergang zum Interleaving Verhalten nicht, d.h. wenn man alle Ausführungen unberücksichtigt lässt, die globalzeitlich gleichzeitige Phasentransitionen in unterschiedlichen Komponenten beinhalten. Die Kantenmenge richtet sich nach dem Interleaving Verhalten des I–Systems. Finden sich zwei Knoten nacheinander als Teiltrace bei einem Element des Interleaving Verhaltens wieder, so existiert eine entsprechende Kante im Graphen. Die Richtung orientiert sich an der Reihenfolge der Knoten in der Teiltrace. Zusammenfassend ergibt sich die nachfolgende Definition.

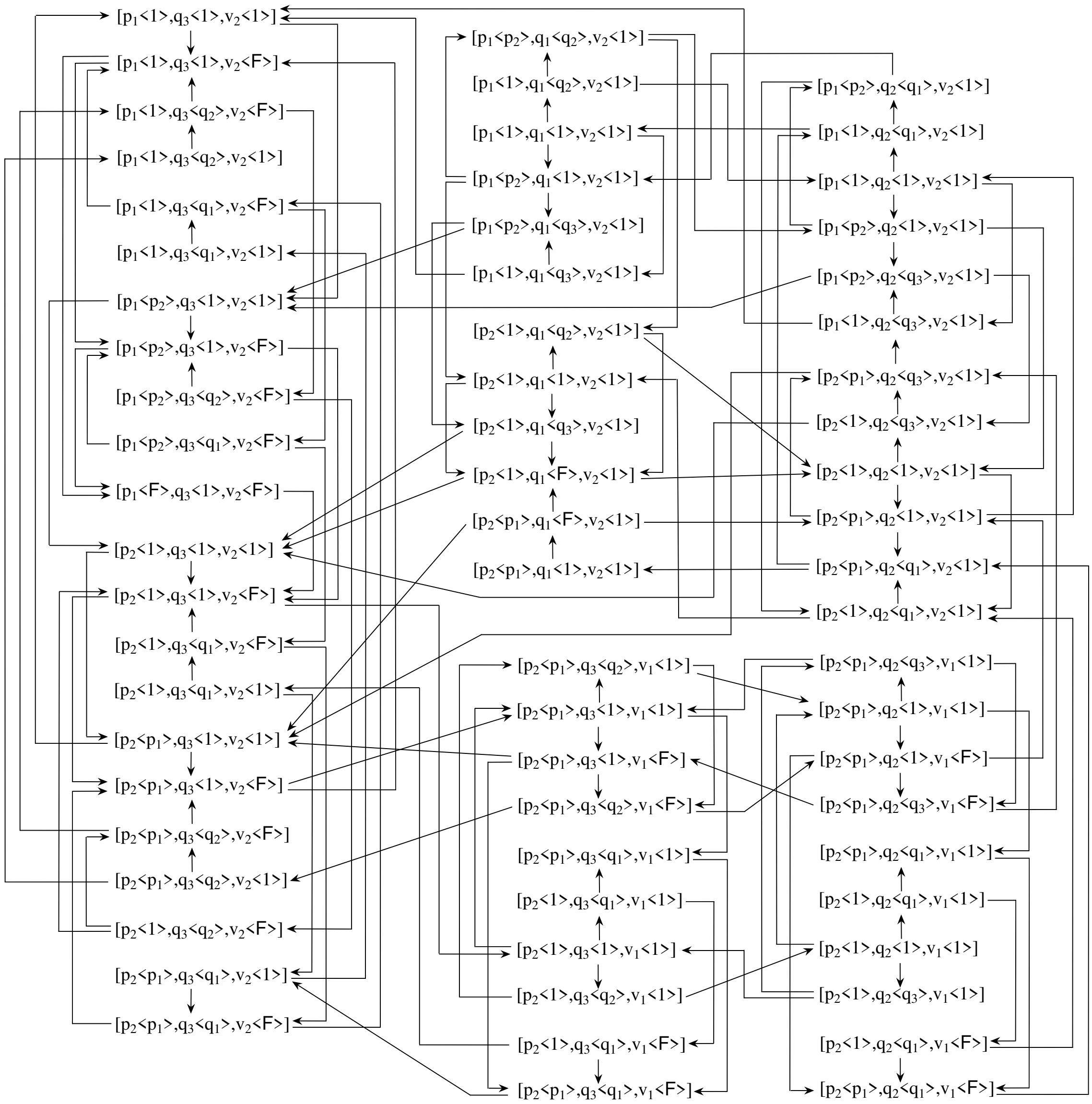


Abbildung 6.1: Verhaltensgraph $VG(IS_1)$

Definition 6.1 (Verhaltensgraph eines I–Systems). Für ein I–System IS ist der *Verhaltensgraph* $VG(IS) := (Z, \rightarrow)$ festgelegt durch:

(1) $Z = RelGZustand(IS)$

(2) $\rightarrow \subseteq Z \times Z$ mit:

$(z_1, z_2) \in\rightarrow$ gdw. $\exists ztr_1, ztr_2 \in GZustand(IS)^* : ztr_1.z_1.z_2.ztr_2 \in \mathcal{V}^{\mathrm{i}}[\![IS]\!]$ □

Da die Menge $RelGZustand(IS)$ definitionsgemäß eine Teilmenge von $GZustand(IS)$ und $GZustand(IS)$ immer endlich ist, ist auch die Knotenmenge des Verhaltensgraphens und damit der Graph selbst endlich.

Beispiel 6.2. Der Verhaltensgraph $VG(IS_1) = (Z, \rightarrow)$ zu dem I–System IS_1 aus Beispiel 2.2 ist in Abbildung 6.1 graphisch dargestellt. Die relevanten globalen Aktivitätszustände von IS_1 bilden die Knotenmenge Z. Die Pfeile geben die Richtung der Kanten an: Es existiert ein Pfeil von Knoten z_1 nach Knoten z_2 gdw. $(z_1, z_2) \in\rightarrow$. Der Verhaltensgraph von IS_1 besitzt 66 Knoten und 138 Kanten. □

Das Beispiel zeigt, dass schon kleine I–Systeme große Verhaltensgraphen mit sich bringen können. Dabei besteht die Gefahr, dass der Vorteil der graphischen Repräsentation von Systemabläufen verloren geht. Um dem entgegen zu wirken, kann man sich auf für den Anwender relevante Ausschnitte des Graphen beschränken oder mit Projektionen auf ausgewählte Bereichsmengen arbeiten. Die Grundlagen zu letzterem werden in Kapitel 9 präsentiert.

Bemerkung 6.3. Die große Komplexität des Verhaltensgraphens ist eine notwendige Konsequenz aus der Reichhaltigkeit der Phasenqualitäten. (Diese selbst waren als notwendige Merkmale lokaler Zustände motiviert worden, wenn man die relevanten Details von Zwängen und autonomem Verhalten vollständig erfassen will.) Sieht man von diesen Qualitäten ab, dann ergibt sich eine sehr stark reduzierte Struktur, der so genannte Casegraph, der in Abschnitt 6.2 definiert wird. Man gibt gerade die Information auf, ob ein Übergang frei/autonom erfolgt oder erzwungen ist (in letzterem Fall, von welcher Seite der Zwang kommt bzw. ob er weitergeleitet ist). Der zu Abbildung 6.1 gehörige Casegraph ist in Abbildung 6.3 dargestellt.

Kanten im Verhaltensgraphen $VG(IS)$ eines I–Systems IS repräsentieren bestimmte Systemereignisse. Da sich die Kanten direkt aus den zweielementigen Teiltraces des Interleaving Verhaltens $\mathcal{V}^{\mathrm{i}}[\![IS]\!]$ ergeben und das Interleaving Verhalten eine Teilmenge des Verhaltens $\mathcal{V}[\![IS]\!]$ ist, ist es der Einheitlichkeit des Sprachgebrauchs wegen sinnvoll, Interpretationen, wie sie für das Verhalten definiert worden sind (siehe Definition 4.5), zu übernehmen.

Definition 6.4 (Ereignisse im Verhaltensgraphen). Sei IS ein I–System, b ein Bereich von IS und $p, q \in b$. Sei $VG(IS) = (Z, \rightarrow)$ der Verhaltensgraph von IS. Eine Kante $(z_1, z_2) \in\rightarrow$ repräsentiert folgende Ereignisse:

a) Bei $z_1(p) = 1$ und $z_2(p) = q$ sagen wir, dass b *eine Entscheidung trifft*, nach q zu wechseln.

b) Bei $z_1(p) = 1$ und $z_2(p) = \mathsf{F}$ sagen wir, dass b *instabil wird*.

c) Bei $z_1(p) = q$ und $z_2(p) = 1$ sagen wir, dass die *Entscheidung* von b, nach q zu wechseln, *zurückgenommen wird*.

d) Bei $z_1(p) = q$ und $z_2(q) = 1$ sagen wir, dass eine *freie Phasentransition* von p nach q in b auftritt.

e) Bei $z_1(p) = \mathsf{F}$ und $z_2(q) = 1$ sagen wir, dass eine *erzwungene Phasentransition* von p nach q in b auftritt. □

Pfade im Verhaltensgraphen eines I–Systems, d.h. Knotenfolgen entlang der gerichteten Kanten, spiegeln, als Aneinanderreihung von oben spezifizierten Ereignissen, eine fortlaufende Aktivität des I–Systems wider. Entsprechend den Traces bei den Trace-Semantiken können sie als Mittel zur Systemanalyse und -verifikation eingesetzt werden.

Mit der Definition des Verhaltensgraphen $VG(IS)$ stellt sich die Frage nach dessen Ausdruckskraft im Vergleich zum Verhalten $\mathcal{V}[\![IS]\!]$ eines I–Systems IS. Nach Definition 6.1 wird der Verhaltensgraph aus dem Interleaving Verhalten $\mathcal{V}^{\mathrm{i}}[\![IS]\!]$ abgeleitet, das wiederum, als Folgerung aus Definition 5.2 und Satz 5.4, von gleicher Ausdruckskraft ist wie das Verhalten. Folglich finden sich alle Informationen, die der Verhaltensgraph enthält, auch im Verhalten wieder. Es bleibt noch zu klären, ob das Interleaving Verhalten aus dem Verhaltensgraphen vollständig zurückgewonnen werden kann. Der folgende Satz zeigt, dass das nicht immer der Fall ist, da das Interleaving Verhalten eine besondere Eigenschaft besitzt, die beim Übergang zum Graphen verloren geht.

Satz 6.5 (Besonderheit des Interleaving Verhaltens). Es existieren I–Systeme IS, für die gilt: $\exists ztr_1, ztr_2, ztr_1', ztr_2' \in GZustand(IS)^*$, $z \in GZustand(IS)$ mit $ztr_1.z.ztr_2 \in \mathcal{V}^{\mathrm{i}}[\![IS]\!]$, $ztr_1'.z.ztr_2' \in \mathcal{V}^{\mathrm{i}}[\![IS]\!]$ und $ztr_1'.z.ztr_2 \notin \mathcal{V}^{\mathrm{i}}[\![IS]\!]$.

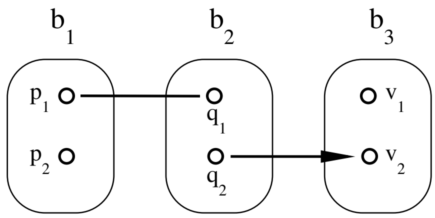


Abbildung 6.2: IS_2

Beweis. Sei IS_2 als das I–System aus Abbildung 6.2 gegeben. Betrachte die folgende Ausführung Π_1 von $V_I System(IS_2)$:

$$\overbrace{[p_1{<}1{>}, q_2{<}1{>}, v_1{<}1{>}]}^{z_0} \xrightarrow{b_2.\mathrm{A3}} \overbrace{[p_1{<}1{>}, q_2{<}q_1{>}, v_1{<}1{>}]}^{z_1} \xrightarrow{b_3.\mathrm{A3}} \overbrace{[p_1{<}1{>}, q_2{<}q_1{>}, v_1{<}v_2{>}]}^{z_2}$$
$$\xrightarrow{b_3.\mathrm{A4}} \overbrace{[p_1{<}1{>}, q_2{<}q_1{>}, v_2{<}1{>}]}^{z_3} \xrightarrow{b_3.\mathrm{A4}} \overbrace{[p_1{<}1{>}, q_2{<}q_1{>}, v_2{<}\mathsf{F}{>}]}^{z_4} \xrightarrow{b_2.\mathrm{A10}}$$
$$\overbrace{[p_1{<}1{>}, q_2{<}1{>}, v_2{<}\mathsf{F}{>}]}^{z_5} \xrightarrow{b_1.\mathrm{A11}} \overbrace{[p_1{<}\mathsf{F}{>}, q_2{<}1{>}, v_2{<}\mathsf{F}{>}]}^{z_6} \xrightarrow{b_3.\mathrm{A5}} \overbrace{[p_1{<}\mathsf{F}{>}, q_2{<}1{>}, v_1{<}1{>}]}^{z_7}$$
$$\xrightarrow{b_1.\mathrm{A12}} \overbrace{[p_1{<}1{>}, q_2{<}1{>}, v_1{<}1{>}]}^{z_8}$$

Somit gilt $z_0z_1z_2z_3z_4z_5z_6z_7z_8 \in \mathcal{V}[\![IS_2]\!]$, und da für $i = 1, \ldots, 8$ gilt: $\exists!\, b \in B$ mit $z_i|_b \neq z_{i-1}|_b$, folgt sogar $z_0z_1z_2z_3z_4z_5z_6z_7z_8 \in \mathcal{V}^{\mathrm{i}}[\![IS_2]\!]$.

Als weitere Ausführung Π_2 von $V_I System(IS_2)$ existiert:

$$\overbrace{[p_2{<}1{>}, q_2{<}1{>}, v_2{<}1{>}]}^{z'_0} \xrightarrow{b_3.\mathrm{A1}} \overbrace{[p_2{<}1{>}, q_2{<}1{>}, v_2{<}\mathsf{F}{>}]}^{z'_1} \xrightarrow{b_1.\mathrm{A3}} \overbrace{[p_2{<}p_1{>}, q_2{<}1{>}, v_2{<}\mathsf{F}{>}]}^{z'_2}$$
$$\xrightarrow{b_1.\mathrm{A4}} \overbrace{[p_1{<}1{>}, q_2{<}1{>}, v_2{<}\mathsf{F}{>}]}^{z'_3} \xrightarrow{b_3.\mathrm{A5}} \overbrace{[p_1{<}1{>}, q_2{<}1{>}, v_1{<}1{>}]}^{z'_4} \xrightarrow{b_1.\mathrm{A3}}$$
$$\overbrace{[p_1{<}p_2{>}, q_2{<}1{>}, v_1{<}1{>}]}^{z'_5} \xrightarrow{b_1.\mathrm{A4}} \overbrace{[p_2{<}1{>}, q_2{<}1{>}, v_1{<}1{>}]}^{z'_6}$$

Es gilt: $z'_0z'_1z'_2z'_3z'_4z'_5z'_6 \in \mathcal{V}^{\mathrm{i}}[\![IS_2]\!]$. Zu beachten ist die Gleichheit der globalen Aktivitätszustände z'_3 und z_5.

Behauptung: $z'_0z'_1z'_2z'_3z_6z_7z_8 \notin \mathcal{V}^{\mathrm{i}}[\![IS_2]\!]$.

Beweis durch Widerspruch.

Annahme: $z'_0z'_1z'_2z'_3z_6z_7z_8 \in \mathcal{V}^{\mathrm{i}}[\![IS_2]\!]$.

Nach Definition des Verhaltens gilt dann (beachte $\mathcal{V}^{\mathrm{i}}[\![IS_2]\!] \subseteq \mathcal{V}[\![IS_2]\!]$): Es existieren eine Ausführung Π_3 von $V_I System(IS_2)$ und Globalzeitpunkte t_a, t_b der Ausführung mit $t_a < t_b$, $z^{\Pi_3,t_a}\langle V_I System(IS_2)\rangle = z'_0$ und $z^{\Pi_3,t_b}\langle V_I System(IS_2)\rangle = z_6$, woraus für die lokale Variablenbelegung der Komponente V_{b_1} $z^{\Pi_3,t_a}\langle V_{b_1}\rangle(p_1) = 1$ sowie $z^{\Pi_3,t_b}\langle V_{b_1}\rangle(p_1) = \mathsf{F}$ folgt.

Es existiert ein Zeitpunkt $t^1 \in [t_a, t_b]$, zu dem ein Wechsel der Phasenqualität von p_1 von p_2 ($b_1 \setminus \{p_1\} = \{p_2\}$) oder 1 nach F bei V_{b_1} stattfindet. Ein Wechsel von $\mathsf{0}$ nach F kann bei den gegebenen Aktionen A1-A13 nie auftreten.

Ein Wechsel der Belegung von $\_z(p_1)$ von p_2 oder 1 nach F erfordert bei der vorhandenen Struktur von IS_2, es gilt $E^{-1}(p_1) = \emptyset$, ein Verhalten nach A13.iii oder A13.iv. Folglich existiert ein Globalzeitpunkt $t^2 \in [t_a, t^1[$ mit $s^{\Pi_3,t^2}\langle V_{b_1}\rangle(q_1) = \mathit{true}$.

$\_s(q_1)$ wird bei V_{b_1} auf *true* gesetzt als Reaktion (A11) auf eine *solicit*-Nachricht (A6) von V_{b_2}.

Der Aufruf von A6 erfolgt durch die Ausführung von A4 oder A5 und setzt gemäß der Aktionsbeschreibungen und unter Beachtung der Struktur von IS_2 einen Globalzeitpunkt $t^3 \in [t_a, t^2[$ voraus, zu dem $z^{\Pi_3,t^3}\langle V_{b_2}\rangle(q_2) = q_1$ (beachte $b_2 \setminus \{q_2\} = \{q_1\}$) oder $z^{\Pi_3,t^3}\langle V_{b_2}\rangle(q_2) = \mathsf{F}$ gilt. Für die z-Globalbelegung zum Zeitpunkt t^3 ergibt sich somit $z^{\Pi_3,t^3}\langle V_I System(IS_2)\rangle(q_2) \in \{q_1, \mathsf{F}\}$.

Die Übertragung auf das Verhalten liefert: $\exists z \in \{z'_0, z'_1, z'_2, z'_3\} : z(q_2) \in \{q_1, \mathsf{F}\}$. Das ist ein Widerspruch zu den gegebenen globalen Aktivitätszuständen, und damit ist die Annahme falsch.

Mit den Festlegungen $ztr_1 := z_0z_1z_2z_3z_4$, $ztr_2 := z_6z_7z_8$, $ztr'_1 := z'_0z'_1z'_2$, $ztr'_2 := z'_4z'_5z'_6z'_7z'_8$, $z := z'_3$ (und damit $z = z_6$) gilt Satz 6.5. □

Satz 6.5 hat entscheidende Auswirkungen auf die Ausdruckskraft des Verhaltensgraphen $VG(IS)$ im Vergleich zum Verhalten $\mathcal{V}[\![IS]\!]$ eines I–Systems IS. Es kann vorkommen, dass bestimmte Folgen von globalen Aktivitätszuständen, die durch Pfade im Verhaltensgraphen gegeben sind und mit einem stabilen globalen Aktivitätszustand beginnen, nicht als Traces im Verhalten existieren. Das zukünftige Geschehen ab einem bestimmten globalen Aktivitätszustand kann abhängig sein von dem dem Aktivitätszustand vorhergegangenen Geschehen. Diese Abhängigkeiten kommen im Verhaltensgraphen nicht zum Ausdruck, da jeder globale Aktivitätszustand nur als *genau ein* Knoten vorkommt. Die möglichen Ausführungen des zugrunde liegenden $\mathrm{V_I}$Systems werden dann nicht präzise dargestellt. Folglich ist es nicht immer möglich, das Verhalten $\mathcal{V}[\![IS]\!]$ eines I–Systems IS exakt zu bestimmen, wenn nur der Verhaltensgraph $VG(IS)$ bekannt ist. Die Charakterisierung der Menge von I–Systemen, bei denen diese Bestimmung möglich ist, d.h. bei denen $\mathcal{V}[\![IS]\!]$ aus $VG(IS)$ berechnet werden kann, ist eine Aufgabe für weiterführende Arbeiten (siehe Kapitel 12.2.2).
Ergänzend sei zu bemerken, dass Folgen von globalen Aktivitätszuständen, die *nicht* als Pfade im Verhaltensgraphen vorkommen, auch *nicht* als Traces im Verhalten existieren. Verhaltensgraphen eignen sich somit zur Bestimmung ausgeschlossener Aktivitäten beim modellierten System, ungeachtet davon, ob die Ausführungen des zugrunde liegenden $\mathrm{V_I}$Systems korrekt wiedergegeben werden.

Eine Hauptursache für die Ungleichheit der Ausdruckskraft von Verhalten und Verhaltensgraph eines I–Systems liegt darin, dass der Ausführungsverlauf des zugeordneten $\mathrm{V_I}$Systems nicht ausschließlich an die z-Globalbelegungen gebunden ist, sondern zusätzlich laufende Nachrichten, die bereits gesendet, aber noch nicht bearbeitet worden sind, mit betrachtet werden müssen. Das Beispiel aus dem Beweis zu Satz 6.5 verdeutlicht dieses. Bei der ersten Ausführung Π_1 zum Zeitpunkt zu z_5 ist noch eine *solicit*-Nachricht auf dem Weg von V_{b_2} nach V_{b_1}. Bei der zweiten Ausführung Π_2 existiert diese Nachricht zum Zeitpunkt zu z'_3 nicht (Erinnerung: $z_5 = z'_3$).
Möchte man das Ausführungsverhalten von den noch laufenden Nachrichten entkoppeln, um damit den globalen Aktivitätszustand als elementare Einheit, aus der die weitere Aktivität im I–System eindeutig abgelesen werden kann, ohne die Vergangenheit berücksichtigen zu müssen, zu etablieren, ist eine Anpassung der der Dynamik eines I–Systems zugrunde liegenden Aktionen des zugeordneten $\mathrm{V_I}$Systems (Kapitel 3.2.2) notwendig. Notwendige Maßnahmen liegen in der Nachrichtenflusskontrolle durch zusätzliche Acknowledgements und der Anpassung der Initialisierung. In jedem Fall müssen die Aktionen um Kontrollstrukturen und Kommunikationsanweisungen erweitert werden.
In dieser Arbeit wird Wert gelegt auf die Handhabbarkeit der Aktionen des $\mathrm{V_I}$Systems. Sie sollen möglichst einfach gehalten sein, um die Akzeptanz der Modellierungsmethode zu stärken. Aus diesem Grund werden alternative Ansätze, die auf eine allgemein gültige Gleichheit der Ausdruckskraft von Verhaltensgraph und Verhalten abzielen, dabei aber zu einer Erweiterung des Aktionensystems führen, hier nicht weiter verfolgt, zumal die erforderlichen zusätzlichen Acknowledgements zwischen Komponenten und die mit ihnen verbundenen Kontroll- und Synchronisationsmechanismen den Grad der Autonomie der einzelnen Komponenten (vgl. [7]) zusätzlich herabsetzen. Dies sollte bei einem verteilten System (d.h. bei verteilter Kontrolle) vermieden werden.

Wegen der im Allgemeinen großen Darstellungskomplexität sind Verhaltensgraphen nur anschaulich handhabbar, wenn die Sicht auf Teilbereiche beschränkt wird. Dazu bedient man sich an Projektionen, wie sie in Kapitel 9.6 eingeführt werden, oder wählt den Übergang zu Casegraphen, die im folgenden Abschnitt behandelt werden.

6.2 Casegraph

Beim *Casegraphen* eines I–Systems wird die Knotenmenge durch die Cases des I–Systems gebildet. Im Gegensatz zur Definition des Verhaltensgraphen ist die Klassifizierung von „relevanten“ Cases überflüssig, da *jeder* Case ein potentieller Start-Case und damit als relevant einzuordnen ist. Die Kantenmenge des Casegraphens ergibt sich aus der Interleaving Casetrace-Semantik des I–Systems. Finden sich zwei Knoten nacheinander als Teiltrace in einem Element der Interleaving Casetrace-Semantik wieder, so existiert eine entsprechende Kante im Graphen. Die Richtung orientiert sich an der Reihenfolge der Knoten in der Teiltrace. Zusammenfassend:

Definition 6.6 (Casegraph eines I–Systems). Für ein I–System IS ist der *Casegraph* $CG(IS) := (C, \rightarrow)$ festgelegt durch:

(1) $C = Case(IS)$

(2) $\rightarrow \subseteq C \times C$ mit:

$(c_1, c_2) \in \rightarrow$ gdw. $\exists ctr_1, ctr_2 \in Case(IS)^* : ctr_1.c_1.c_2.ctr_2 \in \mathcal{CT}^{\mathrm{i}}[\![IS]\!]$ □

Da die Menge $Case(IS)$ immer endlich ist und laut obiger Definition die Knotenmenge des Casegraphens $CG(IS)$ bildet, ist dieser immer endlich.

Beispiel 6.7. Der Casegraph $CG(IS_1) = (C, \rightarrow)$ zu dem I–System IS_1 aus Beispiel 2.2 ist in Abbildung 6.3 graphisch dargestellt.

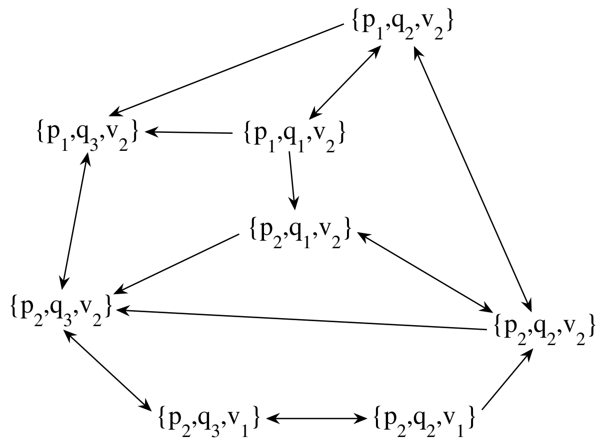


Abbildung 6.3: Casegraph $CG(IS_1)$

Die Cases von IS_1 bilden die Knotenmenge C, die Pfeile geben die Richtung der Kanten an: Es existiert ein Pfeil von Knoten c_1 nach Knoten c_2 gdw. $(c_1, c_2) \in \rightarrow$, es existiert ein Doppelpfeil von c_1 nach c_2 gdw. $(c_1, c_2) \in \rightarrow \wedge (c_2, c_1) \in \rightarrow$. Der Casegraph von IS_1 besitzt 8 Knoten und 18 Kanten. □

Das Beispiel zeigt, dass der Casegraph eines I–Systems wesentlich kleiner und damit handhabbarer als der Verhaltensgraph sein kann. Diese Reduzierung der Darstellungskomplexität ergibt sich aus der Abstraktion von den Phasenqualitäten. Natürlich reduziert dieses in gleicher Weise die Aussagekraft und das Analysepotential des Casegraphens gegenüber dem Verhaltensgraphen, wie es schon beim Übergang von Verhalten zur Casetrace-Semantik (siehe Kapitel 4.3) vermerkt wurde. Es bleibt dem Anwender überlassen, die für seinen Zweck angemessene Darstellung zu wählen.

Kanten im Casegraphen $CG(IS)$ eines I–Systems IS repräsentieren bestimmte Ereignisse im I–System. Da sich die Kanten direkt aus den zweielementigen Teiltraces der Interleaving Casetrace-Semantik $\mathcal{CT}^{\mathrm{i}}[\![IS]\!]$ ergeben und die Interleaving Casetrace-Semantik eine Teilmenge der

Casetrace-Semantik $\mathcal{CT}[\![IS]\!]$ ist, ist es der Einheitlichkeit wegen sinnvoll, Interpretationen, wie sie für die Casetrace-Semantik definiert worden sind (siehe Definition 4.11), für den Casegraphen zu übernehmen.

Definition 6.8 (Phasentransition im Casegraphen). Sei IS ein I–System, b ein Bereich von IS und $p, q \in b$. Sei $CG(IS) = (C, \rightarrow)$ der Casegraph von IS. Eine Kante $(c_1, c_2) \in \rightarrow$ repräsentiert folgendes Ereignis: Bei $p \in \{c_1 \setminus c_2\}$ und $q \in \{c_2 \setminus c_1\}$ sagen wir, dass eine *Phasentransition* von p nach q in b auftritt. □

Pfade im Casegraphen eines I–Systems beschreiben, als Aneinanderreihung von Phasentransitionen, eine durchgängige Aktivität des I–Systems. Entsprechend den Traces bei den Trace-Semantiken können sie als Mittel zur Systemanalyse und Systemverifikation eingesetzt werden.

Es stellt sich nun die Frage nach der Ausdruckskraft des Casegraphens $CG(IS)$ im Vergleich zur Ausdruckskraft der Casetrace-Semantik $\mathcal{CT}[\![IS]\!]$ eines I–Systems IS. Definition 6.6 beschreibt die Berechnung des Casegraphen aus der Interleaving Casetrace-Semantik, welche (als Folgerung aus Definition 5.13 und Satz 5.15) von der Ausdruckskraft her gleichmächtig zur Casetrace-Semantik ist. Somit kann der Casegraph höchstens gleichmächtig zur Casetrace-Semantik sein. Sollte allgemein Gleichheit herrschen, muss für jedes beliebige I–System dessen Interleaving Casetrace-Semantik (und damit auch die Casetrace-Semantik) aus dem Casegraphen zurückgewonnen werden können. Nach den Ergebnissen aus Abschnitt 6.1 ist nun zu klären, ob durch die Vernachlässigung von Phasenqualitäten Fälle, die eine Ungleichheit des Informationsgehaltes von Verhalten und Verhaltensgraph beinhalten (wie z.B. im Beweis von Satz 6.5), auf Caseebene ausgeschlossen sind. Aus dem folgenden Satz folgt, dass auch die Interleaving Casetrace-Semantik besondere Eigenschaften besitzt, die beim Casegraphen nicht vorliegen können.

Satz 6.9 (Besonderheit der Interleaving Casetrace-Semantik). Es existieren I–Systeme IS, für die gilt: $\exists ctr_1, ctr_2, ctr'_1, ctr'_2 \in Case(IS)^*$, $c \in Case(IS)$ mit $ctr_1.c.ctr_2 \in \mathcal{CT}^{\mathrm{i}}[\![IS]\!]$, $ctr'_1.c.ctr'_2 \in \mathcal{CT}^{\mathrm{i}}[\![IS]\!]$ und $ctr'_1.c.ctr_2 \notin \mathcal{CT}^{\mathrm{i}}[\![IS]\!]$.

Beweis.

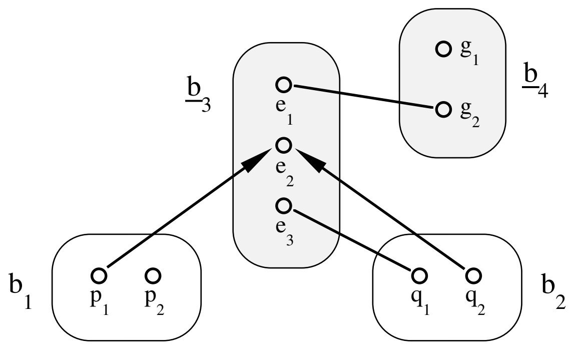


Abbildung 6.4: IS_3

Sei IS_3 als das I–System aus Abbildung 6.4 gegeben. Betrachte die folgende Ausführung Π_1 von $V_I System(IS_3)$:

$$\overbrace{[p_2{<}1{>}, q_1{<}1{>}, e_2{<}1{>}, g_2{<}1{>}]}^{z_0} \xrightarrow{b_1.\mathrm{A3}} \overbrace{[p_2{<}p_1{>}, q_1{<}1{>}, e_2{<}1{>}, g_2{<}1{>}]}^{z_1} \xrightarrow{b_1.\mathrm{A4}}$$

$$\overbrace{[p_1{<}1{>}, q_1{<}1{>}, e_2{<}1{>}, g_2{<}1{>}]}^{z_2} \xrightarrow{\underline{b}_3.\mathrm{A10}} \overbrace{[p_1{<}1{>}, q_1{<}1{>}, e_2{<}\mathsf{F}{>}, g_2{<}1{>}]}^{z_3} \xrightarrow{b_2.\mathrm{A11}, \underline{b}_4.\mathrm{A11}}$$

$$\overbrace{[p_1{<}1{>}, q_1{<}\mathsf{F}{>}, e_2{<}\mathsf{F}{>}, g_2{<}\mathsf{F}{>}]}^{z_4} \xrightarrow{b_2.\mathrm{A5}} \overbrace{[p_1{<}1{>}, q_2{<}1{>}, e_2{<}\mathsf{F}{>}, g_2{<}\mathsf{F}{>}]}^{z_5} \xrightarrow{\underline{b}_4.\mathrm{A5}}$$

$$\overbrace{[p_1{<}1{>}, q_2{<}1{>}, e_2{<}\mathsf{F}{>}, g_1{<}1{>}]}^{z_6} \xrightarrow{\underline{b}_3.\mathrm{A5}} \overbrace{[p_1{<}1{>}, q_2{<}1{>}, e_1{<}1{>}, g_1{<}1{>}]}^{z_7}$$

Die den einzelnen Aktivitätszuständen zugeordneten Cases sind:
$zc(z_0) = zc(z_1) = \{p_2, q_1, e_2, g_2\} =: c_0$,
$zc(z_2) = zc(z_3) = zc(z_4) = \{p_1, q_1, e_2, g_2\} =: c_1$,
$zc(z_5) = \{p_1, q_2, e_2, g_2\} =: c_2$,
$zc(z_6) = \{p_1, q_2, e_2, g_1\} =: c_3$,
$zc(z_7) = \{p_1, q_2, e_1, g_1\} =: c_4$.

Somit gilt $c_0c_1c_2c_3c_4 \in \mathcal{CT}[\![IS_3]\!]$ und wegen $|c_i \backslash c_{i-1}| = 1$ für $i = 1, 2, 3$ auch $c_0c_1c_2c_3c_4 \in \mathcal{CT}^{\mathrm{i}}[\![IS_3]\!]$.

Als weitere Ausführung Π_2 von $V_I System(IS_3)$ existiert:

$$\overbrace{[p_2{<}1{>}, q_1{<}1{>}, e_2{<}1{>}, g_2{<}1{>}]}^{z'_0} \xrightarrow{b_2.\mathrm{A3}} \overbrace{[p_2{<}1{>}, q_1{<}q_2{>}, e_2{<}1{>}, g_2{<}1{>}]}^{z'_1} \xrightarrow{b_2.\mathrm{A4}}$$
$$\overbrace{[p_2{<}1{>}, q_2{<}1{>}, e_2{<}1{>}, g_2{<}1{>}]}^{z'_2} \xrightarrow{b_1.\mathrm{A3}, \underline{b}_3.\mathrm{A10}} \overbrace{[p_2{<}p_1{>}, q_2{<}1{>}, e_2{<}\mathsf{F}{>}, g_2{<}1{>}]}^{z'_3} \xrightarrow{b_1.\mathrm{A4}}$$
$$\overbrace{[p_1{<}1{>}, q_2{<}1{>}, e_2{<}\mathsf{F}{>}, g_2{<}1{>}]}^{z'_4} \xrightarrow{\underline{b}_3.\mathrm{A5}} \overbrace{[p_1{<}1{>}, q_2{<}1{>}, e_3{<}1{>}, g_2{<}1{>}]}^{z'_5}$$

Die den Aktivitätszuständen zugeordneten Cases sind:
$zc(z'_0) = zc(z'_1) = \{p_2, q_1, e_2, g_2\} =: c'_0$,
$zc(z'_2) = zc(z'_3) = \{p_2, q_2, e_2, g_2\} =: c'_1$,
$zc(z'_4) = \{p_1, q_2, e_2, g_2\} =: c'_2$,
$zc(z'_5) = \{p_1, q_2, e_3, g_2\} =: c'_3$.

Es gilt: $c'_0c'_1c'_2c'_3 \in \mathcal{CT}^{\mathrm{i}}[\![IS_3]\!]$. Zu beachten ist die Gleichheit der Cases c_2 und c'_2.

Behauptung: $c'_0c'_1c'_2c_3c_4 \notin \mathcal{CT}^{\mathrm{i}}[\![IS_3]\!]$.

Beweis durch Widerspruch.

Annahme: $c'_0c'_1c'_2c_3c_4 \in \mathcal{CT}^{\mathrm{i}}[\![IS_3]\!]$.

Aus der Definition der Interleaving Casetrace-Semantik folgt dann: $\exists ztr \in \mathcal{V}[\![IS_3]\!]$ mit $\lfloor ztr \rfloor = c'_0c'_1c'_2c_3c_4$.
Nach Definition des Verhaltens gilt: Es existieren eine Ausführung Π_3 von $V_I System(IS_3)$ und Globalzeitpunkte t_a, t_b der Ausführung mit $t_a < t_b$, $z^{\Pi_3, t_a}\langle V_I System(IS_3)\rangle = first(ztr)$ und $z^{\Pi_3, t_b}\langle V_I System(IS_3)\rangle = last(ztr)$.

Aus $g_2 \in c'_0$ und $g_1 \in c_4$ folgt: $z^{\Pi_3, t_a}\langle V_{\underline{b}_4}\rangle(g_1) = \mathsf{0}$, $z^{\Pi_3, t_a}\langle V_{\underline{b}_4}\rangle(g_2) = \mathsf{1}$, $z^{\Pi_3, t_b}\langle V_{\underline{b}_4}\rangle(g_1) \neq \mathsf{0}$, $z^{\Pi_3, t_b}\langle V_{\underline{b}_4}\rangle(g_2) = \mathsf{0}$.
Da $\underline{b}_4$ aus zwei Phasen besteht, existiert ein Zeitpunkt $t^1 \in [t_a, t_b]$, zu dem eine Phasentransition $g_2 \rightarrow g_1$ bei $V_{\underline{b}_4}$ stattfindet.

Da $\underline{b}_4$ ein träger Bereich ist, kann die Phasentransition nur durch Aktion A5 (Phasentransition wegen Erregung) bewirkt werden. (Aktion A4 erforderte eine vorherige Ausführung von A3 mit der Voraussetzung der Nicht-Trägheit des Bereiches.) Gemäß der Aktionsbeschreibung von A5 muss ein Globalzeitpunkt $t^2 \in [t_a, t_b]$ existieren, für den $z^{\Pi_3, t^2}\langle V_{\underline{b}_4}\rangle(g_2) = \mathsf{F}$ gilt.

Ein Wechsel der Belegung von $\_z(g_2)$ von $\mathsf{1}$ nach F erfordert bei der vorhandenen Struktur von IS_3, es gilt $E^{-1}(g_2) = \emptyset$, ein Verhalten nach A13.iv. Folglich existiert ein Globalzeitpunkt $t^3 \in [t_a, t^2]$ mit $s^{\Pi_3, t^3}\langle V_{\underline{b}_4}\rangle(e_1) = true$.

$\_s(e_1)$ wird bei $V_{\underline{b}_4}$ auf *true* gesetzt als Reaktion (A11) auf eine *solicit*-Nachricht (A6) von $V_{\underline{b}_3}$. Da $\underline{b}_3$ ein träger Bereich ist, erfolgt der Aufruf von A6 durch die Ausführung von A5 und setzt gemäß der Aktionsbeschreibung und unter Beachtung der Struktur von IS_3 einen Globalzeitpunkt $t^4 \in [t_a, t^3]$ mit $z^{\Pi_3, t^4}\langle V_{\underline{b}_3}\rangle(e_2) = \mathsf{F}$ und $k^{\Pi_3, t^4}\langle V_{\underline{b}_3}\rangle(e_3) = true$ voraus.

Wegen $z^{\Pi_3, t_a}\langle V_{b_2}\rangle(q_1) = \mathsf{1}$ zum Ausführungsbeginn gilt: $z^{\Pi_3, t}\langle V_{b_2}\rangle(q_1) \neq \mathsf{0}$ für alle $t \in [t_a, t^4]$. $(*)$
Die andere Möglichkeit $z^{\Pi_3, t}\langle V_{b_2}\rangle(q_2) = q_1$ ist aufgrund der Aktionsbeschreibung von A3 nicht möglich.

Wegen $(*)$ muss ein Wechsel der Belegung von $\_z(e_2)$ von 1 nach F bei $V_{\underline{b}_3}$ erfolgen mittels Aktion A13, als Reaktion auf die Nachricht $done(p_2 \rightarrow p_1)$ von V_{b_1} (A10). Eine (weitere) Phasentransition $p_1 \rightarrow p_2$ ist nicht möglich, solange $\_z(e_2) \neq 0$ bei $V_{\underline{b}_3}$ gilt. Es folgt: $\exists t_x \in [t_a, t^4]$ mit $z^{\Pi_3,t_x}\langle V_{b_1}\rangle(p_1) \neq 0$. $(**)$

Aus $(*)$ und $(**)$ folgt für die z-Globalbelegung: $\exists t_x \in [t_a, t^4]$ mit $z^{\Pi_3,t_x}\langle V_I System(IS_3)\rangle(q_1) \neq 0$ und $z^{\Pi_3,t_x}\langle V_I System(IS_3)\rangle(p_1) \neq 0$.

Die Übertragung auf die Interleaving Casetrace-Semantik liefert: $\exists c \in \{c_0', c_1', c_2', c_3, c_4\} : \{p_1, q_1\} \subseteq c$. Das ist ein Widerspruch zu den gegebenen Cases, und damit ist die Annahme falsch.

Mit den Festlegungen $ctr_1 := c_0c_1$, $ctr_2 := c_3c_4$, $ctr_1' := c_0'c_1'$, $ctr_2' := c_3'$, $c := c_2'$ (und damit $c = c_2$) gilt Satz 6.9. $\square$

In Verbindung mit Definition 6.6 folgt aus Satz 6.9, dass es I–Systeme IS gibt, bei denen Casefolgen, die durch Pfade im Casegraphen $CG(IS)$ gegeben sind, nicht als Traces in der Interleaving Casetrace-Semantik $\mathcal{CT}^{\mathrm{i}}[\![IS]\!]$, und damit in der Casetrace-Semantik $\mathcal{CT}[\![IS]\!]$, existieren. Das zukünftige Geschehen ab einem bestimmten Case, das bestimmt wird durch die Ausführungen des zugeordneten $\mathrm{V_I}$Systems $V_I System(IS)$, kann abhängig sein von dem dem Case vorangegangenen Geschehen. Im Casegraphen werden diese Abhängigkeiten nicht erfasst, da jeder Case nur als *genau ein* Knoten vorkommt. Folglich ist die (Interleaving) Casetrace-Semantik $\mathcal{CT}[\![IS]\!]$ ($\mathcal{CT}^{\mathrm{i}}[\![IS]\!]$) nicht in jedem Fall aus dem Casegraphen $CG(IS)$ ableitbar, was bedeutet, dass die Gleichheit im Informationsgehalt von Casetrace-Semantik und Casegraph eines I–Systems nicht vorausgesetzt werden kann.
Ergänzend ist zu bemerken, dass Folgen von Cases, die *nicht* als Pfade im Casegraphen vorkommen, auch *nicht* als Traces in der Casetrace-Semantik existieren. Casegraphen eignen sich somit zur Bestimmung *ausgeschlossener* Systemaktivitäten, ungeachtet davon, ob die Ausführungen des zugeordneten $\mathrm{V_I}$Systems korrekt erfasst werden.

Eine Ursache, die zu der in Satz 6.9 beschriebenen Eigenschaft und damit zur Ungleichheit der Ausdruckskräfte von Casetrace-Semantik und Casegraph eines I–Systems führt, liegt bei den Kontrollstrukturen innerhalb der die Dynamik bestimmenden Aktionen A1-A13 in Verbindung mit der Autonomie einzelner Komponenten des zugeordneten $\mathrm{V_I}$Systems, sowie der zeitlichen Varianz bei den einzelnen Aktionsausführungen. Beim I–System IS_3 in dem Beweis zu Satz 6.9 sind die Auswirkungen der Erregung von e_2 (d.h. $\_e_{in}(e_2) = true$) bei $V_{\underline{b}_3}$ abhängig vom Verhalten von V_{b_2}. Findet dort die Phasentransition $q_1 \rightarrow q_2$ „rechtzeitig“ statt, erfolgt kein Solicitation-Aufruf (Aktion A5, Fall *not* a) und *not* b) und *not* c)), im Rahmen dessen eine *solicit*-Nachricht $V_{\underline{b}_4}$ erreicht. Es unterliegt der Autonomie von V_{b_2}, diese Phasentransition durchzuführen und $V_{\underline{b}_3}$ hat keine Informationen darüber, ob und wann das geschieht.
Möchte man die besondere Eigenschaft der (Interleaving) Casetrace-Semantik eines I–Systems aufheben, d.h. die Möglichkeit schaffen, ausgehend von einem Case an einer beliebigen Stelle einer beliebigen Trace der Casetrace-Semantik die weitere Ausführung (gemäß Definition 4.8) des zugeordneten $\mathrm{V_I}$Systems angeben zu können, ohne Vorgängercases berücksichtigen zu müssen, sind zusätzliche Kommunikationsanweisungen notwendig, über die Komponenten des $\mathrm{V_I}$Systems genaue Auskünfte über momentanes und geplantes Verhalten der Nachbarkomponenten erhalten. Die Kontrollstrukturen innerhalb der Aktionen benötigen eine entsprechende Anpassung. Wenn, im obigen Szenario, $V_{\underline{b}_3}$ weiß, dass V_{b_2} die Phasentransition $q_1 \rightarrow q_2$ durchführen wird, könnte auf den Solicitation-Aufruf verzichtet werden. Die einzelnen Komponenten sind dann an ihre Auskünfte gebunden.
Da in dieser Arbeit Wert gelegt wird auf möglichst einfache, kurz gehaltene Aktionsbeschreibungen, die den lokalen Aktivitäten der Komponenten des $\mathrm{V_I}$Systems möglichst wenig Restriktionen auferlegen, werden Ansätze, die auf eine allgemein gültige Gleichheit der Ausdrucksstärke von Casegraph und Casetrace-Semantik abzielen, dabei aber zu einer Erweiterung des Aktionensystems führen, hier nicht weiter verfolgt. In Kapitel 8.3 wird hingegen eine spezielle Klasse von I–Systemen angegeben, für die die Gleichheit der Ausdrucksstärke im bestehenden Modell gilt.

Der Casegraph bietet sich, wie schon die Casetrace-Semantik, an, wenn insbesondere Sicherheitseigenschaften (z.B. wechselseitiger Ausschluss) und keine Fortschrittseigenschaften im Mittelpunkt

der Systemanalyse stehen. Er verkörpert dabei naturgemäß die anschaulichere Repräsentationsform und ist gut zur Beobachtung von Phasentransitionen und deren kausalen Abhängigkeiten, erlaubt aber keine Aussagen über die Garantie des Eintreffens bestimmter Ereignisse. Für einen Knoten, der mindestens einen Nachfolgeknoten besitzt, kann anhand des Graphen nicht festgestellt werden, ob er ausschließlich, nie oder nur in bestimmten Fällen Endpunkt eines Pfades ist, der eine mögliche Ausführung des zugrunde liegenden V_ISystems beschreibt. Ereignisse (Phasentransitionen), die garantiert eintreten werden, können nicht ermittelt werden. Hierzu sind Zusatzinformationen notwendig, wie sie der Erweiterte Casegraph, der im nächsten Abschnitt vorgestellt wird, bietet.
Obwohl der Casegraph eines I–Systems in der Darstellung wesentlich kleiner ist als der Verhaltensgraph, kann auch er für reale Anwendungen Dimensionen annehmen, die anschaulich nicht mehr handhabbar sind. Um dann die Sicht auf Teilbereiche zu beschränken, bedient man sich der Projektionen, die in Kapitel 9.6 behandelt werden.

6.3 Erweiterter Casegraph

Wie beim Casegraphen entspricht auch beim *Erweiterten Casegraphen* eines I–Systems die Knotenmenge der Menge der Cases. Neu ist, dass zwei Mengen von Kanten existieren zur Klassifizierung der Übergänge. Die Kantenmengen ergeben sich aus der Erweiterten Interleaving Casetrace-Semantik des I–Systems. Finden sich zwei Knoten nacheinander und nur getrennt durch eine Bereichsmenge als Teiltrace in einem Element der Erweiterten Interleaving Casetrace-Semantik wieder, so existiert eine entsprechende Kante im Graphen. Die Richtung orientiert sich an der Reihenfolge der Knoten in der Teiltrace. Die Zugehörigkeit zu einer der beiden Kantenmengen wird durch die Bereichsmenge festgelegt. Ist die Bereichsmenge nicht leer, dann gehört die Kante zur ersten Kantenmenge. In diesem Fall repräsentiert die Teiltrace der Erweiterten Interleaving Casetrace-Semantik eine freie Phasentransition. Ist die Bereichsmenge leer, dann gehört die Kante zur zweiten Kantenmenge. In diesem Fall repräsentiert die Teiltrace der Erweiterten Interleaving Casetrace-Semantik eine erzwungene Phasentransition. Beide Kantenmengen sind nicht notwendigerweise disjunkt. Es kann vorkommen, dass der gleiche Caseübergang einerseits als freie und andererseits als erzwungene Phasentransition in unterschiedlichen Teiltraces der Erweiterten Interleaving Casetrace-Semantik vorkommt. In diesem Fall existieren zwischen den korrespondierenden Knoten im Graphen zwei Kanten, eine aus jeder Kantenmenge. Zusammenfassend:

Definition 6.10 (Erweiterter Casegraph eines I–Systems). Für ein I–System IS mit Bereichsmenge B ist der *Erweiterte Casegraph* $ECG(IS) := (C, \rightarrow_1, \rightarrow_2)$ festgelegt durch:

(1) $C = Case(IS)$

(2) $\rightarrow_1 \subseteq C \times C$ mit:

$(c_1, c_2) \in \rightarrow_1$ gdw. $\exists ectr_1 \in (Case(IS).\mathcal{P}(B))^*, ectr_2 \in (\mathcal{P}(B).Case(IS))^*, \delta \subseteq B :$

$ectr_1.c_1.\delta.c_2.ctr_2 \in \mathcal{ECT}^{\mathrm{i}}[\![IS]\!]$ und $\delta \neq \emptyset$

(3) $\rightarrow_2 \subseteq C \times C$ mit:

$(c_1, c_2) \in \rightarrow_2$ gdw. $\exists ectr_1 \in (Case(IS).\mathcal{P}(B))^*, ectr_2 \in (\mathcal{P}(B).Case(IS))^*, \delta \subseteq B :$

$ectr_1.c_1.\delta.c_2.ctr_2 \in \mathcal{ECT}^{\mathrm{i}}[\![IS]\!]$ und $\delta = \emptyset$ □

Die vorgestellte Definition des Erweiterten Casegraphen macht Sinn durch den Rückgriff auf die Erweiterte *Interleaving* Casetrace-Semantik. Da $ectr_1.c_1.\delta.c_2.ctr_2$ ein Element der Erweiterten Interleaving Casetrace-Semantik ist, ist δ maximal einelementig und ergibt sich eindeutig aus den umgebenden Cases, sofern nur die Ursache des Übergangs bekannt ist (vgl. Bemerkung 5.11). Jedes δ klassifiziert somit genau eine Phasentransition und damit eine Kante. Es ist denkbar, einen dritten Kantentyp einzuführen, der die Fälle der doppelten Kanten aufnimmt. Bei der obigen Definition des Erweiterten Casegraphen wird allerdings darauf verzichtet, um auch anschaulich eine mögliche Alternative zu unterstreichen.

Der Erweiterte Casegraph ist, bei gleicher Knotenmenge wie der Casegraph, ebenfalls immer endlich.

Beispiel 6.11. Der Erweiterte Casegraph $ECG(IS_1) = (C, \rightarrow_1, \rightarrow_2)$ zu dem I–System IS_1 aus Beispiel 2.2 ist in Abbildung 6.5 graphisch dargestellt.

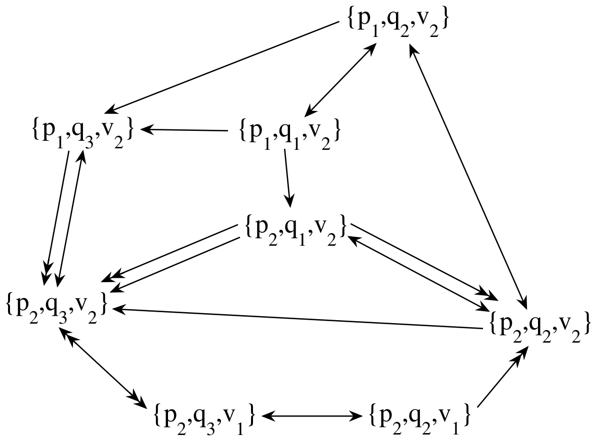


Abbildung 6.5: Erweiterter Casegraph $ECG(IS_1)$

Die Cases von IS_1 bilden die Knotenmenge C, die Pfeile geben die Richtung der Kanten an und klassifizieren je nach Pfeiltyp die Übergänge: Es existiert ein einspitziger Pfeil von Knoten c_1 nach Knoten c_2 gdw. $(c_1, c_2) \in\rightarrow_1$; es existiert ein einspitziger Doppelpfeil von c_1 nach c_2 gdw. $(c_1, c_2) \in\rightarrow_1 \wedge (c_2, c_1) \in\rightarrow_1$. Dieser einspitzige Pfeiltyp repräsentiert freie Phasentransitionen. Es existiert ein zweispitziger Pfeil von c_1 nach c_2 gdw. $(c_1, c_2) \in\rightarrow_2$; es existiert ein zweispitziger Doppelpfeil von c_1 nach c_2 gdw. $(c_1, c_2) \in\rightarrow_2 \wedge (c_2, c_1) \in\rightarrow_2$. Dieser zweispitzige Pfeiltyp repräsentiert erzwungene Phasentransitionen. Zwischen zwei Knoten können auch beide Kantentypen vertreten sein (z.B von $\{p_2, q_1, v_2\}$ nach $\{p_2, q_3, v_2\}$), d.h. die Phasentransition kann sowohl frei als auch erzwungen auftreten, je nach Systemverhalten. Der Erweiterte Casegraph von IS_1 besitzt 8 Knoten und insgesamt 21 Kanten. □

Der Erweiterte Casegraph besitzt bei gleicher Knotenanzahl in der Regel mehr Kanten (maximal doppelt so viele) als der Casegraph. Der Grund liegt in der Nicht-Disjunktheit der beiden Kantenmengen $\rightarrow_1$ und $\rightarrow_2$ in Definition 6.10. So besitzt der Erweiterte Casegraph $ECG(IS_1)$ in Abbildung 6.5 21 Kanten gegenüber 18 Kanten beim „normalen" Casegraph $CG(IS_1)$ aus Abbildung 6.3.

Wie schon erwähnt wurde, erlaubt die Einführung von zwei Kantenmengen eine Klassifizierung der Ereignisse, die durch die Kanten in Verbindung mit den Quell- und Zielknoten repräsentiert werden. Dieses erhöht die Aussagekraft und das Analysepotential des Erweiterten Casegraphens gegenüber dem Casegraphen in gleicher Weise, wie es schon beim Übergang von der Casetrace-Semantik zur Erweiterten Casetrace-Semantik (siehe Kapitel 4.4) diskutiert wurde. Da sich die Kanten des Erweiterten Casegraphens $ECG(IS)$ eines I–Systems IS direkt aus dreielementigen Teiltraces der Erweiterten Interleaving Casetrace-Semantik $\mathcal{ECT}^{\mathrm{i}}[\![IS]\!]$ ergeben und die Erweiterte Interleaving Casetrace-Semantik eine Teilmenge der Erweiterten Casetrace-Semantik $\mathcal{ECT}[\![IS]\!]$ ist, werden der Einheitlichkeit wegen Bezeichnungen für die Ereignisse, wie sie für die Erweiterte Casetrace-Semantik definiert worden sind (siehe Definition 4.18), übernommen.

Definition 6.12 (Freie/Erzwungene Phasentransition im Erweiterten Casegraphen). Sei IS ein I–System, b ein Bereich von IS und $p, q \in b$. Sei $ECG(IS) = (C, \rightarrow_1, \rightarrow_2)$ der Erweiterte Casegraph von IS. Kanten $(c_1, c_2) \in\rightarrow_1$ und $(c_1', c_2') \in\rightarrow_2$ repräsentieren folgende Ereignisse:

a) Bei $p \in \{c_1 \setminus c_2\} \wedge q \in \{c_2 \setminus c_1\}$ sagen wir, dass eine *freie Phasentransition* von p nach q in b auftritt.

b) Bei $p \in \{c'_1 \setminus c'_2\} \wedge q \in \{c'_2 \setminus c'_1\}$ sagen wir, dass eine *erzwungene Phasentransition* von p nach q in b auftritt. □

Ein Pfad im Erweiterten Casegraphen eines I–Systems repräsentiert eine Folge von Phasentransitionen, von denen jede einzelne als entweder frei oder erzwungen klassifiziert wird, je nach Kantentyp. Existieren zwischen zwei Knoten zwei Kanten gleicher Ausrichtung (aus unterschiedlichen Kantenmengen) wird beliebig eine Kante zur Bildung des Pfades herangezogen. Die Menge aller Pfade spiegelt die Aktivität des I–Systems wider.

Bei dem Vergleich der Ausdruckskraft des Erweiterten Casegraphens $ECG(IS)$ und der Ausdruckskraft der Erweiterten Casetrace-Semantik $\mathcal{ECT}[\![IS]\!]$ eines I–Systems IS treten die gleichen Unterschiede auf, wie schon bei dem Casegraphen $CG(IS)$ und der Casetrace-Semantik $\mathcal{CT}[\![IS]\!]$. Definition 6.10 liefert die Herleitung des Erweiterten Casegraphen aus der Erweiterten Interleaving Casetrace-Semantik, welche (als Folgerung von Definition 5.7 und Satz 5.9) von der Aussagekraft her als gleichmächtig zur Erweiterten Casetrace-Semantik anzusehen ist. Die Rückrichtung, d.h. die Berechnung der Erweiterten Interleaving Casetrace-Semantik aus dem Erweiterten Casegraphen, ist in bestimmten Fällen allerdings nicht möglich. Die Ursache liegt in der durch den folgenden Satz aufgezeigten Eigenschaft der Erweiterten Interleaving Casetrace-Semantik.

Satz 6.13 (Besonderheit der Erweiterten Interleaving Casetrace-Semantik). Es existieren I–Systeme IS, für die gilt: $\exists ectr_1, ectr'_1 \in (Case(IS).\mathcal{P}(B))^*$, $ectr_2, ectr'_2 \in (\mathcal{P}(B).Case(IS))^*$, $c \in Case(IS)$ mit $ectr_1.c.ectr_2 \in \mathcal{ECT}^{\mathrm{i}}[\![IS]\!]$, $ectr'_1.c.ectr'_2 \in \mathcal{ECT}^{\mathrm{i}}[\![IS]\!]$ und $ectr'_1.c.ectr_2 \notin \mathcal{ECT}^{\mathrm{i}}[\![IS]\!]$.

Beweis. Der Satz ergibt sich direkt aus dem Beweis von Satz 6.9 in Kombination mit Satz 4.20.d. Bezieht man sich auf das im Beweis von Satz 6.9 verwendete I–System IS_3 (Abbildung 6.4), die dortigen Ausführungen und Casebezeichnungen, dann ergibt sich:

$\exists\, \delta_1, \delta_2, \delta_3, \delta_4 \subseteq B : c_0.\delta_1.c_1.\delta_2.c_2.\delta_3.c_3.\delta_4.c_4 \in \mathcal{ECT}^{\mathrm{i}}[\![IS_3]\!]$,
$\exists\, \delta'_1, \delta'_2, \delta'_3 \subseteq B : c'_0.\delta'_1.c'_1.\delta'_2.c'_2.\delta'_3.c'_3 \in \mathcal{ECT}^{\mathrm{i}}[\![IS_3]\!]$,
$\nexists\, \delta''_1, \delta''_2, \delta''_3, \delta''_4 \subseteq B : c'_0.\delta''_1.c'_1.\delta''_2.c'_2.\delta''_3.c_3.\delta''_4.c_4 \in \mathcal{ECT}^{\mathrm{i}}[\![IS_3]\!]$.

Demnach gilt insbesondere (beachte $c_2 = c'_2$): $\exists\, \delta_1, \delta_2, \delta_3, \delta_4, \delta'_1, \delta'_2, \delta'_3 \subseteq B$:

$$\overbrace{c_0.\delta_1.c_1.\delta_2}^{ectr_1}.\overbrace{c_2}^{c}.\overbrace{\delta_3.c_3.\delta_4.c_4}^{ectr_2} \in \mathcal{ECT}^{\mathrm{i}}[\![IS_3]\!],$$
$$\overbrace{c'_0.\delta'_1.c'_1.\delta'_2}^{ectr'_1}.\overbrace{c'_2}^{c}.\overbrace{\delta'_3.c'_3}^{ectr'_2} \in \mathcal{ECT}^{\mathrm{i}}[\![IS_3]\!],$$
$$\overbrace{c'_0.\delta'_1.c'_1.\delta'_2}^{ectr'_1}.\overbrace{c'_2}^{c}.\overbrace{\delta_3.c_3.\delta_4.c_4}^{ectr_2} \notin \mathcal{ECT}^{\mathrm{i}}[\![IS_3]\!].$$

Die Bereichsmengen bieten nur eine Zusatzinformation beim Übergang von der Casetrace- zur Erweiterten Casetrace-Semantik. Die erzeugten Casefolgen bleiben dabei erhalten (vgl. Definition 4.8 und Definition 4.16), und damit auch die Besonderheiten, die sich auf die Casefolgen abstützen. □

Die Bedeutung von Satz 6.13 für die Ausdruckskraft des Erweiterten Casegraphen entspricht der Bedeutung von Satz 6.9 für die Ausdruckskraft des Casegraphens. Es folgt nämlich, dass I–Systeme IS existieren, bei denen Casefolgen, die durch Pfade im Erweiterten Casegraphen $ECG(IS)$ gegeben sind, nicht als Traces mit Bereichsmengen in der Erweiterten Interleaving Casetrace-Semantik $\mathcal{ECT}^{\mathrm{i}}[\![IS]\!]$, und damit nicht in der Erweiterten Casetrace-Semantik $\mathcal{ECT}[\![IS]\!]$, vorkommen. Das zukünftige Geschehen ab einem bestimmten Case, das bestimmt wird durch die Ausführungen des zugeordneten $\mathrm{V_I}$Systems $V_I System(IS)$, kann abhängig sein von dem dem Case vorangegangenen Geschehen. Im Erweiterten Casegraphen werden diese Abhängigkeiten nicht erfasst, da jeder Case nur als *genau ein* Knoten vorkommt. Folglich ist die Erweiterte (Interleaving) Casetrace-Semantik $\mathcal{ECT}[\![IS]\!]$ ($\mathcal{ECT}^{\mathrm{i}}[\![IS]\!]$) nicht in jedem Fall aus dem Erweiterten Casegraphen $ECG(IS)$ ableitbar, was bedeutet, dass die Gleichheit der Ausdrucksstärken von Erweiterter Casetrace-Semantik und Erweitertem Casegraphen eines I–Systems nicht vorausgesetzt werden kann.

Die Ursachen, die zu der in in Satz 6.13 präsentierten Eigenschaft führen, sind die gleichen wie bei Satz 6.9: spezielle Kontrollstrukturen innerhalb der die Dynamik bestimmenden Aktionen A1-A13 in Verbindung mit der Autonomie einzelner Komponenten des zugeordneten V_ISystems, sowie der zeitlichen Varianz bei den einzelnen Aktionsausführungen. Die Aufhebung der Eigenschaft erfordert die in Abschnitt 6.2 beschriebenen zusätzlichen Kommunikationsmechanismen, zu Lasten der Kürze und Einfachheit der Aktionsbeschreibungen sowie des Grades an Autonomie der Komponenten.

Im Gegensatz zum Casegraphen bietet der Erweiterte Casegraph zusätzlich an, neben Sicherheits- auch Fortschrittseigenschaften untersuchen zu können. Durch die Zweiteilung der Kantenmenge kann zwischen Ereignissen (Phasentransitionen) unterschieden werden, die eintreten können, und welchen, die eintreten werden. Der Erweiterte Casegraph eines I–Systems ist somit als Grundlage für eine Systemanalyse oder Systemverifikation eine Alternative zur Erweiterten Casetrace-Semantik, mit dem Vorteil der anschaulicheren Darstellungsform und dem Nachteil, dass Ausführungen des zugeordneten V_ISystems in bestimmten Fällen nicht korrekt erfasst werden, was allerdings durch das formale Modell auch nicht gefordert wird, da das V_ISystem unabhängig ist von dem zu modellierenden Anwendungssystem. In Abschnitt 6.4 wird noch einmal auf diese Beziehungen eingegangen. Ergänzend sei zu bemerken, dass Folgen von Cases, die nicht als Pfade im Erweiterten Casegraphen vorkommen, auch nicht als Traces mit Bereichsmengen in der Erweiterten Casetrace-Semantik existieren. Wie schon Verhaltensgraphen und Casegraphen eignen sich Erweiterte Casegraphen somit zur Bestimmung ausgeschlossener Aktivitäten beim modellierten System, ungeachtet davon, ob die Ausführungen des zugeordneten V_ISystems korrekt erfasst werden.
Analog zum Verhaltens- und Casegraphen kann auch beim Erweiterten Casegraphen eines I–Systems, der von der Darstellungskomplexität zwischen den beiden ersten liegt, die Sicht auf Teilbereiche beschränkt werden, indem Projektionen verwendet werden. Die notwendigen Techniken werden in Kapitel 9.6 vorgestellt.

6.4 Beziehungen

In diesem Abschnitt werden in Form eines Satzes die Übergänge zwischen den einzelnen Typen von Zustandsgraphen eines I–Systems IS aufgeführt. Aus dem Verhaltensgraphen $VG(IS)$ kann der Erweiterte Casegraph $ECG(IS)$ und daraus der Casegraph $CG(IS)$ berechnet werden. Die Rückrichtungen sind nicht möglich, da vom Verhaltensgraphen zum Erweiterten Casegraphen die Phasenqualitäten und vom Erweiterten Casegraphen zum Casegraphen die Kantenklassifizierungen verloren gehen.

Satz 6.14 (Graphenabhängigkeiten). Sei IS ein I–System und sei $VG(IS) = \{Z, \rightarrow_Z\}$ der Verhaltensgraph, $ECG(IS) = \{C, \rightarrow_{C_1}, \rightarrow_{C_2}\}$ der Erweiterte Casegraph, und $CG(IS) = \{\bar{C}, \rightarrow_{\bar{C}}\}$ der Casegraph von IS. Es gilt:

a) *Berechnung von ECG(IS) aus VG(IS)*
$C = \{zc(z) \mid z \in Z\}$,
$(c_1, c_2) \in \rightarrow_{C_1}$ gdw. $\exists (z_1, z_2) \in \rightarrow_Z, p \in c_1 : z_1(p) \in b(p) \setminus \{p\} \wedge z_2(p) = 0$,
$(c_1, c_2) \in \rightarrow_{C_2}$ gdw. $\exists (z_1, z_2) \in \rightarrow_Z, p \in c_1 : z_1(p) = \mathsf{F} \wedge z_2(p) = 0$

b) *Berechnung von CG(IS) aus VG(IS)*
$\bar{C} = \{zc(z) \mid z \in Z\}$,
$(c_1, c_2) \in \rightarrow_{\bar{C}}$ gdw. $\exists (z_1, z_2) \in \rightarrow_Z : zc(z_1) = c_1, zc(z_2) = c_2, c_1 \neq c_2$

c) *Berechnung von CG(IS) aus ECG(IS)*
$\bar{C} = C$,
$(c_1, c_2) \in \rightarrow_{\bar{C}}$ gdw. $(c_1, c_2) \in \rightarrow_{C_1} \cup \rightarrow_{C_2}$

Beweis. Der Satz ist eine direkte Folgerung aus den einzelnen Graphen-Definitionen 6.1, 6.6 und 6.10, unter Beachtung der Abhängigkeiten von $\mathcal{V}[\![IS]\!]$, $\mathcal{CT}[\![IS]\!]$, $\mathcal{ECT}[\![IS]\!]$ aus den Satz 4.20. □

Bei der Verwendung von Zustandsgraphen eines I–Systems zur Systembeschreibung und -analyse geht man davon aus, dass das Systemverhalten einer modellierten realen Anwendung durch die Pfade in dem Graphen korrekt (im Sinne der Systemanalyse) repräsentiert wird. In dem formalen Modell der I–Systeme wird nicht gefordert, dass gleichzeitig eine 1:1 Beziehung zwischen den Pfaden im Graphen und den Ausführungen des zugeordneten $\mathrm{V_I}$Systems bestehen muss, derart, dass eine Folge von Systemzuständen genau dann als Pfad im Graphen existiert, wenn es eine Ausführung des $\mathrm{V_I}$Systems gibt, aus deren aufeinander folgenden z-Globalbelegungen sich die Folge ergibt. Die Anwendung und das $\mathrm{V_I}$System vertreten unterschiedliche Modellierungsebenen, die nicht direkt miteinander verbunden sind (siehe hierzu auch Kapitel 7). In den Beweisen zu Satz 6.5 und Satz 6.9 wurden Beispiele aufgezeigt, bei denen obige 1:1 Beziehung auch tatsächlich nicht besteht, woraus sich dann ein Unterschied in der Ausdruckskraft von Zustandsgraphen und Trace-Semantiken ergeben hat.

Es ist ein offenes Problem, ob und wie eine 1:1 Beziehung zwischen den Pfaden im Graphen und den Ausführungen des $\mathrm{V_I}$Systems durch eine geeignete Modellanpassung erreicht wird, womit dann auch die Gleichheit in der Ausdruckskraft von Graphen und Trace-Semantiken verbunden ist. Weiterführende Arbeiten werden sich gezielt mit der Lösung des Problems befassen (siehe Kapitel 12.2.2).

Kapitel 7

Modellierungsmethodik

In Kapitel 1 wurde festgestellt, dass sich bei der Arbeit mit I–Systemen verschiedene Modellierungsebenen unterscheiden lassen. Ein Hauptziel dieser Arbeit ist es, diese Modellierungsebenen herauszuarbeiten und als Bestandteil des formalen Modells zu etablieren. In den vorangegangenen Kapiteln ist dieses Ziel bereits durchgehend beachtet worden. Die einzelnen Teile (d.h. Syntax, Dynamik, Semantik) des formalen Modells I–System sind bestimmten Ebenen zugeordnet. Im ersten Abschnitt dieses Kapitels soll die Strukturierung noch einmal deutlich gemacht werden.

In den Kapiteln 4, 5 und 6 wurden verschiedene Beschreibungsformen für Semantiken für I–Systeme eingeführt (Trace-Semantiken, Interleaving Trace-Semantiken, Zustandsgraphen). Die einzelnen Formen unterscheiden sich in puncto Darstellungskomplexität und Informationsgehalt. Um einen Überblick zu geben, werden im zweiten Teil dieses Kapitels die verschiedenen Beschreibungsformen, deren Abhängigkeiten und Aussagekräfte zusammenfassend präsentiert.

7.1 Modellierungsebenen

Bei der Modellierung eines verteilten Systems mit I–Systemen bewegt man sich innerhalb unterschiedlicher Beschreibungsebenen. Jede Ebene repräsentiert eine bestimmte Betrachtungsweise auf das Gesamtmodell. Sie beinhaltet eigene Begriffe, Notationen, Eigenschaften und Beweismethoden. Zwischen den einzelnen Ebenen bestehen fest vorgegebene Verbindungen/Schnittstellen. Dadurch werden zum einen Anwendern Orientierungshilfen bei dem Design und der Analyse von I–Systemen und zum anderen den Entwicklern der Theorie der I–Systeme präzise Ansatzpunkte für Modellerweiterungen bzw. Modellanpassungen geboten.
In Abbildung 7.1 sind die verschiedenen Modellierungsebenen dargestellt. Sie werden im weiteren der Reihe nach betrachtet und es werden die Bezüge zu den vorangegangenen Kapiteln angegeben.

7.1.1 Anwendungsebene

Ausgegangen wird von einem realen verteilten System, das formal modelliert und analysiert bzw. für das Systemanforderungen verifiziert werden sollen (z.B. das Fußgänger-Leitsystem LS aus Kapitel 1.1). Da als formales Modell I–Systeme eingesetzt werden, müssen die formalen Komponenten eines I–Systems den realen Komponenten der Anwendung zugeordnet werden, d.h. die mathematische Struktur $(P, B, \underline{B}, K, E,)$ wird auf der Anwendungsebene interpretiert. Gleiches gilt für die formalen Semantiken der I–Systeme, die in Abbildung 7.2 zusammengefasst sind. Die Interpretationen ergeben sich aus der Art der Anwendung und unterliegen a priori keinen Einschränkungen. So ist z.B. eine geographische Verteiltheit von vornherein nicht gefordert, Nachrichtenaustauschmechanismen können, müssen aber nicht vorhanden sein. Durch Analyse- oder Verifikationsverfahren auf der formalen Ebene festgestellte Eigenschaften werden im Kontext der Interpretationen auf das reale System als vorhandene Systemeigenschaften übertragen.

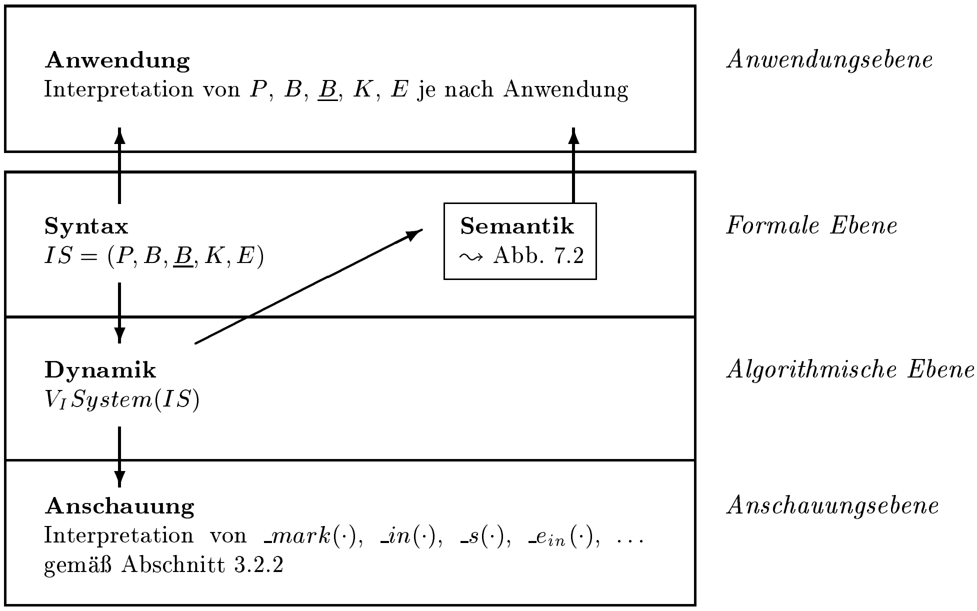


Abbildung 7.1: Modellierungsebenen

Die Eignung von I–Systemen als vielseitiges Modellierungs- und Analysewerkzeug in unterschiedlichen Anwendungsbereichen hat sich bereits in vorangegangenen Arbeiten gezeigt. Betrachtet wurden dabei unter anderem Hardwarekomponenten, das Resource-Management bei Betriebssystemen, mechanische Konstruktionen, Abläufe innerhalb einer Büroorganisation [14, 56, 85, 91]. Von einer konkreten Anwendung wird in dieser Arbeit bewusst abstrahiert, um eine Allgemeingültigkeit der entwickelten formalen Ansätze anzustreben. Weiterführende Arbeiten werden sich gezielt mit der Anwendungsebene beschäftigen (siehe Kapitel 12.2.8).

7.1.2 Formale Ebene

Bewegt man sich innerhalb der formalen Ebene, abstrahiert man von konkreten Anwendungen oder Anschauungen. Von Interesse sind ausschließlich das I–System und dessen Semantiken als mathematische Konstrukte. In diesem Sinne ist ein I–System IS ein 5-Tupel mit einer endlichen Menge von Bezeichnern (P), Mengen von Mengen von Bezeichnern (B und $\underline{B}$) und Mengen von Tupeln von Bezeichnern (K und E) als Komponenten. Die Semantiken sind gegeben als Folgen von Funktionen oder Mengen. Des Weiteren gibt es Graphen in den üblichen mathematischen Notationen.
Alle in dieser Arbeit eingeführten Semantiken inklusive der Zustandsgraphen sind in Abbildung 7.2 dargestellt. Zur Bestimmung der grundlegenden Trace-Semantiken Verhalten $\mathcal{V}[\![IS]\!]$, Casetrace-Semantik $\mathcal{CT}[\![IS]\!]$ und Erweiterte Casetrace-Semantik $\mathcal{ECT}[\![IS]\!]$ (eingerahmt durch das dünne Rechteck) wird auf die algorithmische Ebene zugegriffen. Dort werden die relevanten Traces aus den Ausführungen des zugeordneten $\mathrm{V_I}$Systems $V_I System(IS)$ abgeleitet. Die entscheidenden Festlegungen hierzu finden sich in Definition 4.1 für $\mathcal{V}[\![IS]\!]$, Definition 4.8 für $\mathcal{CT}[\![IS]\!]$ und Definition 4.16 für $\mathcal{ECT}[\![IS]\!]$. Alle weiteren Semantiken einschließlich der Zustandsgraphen können aus bereits bekannten Semantiken berechnet werden, ohne auf die Ausführungen von $V_I System(IS)$ zurückzugreifen. Die formale Ebene wird in diesen Fällen nicht verlassen. In Abschnitt 7.2 wird ein Überblick über die betreffenden Definitionen gegeben.
Die formale Ebene bildet die Basis für modulare Analyse- und Verifikationsverfahren bei I–Systemen (siehe Kapitel 9 ff.). Äquivalenzdefinitionen und Transformationsregeln (einschließlich der Korrektheitsbeweise) beziehen sich ebenfalls auf die mathematischen Konstrukte (siehe Kapitel 11). Man bewegt sich auf einer Ebene, auf der auch andere formale Modelle, z.B Petri-Netze und Statecharts, einzuordnen sind.

7.1.3 Algorithmische Ebene

Eine Besonderheit bei dem Modell der I–Systeme besteht in der Spezifikation der Dynamik. Bei Petri-Netzen oder Statecharts erfolgt die Spezifikation der Dynamik auf der formalen Ebene. Es existieren dort Übergangsfunktionen oder Mengenoperationen auf Multimengen (Markierungen), um Folgezustände eines aktuellen Systemzustandes festzulegen. Die mathematischen Konstrukte der formalen Ebenen reichen hierbei zur Berechnung aus. Bei I–Systemen wird zur Spezifizierung die formale Ebene verlassen und eine algorithmische Ebene eingeführt. Die erreichbaren Systemzustände (Cases oder globale Aktivitätszustände) und die jeweils möglichen Folgezustände werden aus den Ausführungen eines so genannten $\mathrm{V_I}$System abgeleitet. Das einem I–System IS durch Definition 3.9 zugeordnete $\mathrm{V_I}$System $V_I System(IS)$ unterliegt den in Abschnitt 3.2.1 vorgegebenen Verhaltensaxiomen VA1, VA2, VA3. Die Aktivitäten der einzelnen Komponenten werden durch die algorithmisch formulierten Aktionen A1-A13 (Abschnitt 3.2.2) festgelegt.
Bei der Modellierung eines verteilten Systems spielt das Erfassen von Ungewissheiten in den Aktivitäten der lokalen Komponenten, hervorgerufen durch die verteilte Kontrolle und durch Nachrichtenverzögerungen, eine zentrale Rolle. $V_I System(IS)$ bietet durch die integrierten Kommunikationsmechanismen die Möglichkeit, solche Ungewissheiten modellinhärent zu erfassen. Es vermittelt auf diese Weise ein anschauliches realitätsbezogenes Verständnis der verteilten Abläufe und notwendigen Synchronisationsmechanismen. Der operationelle programmiersprachliche Ansatz, die Abläufe innerhalb der Komponenten zu beschreiben, soll dem Benutzer helfen, das gegenüber Petri-Netzen oder Statecharts in der Beschreibung komplexere Systemverhalten leicht und eindeutig nachvollziehen zu können. Der Ansatz erlaubt zudem eine Unterscheidung von Aktionen, die eintreten werden, und Aktionen, die eintreten können, ohne zusätzliche Informationen am Modell (z.B. Zusatzbeschriftungen) angeben zu müssen.

7.1.4 Anschauungsebene

Der programmiersprachliche Ansatz innerhalb der algorithmischen Ebene erlaubt eine eindeutige und kurze Spezifikation der Aktionen A1-A13 (Abschnitt 3.2.2), erfordert aber die Fähigkeit, mit Variablen, Nachrichten und Programmbefehlen umgehen, d.h. deren Syntax und Semantik verstehen zu können. Folglich kann es vorkommen, dass der Zweck gewisser Kontrollstrukturen für einen ungeübten Anwender nicht offensichtlich ist. Das erschwert die Einarbeitung und mindert die allgemeine Akzeptanz, I–Systeme zur Modellierung einzusetzen. Aus diesem Grund ist es hilfreich, den einzelnen lokalen Variablen der Komponenten eines $\mathrm{V_I}$Systems sowie den Befehlsfolgen der Aktionen anschauliche Interpretationen zuzuordnen. Das führt dann zu einem leichteren Verständnis der Aktionen und dient zugleich der Rechtfertigung der einzelnen verteilten Algorithmen.
Die Interpretationsinhalte sind motiviert durch die Schlüsselphänomene aus Kapitel 1.1. Sie umfassen Einflüsse, Zwänge, Entscheidungen, Erregungen, In-/Stabilität, sowie deren Auswirkungen. Die Zuordnung erfolgt in Abschnitt 3.2.2. Dort werden den lokalen Variablen $\_mark(\cdot)$, $\_in(\cdot)$, $\_s(\cdot)$, $\_e_{in}(\cdot)$, $\_e_{out}(\cdot)$, $\_k(\cdot)$, $\_z(\cdot)$ Bedeutungen zugewiesen, gefolgt von Beschreibungen zu den Aktionen A1-A13.
Die allgemein gehaltenen Interpretationen auf der Anschauungsebene erlauben einen anschaulichen Bezug zu den meisten Anwendungen auf der Anwendungsebene herzustellen, obwohl dieser nicht explizit durch die Modellierungsebenen vorgegeben ist. So lassen sich die einzelnen Begriffe auf unterschiedliche Weise auf bestimmte Anwendungen übertragen. Denkbare Zuordnungen sind z.B.: Zwänge entsprechen mechanischen Kräften, Signalen oder organisatorischen Vorschriften; Instabilität entspricht mechanischem Vibrieren, einer Programmprozedur während der Abarbeitung eines Fehlerfalls, dem Warten auf eine schriftliche Bestätigung; eine Entscheidung treffen entspricht der gezielten Interaktion mit einer Steuereinheit, dem Einleiten bestimmter Verwaltungsvorgänge; usw.. Diese Deutungen ergeben sich aus der Art der Anwendung und sind letztendlich dem Anwender überlassen. Sie werden nicht durch das formale Modell der I–Systeme vorgegeben, um die Anwendungsgebiete für das Modell nicht unnötig einzugrenzen. Folglich gibt es in Abbildung 7.1 keine explizite Verbindung zwischen der Anschauungsebene und der Anwendungsebene.

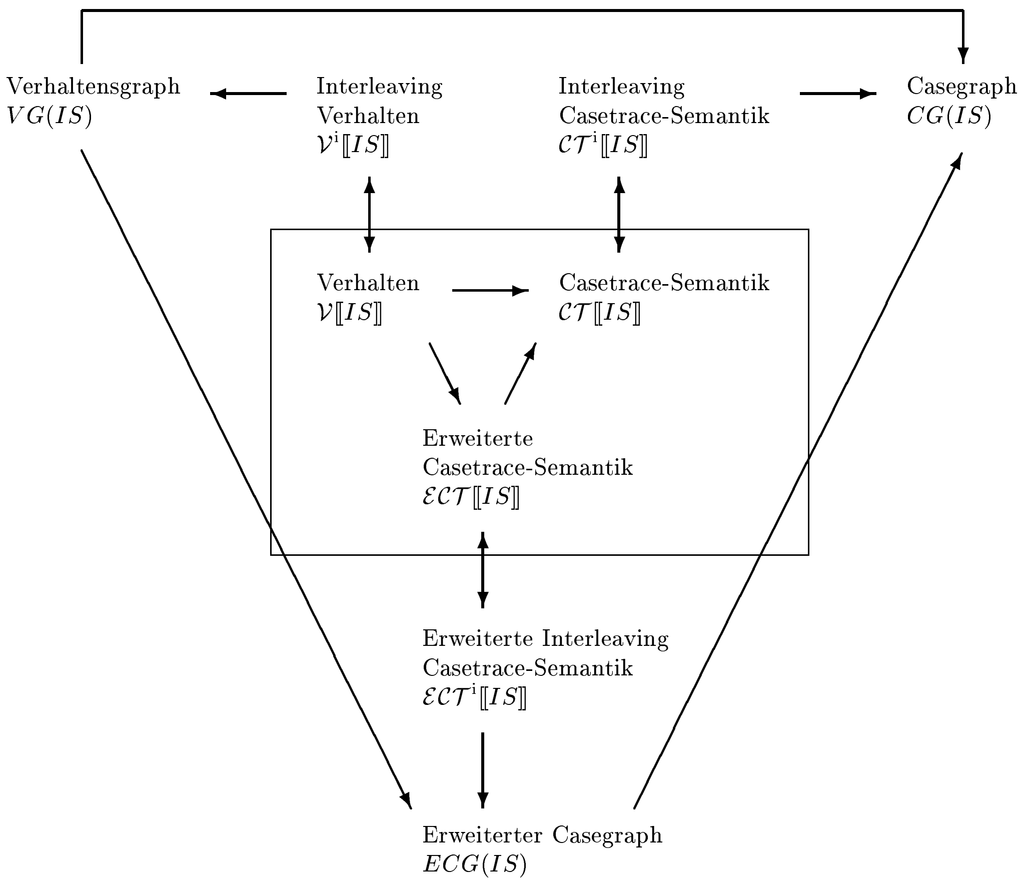


Abbildung 7.2: Semantische Beziehungen

7.2 Semantische Zusammenhänge

In Abschnitt 7.1 wurde bereits die Zuordnung der Trace-Semantiken und Zustandsgraphen eines I–Systems zur formalen Ebene aufgezeigt. Im Folgenden erfolgt eine genauere Unterteilung der verschiedenen Beschreibungsformen, wobei die Beziehungen untereinander verdeutlicht werden sollen. Je nach Interesse bzw. Bedarf des Anwenders stehen verschiedene Typen von Trace-Semantiken und Zustandsgraphen unterschiedlicher Ausdrucksstärke zur Auswahl. Ein Mehr an Ausdruckskraft bedeutet mehr Möglichkeiten zur Systemanalyse und Systemverifikation, allerdings zu Lasten der Komplexität der Darstellung. Die Abwägung der jeweiligen Kosten obliegt dem Benutzer. Motiviert wurden die einzelnen Trace-Semantiken und Zustandsgraphen ausführlich in den vorangegangenen Kapiteln. Die Charakteristika werden im weiteren Verlauf deshalb nur noch einmal kurz skizziert.

Abbildung 7.2 zeigt die bisher eingeführten Semantiken eines I–Systems IS. Die verschiedenen Zustandsgraphen sind mit aufgenommen. Das Rechteck umschließt dabei die drei Trace-Semantiken, die über das IS zugeordnete $\mathrm{V_I}$System $V_I System(IS)$ bestimmt werden und für die damit eine direkte Verbindung zur algorithmischen Ebene besteht (vgl. Abbildung 7.1). Diese Verbindung wird hergestellt durch die Definition 4.1 für das Verhalten $\mathcal{V}[\![IS]\!]$, Definition 4.8 für die Casetrace-Semantik $\mathcal{CT}[\![IS]\!]$ und Definition 4.16 für die Erweiterte Casetrace-Semantik $\mathcal{ECT}[\![IS]\!]$. Existiert ein Pfeil von Semantik/Graph X zu Semantik/Graph Y, bedeutet das, dass Y berechnet werden kann, sofern X bekannt ist. Doppelpfeile stehen für eine wechselseitige Berechnungsmöglichkeit. Existiert nur ein Einfachpfeil (und kein Doppelpfeil), dann ist eine Berechnung in der Rückrichtung im Allgemeinen nicht möglich. Die Korrektheit der einzelnen Pfeile ist durch entsprechende Definitionen und Sätze, die im Folgenden angegeben werden, gesichert.

Das Verhalten $\mathcal{V}[\![IS]\!]$ ist die ausdrucksstärkste und von der Darstellung komplexeste Semantik von IS. Ihr liegt das Interesse zugrunde, nicht nur das Einnehmen einzelner Phasen in den Bereichen des I–Systems zu erfassen, sondern auch eine Qualität der Aktivität, deren Abhängigkeiten und Entwicklung. Aus diesem Grund werden Folgen globaler Aktivitätszustände betrachtet. Die Casetrace-Semantik $\mathcal{CT}[\![IS]\!]$ ist vom Verhalten durch Vernachlässigung der Phasenqualitäten ableitbar, gemäß Satz 4.14. Es erfolgt bei den mathematischen Strukturelementen ein Übergang von Funktionen nach Mengen. Von Bedeutung sind nur noch die in den Bereichen aus globaler Sicht eingenommenen Phasen und die auftretenden Phasentransitionen. Ähnliches gilt für die Ableitung der Erweiterten Casetrace-Semantik $\mathcal{ECT}[\![IS]\!]$ aus dem Verhalten, gemäß Satz 4.20.b. Gegenüber der Casetrace-Semantik bleibt hierbei allerdings noch die Information (in Form von Bereichsmengen) erhalten, ob auftretende Phasentransitionen als frei oder als erzwungen einzuordnen sind. Der Übergang von der Erweiterten Casetrace-Semantik zur Casetrace-Semantik ist offensichtlich. Gemäß Satz 4.20.d werden nur die Zusatzinformationen über den Typ der Phasentransitionen (d.h. die Bereichsmengen) weggelassen.
Da bei den Übergängen vom Verhalten zur Casetrace-Semantik und zur Erweiterten Casetrace-Semantik Informationen über die Phasenqualitäten, und beim Übergang von der Erweiterten Casetrace-Semantik zur Casetrace-Semantik Informationen über die Typen der Phasentransitionen verloren gehen, existieren in diesen Fällen keine Berechnungsrückrichtungen.

Für die drei bisher angesprochenen Trace-Semantiken gibt es jeweils Interleaving-Varianten, bei denen sich aufeinander folgende Aktivitätszustände bzw. Cases in den Traces nur minimal unterscheiden. Globalzeitlich gleichzeitig eintretende Ereignisse (Wechsel von Phasenqualitäten, Phasentransitionen) in unterschiedlichen Bereichen des I–Systems werden nicht berücksichtigt. Nebenläufigkeit wird auf Kombinationen von sequentiellen Abläufen zurückgeführt.
Beim Übergang von einer Trace-Semantik zu seiner Interleaving-Variante werden alle Traces, die an mindestens einer Stelle globalzeitlich parallele Ereignisse repräsentieren, weggelassen. Dies zeigt sich in der Definition 5.2 für das Interleaving Verhalten $\mathcal{V}^{\mathrm{i}}[\![IS]\!]$, Definition 5.13 für die Interleaving Casetrace-Semantik $\mathcal{CT}^{\mathrm{i}}[\![IS]\!]$ und Definition 5.7 für die Erweiterte Interleaving Casetrace-Semantik $\mathcal{ECT}^{\mathrm{i}}[\![IS]\!]$. Die Rückkonstruktionen, d.h. die Berechnung der Trace-Semantiken aus ihren Interleaving-Varianten, sind in Kapitel 5 ebenfalls gezeigt worden. Die entscheidenden Aussagen finden sich in Satz 5.4 für das Verhalten, Satz 5.15 für die Casetrace-Semantik und Satz 5.9 für die Erweiterte Casetrace-Semantik.

Als anschauliche endliche Darstellungsformen für das Systemverhalten eines I–Systems IS wurden verschiedene Zustandsgraphen eingeführt, deren Einordnung in Abbildung 7.2 sich aus den Erkenntnissen aus Kapitel 6 ergibt. Die Knotenmenge des Verhaltensgraphens $VG(IS)$ entspricht, gemäß Definition 6.1, den innerhalb der Traces des Interleaving-Verhaltens auftretenden globalen Aktivitätszuständen. Die Kanten verdeutlichen direktes aufeinander folgen der verbundenen Aktivitätszustände in mindestens einer der Traces. Die Konstruktion des Casegraphen $CG(IS)$ ergibt sich in gleicher Weise aus der Interleaving Casetrace-Semantik, mit den Cases als Knoten, entsprechend Definition 6.6. Beim Erweiterten Casegraphen $ECG(IS)$, Definition 6.10, erfolgt zusätzlich eine Zweiteilung der Kantenmenge, je nachdem ob Caseübergänge eine freie oder eine erzwungene Phasentransition repräsentieren. Diese Information ist aus den Bereichsmengen in den Traces der Erweiterten Interleaving Casetrace-Semantik ablesbar.
Somit ist der Verhaltensgraph aus dem Interleaving Verhalten, der Casegraph aus der Interleaving Casetrace-Semantik und der Erweiterte Casegraph aus der Erweiterten Interleaving Casetrace-Semantik berechenbar. In Kapitel 6 wurde gezeigt, dass die Rückrichtungen, d.h. die Rekonstruktionen der Interleaving Trace-Semantiken und damit der allgemeinen Trace-Semantiken aus den Zustandsgraphen, nicht in jedem Fall möglich sind. Abschnitt 6.1 liefert den Nachweis, dass es vorkommen kann, dass bestimmte Folgen von globalen Aktivitätszuständen, die durch Pfade im Verhaltensgraphen gegeben sind und mit einem stabilen globalen Aktivitätszustand beginnen, nicht als Traces im Interleaving Verhalten und damit auch nicht als Traces im Verhalten existieren. Diese in Bezug auf die Rekonstruktion „falschen“ Pfade im Graphen sind nicht identifizierbar. Gleiches gilt für den Casegraphen und die Berechnung der (Interleaving) Casetrace-Semantik, Abschnitt 6.2, sowie für den Erweiterten Casegraph und die Berechnung der Erweiterten (Interleaving) Casetrace-Semantik, Abschnitt 6.3. Jeweils lassen sich Fälle angeben, in denen die Rekonstruktion der (Interleaving) Trace-Semantiken nicht möglich ist. Folglich

sind in Abbildung 7.2 zwischen den Interleaving Trace-Semantiken und den daraus abgeleiteten Zustandsgraphen nur Pfeile in eine Richtung eingezeichnet.
Die Übergänge zwischen den einzelnen Graphentypen ergeben sich aus Satz 6.14. Sowohl der Casegraph als auch der Erweiterte Casegraph sind aus dem Verhaltensgraphen berechenbar. In beiden Fällen ergibt sich die Knotenmenge aus der Knotenmenge des Verhaltensgraphens, indem der Übergang von globalen Aktivitätszuständen nach Cases vollzogen wird. Kanten des Verhaltensgraphens werden genau dann übernommen, wenn sie freie oder erzwungene Phasentransitionen repräsentieren. Beim Erweiterten Casegraphen wird der Unterschied zwischen frei und erzwungen durch zwei Kantenmengen berücksichtigt. Beim Casegraphen wird nur eine Kantenmenge gebildet. Eine Rückberechnung des Verhaltensgraphen ist in beiden Fällen nicht möglich, da die Phasenqualitäten in den Knoten des Verhaltensgraphens nicht rekonstruiert werden können. Die Bestimmung des Casegraphen aus dem Erweiterten Casegraphen erfolgt durch Vereinigung der beiden Kantenmengen bei identischer Knotenmenge. Durch die Vereinigung gehen die Informationen über die Kantentypen verloren, weshalb eine Rückberechnung des Erweiterten Casegraphen aus dem Casegraphen im Allgemeinen nicht möglich ist.

In der Abbildung 7.2 sind nur die Übergänge aufgenommen, deren Berechtigungen durch entsprechende Berechnungsvorschriften und Sätze in dieser Arbeit nachgewiesen worden sind. Genau die fehlenden Übergänge erhält man durch Bildung des transitiven Abschlusses. Z.B. dürfte in Abbildung 7.2 ein Pfeil von der Erweiterten Casetrace-Semantik $\mathcal{ECT}[\![IS]\!]$ zum Casegraphen $CG(IS)$ gezogen werden, da bereits Pfeile von der Erweiterten Casetrace-Semantik zur Casetrace-Semantik $\mathcal{CT}[\![IS]\!]$, von dort zur Interleaving Casetrace-Semantik $\mathcal{CT}^{\mathrm{i}}[\![IS]\!]$, und von dort zum Casegraphen eingezeichnet sind. Ein Doppelpfeil wäre hingegen nicht korrekt.

Der folgende Satz baut auf die in Abschnitt 7.2 präsentierten Beziehungen zwischen Trace-Semantiken und Zustandsgraphen von I–Systemen auf.

Satz 7.1 (Semantische Gleichheiten). Seien IS_a und IS_b zwei I–Systeme mit $IS_a \neq IS_b$. Es gelten die folgenden Implikationen und Äquivalenzen:

$$\begin{array}{ccccc}
\mathcal{V}[\![IS_a]\!] = \mathcal{V}[\![IS_b]\!] & \Leftrightarrow & \mathcal{V}^{\mathrm{i}}[\![IS_a]\!] = \mathcal{V}^{\mathrm{i}}[\![IS_b]\!] & \Rightarrow & VG(IS_a) = VG(IS_b) \\
\Downarrow & & \Downarrow & & \Downarrow \\
\mathcal{ECT}[\![IS_a]\!] = \mathcal{ECT}[\![IS_b]\!] & \Leftrightarrow & \mathcal{ECT}^{\mathrm{i}}[\![IS_a]\!] = \mathcal{ECT}^{\mathrm{i}}[\![IS_b]\!] & \Rightarrow & ECG(IS_a) = ECG(IS_b) \\
\Downarrow & & \Downarrow & & \Downarrow \\
\mathcal{CT}[\![IS_a]\!] = \mathcal{CT}[\![IS_b]\!] & \Leftrightarrow & \mathcal{CT}^{\mathrm{i}}[\![IS_a]\!] = \mathcal{CT}^{\mathrm{i}}[\![IS_b]\!] & \Rightarrow & CG(IS_a) = CG(IS_b)
\end{array}$$

Beweis. Die Implikationen und Äquivalenzen folgen direkt aus Abbildung 7.2 und den zugehörigen Erklärungen im Text. Das Beweisvorgehen wird exemplarisch an der Implikation $(\mathcal{V}[\![IS_a]\!] = \mathcal{V}[\![IS_b]\!]) \Rightarrow (\mathcal{ECT}[\![IS_a]\!] = \mathcal{ECT}[\![IS_b]\!])$ gezeigt.

Sei $ectr \in \mathcal{ECT}[\![IS_a]\!]$

$\Rightarrow$ {Die Erweiterte Casetrace-Semantik $\mathcal{ECT}[\![IS_a]\!]$ kann aus dem Verhalten $\mathcal{V}[\![IS_a]\!]$ berechnet werden; Satz 4.20.b}
$\exists ztr \in \mathcal{V}[\![IS_a]\!] : ectr = \lfloor ztr \rfloor^{\mathrm{e}}$

$\Rightarrow$ {vorausgesetzte Gleichheit von $\mathcal{V}[\![IS_a]\!]$ und $\mathcal{V}[\![IS_b]\!]$}
$\exists ztr \in \mathcal{V}[\![IS_b]\!] : ectr = \lfloor ztr \rfloor^{\mathrm{e}}$

$\Rightarrow$ {Berechnung von $\mathcal{ECT}[\![IS_b]\!]$ aus $\mathcal{V}[\![IS_b]\!]$; Satz 4.20.b}
$ectr \in \mathcal{ECT}[\![IS_b]\!]$.

Symmetrisch folgt $ectr \in \mathcal{ECT}[\![IS_a]\!]$ aus $ectr \in \mathcal{ECT}[\![IS_b]\!]$, und damit gilt die Gleichheit $\mathcal{ECT}[\![IS_a]\!] = \mathcal{ECT}[\![IS_b]\!]$.

Der Beweis der restlichen Implikationen aus dem Satz verläuft analog. Bei den Äquivalenzen werden beide Richtungen separat betrachtet. □

Kapitel 8

Beziehung zwischen I–Systemen und Lose Gekoppelten Systemen

In Kapitel 1.3 wurde bereits beschrieben, dass Lose Gekoppelte Systeme (LCS) als formales Modell für verteilte Systemabläufe der Entwicklung der I–Systeme vorausgingen. Die Aufgabe von Lose Gekoppelten Systemen ist die Darstellung von Abhängigkeiten zwischen Prozessen in Teilsystemen. Ziel ist die vollständige Charakterisierung von Verhaltenseigenschaften, auch unter Annahme von Verhaltenskonflikten. Es wird der Versuch gemacht, wesentliche Aspekte des Systemverhaltens zu erfassen allein durch Berücksichtigung der Einflüsse der Teilsysteme aufeinander, ohne Eingehen auf deren innere Struktur. Das legt die Einführung eines Ereignisbegriffs nahe, der sich darauf stützt, dass die betrachteten Systemkomponenten das Verhalten jeder ihrer Nachbarkomponenten nicht zu sehr einengen dürfen, sonder nur *lose* daran gekoppelt sind. Die Philosophie des Modells wählt zur Strukturbildung die Verknüpfungen, mit denen zueinander unverträgliche Elemente benannt werden können. Einen Überblick über die Theorie der Lose Gekoppelten Systeme liefern [16, 65, 83].
Um die Modellierungsmöglichkeiten auszubauen, wurden Lose Gekoppelte Systeme zu Interaktionssystemen und schließlich zu I–Systemen weiterentwickelt. Berücksichtigt werden sollten unter anderem eine lokale Autonomie einzelner Systemkomponenten sowie der Aspekt lokaler Zwänge, die durch andere Komponenten initiiert sind, und deren Auswirkungen. Ein Ziel ist dabei die Erfassung globaler Effekte lokaler Anforderungen und Gegebenheiten. Im Gegensatz zu Lose Gekoppelten Systemen erlauben Interaktionssysteme und I–Systeme die Beschreibung asymmetrischer Ereignisstrukturen. Auf die Modellierungsmöglichkeiten von I–Systemen wurde bereits in den Kapiteln 3 und 4 ausführlich eingegangen.
Die Modellerweiterungen hin zu I–Systemen führten insbesondere zu einem Ausbau der zugrunde liegenden formalen Struktur und zur Integration eines neuen Ansatzes bei der Spezifikation der Dynamik. Dabei war es durchgehend das Ziel, die Lose Gekoppelten Systeme syntaktisch und semantisch in die I–Systeme einzubetten. In diesem Kapitel wird gezeigt, dass die Einbettung vollzogen worden ist, derart, dass Lose Gekoppelte Systeme speziellen I–Systemen entsprechen, bei denen es keine trägen Bereiche und eine leere Erregungsrelation gibt.

8.1 Lose Gekoppelte Systeme

Um die Voraussetzungen für das Verständnis der in den nächsten Abschnitten präsentierten Sätze und deren Beweise zu schaffen, werden im Folgenden die notwendigen formalen Grundlagen der Lose Gekoppelten Systeme vorgestellt und kurz auf die wichtigsten Unterschiede zu I–Systemen hingewiesen.

Formal wird ein Lose Gekoppeltes System durch ein 4-Tupel beschrieben, dessen Komponenten bestimmte Elemente eines verteilten Systems repräsentieren. Wie bei I–Systemen werden die Systemkomponenten/Knoten des verteilten Systems durch so genannte *Bereiche* modelliert. Jeder Bereich zeichnet sich aus durch eine endliche Anzahl von für die Modellierung des Systemverhaltens relevanter lokaler Zustände, die *Phasen* genannt werden. Das einzige, was über die Bereiche angenommen wird, ist, dass sie sich zu jeder Zeit in genau einer Phase befinden. Die Einflüsse eines Bereiches b_1 auf einen anderen Bereich b_2 werden durch eine zweistellige symmetrische *Kopplungsrelation* $K\langle b_1, b_2\rangle$ beschrieben. Sie wird als Unverträglichkeit interpretiert, d.h. ein Paar (p_1, p_2) von Phasen aus b_1 bzw. b_2 gehört gerade dann zu dieser Relation, wenn der Bereich b_1 in der Phase p_1 den Bereich b_2 in der Phase p_2 stört, so dass die gleichzeitige Gültigkeit von p_1 und p_2 auszuschließen ist. Verschiedene Phasen eines Bereiches schließen sich immer gegenseitig aus. Im Gegensatz zu I–Systemen existieren bei Lose Gekoppelten Systemen keine ausgezeichnete Menge von trägen Bereichen und keine Erregungsrelation. Hinzu kommt hingegen eine Menge von so genannten *Cases*, die mögliche globale Systemzustände repräsentieren. Ein Case ist eine Menge von Phasen, jeweils genau eine aus jedem Bereich, die sich paarweise nicht wechselseitig ausschließen. Bei I–Systemen wurde auf die Hinzunahme der Casemenge verzichtet, da sie sich eindeutig aus den anderen Komponenten berechnen lässt und somit keine Zusatzinformation darstellt. Zusammenfassend definiert sich ein Lose Gekoppeltes System wie folgt:

Definition 8.1 (Lose Gekoppeltes System). Ein 4-Tupel $LCS = (P, B, C, K)$ heißt *Lose Gekoppeltes System*, wenn gilt:

(1) P ist eine endliche Menge von *Phasen*.

(2) B ist eine Menge von *Bereichen* mit

  a) $\forall b \in B : b \subseteq P$

  b) $\bigcup_{b \in B} b = P$

  c) $\forall b_1, b_2 \in B, b_1 \neq b_2 : b_1 \cap b_2 = \emptyset$

(3) C ist die Menge der *Cases* mit $c \in C$ gdw.

  a) $c \subseteq P$

  b) $\forall b \in B : |c \cap b| = 1$

  c) $\forall p_1, p_2 \in c : (p_1, p_2) \notin K$

(4) $K \subseteq P \times P$ ist die *Kopplungsrelation von* LCS und ist definiert als $K := \bigcup_{b_1, b_2 \in B} K\langle b_1, b_2\rangle$, wobei $K\langle b_1, b_2\rangle \subseteq b_1 \times b_2$ die *Kopplungsrelation zwischen* den Bereichen b_1 *und* b_2 ist. Hierbei gilt:

  a) $K\langle b_2, b_1\rangle^{-1} = K\langle b_1, b_2\rangle$

  b) für $b_1 = b_2$ wird festgelegt: $K\langle b_1, b_2\rangle := \{(p, p') \mid p, p' \in b_1,\ p \neq p'\}$

Mit $LCSystem$ werde die Menge aller Lose Gekoppelten Systeme bezeichnet. □

Bemerkung 8.2. Die graphische Repräsentation eines Lose Gekoppelten Systems (P, B, C, K) entspricht der des I–Systems $(P, B, \emptyset, K, \emptyset)$, d.h. es entfallen schraffierte/eingefärbte abgerundete Rechtecke zur Kennzeichnung träger Bereiche und Pfeile zur Darstellung der Erregungsrelation.

Die Dynamik bei Lose Gekoppelten Systemen leitet sich vollständig aus der formalen Struktur aus Definition 8.1 ab. Die Basis bilden die Cases (Punkt 3 der Definition) als diskrete globale Systemzustände. Paare von Cases definieren *Ereignisse*, die aus Sicht der Systemdynamik Zustände und mögliche Folgezustände repräsentieren. Für Ereignisse wird gefordert, dass sie sich aus so genannten *Elementarereignissen* zusammensetzen lassen. Jedes Elementarereignis beschreibt einen Wechsel der eingenommenen Phase in *genau einem* Bereich des Lose Gekoppelten Systems. Es findet dort eine *Phasentransition* statt, wobei sich die Ausgangs- und Zielphase aus den beiden Cases, die das Elementarereignis bilden, ergeben.

Definition 8.3 (Elementarereignis, Phasentransition). Sei LCS ein Lose Gekoppeltes System mit Casemenge C und seien $c_1, c_2 \in C$.

a) Das Tuple (c_1, c_2) repräsentiert ein *Elementarereignis* in LCS gdw. $|c_1 \setminus c_2| = |c_2 \setminus c_1| = 1$.

b) Sei (c_1, c_2) ein Elementarereignis in LCS und $\{p\} := c_1 \setminus c_2$, $\{q\} := c_2 \setminus c_1$. Dann beschreibt (c_1, c_2) die *Phasentransition* $p \to q$ in c_1.

c) Das Tuple (c_1, c_2) repräsentiert ein *Ereignis* in LCS gdw. $\exists n \in I\!N$, $\exists c'_1, \ldots, c'_n \in C$: $c_1 = c'_1 \wedge c_2 = c'_n \wedge (\forall i = 1, \ldots, n-1 : (c_i, c_{i+1})$ repräsentiert ein Elementarereignis in $LCS)$. □

Die Forderung für Ereignisse nach Zusammensetzbarkeit aus Elementarereignissen verhindert das Auftreten von koinzidenten Phasentransitionen bei über Kreuz wechselseitig ausgeschlossenen Phasen in benachbarten Bereichen. Diese Situation wurde bereits für I–Systeme in den Erläuterungen zu Satz 4.12.b ausführlicher diskutiert. Da die globale Zeitgleichheit zweier lokaler Zustandswechsel in unterschiedlichen Komponenten eines verteilten Systems (bei verteilter Kontrolle) nicht garantiert werden kann, ist die Forderung sinnvoll. Für weitere Details sei auf [16, 83] verwiesen. Man beachte, dass solch eine Forderung nach Zusammensetzbarkeit aus Elementarereignissen bei I–Systemen nicht vorliegt. Der Ausschluss von Koinzidenz in obiger Situation ergibt sich für ein I–System implizit aus dessen Dynamik (d.h. aus den Aktionen A1-A13 des zugeordneten $\mathrm{V_I}$Systems) und spiegelt sich in den Charakterisierungssätzen der Semantiken wider (Sätze 4.6.c, 4.12.b, 4.22.b). Bei Lose Gekoppelten Systemen sind vergleichbare Aussagen aus deren Dynamik nicht ableitbar. Die Forderung muss somit explizit angegeben werden.

Die Art der Spezifikation der Dynamik bei Lose Gekoppelten Systemen unterscheidet sich wesentlich von der bei I–Systemen. Zur Bestimmung der Ereignisse bei Lose Gekoppelten Systemen reicht es, deren formale Struktur (aus Definition 8.1) zu kennen. Die Berechnung von Folge-Cases ist dabei vergleichbar mit der Berechnung von Folge-Markierungen bei Petri-Netzen. Betrachtet man ein Modellierungsschema wie in Abbildung 7.1, dann bewegt man sich bei der Definition der Dynamik eines Lose Gekoppelten Systems ausschließlich auf der formalen Ebene. Hingegen wurde im Modell der I–Systeme eine zusätzliche algorithmische Ebene eingeführt. Die Dynamik eines I–Systems ergibt sich aus den Ausführungen des zugeordneten $\mathrm{V_I}$Systems. In Kapitel 7 wurde auf diese Modellierungsmethodik eingegangen.

Als Beschreibungsform für die Semantik eines Lose Gekoppelten Systems LCS wird ein *Casegraph* verwendet. Dessen Knotenmenge entspricht der Menge der Cases von LCS und die Kantenmenge ergibt sich aus den Elementarereignissen in LCS. Pfade in dem Casegraphen spiegeln das Systemverhalten wider.

Definition 8.4 (Casegraph). Sei $LCS = (P, B, C, K)$ ein Lose Gekoppeltes System. Der LCS zugeordnete gerichtete Graph $CG(LCS)$ heißt *Casegraph* von LCS und wird festgelegt als $CG(LCS) := (C, \rightarrow)$ mit $(c_1, c_2) \in \rightarrow$ gdw. (c_1, c_2) repräsentiert ein Elementarereignis in LCS gemäß Definition 8.3.a. □

Bemerkung 8.5. In Definition 8.3.a ist offensichtlich, dass, wenn (c_1, c_2) ein Elementarereignis in LCS repräsentiert, dann repräsentiert auch (c_2, c_1) ein Elementarereignis in LCS, oder mit anderen Worten: die Ereignisstruktur ist symmetrisch. Für Definition 8.4 bedeutet das, dass, wenn $(c_1, c_2) \in \rightarrow$ gilt, dann gilt auch $(c_2, c_1) \in \rightarrow$. Es ist somit möglich, $CG(LCS)$ alternativ als ungerichteten Graphen zu definieren (wie z.B. in [65]), ohne den Informationsgehalt zu verändern. In diesem Kapitel wird die gerichtete Variante verwendet, um Vergleiche mit den (immer gerichteten) Casegraphen von I–Systemen direkt durchführen zu können.

8.2 Casegraphen bei Lose Gekoppelten Systemen und I–Systemen

In Abschnitt 8.1 wurde durch die Bemerkung 8.2 eine syntaktische Einbettung der Lose Gekoppelten Systeme in die I–Systeme präsentiert, derart, dass die graphischen Darstellungen identisch sind. Hierzu werden, beim Übergang vom Lose Gekoppelten System zum I–System, die Mengen der Phasen und der Bereiche sowie die Kopplungsrelation übernommen. Des Weiteren entfällt die Menge der Cases, und die Komponenten, die ein I–System gegenüber einem Lose Gekoppelten System mehr hat, d.h. die Menge der trägen Bereiche und die Erregungsrelation, werden als leere Mengen angenommen.

Nach der syntaktischen Einbettung bleibt zu untersuchen, ob ein Lose Gekoppeltes System und ein I–System, die beide in ihrer graphischen Darstellung identisch sind, auch semantisch äquivalent sind. Dabei sind die Systeme als semantisch äquivalent anzusehen, wenn ihre Casegraphen (die für beide Modelle durch die Definitionen 6.6 und 8.4 definiert sind) übereinstimmen. Die Übereinstimmung ist nicht offensichtlich, da sich die Spezifikationen der Dynamik bei den formalen Modellen der Lose Gekoppelten Systeme und der I–Systeme wesentlich unterscheiden. (Im vorangegangenen Abschnitt wurde bereits darauf eingegangen.) Der folgende Satz sagt aus, dass die Gleichheit der Casegraphen im obigen Fall gilt.

Satz 8.6 (Gleichheit von $CG(LCS)$ und $CG(IS)$). Gegeben seien das Lose Gekoppelte System $LCS = (P, B, C, K)$ und das I–System $IS = (P, B, \emptyset, K, \emptyset)$. Es gilt:

$$CG(LCS) = CG(IS)$$

Beweis. Sei $CG(LCS) = (C, \rightarrow)$ und sei $CG(IS) = (C', \rightarrow')$.
Definitionsgemäß entspricht die Knotenmenge C der Menge der Cases von LCS und C' entspricht der Menge der Cases von IS. Die Gleichheit dieser beiden Mengen, d.h. $C = C'$, lässt sich direkt nachrechnen.
Zu jeder Kante $(c_1, c_2) \in\rightarrow$ lässt sich eine Ausführung Π von $V_I System(IS)$ konstruieren, bei der zwei aufeinander folgende z-Globalbelegungen z_1, z_2 mit $zc(z_1) = c_1$ und $zc(z_2) = c_2$ auftreten. Folglich existieren $ctr_1, ctr_2 \in Case(IS)^*$, so dass $ctr_1.c_1.c_2.ctr_2 \in \mathcal{CT}^{\mathrm{i}}[\![IS]\!]$ gilt. Aus Definition 6.6 folgt dann $(c_1, c_2) \in\rightarrow'$. Durch logische Umformungen lässt sich zeigen, dass eine Kante $(c'_1, c'_2) \in\rightarrow'$ ein Elementarereignis in LCS (gemäß Definition 8.3.a) repräsentiert. Aus Definition 8.4 folgt $(c'_1, c'_2) \in\rightarrow$. Zusammenfassend ergibt sich $\rightarrow=\rightarrow'$.

Der vollständige Beweis befindet sich im Anhang A.2 (Seite 171 ff.). □

Der Satz gewährleistet, dass Lose Gekoppelte Systeme und deren Dynamik durch spezielle I–Systeme mit einer leeren Menge von trägen Bereichen und leerer Erregungsrelation simuliert werden können. Die Lose Gekoppelten Systeme sind auf diese Weise vollständig in die I–Systeme eingebettet, oder anders ausgedrückt, die I–Systeme sind eine wirkliche Erweiterung der Lose Gekoppelten Systeme. Keine Modellierungseigenschaften der Lose Gekoppelten Systeme gehen durch eine Verlagerung der Modellierung auf die I–Systeme verloren. Alle semantischen Aussagen über Lose Gekoppelte Systeme, die sich auf den Casegraphen abstützen, können in dem Fall fehlender träger Bereiche und leerer Erregungsrelation auf I–Systeme übertragen werden. Dabei ist es dann nicht mehr notwendig, die Korrektheit der Aussagen anhand einer Analyse der (in der Spezifikation komplexeren) Dynamik der I–Systeme zu verifizieren. Bei einer nichtleeren Menge von trägen Bereichen oder einer nichtleeren Erregungsrelation bei einem I–System IS gilt die Gleichheit der Casegraphen in Satz 8.6 in der Regel nicht, was das folgende Beispiel verdeutlicht.

Beispiel 8.7.

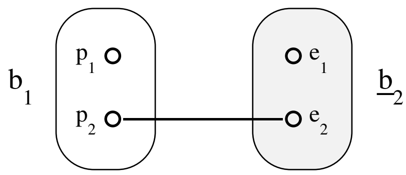


Abbildung 8.1: IS_4 mit $\underline{B} \neq \emptyset$

Abbildung 8.1 zeigt das I–System $IS_4 = (P, B, \underline{B}, K, \emptyset)$. Es gilt $\underline{b_2} \in \underline{B}$, also $\underline{B} \neq \emptyset$. Betrachtet man das Lose Gekoppelte System $LCS_1 = (P, B, C, K)$ und die Casegraphen $CG(LCS_1) = (C, \rightarrow_1)$ sowie $CG(IS_4) = (C, \rightarrow'_1)$, dann gilt zum einen $(\{p_1, e_1\}, \{p_1, e_2\}) \in\rightarrow_1$ und andererseits $(\{p_1, e_1\}, \{p_1, e_2\}) \notin\rightarrow'_1$. Daraus folgt $CG(LCS_1) \neq CG(IS_4)$.

Abbildung 8.2 zeigt das I–System $IS_5 = (P, B, \emptyset, K, E)$. Es gilt $(p_1, e_1) \in E$, also $E \neq \emptyset$. Betrachtet man das Lose Gekoppelte System $LCS_2 = (P, B, C, K)$ und die Casegraphen

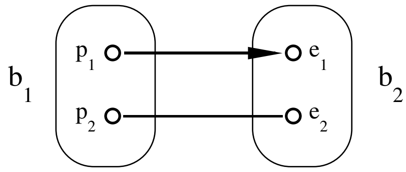


Abbildung 8.2: IS_5 mit $E \neq \emptyset$

$CG(LCS_2) = (C, \rightarrow_2)$ sowie $CG(IS_5) = (C, \rightarrow'_2)$, dann gilt zum einen $(\{p_1, e_1\}, \{p_2, e_1\}) \in \rightarrow_2$ und andererseits $(\{p_1, e_1\}, \{p_2, e_1\}) \notin \rightarrow'_2$. Daraus folgt $CG(LCS_2) \neq CG(IS_5)$. □

Nachdem die Gleichheit der Casegraphen für ein Lose Gekoppeltes System $LCS = (P, B, C, K)$ und das I–System $IS = (P, B, \emptyset, K, \emptyset)$ gezeigt wurde, soll als nächstes untersucht werden, in welcher Beziehung der Casegraph $CG(LCS)$ von LCS zur Interleaving Casetrace-Semantik $\mathcal{CT}^{\mathrm{i}}[\![IS]\!]$ von IS steht. In Kapitel 6.2 wurde beschrieben, dass es I–Systeme gibt, bei denen Casefolgen, die durch Pfade im zugehörigen Casegraphen gegeben sind, nicht als Traces in der zugehörigen Interleaving Casetrace-Semantik existieren. Träfe diese Eigenschaft für IS zu, dann gäbe es folglich (da nach Satz 8.6 $CG(LCS) = CG(IS)$) gilt) Casefolgen, die durch Pfade in $CG(LCS)$ gegeben sind, die aber nicht Elemente von $\mathcal{CT}^{\mathrm{i}}[\![IS]\!]$ sind. Der folgende Satz zeigt, dass dies nie der Fall ist.

Satz 8.8 (Gleichheit von $CG(LCS)$ und $\mathcal{CT}^{\mathrm{i}}[\![IS]\!]$). Gegeben seien das Lose Gekoppelte System $LCS = (P, B, C, K)$ und das I–System $IS = (P, B, \emptyset, K, \emptyset)$. Es gilt:

$$\{c_0 c_1 c_2 \ldots \mid \langle c_0, c_1, c_2, \ldots \rangle \text{ ist Pfad in CG(LCS)}\} = \mathcal{CT}^{\mathrm{i}}[\![IS]\!]$$

Beweis.
$\subseteq$: Zu jedem Pfad $\langle c_0, c_1, c_2, \ldots \rangle$ in $CG(LCS)$ lässt sich eine Ausführung Π von $V_I System(IS)$ konstruieren, bei der für die direkt nacheinander auftretenden z-Globalbelegungen $z_0, z_1, z_2, \ldots$ gilt: $\lfloor z_0 z_1 z_2 \ldots \rfloor = c_0 c_1 c_2 \ldots$. Da jede Teiltrace $c_{i-1} c_i$ für $i = 1, 2, \ldots$ jeweils nur eine einzelne Phasentransition repräsentiert, folgt $c_0 c_1 c_2 \ldots \in \mathcal{CT}^{\mathrm{i}}[\![IS]\!]$.
$\supseteq$: Durch logische Umformungen lässt sich zeigen, dass bei einer beliebigen Trace $c'_0 c'_1 c'_2 \ldots \in \mathcal{CT}^{\mathrm{i}}[\![IS]\!]$ jedes Paar (c'_{j-1}, c'_j), $j = 1, 2, \ldots$, ein Elementarereignis in LCS (gemäß Definition 8.3.a) repräsentiert. Somit ist (c'_{j-1}, c'_j) eine Kante in $CG(LCS)$ (siehe Definition 8.4) und folglich ist $\langle c'_0, c'_1, c'_2, \ldots \rangle$ ein Pfad in $CG(LCS)$.

Der vollständige Beweis befindet sich im Anhang A.2 (Seite 174 ff.). □

Aufgrund des Satzes können Strukturaussagen, die für ein Lose Gekoppeltes System gelten und sich auf dessen Casegraphen (genauer: auf die Folgen von Cases, die sich durch Pfade im Graphen ergeben) beziehen, nun direkt auf die Interleaving Casetrace-Semantik des in der graphischen Darstellung identischen I–Systems (ohne träge Bereiche und mit leerer Erregungsrelation) übertragen werden. Zum Beweis brauchen nicht die Verhaltensaxiome und Aktionen des die Dynamik des I–Systems bestimmenden zugeordneten $\mathrm{V_I}$Systems analysiert zu werden. Andererseits kann jede Aussage, die für die Interleaving Casetrace-Semantik des speziellen I–Systems formuliert wird, auch für den Casegraphen des korrespondierenden Lose Gekoppelten Systems übernommen werden.

Bei einer nichtleeren Menge von trägen Bereichen oder einer nichtleeren Erregungsrelation bei einem I–System IS gilt die Gleichheit aus Satz 8.8 in der Regel nicht. Das lässt sich an Beispiel 8.7 zeigen. Dort gilt: $\langle \{p_1, e_1\}, \{p_1, e_2\} \rangle$ ist Pfad in $CG(LCS_1)$, aber $\{p_2, e_1\}.\{p_1, e_1\} \notin \mathcal{CT}^{\mathrm{i}}[\![IS_4]\!]$. Und $\langle \{p_2, e_1\}, \{p_1, e_1\} \rangle$ ist Pfad in $CG(LCS_2)$, aber $\{p_2, e_1\}.\{p_1, e_1\} \notin \mathcal{CT}^{\mathrm{i}}[\![IS_5]\!]$. ($\{p_1, e_1\}$ ist kein möglicher End-Case, da danach noch die Phasentransition $e_1 \rightarrow e_2$ stattfinden wird).

8.3 ISystem$_{\mathbf{LCS}}$

Als ein wichtiges Nebenergebnis der Untersuchung der Beziehungen zwischen Lose Gekoppelten Systemen und I–Systemen lässt sich nun eine erste Klasse von I–Systemen angeben, bei denen der Casegraph und die Interleaving Casetrace-Semantik, und damit auch die Casetrace-Semantik, eines I–Systems gleichwertig sind in der Erfassung des Systemverhaltens des zugeordneten V$_{\mathrm{I}}$Systems. Die Konsequenz: Ist der Casegraph bekannt, kann die (Interleaving) Casetrace-Semantik berechnet werden, und ist die (Interleaving) Casetrace-Semantik bekannt, kann der Casegraph berechnet werden. Nach den Erläuterungen in Kapitel 6.2, resultierend aus Satz 6.9, ist diese Eigenschaft für beliebige I–Systeme nicht von vornherein gewährleistet.

Definition 8.9 (*$ISystem_{LCS}$*). Die Menge $ISystem_{LCS}$ aller I–Systeme ohne träge Bereiche und mit leerer Erregungsrelation ist wie folgt definiert:

$ISystem_{LCS} := \{(P, B, \underline{B}, K, E) \in ISystem \mid \underline{B} = \emptyset \wedge E = \emptyset\}$ □

Satz 8.10 (Berechnungen). Sei $IS \in ISystem_{LCS}$. Es gilt:

a) Aus $\mathcal{CT}[\![IS]\!]$ und aus $\mathcal{CT}^{\mathrm{i}}[\![IS]\!]$ kann $CG(IS)$ berechnet werden.

b) Aus $CG(IS)$ können $\mathcal{CT}^{\mathrm{i}}[\![IS]\!]$ und $\mathcal{CT}[\![IS]\!]$ berechnet werden.

Beweis.

Zu **a)**.
Ist für $IS \in ISystem_{LCS}$ die Casetrace-Semantik $\mathcal{CT}[\![IS]\!]$ bekannt, kann daraus die Interleaving Casetrace-Semantik $\mathcal{CT}^{\mathrm{i}}[\![IS]\!]$ bestimmt werden mit Definition 5.13. Aus $\mathcal{CT}^{\mathrm{i}}[\![IS]\!]$ kann der Casegraph $CG(IS)$ hergeleitet werden gemäß Definition 6.6.

Zu **b)**.
Ist $CG(IS)$ bekannt, dann liefert die Kombination der Sätze 8.6 und 8.8 die Berechnung von $\mathcal{CT}^{\mathrm{i}}[\![IS]\!]$ wie folgt: Satz 8.6 garantiert die Gleichheit von $CG(IS)$ und $CG(LCS)$, wobei LCS das zu IS graphisch identische Lose Gekoppelte System ist. Die Menge aller Case-Folgen, welche sich durch Pfade in $CG(LCS)$ (oder, aufgrund der Gleichheit, in $CG(IS)$) ergeben, entspricht $\mathcal{CT}^{\mathrm{i}}[\![IS]\!]$ nach Satz 8.8. Die Herleitung von $\mathcal{CT}[\![IS]\!]$ aus $\mathcal{CT}^{\mathrm{i}}[\![IS]\!]$ beschreibt Satz 5.15. □

Somit vermittelt der Casegraph eines I–Systems aus $ISystem_{LCS}$ genau die gleichen Informationen über ein modelliertes Systemverhalten wie dessen Interleaving Casetrace-Semantik und Casetrace-Semantik. Der Graph kann als anschauliche graphische Darstellungsform verwendet werden, ohne die in Kapitel 6.2 beschriebenen Einschränkungen in der Aussagekraft berücksichtigen zu müssen. Betrachtet man die Abbildung 7.2 in Kapitel 7.2 über die einzelnen Semantiken von I–Systemen und deren Beziehungen zueinander, dann müsste bei einer Beschränkung von IS auf Zugehörigkeit zu $ISystem_{LCS}$ zusätzlich ein Pfeil vom Casegraphen zur Interleaving Casetrace-Semantik eingezeichnet werden. Die Berechnung der Interleaving Casetrace-Semantik aus dem Casegraphen wurde im Beweis zu Satz 8.10.b aufgezeigt. Gegenstand laufender Arbeiten ist die Klärung der Frage, ob im Fall der Beschränkung auf $ISystem_{LCS}$ in Abbildung 7.2 ebenfalls Pfeile vom Erweiterten Casegraphen zur Erweiterten Interleaving Casetrace-Semantik und vom Verhaltensgraphen zum Interleaving Verhalten ergänzt werden können.

Kapitel 9

Semantische Projektionen

Große verteilte Systeme (z.B. das Leitsystem für Fußgänger im Straßenverkehr aus Kapitel 1.1), die aus einer Vielzahl interagierender Komponenten bestehen, können in der Praxis naturgemäß nicht im Ganzen modelliert und analysiert werden. Daran ist man in der Regel aber auch nicht interessiert. Vielmehr konzentriert sich das Interesse auf bestimmte Teilsysteme (z.B. die Positionsbestimmung oder das Alarmsystem) und deren Verhalten innerhalb des Gesamtsystems. Für die Theorie der I–Systeme bedeutet das, dass Formalismen bereitgestellt werden sollten, die innerhalb der Semantiken die „Sicht“ auf ausgezeichnete Teilsysteme ermöglichen. Das Ziel ist das Herausfiltern genau der semantischen Informationen, die für die Systemanalyse wichtig sind. In diesem Kapitel werden die notwendigen Grundlagen geschaffen, um mit solchen Sichten auf Teilsysteme arbeiten zu können. Dazu werden zur Klassifizierung der Bereiche eines I–Systems die Begriffe „Relevanzbereich“ und „Kontrollbereich“ aus [91] übernommen und präzisiert. Darauf aufbauend werden Projektionen der unterschiedlichen Semantiken (inkl. Graphen) auf Relevanz- oder Kontrollbereiche eingeführt. Ein beabsichtigter Nebeneffekt bei den Projektionen ist die Verringerung der Darstellungskomplexität der Beschreibungsformen für die analyserelevanten Systemabläufe (d.h. kürzere und weniger Traces, kleinere Zustandsgraphen), ein Anliegen, auf das bereits in den vorangegangenen Kapiteln hingewiesen wurde. Als Beispiel für eine Sicht auf einen einzelnen Bereich eines I–Systems wird in der zweiten Hälfte dieses Kapitels die Modellierung von lokalen Ereignisstrukturen und interner Zwänge mittels externer Einflüsse behandelt.

9.1 Relevanzbereiche und Kontrollbereiche

Betrachtet man den Aufbau eines I–Systems in Zusammenhang mit dem modellierten Anwendungssystem, dann kann man die Bereichsmenge zweiteilen. Es existieren so genannte *Relevanzbereiche*, die im Mittelpunkt der Systemanalyse stehen und für die bestimmte Strukturen und Eigenschaften nachgewiesen bzw. garantiert werden sollen. Anforderungen an die Aktivitäten in den Komponenten des Anwendungssystems werden über die dynamischen Abläufe in den Relevanzbereichen verifiziert. Neben den Relevanzbereichen gibt es *Kontrollbereiche*. Deren Existenz dient der Beeinflussung der Abläufe in den Relevanzbereichen, um gewünschte Systemanforderungen zu erzielen. Die Kontrollbereiche sind untereinander und mit den Relevanzbereichen über die Kopplungs- und Erregungsrelation des I–Systems verbunden.

Definition 9.1 (Relevanzbereich / Kontrollbereich). Sei IS ein I–System und b ein Bereich von IS.

a) b wird als *Relevanzbereich* bezeichnet, falls sich seine Existenz direkt aus der zu modellierenden Anwendung ergibt, d.h. b repräsentiert eine vorgegebene Anwendungskomponente, für die Systemanforderungen vorliegen, die mit Hilfe des formalen Modells realisiert werden sollen. $Relevanz(IS)$ bezeichnet die Menge aller Relevanzbereiche von IS.

b) Ist b kein Relevanzbereich, so wird er als *Kontrollbereich* bezeichnet. $Kontroll(IS)$ bezeichnet die Menge aller Kontrollbereiche von IS. □

Bemerkung 9.2. Aus der Definition 9.1 folgt direkt $Relevanz(IS) \cap Kontroll(IS) = \emptyset$ für jedes I–System IS. □

Während die Anzahl der Relevanzbereiche durch die zu modellierende Anwendung implizit vorgegeben ist, bestehen für die Anzahl und genaue Struktur der Kontrollbereiche vorrangig keine Vorgaben, sofern nur erforderliche Einflüsse auf die Relevanzbereiche gewährleistet sind. Die gesamte Menge der benötigten Kontrollbereiche ergibt sich in vielen Fällen erst im Laufe eines inkrementellen Modellierungsprozesses (siehe Kapitel 10.4). Im Hinblick auf eine effiziente Implementierung des I–Systems kann es allerdings von Interesse sein, die Anzahl und/oder Größe (Anzahl der Phasen) der Kontrollbereiche zu limitieren. Hieraus ergeben sich dann Fragen nach z.B. Äquivalenzbegriffen und Minimierungsstrategien. Auf die Entwicklung diesbezüglicher Konzepte wird in Kapitel 11 eingegangen.

9.2 Verhalten mit Sicht auf eine Bereichsmenge

Das Verhalten $\mathcal{V}[\![IS]\!]$ eines I–Systems IS basiert auf Folgen von globalen Aktivitätszuständen (gemäß Kapitel 4.2). Für einen solchen globalen Aktivitätszustand wird nun die *Sicht* auf eine ausgezeichnete Menge von Bereichen definiert. Dazu wird der Definitionsbereich der Abbildung auf die Phasen der angegebenen Bereiche begrenzt. Das Ergebnis der Abbildung wird für diese Phasen übernommen.

Definition 9.3 (Sicht bei globalen Aktivitätszuständen). Sei IS ein I–System mit Bereichsmenge B. Sei T eine nichtleere Teilmenge von B und $z \in GZustand(IS)$ ein globaler Aktivitätszustand. Die Abbildung

$$z\lfloor_T \quad : \quad \bigcup_{b \in T} b \longrightarrow \bigcup_{b \in T} b \cup \{0, 1, \mathsf{F}\}$$

mit:

$$z\lfloor_T (p) := z(p)$$

bezeichnet z *mit Sicht auf* T.

$GZustand(IS)\lfloor_T := \{z\lfloor_T \mid z \in GZustand(IS)\}$ ist die Menge der globalen Aktivitätszustände von IS mit Sicht auf T.
$RelGZustand(IS)\lfloor_T := \{z\lfloor_T \mid z \in RelGZustand(IS)\}$ ist die Menge der relevanten globalen Aktivitätszustände von IS mit Sicht auf T. □

Die obige Definition beschreibt Arten von Projektionen von globalen Aktivitätszuständen, und Mengen davon, auf Bereiche. Bei den Bereichen der Menge T kann es sich um Relevanz- oder Kontrollbereiche des I–Systems handeln, je nachdem, welche Analysen, bezogen auf welche Bereiche, durchgeführt werden sollen.
Die Definition der Sicht, wie sie für einen globalen Aktivitätszustand festgelegt ist, kann nun auf Folgen von globalen Aktivitätszuständen und damit auf das Verhalten eines I–Systems fortgesetzt werden.

Definition 9.4 (Sicht beim Verhalten). Sei IS ein I–System mit Bereichsmenge B. Sei $\emptyset \neq T \subseteq B$ und $z_0, z_1, z_2, \ldots \in GZustand(IS)$, z_0 stabil.

a) $(z_0.z_1.z_2\ldots)\lfloor_T := z_0\lfloor_T .z_{i_1}\lfloor_T .z_{i_2}\lfloor_T \ldots$ mit $i_1, i_2, \ldots \in I\!N$, $i_1 < i_2 < \ldots$ und $(j \in \{i_1, i_2, \ldots\}$ gdw. $z_j\lfloor_T \neq z_{j-1}\lfloor_T)$ definiert $z_0.z_1.z_2\ldots$ *mit Sicht auf* T.

b) $\mathcal{V}[\![IS]\!](z_0)\lfloor_T := \{ztr\lfloor_T \mid ztr \in \mathcal{V}[\![IS]\!](z_0)\}$ ist das Verhalten von IS bzgl. z_0 mit Sicht auf Bereichsmenge T.

c) $\mathcal{V}[\![IS]\!]\lfloor_T := \{ztr\lfloor_T \mid ztr \in \mathcal{V}[\![IS]\!]\}$ ist das Verhalten von IS mit Sicht auf T. □

Teil a) der Definition beschreibt die Projektion einer Folge von globalen Aktivitätszuständen auf eine Menge T von Bereichen. Die in der Folge auftretenden Zustände werden der Reihe nach einzeln projiziert. Zusätzlich zur Beschränkung des Definitionsbereichs bei den Ergebnisabbildungen auf die Phasen der Bereiche in T (siehe Definition 9.3) werden aufeinander folgende gleiche Ergebnisse auf ein Vorkommen reduziert. Die Teile b) und c) nutzen die Festlegungen aus Teil a), um über die Projektionen der Elemente (Folgen von globalen Aktivitätszuständen) des Verhaltens $\mathcal{V}[\![IS]\!]$ eine Sicht für diese Semantik zu definieren. In Teil b) wird gegenüber c) von einem ausgezeichneten Startzustand ausgegangen.

Beispiel 9.5. Als Grundlage dient das I–System IS_1 aus Beispiel 2.2 und die in Beispiel 4.2 angegebenen Ausführungen und globalen Aktivitätszustände. Aus Definition 9.4 folgt nun:

$(z_0 z_1 z_2 z_3 z_4 z_5 z_6)\lfloor_{\{b_1,b_2\}} = [p_2{<}1{>}, q_2{<}1{>}].[p_2{<}p_1{>}, q_2{<}q_3{>}].[p_1{<}1{>}, q_2{<}q_3{>}].[p_1{<}p_2{>}, q_3{<}1{>}].$
$[p_2{<}1{>}, q_3{<}1{>}] \in \mathcal{V}[\![IS_1]\!]\lfloor_{\{b_1,b_2\}}$

$(z'_0 z'_1 z'_2 z'_3 z'_4)\lfloor_{\{b_1,b_2\}} = [p_1{<}1{>}, q_1{<}1{>}].[p_1{<}p_2{>}, q_1{<}1{>}].[p_2{<}1{>}, q_1{<}1{>}].[p_2{<}1{>}, q_1{<}\mathsf{F}{>}].$
$[p_2{<}1{>}, q_2{<}1{>}] \in \mathcal{V}[\![IS_1]\!]\lfloor_{\{b_1,b_2\}}$

$(z'_0 z'_1 z'_2 z'_3 z'_4 z_1 z_2 z_3 z_4 z_5 z_6)\lfloor_{\{b_1,b_2\}} = [p_1{<}1{>}, q_1{<}1{>}].[p_1{<}p_2{>}, q_1{<}1{>}].[p_2{<}1{>}, q_1{<}1{>}].$
$[p_2{<}1{>}, q_1{<}\mathsf{F}{>}].[p_2{<}1{>}, q_2{<}1{>}].[p_2{<}p_1{>}, q_2{<}q_3{>}].[p_1{<}1{>}, q_2{<}q_3{>}].[p_1{<}p_2{>}, q_3{<}1{>}].$
$[p_2{<}1{>}, q_3{<}1{>}] \in \mathcal{V}[\![IS_1]\!]\lfloor_{\{b_1,b_2\}}$ □

Bemerkung 9.6. Für ein I–System IS und eine Teilmenge T der Bereichsmenge ist das Verhalten des Teilsystems $IS\lfloor_T$ (Definition 2.6) nicht gleichzusetzen mit der Sicht des Verhaltens von IS auf T (Definition 9.4), d.h. im Allgemeinen gilt nicht: $\mathcal{V}[\![IS\lfloor_T]\!] = \mathcal{V}[\![IS]\!]\lfloor_T$. □

Die Bemerkung 9.6 lässt sich an dem I–System IS_1 nachvollziehen. Man betrachte dazu folgende Ausführung 4 von $V_I System(IS_1)$:

$$\overbrace{[p_1{<}1{>}, q_3{<}1{>}, v_2{<}1{>}]}^{z''_0} \xrightarrow{b_3.\mathrm{A1}} \overbrace{[p_1{<}1{>}, q_3{<}1{>}, v_2{<}\mathsf{F}{>}]}^{z''_1} \xrightarrow{b_1.\mathrm{A11}} \overbrace{[p_1{<}\mathsf{F}{>}, q_3{<}1{>}, v_2{<}\mathsf{F}{>}]}^{z''_2}$$
$$\xrightarrow{b_1.\mathrm{A5}} \overbrace{[p_2{<}1{>}, q_3{<}1{>}, v_2{<}\mathsf{F}{>}]}^{z''_3} \xrightarrow{b_3.\mathrm{A5}} \overbrace{[p_2{<}1{>}, q_3{<}1{>}, v_1{<}1{>}]}^{z''_4}$$

Somit gilt $z''_0 z''_1 z''_2 z''_3 z''_4 \in \mathcal{V}[\![IS_1]\!]$ und mit Sicht auf $\{b_1, b_2\}$:

$$(z''_0 z''_1 z''_2 z''_3 z''_4)\lfloor_{\{b_1,b_2\}} = \overbrace{[p_1{<}1{>}, q_3{<}1{>}].[p_1{<}\mathsf{F}{>}, q_3{<}1{>}].[p_2{<}1{>}, q_3{<}1{>}]}^{\overline{ztr}} \in \mathcal{V}[\![IS_1]\!]\lfloor_{\{b_1,b_2\}}. \qquad (*)$$

Betrachtet man nun $IS_1\lfloor_{\{b_1,b_2\}}$ (d.h. nach Definition 2.6 nur die Bereiche b_1 und b_2, deren Phasen, die leere Kopplungsrelation und die Erregungsrelation $E' = \{(p_2, q_1)\}$), dann ergibt sich für das zugeordnete V$_\mathrm{I}$System $V_I System(IS_1\lfloor_{\{b_1,b_2\}})$, dass es dort *keine* Ausführung

$$[p_1{<}1{>}, q_3{<}1{>}] \xrightarrow{b_1.\mathrm{A?}} [p_1{<}\mathsf{F}{>}, q_3{<}1{>}] \xrightarrow{b_1.\mathrm{A5}} [p_2{<}1{>}, q_3{<}1{>}]$$

gibt. Es gibt keine Aktion innerhalb von A1-A13, die den ersten Übergang der Ausführung bewirken kann. Es wäre nur möglich, falls $(q_3, p_1) \in E$ gelte, was aber nicht der Fall ist. Folglich gilt $\overline{ztr} \notin \mathcal{V}[\![IS_1\lfloor_{\{b_1,b_2\}}]\!]$ und mit $(*)$ folgt $\mathcal{V}[\![IS_1\lfloor_{\{b_1,b_2\}}]\!] \neq \mathcal{V}[\![IS_1]\!]\lfloor_{\{b_1,b_2\}}$, als ein Beispiel, bei dem die Gleichheit aus Bemerkung 9.6 nicht gilt.

9.3 Casetrace-Semantik mit Sicht auf eine Bereichsmenge

Die Casetrace-Semantik $\mathcal{CT}[\![IS]\!]$ eines I–Systems IS besteht aus Folgen von Cases von IS (gemäß Kapitel 4.3). Die *Sicht* eines Cases auf eine ausgezeichnete Menge von Bereichen ist eine Teilmenge des Cases. Diese Teilmenge enthält genau jede Phase des Cases, die auch als Element in einer der die Sicht spezifizierenden Bereiche vorkommt.

Definition 9.7 (Sicht bei Cases). Sei IS ein I–System mit Bereichsmenge B. Sei T eine nichtleere Teilmenge von B und $c \in Case(IS)$ ein Case.

Die Menge $c\lfloor_T := c \cap \bigcup_{b \in T} b$ bezeichnet c *mit Sicht auf* T.

$Case(IS)\lfloor_T := \{c\lfloor_T \mid c \in Case(IS)\}$ ist die Menge der Cases von IS mit Sicht auf T. □

Analog zu Abschnitt 9.2 kann man die Sichten, wie sie in der vorangegangenen Definition festgelegt sind, als Projektionen von Cases, und Mengen davon, auf Bereiche betrachten. Bei den Bereichen der Menge T kann es sich auch hier um Relevanz- oder Kontrollbereiche handeln.
Die Fortsetzung der Definition der Sicht von Cases auf Folgen von Cases beinhaltet die folgende Definition. Darauf aufbauend lassen sich Sichten bei der Casetrace-Semantik definieren.

Definition 9.8 (Sicht bei der Casetrace-Semantik). Sei IS ein I–System mit Bereichsmenge B. Sei $\emptyset \neq T \subseteq B$ und $c_0, c_1, c_2, \ldots \in Case(IS)$.

a) $(c_0.c_1.c_2 \ldots)\lfloor_T := c_0\lfloor_T .c_{i_1}\lfloor_T .c_{i_2}\lfloor_T \ldots$ mit $i_1, i_2, \ldots \in I\!N$, $i_1 < i_2 < \ldots$ und $(j \in \{i_1, i_2, \ldots\}$ gdw. $c_j\lfloor_T \neq c_{j-1}\lfloor_T)$ definiert $c_0.c_1.c_2 \ldots$ *mit Sicht auf* T.

b) $\mathcal{CT}[\![IS]\!](c_0)\lfloor_T := \{ctr\lfloor_T \mid ctr \in \mathcal{CT}[\![IS]\!](c_0)\}$ ist die Casetrace-Semantik von IS bzgl. c_0 mit Sicht auf T.

c) $\mathcal{CT}[\![IS]\!]\lfloor_T := \{ctr\lfloor_T \mid ctr \in \mathcal{CT}[\![IS]\!]\}$ ist die Casetrace-Semantik von IS mit Sicht auf T. □

Teil a) der Definition beschreibt die Projektion einer Folge von Cases auf eine Menge T von Bereichen. Die in der Folge auftretenden Cases werden der Reihe nach einzeln projiziert. Zusätzlich zur Durchschnittsbildung der Cases mit der Vereinigung der Bereiche aus T (siehe Definition 9.7) werden aufeinander folgende gleiche Ergebnismengen auf ein Vorkommen reduziert. Die Teile b) und c) nutzen die Festlegungen aus Teil a), um über die Projektionen der Elemente (Folgen von Cases) der Casetrace-Semantik $\mathcal{CT}[\![IS]\!]$ eine Sicht für diese Semantik zu definieren. In Teil b) wird gegenüber c) von einem ausgezeichneten Start-Case ausgegangen.

Beispiel 9.9. Als Grundlage dienen die Casetraces aus Beispiel 4.9 der Casetrace-Semantik des I–Systems IS_1 aus Beispiel 2.2. Nach Definition 9.8 gilt:

$(c_0 c_1 c_2 c_3 c_4)\lfloor_{\{b_1,b_2\}} = \{p_2,q_2\}.\{p_1,q_2\}.\{p_1,q_3\}.\{p_2,q_3\} \in \mathcal{CT}[\![IS_1]\!]\lfloor_{\{b_1,b_2\}}$

$(c_0' c_1' c_2')\lfloor_{\{b_1,b_2\}} = \{p_1,q_1\}.\{p_2,q_1\}.\{p_2,q_2\} \in \mathcal{CT}[\![IS_1]\!]\lfloor_{\{b_1,b_2\}}$

$(c_0' c_1' c_2' c_1 c_2 c_3 c_4)\lfloor_{\{b_1,b_2\}} = \{p_1,q_1\}.\{p_2,q_1\}.\{p_2,q_2\}.\{p_1,q_2\}.\{p_1,q_3\}.\{p_2,q_3\} \in \mathcal{CT}[\![IS_1]\!]\lfloor_{\{b_1,b_2\}}$ □

Bemerkung 9.10. Für ein I–System IS und eine Teilmenge T der Bereichsmenge ist die Casetrace-Semantik des Teilsystems $IS\lfloor_T$ (Definition 2.6) nicht gleichzusetzen mit der Sicht der Casetrace-Semantik von IS auf T (Definition 9.7), d.h. im Allgemeinen gilt nicht: $\mathcal{CT}[\![IS\lfloor_T]\!] = \mathcal{CT}[\![IS]\!]\lfloor_T$. □

Die Bemerkung 9.10 korrespondiert zu der Bemerkung 9.6 aus Abschnitt 9.2, diesmal allerdings mit Bezug auf die Casetrace-Semantik. Als Beispiel kann man wiederum IS_1 nehmen. Für den speziellen Start-Case $\{p_1,q_3,v_2\}$ und eine beliebige Casefolge $ctr \in Case(IS_1)^*$ folgt $ctr \neq \epsilon$ aus $\{p_1,q_3,v_2\}.ctr \in \mathcal{CT}[\![IS_1]\!]$. Die Aktionen A1-A13 bewirken, dass nach dem Start-Case noch mindestens eine Phasentransition (in b_1) stattfindet. Die Definition von $\mathcal{CT}[\![\cdot]\!]$ liefert $first(ctr) \neq \{p_1,q_3,v_2\}$ und die Definition von „Case" liefert $first(ctr) \neq \{p_1,q_3,v_1\}$. Wegen $\underline{b}_3 = \{v_1,v_2\}$ folgt $p_1 \notin first(ctr) \vee q_3 \notin first(ctr)$. Nach Definition 9.8 muss somit gelten: $\{p_1,q_3\} \notin \mathcal{CT}[\![IS_1]\!]\lfloor_{\{b_1,b_2\}}$. $(*)$
Betrachtet man das Teilsystem $IS_1\lfloor_{\{b_1,b_2\}}$, dann stellt sich heraus, dass $\{p_1,q_3\}$ ein möglicher End-Case innerhalb der Elemente (Traces) der Casetrace-Semantik ist. Da für das Teilsystem $K(p_1) = E^{-1}(p_1) = K(q_3) = E^{-1}(q_3) = \emptyset$ gilt, gibt es keine Aktion von A1-A13, die innerhalb der Ausführungen von $V_I System(IS_1\lfloor_{\{b_1,b_2\}})$ die Variablenbelegungen $\_z(p_1) = \mathsf{F}$ oder $\_z(q_3) = \mathsf{F}$ erwirkt und dadurch eine Phasentransition (gemäß A5) erzwingen kann. Somit gilt insbesondere $\{p_1,q_3\} \in \mathcal{CT}[\![IS\lfloor_{\{b_1,b_2\}}]\!]$. Zusammen mit $(*)$ folgt $\mathcal{CT}[\![IS_1]\!]\lfloor_{\{b_1,b_2\}} \neq \mathcal{CT}[\![IS_1\lfloor_{\{b_1,b_2\}}]\!]$, als ein Beispiel für eine Ungleichheit.

9.4 Erweiterte Casetrace-Semantik mit Sicht auf eine Bereichsmenge

Die Erweiterung bei der Erweiterten Casetrace-Semantik $\mathcal{ECT}[\![IS]\!]$ gegenüber der Casetrace-Semantik $\mathcal{CT}[\![IS]\!]$ besteht in der Hinzunahme von Bereichsmengen in die Traces der Semantik zur Klassifizierung von Phasentransitionen. Aufeinander folgende Cases in einer Trace repräsentieren eine oder mehrere Phasentransitionen, und eine dazwischenliegende Bereichsmenge gibt die Bereiche an, in denen eine Phasentransition stattfindet, die als frei bezeichnet wird. Die verbleibenden Phasentransitionen gelten dann als erzwungen. In Kapitel 4.4 wurde ausführlich darauf eingegangen.
Um die *Sicht* der Erweiterten Casetrace-Semantik auf eine ausgezeichnete Menge von Bereichen zu definieren, wird als erstes die Sicht von einer alternierenden Folge von Cases und Bereichsmengen auf die ausgezeichnete Menge von Bereichen festgelegt. Grundlage hierzu ist die Definition 9.8.a mit der Projektion einer Folge von Cases. Die zusätzlichen Bereichsmengen ergeben sich aus einer einfachen Schnittmengenbildung zwischen den Mengen, die die Phasentransitionen in der Ausgangsfolge klassifizieren, und den Mengen, die die Sicht spezifizieren.

Definition 9.11 (Sicht bei der Erweiterten Casetrace-Semantik). Sei IS ein I–System mit Bereichsmenge B. Sei $\emptyset \neq T \subseteq B$ und $c_0, c_1, c_2, \ldots \in Case(IS)$ sowie $\delta_1, \delta_2, \ldots \in \mathcal{P}(B)$.

a) $(c_0.\delta_1.c_1.\delta_2.c_2\ldots)\lfloor_T := c_0\lfloor_T .(\delta_{i_1} \cap T).c_{i_1}\lfloor_T .(\delta_{i_2} \cap T).c_{i_2}\lfloor_T \ldots$ mit $i_1, i_2, \ldots \in I\!N$, $i_1 < i_2 < \ldots$ und $(j \in \{i_1, i_2, \ldots\}$ gdw. $c_j\lfloor_T \neq c_{j-1}\lfloor_T)$ definiert $c_0.\delta_1.c_1.\delta_2.c_2\ldots$ *mit Sicht auf* T.

b) $\mathcal{ECT}[\![IS]\!](c_0)\lfloor_T := \{ectr\lfloor_T \mid ectr \in \mathcal{ECT}[\![IS]\!](c_0)\}$ ist die Erweiterte Casetrace-Semantik von IS bzgl. c_0 mit Sicht auf T.

c) $\mathcal{ECT}[\![IS]\!]\lfloor_T := \{ectr\lfloor_T \mid ectr \in \mathcal{ECT}[\![IS]\!]\}$ ist die Erweiterte Casetrace-Semantik von IS mit Sicht auf T. □

Teil a) der Definition beschreibt die Projektion einer Folge von alternierenden Cases und Bereichsmengen auf eine ausgezeichnete Menge T von Bereichen. Aus dem Blickpunkt der in der Folge auftretenden Cases werden diese der Reihe nach einzeln projiziert. Zusätzlich zur Durchschnittsbildung der Cases mit der Vereinigung der Bereiche aus T werden aufeinander folgende gleiche Ergebnismengen auf ein Vorkommen reduziert (analog zu Definition 9.8.a). Aus dem Blickpunkt der in der Folge auftretenden Bereichsmengen sind genau die Bereichsmengen von Interesse, bei denen die Projektion des jeweils nachfolgenden Cases in der Ergebnisfolge vorkommt. Für diese Bereichsmengen werden deren Schnittmengen mit T als Zwischenelemente in die Folge der Projektionen der Cases übernommen. Die Teile b) und c) nutzen die Festlegungen aus Teil a), um über die Projektionen der Elemente (Folgen von alternierenden Cases und Bereichsmengen) der Erweiterten Casetrace-Semantik $\mathcal{ECT}[\![IS]\!]$ eine Sicht für diese Semantik zu definieren. In Teil b) wird gegenüber c) von einem ausgezeichneten Start-Case ausgegangen.

Beispiel 9.12. Als Grundlage dient die Ausführung 3 aus Beispiel 4.17 und die daraus abgeleitete Erweiterte Casetrace. Es gilt mit Definition 9.11:

$(c_0\delta_1c_1\delta_2c_2\delta_3c_3\delta_4c_4\delta_5c_5\delta_6c_6\delta_7c_7)\lfloor_{\{b_1,b_2\}} = \{p_1,q_2\}.\{b_1,b_2\}.\{p_2,q_3\}.\{b_2\}.\{p_2,q_2\}.\{b_2\}.\{p_2,q_3\}.\{b_1\}.$
$\{p_1,q_3\}.\{\}.\{p_2,q_3\} \in \mathcal{ECT}[\![IS_1]\!]\lfloor_{\{b_1,b_2\}}$ □

Die Übertragung von Bemerkung 9.6 bzw. Bemerkung 9.10 auf die Erweiterte Casetrace-Semantik mit dem Ziel, den Unterschied zwischen Sichten und Teilsystemen aufzuzeigen, führt zu folgender Bemerkung.

Bemerkung 9.13. Für ein I–System IS und eine Teilmenge T der Bereichsmenge ist die Erweiterte Casetrace-Semantik des Teilsystems $IS\lfloor_T$ (Definition 2.6) nicht gleichzusetzen mit der Sicht der Erweiterten Casetrace-Semantik von IS auf T (Definition 9.11), d.h. im Allgemeinen gilt nicht: $\mathcal{ECT}[\![IS\lfloor_T]\!] = \mathcal{ECT}[\![IS]\!]\lfloor_T$. □

Als ein Beispiel für ein I–System, bei dem die Gleichheit aus Bemerkung 9.13 nicht gilt, bietet sich wiederum das I–System IS_1 an. Die in Abschnitt 9.2 vorgestellte Ausführung 4 von $V_I System(IS_1)$ liefert mit Definition 4.16:
$\{p_1, q_3, v_2\}.\{\}.\{p_2, q_3, v_2\}.\{\}.\{p_2, q_3, v_1\} \in \mathcal{ECT}[\![IS_1]\!]$ und mit Sicht auf $\{b_1, b_2\}$:
$\{p_1, q_3\}.\{\}.\{p_2, q_3\} \in \mathcal{ECT}[\![IS_1]\!]\lfloor_{\{b_1,b_2\}}$ (∗)
Nun wurde in Abschnitt 9.2 bereits erwähnt, dass eine Ausführung
$[p_1{<}1{>}, q_3{<}1{>}] \xrightarrow{b_1.\mathrm{A?}} [p_1{<}\mathsf{F}{>}, q_3{<}1{>}] \xrightarrow{b_1.\mathrm{A5}} [p_2{<}1{>}, q_3{<}1{>}]$
bei $V_I System(IS_1\lfloor_{\{b_1,b_2\}})$ nicht vorkommen kann. Für die Erweiterte Casetrace-Semantik von $IS_1\lfloor_{\{b_1,b_2\}}$ bedeutet das: Aus $\{p_1, q_3\}.\delta.\{p_2, q_3\} \in \mathcal{ECT}[\![IS_1\lfloor_{\{b_1,b_2\}}]\!]$ folgt $\delta \neq \{\}$. Zusammen mit (∗) ergibt sich die Ungleichung $\mathcal{ECT}[\![IS_1]\!]\lfloor_{\{b_1,b_2\}} \neq \mathcal{ECT}[\![IS_1\lfloor_{\{b_1,b_2\}}]\!]$.

9.5 Interleaving Trace-Semantiken mit Sicht auf eine Bereichsmenge

In Kapitel 5 wurden für ein I–System IS das Interleaving Verhalten $\mathcal{V}^{\mathrm{i}}[\![IS]\!]$, die Interleaving Casetrace-Semantik $\mathcal{CT}^{\mathrm{i}}[\![IS]\!]$ und die Erweiterte Interleaving Casetrace-Semantik $\mathcal{ECT}^{\mathrm{i}}[\![IS]\!]$ präsentiert. Bei diesen Interleaving Trace-Semantiken handelt es sich um Teilmengen der ohne den Präfix „Interleaving" gleichnamigen Trace-Semantiken. Traces, bei denen aufeinander folgende Elemente parallel stattfindende Ereignisse in unterschiedlichen Komponenten des modellierten Systems repräsentieren, werden nicht mit aufgenommen. Um die *Sicht* einer Interleaving Trace-Semantik auf eine ausgezeichnete Menge von Bereichen zu definieren, werden deren Elemente (Traces) einzeln projiziert. Die notwendigen Projektionen sind in den vorangegangenen Abschnitten bereits definiert worden. Zusammenfassend ergibt sich für alle bisher eingeführten Interleaving Trace-Semantiken:

Definition 9.14 (Sichten bei Interleaving Trace-Semantiken). Sei IS ein I–System mit Bereichsmenge B. Sei $\emptyset \neq T \subseteq B$ und $z_0 \in GZustand(IS)$, z_0 stabil, $c_0 \in Case(IS)$.

a) $\mathcal{V}^{\mathrm{i}}[\![IS]\!](z_0)\lfloor_T := \{ztr\lfloor_T \mid ztr \in \mathcal{V}^{\mathrm{i}}[\![IS]\!](c_0)\}$ ist das Interleaving Verhalten von IS bzgl. z_0 mit Sicht auf T.

b) $\mathcal{V}^{\mathrm{i}}[\![IS]\!]\lfloor_T := \{ztr\lfloor_T \mid ztr \in \mathcal{V}^{\mathrm{i}}[\![IS]\!]\}$ ist das Interleaving Verhalten von IS mit Sicht auf T.

c) $\mathcal{CT}^{\mathrm{i}}[\![IS]\!](c_0)\lfloor_T := \{ctr\lfloor_T \mid ctr \in \mathcal{CT}^{\mathrm{i}}[\![IS]\!](c_0)\}$ ist die Interleaving Casetrace-Semantik von IS bzgl. c_0 mit Sicht auf T.

d) $\mathcal{CT}^{\mathrm{i}}[\![IS]\!]\lfloor_T := \{ctr\lfloor_T \mid ctr \in \mathcal{CT}^{\mathrm{i}}[\![IS]\!]\}$ ist die Interleaving Casetrace-Semantik von IS mit Sicht auf T.

e) $\mathcal{ECT}^{\mathrm{i}}[\![IS]\!](c_0)\lfloor_T := \{ectr\lfloor_T \mid ectr \in \mathcal{ECT}^{\mathrm{i}}[\![IS]\!](c_0)\}$ ist die Erweiterte Interleaving Casetrace-Semantik von IS bzgl. c_0 mit Sicht auf T.

f) $\mathcal{ECT}^{\mathrm{i}}[\![IS]\!]\lfloor_T := \{ectr\lfloor_T \mid ectr \in \mathcal{ECT}^{\mathrm{i}}[\![IS]\!]\}$ ist die Erweiterte Interleaving Casetrace-Semantik von IS mit Sicht auf T. □

Die oben erwähnte Teilmengenbeziehung zwischen den Interleaving Trace-Semantiken und den gleichnamigen allgemeineren Semantiken gewährleistet die Wohldefiniertheit der einzelnen Teildefinitionen a)-f). Die Sicht von *ztr* auf T (in a) und b)) wird durch Definition 9.4, die Sicht von *ctr* auf T (in c) und d)) durch Definition 9.8, und die Sicht von *ectr* auf T (in e) und f)) durch Definition 9.11 abgedeckt.

Beispiel 9.15. In Kapitel 5 wurden die Interleaving-Varianten der vorgestellten Trace-Semantiken eingeführt. Die in den Beispielen 5.3, 5.8 und 5.14 präsentierten Traces können nun mit Sicht auf $\{b_1, b_2\}$ betrachtet werden, unter Anwendung von Definition 9.14. Es ergibt sich exemplarisch:

$ztr_a\lfloor_{\{b_1,b_2\}} = [p_2{<}1{>}, q_2{<}1{>}].[p_2{<}p_1{>}, q_2{<}1{>}].[p_2{<}p_1{>}, q_2{<}q_3{>}].[p_1{<}1{>}, q_2{<}q_3{>}].$
$[p_1{<}p_2{>}, q_2{<}q_3{>}].[p_1{<}p_2{>}, q_3{<}1{>}].[p_2{<}1{>}, q_3{<}1{>}] \in \mathcal{V}^{\mathrm{i}}[\![IS_1]\!]\lfloor_{\{b_1,b_2\}}$,

$ctr_a\lfloor_{\{b_1,b_2\}} = \{p_1, q_2\}.\{p_2, q_2\}.\{p_2, q_3\}.\{p_2, q_2\}.\{p_2, q_3\}.\{p_1, q_3\}.\{p_2, q_3\} \in \mathcal{CT}^{\mathrm{i}}[\![IS_1]\!]\lfloor_{\{b_1,b_2\}}$,

$ectr_a\lfloor_{\{b_1,b_2\}}= \{p_1, q_2\}.\{b_1\}.\{p_2, q_2\}.\{b_2\}.\{p_2, q_3\}.\{b_2\}.\{p_2, q_2\}.\{b_2\}.\{p_2, q_3\}.\{b_1\}.\{p_1, q_3\}.\{\}.\{p_2, q_3\} \in \mathcal{ECT}[\![IS_1]\!]\lfloor_{\{b_1,b_2\}}$.

Die verbleibenden Traces $ztr_b, ztr_c, ztr_d, ctr_b, ctr_c, ctr_d, ectr_b, ectr_c, ectr_d$ können analog projiziert werden. □

In Weiterführung der Bemerkungen aus den vorangegangenen Abschnitten ist auch bei den Interleaving Trace-Semantiken zwischen der Semantik eines Teilsystems und der Sicht auf die Menge der Bereiche des Teilsystems zu unterscheiden.

Bemerkung 9.16. Für ein I–System IS und eine Teilmenge T der Bereichsmenge von IS ist das Interleaving Verhalten des Teilsystems $IS\lfloor_T$ nicht gleichzusetzen mit der Sicht des Interleaving Verhaltens von IS auf T. Gleiches trifft für die Interleaving Casetrace-Semantik und die Erweiterte Interleaving Casetrace-Semantik zu. Zusammenfassend, im Allgemeinen gilt nicht: $\mathcal{V}^{\mathrm{i}}[\![IS\lfloor_T]\!] = \mathcal{V}^{\mathrm{i}}[\![IS]\!]\lfloor_T$, $\mathcal{CT}^{\mathrm{i}}[\![IS\lfloor_T]\!] = \mathcal{CT}^{\mathrm{i}}[\![IS]\!]\lfloor_T$, $\mathcal{ECT}^{\mathrm{i}}[\![IS\lfloor_T]\!] = \mathcal{ECT}^{\mathrm{i}}[\![IS]\!]\lfloor_T$. □

Zur Veranschaulichung der Bemerkung lassen sich die Beispiele zu den Bemerkungen 9.6, 9.10 und 9.13 direkt für die entsprechenden Interleaving Varianten übernehmen. Die Beispiele sind so konstruiert, dass dort nur sequentielle Zustandswechsel betrachtet werden.

9.6 Zustandsgraphen mit Sicht auf eine Bereichsmenge

Als mögliche Zustandsgraphen eines I–Systems IS wurden in Kapitel 6 der Verhaltensgraph $VG(IS)$, der Casegraph $CG(IS)$ sowie der Erweiterte Casegraph $ECG(IS)$ vorgestellt und in ihrer Ausdrucksstärke mit den Trace-Semantiken verglichen. Die Knoten- und Kantenmengen der einzelnen Zustandsgraphen leiten sich aus den Traces der gleichnamigen Interleaving Trace-Semantiken ab. Um für Zustandsgraphen *Sichten* auf eine ausgezeichnete Menge von Bereichen zu definieren, werden die jeweiligen Definitionen der Zustandsgraphen aus Kapitel 6 übernommen und anstelle der Interleaving Trace-Semantiken die in Abschnitt 9.5 definierten Sichten dieser Semantiken auf die betreffende Bereichsmenge eingesetzt.

Definition 9.17 (Sichten bei Zustandsgraphen). Sei IS ein I–System mit Bereichsmenge B und $\emptyset \neq T \subseteq B$.

a) Der *Verhaltensgraph* $VG(IS)\lfloor_T = (Z, \rightarrow)$ *mit Sicht auf* T ist festgelegt durch:

(1) $Z = RelGZustand(IS)\lfloor_T$

(2) $\rightarrow \subseteq Z \times Z$ mit:

$(z_1, z_2) \in \rightarrow$ gdw. $\exists ztr_1, ztr_2 \in (GZustand(IS)\lfloor_T)^* : ztr_1.z_1.z_2.ztr_2 \in \mathcal{V}^{\mathrm{i}}[\![IS]\!]\lfloor_T$

b) Der *Casegraph* $CG(IS)\lfloor_T = (C, \rightarrow)$ *mit Sicht auf* T ist festgelegt durch:

(1) $C = Case(IS)\lfloor_T$

(2) $\rightarrow \subseteq C \times C$ mit:

$(c_1, c_2) \in \rightarrow$ gdw. $\exists ctr_1, ctr_2 \in (Case(IS)\lfloor_T)^* : ctr_1.c_1.c_2.ctr_2 \in \mathcal{CT}^{\mathrm{i}}[\![IS]\!]\lfloor_T$

c) Der *Erweiterte Casegraph* $ECG(IS)\lfloor_T = (C, \rightarrow_1, \rightarrow_2)$ *mit Sicht auf* T ist festgelegt durch:

(1) $C = Case(IS)\lfloor_T$

(2) $\rightarrow_1 \subseteq C \times C$ mit:

$(c_1, c_2) \in \rightarrow_1$ gdw. $\exists ectr_1 \in (Case(IS)\lfloor_T .\mathcal{P}(T))^*, ectr_2 \in (\mathcal{P}(T).Case(IS)\lfloor_T)^*, \delta \subseteq T : ectr_1.c_1.\delta.c_2.ctr_2 \in \mathcal{ECT}^{\mathrm{i}}[\![IS]\!]\lfloor_T$ und $\delta \neq \emptyset$

(3) $\rightarrow_2 \subseteq C \times C$ mit:

$(c_1, c_2) \in \rightarrow_2$ gdw. $\exists ectr_1 \in (Case(IS)\lfloor_T .\mathcal{P}(T))^*, ectr_2 \in (\mathcal{P}(T).Case(IS)\lfloor_T)^*, \delta \subseteq B : ectr_1.c_1.\delta.c_2.ctr_2 \in \mathcal{ECT}^{\mathrm{i}}[\![IS]\!]\lfloor_T$ und $\delta = \emptyset$

Die Wohldefiniertheit der einzelnen Graphendefinitionen ist gewährleistet durch die Wohldefiniertheit von $\mathit{RelGZustand}(IS)\lfloor_T$, $\mathit{GZustand}(IS)\lfloor_T$, $\mathcal{V}^{\mathrm{i}}[\![IS]\!]\lfloor_T$, $\mathit{Case}(IS)\lfloor_T$, $\mathcal{CT}^{\mathrm{i}}[\![IS]\!]\lfloor_T$ und $\mathcal{ECT}^{\mathrm{i}}[\![IS]\!]\lfloor_T$ in den vorangegangenen Abschnitten. Letztere Wohldefiniertheit ergibt sich direkt aus den zugehörigen Erläuterungen.

Beispiel 9.18. Der Verhaltensgraph von IS_1 mit Sicht auf $\{b_1, b_2\}$, der Casegraph von IS_1 mit Sicht auf $\{b_1, b_2\}$ und der Erweiterte Casegraph von IS_1 mit Sicht auf $\{b_1, b_2\}$ des I–Systems IS_1 aus Beispiel 2.2 sind in den Abbildungen 9.1, 9.2 und 9.3 graphisch dargestellt. □

Im Beispiel zeigt sich, dass sich durch die Beschränkung auf für eine modulare Systemanalyse wesentliche semantische Informationen mittels Sichten die Darstellungskomplexität der verwendeten Graphen reduziert werden kann. So besitzt z.B. der Verhaltensgraph von IS_1 mit Sicht auf $\{b_1, b_2\}$ aus Abbildung 9.1 37 Knoten und 85 Kanten. Der komplette Verhaltensgraph, dargestellt in Abbildung 6.1, besitzt gemäß den Angaben in Beispiel 6.2 66 Knoten und 138 Kanten.

In Übereinstimmung mit den vorhergehenden Abschnitten soll als nächstes auf die Unterscheidung zwischen Zustandsgraphen von Teilsystemen eines I–Systems und Sichten von Zustandsgraphen auf die Bereiche der Teilsysteme eingegangen werden.

Bemerkung 9.19. Für ein I–System IS und eine Teilmenge T der Bereichsmenge von IS ist der Verhaltensgraph des Teilsystems $IS\lfloor_T$ nicht gleichzusetzen mit der Sicht des Verhaltensgraphens von IS auf T. Gleiches trifft für den Casegraphen und den Erweiterten Casegraphen zu. Zusammenfassend, im Allgemeinen gilt nicht: $VG(IS\lfloor_T) = VG(IS)\lfloor_T$, $CG(IS\lfloor_T) = CG(IS)\lfloor_T$, $ECG(IS\lfloor_T) = ECG(IS)\lfloor_T$. □

Das I–System IS_1 lässt sich erneut zur Veranschaulichung der Bemerkung heranziehen. Dabei können die bisherigen Ergebnisse aus den Abschnitten 9.2 - 9.5 mit einbezogen werden.

Für $\overbrace{[p_1{<}1{>}, q_3{<}1{>}]}^{\overline{z}_0}.\overbrace{[p_1{<}\mathsf{F}{>}, q_3{<}1{>}]}^{\overline{z}_1}.\overbrace{[p_2{<}1{>}, q_3{<}1{>}]}^{\overline{z}_2}$ gilt: $\overline{z}_0\overline{z}_1\overline{z}_2 \in \mathcal{V}^{\mathrm{i}}[\![IS_1]\!]\lfloor_{\{b_1,b_2\}}$ und $\overline{z}_0\overline{z}_1\overline{z}_2 \notin \mathcal{V}^{\mathrm{i}}[\![IS_1\lfloor_{\{b_1,b_2\}}]\!]$. Betrachtet man $VG(IS_1)\lfloor_{\{b_1,b_2\}} = (Z, \rightarrow_1)$ und $VG(IS_1\lfloor_{\{b_1,b_2\}}) = (Z, \rightarrow_2)$, dann folgt mit Definition 9.17.a: $((\overline{z}_0, \overline{z}_1) \in\rightarrow_1 \wedge (\overline{z}_0, \overline{z}_1) \notin\rightarrow_2) \vee ((\overline{z}_1, \overline{z}_2) \in\rightarrow_1 \wedge (\overline{z}_1, \overline{z}_2) \notin\rightarrow_2)$. Aufgrund der Ungleichheit von $\rightarrow_1$ und $\rightarrow_2$ gilt $VG(IS_1)\lfloor_{\{b_1,b_2\}} \neq VG(IS_1\lfloor_{\{b_1,b_2\}})$.
Das Teilsystem $IS_1\lfloor_{\{b_1,b_2\}}$ besitzt eine leere Kopplungsrelation, und nur (p_2, q_1) ist Bestandteil der Erregungsrelation. Damit fehlen notwendige strukturelle Voraussetzungen, um eine Phasentransition von q_3 nach q_1 in dem autonomen Bereich b_2 zu verhindern (überprüfbar an den Aktionen A1-13). Für $CG(IS_1\lfloor_{\{b_1,b_2\}} = (C, \rightarrow'_1)$ bedeutet das: $(\{p_1, q_3\}, \{p_1, q_1\}) \in\rightarrow'_1$. Hingegen, bei $CG(IS_1)\lfloor_{\{b_1,b_2\}} = (C, \rightarrow'_2)$, dargestellt in Abbildung 9.2, gilt $(\{p_1, q_3\}, \{p_1, q_1\}) \notin\rightarrow'_2$. Die Ungleichheit von $\rightarrow'_1$ und $\rightarrow'_2$ liefert $CG(IS_1)\lfloor_{\{b_1,b_2\}} \neq CG(IS_1\lfloor_{\{b_1,b_2\}})$.

Für $\overbrace{\{p_1, q_3\}}^{\overline{c}_0}.\{\}.\overbrace{\{p_2, q_3\}}^{\overline{c}_1}$ gilt: $\overline{c}_0.\{\}.\overline{c}_1 \in \mathcal{ECT}^{\mathrm{i}}[\![IS_1]\!]\lfloor_{\{b_1,b_2\}}$ und $\overline{c}_0.\{\}.\overline{c}_1 \notin \mathcal{ECT}^{\mathrm{i}}[\![IS_1\lfloor\{b_1, b_2\}]\!]$. Betrachtet man $ECG(IS_1)\lfloor_{\{b_1,b_2\}} = (C', \rightarrow_{11}, \rightarrow_{12})$ und $ECG(IS_1\lfloor_{\{b_1,b_2\}}) = (C', \rightarrow_{21}, \rightarrow_{22})$, dann folgt mit Definition 9.17.c: $(\overline{c}_0, \overline{c}_1) \in\rightarrow_{12} \wedge (\overline{c}_0, \overline{c}_1) \notin\rightarrow_{22}$. Aufgrund der Ungleichheit von $\rightarrow_{12}$ und $\rightarrow_{22}$ gilt $ECG(IS_1)\lfloor_{\{b_1,b_2\}} \neq ECG(IS_1\lfloor_{\{b_1,b_2\}})$.

In den letzten Bemerkungen ab 9.6 wurden die in der Regel vorliegenden Ungleichheiten zwischen einerseits den Trace-Semantiken/Zustandsgraphen eines Teilsystems eines I–Systems und andererseits den Sichten der gleichen Trace-Semantiken/Zustandsgraphen auf die Bereiche des Teilsystems betont. Diese Ungleichheiten gelten allerdings nicht immer. Mit z.B. IS_3 aus Kapitel 6.2 liegt ein Beispiel für eine Gleichheit vor. Es gilt nämlich (ohne Beweis): $\mathcal{CT}[\![IS_3]\!]\lfloor_{\{b_1,b_2\}} = \mathcal{CT}[\![IS_3\lfloor_{\{b_1,b_2\}}]\!]$, $\mathcal{CT}^{\mathrm{i}}[\![IS_3]\!]\lfloor_{\{b_1,b_2\}} = \mathcal{CT}^{\mathrm{i}}[\![IS_3\lfloor_{\{b_1,b_2\}}]\!]$ und $CG(IS_3)\lfloor_{\{b_1,b_2\}} = CG(IS_3\lfloor_{\{b_1,b_2\}})$. Bei Verwendung des (Interleaving) Verhaltens, der Erweiterten (Interleaving) Casetrace-Semantik, des Verhaltensgraphens oder des Erweiterten Casegraphens gilt für dieses Beispiel weiterhin die Ungleichheit. Erweitert man IS_3 zu IS'_3, indem zu $\underline{b}_3$ eine weitere Phase e_4 mit $(e_4, g_1) \in K$ hinzugefügt wird, dann ergibt sich sogar die Gleichheit für alle (Interleaving) Trace-Semantiken und Zustandsgraphen, d.h für alle $S \in \{\mathcal{V}, \mathcal{V}^{\mathrm{i}}, \mathcal{CT}, \mathcal{CT}^{\mathrm{i}}, \mathcal{ECT}, \mathcal{ECT}^{\mathrm{i}}\}$ gilt $S[\![IS'_3]\!]\lfloor_{\{b_1,b_2\}} = S[\![IS'_3\lfloor_{\{b_1,b_2\}}]\!]$, und für alle $G \in \{VG, CG, ECG\}$ gilt $G(IS_3)\lfloor_{\{b_1,b_2\}} = G(IS_3\lfloor_{\{b_1,b_2\}})$.

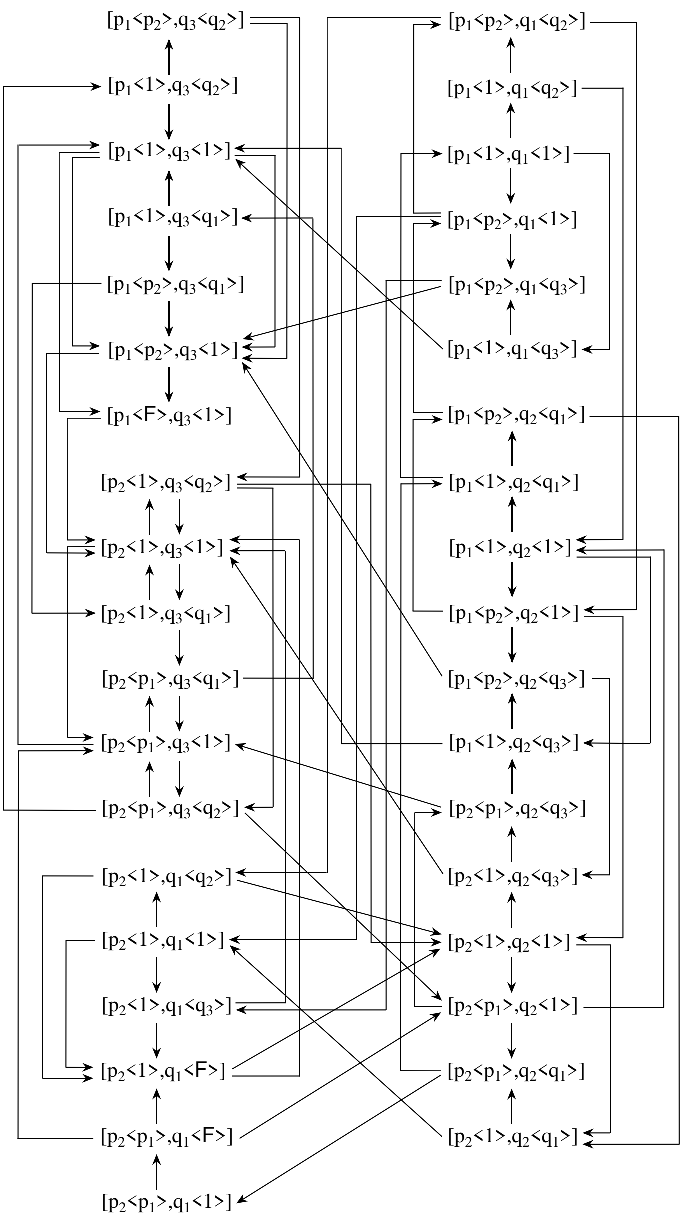


Abbildung 9.1: $VG(IS_1)\lfloor_{\{b_1,b_2\}}$

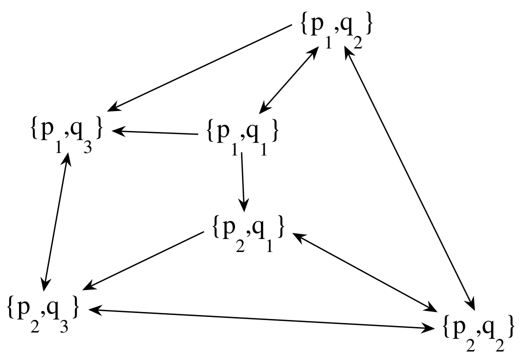


Abbildung 9.2: $CG(IS_1)\lfloor_{\{b_1,b_2\}}$

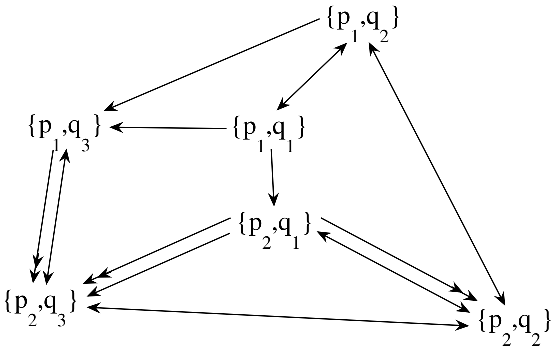


Abbildung 9.3: $ECG(IS_1)\lfloor_{\{b_1,b_2\}}$

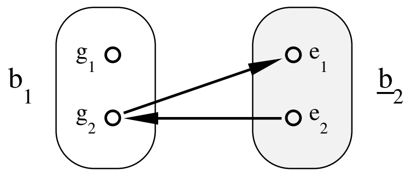


Abbildung 9.4: IS_6

Abschließend soll deutlich gemacht werden, dass, wenn eine wie oben beschriebene Ungleichheit/Gleichheit bei Verwendung einer Trace-Semantik vorliegt, dann gilt diese Eigenschaft nicht automatisch auch für die gleichnamigen Zustandsgraphen (und andersherum). Betrachtet man z.B. das I–System IS_6 aus Abbildung 9.4, dann gilt unter Verwendung der Casetrace-Semantik:

$$\begin{aligned}\mathcal{CT}[\![IS_6]\!]\lfloor_{\{b_1\}} = \mathcal{CT}^{\mathrm{i}}[\![IS_6]\!]\lfloor_{\{b_1\}} &= \{\{g_1\},\{g_1\}\{g_2\}\{g_1\},\{g_1\}\{g_2\}\{g_1\}\{g_2\}\{g_1\},\ldots\}\cup\\ &\quad \{\{g_2\}\{g_1\},\{g_2\}\{g_1\}\{g_2\}\{g_1\},\{g_2\}\{g_1\}\{g_2\}\{g_1\}\{g_2\}\{g_1\},\ldots\}\\ \mathcal{CT}[\![IS_3\lfloor_{\{b_1\}}]\!] = \mathcal{CT}^{\mathrm{i}}[\![IS_3\lfloor_{\{b_1\}}]\!] &= \{\{g_1\},\{g_1\}\{g_2\},\{g_1\}\{g_2\}\{g_1\},\{g_1\}\{g_2\}\{g_1\}\{g_2\},\ldots\}\cup\\ &\quad \{\{g_2\},\{g_2\}\{g_1\},\{g_2\}\{g_1\}\{g_2\},\{g_2\}\{g_1\}\{g_2\}\{g_1\},\ldots\}\end{aligned}$$

Offensichtlich gilt die Ungleichung $\mathcal{CT}[\![IS_6]\!]\lfloor_{\{b_1\}} \neq \mathcal{CT}[\![IS_3\lfloor_{\{b_1\}}]\!]$.
Demgegenüber liegt bei der Betrachtung der beiden entsprechenden Casegraphen (der erste mit Sicht auf b_1) eine Gleichheit vor, denn:
$CG(IS_6)\lfloor_{\{b_1\}} = (\{\{g_1\},\{g_2\}\},\{(\{g_1\},\{g_2\}),(\{g_1\},\{g_2\})\}) = CG(IS_6\lfloor_{\{b_1\}})$.

In Anlehnung an die Diskussionen in Kapitel 6 zeigt das Beispiel mit IS_6 erneut, dass es notwendig ist, zwischen der Ausdruckskraft einer Trace-Semantik und der Ausdruckskraft des Zustandsgraphens, der sich aus der Interleaving Variante der Trace-Semantik ableitet, zu unterscheiden. Die Einführung von Sichten, mit dem Ziel der Fokussierung innerhalb einer Semantik eines I–Systems auf eine augezeichnete Menge von Bereichen, macht die Unterscheidung nicht hinfällig.

9.7 Beispiel: Realisierung lokaler Ereignisstrukturen

Anhand eines Beispiels soll in diesem Abschnitt der Einsatz von Sichten auf Relevanzbereiche verdeutlicht werden. Das Beispiel wird in vereinfachender formaler Form auch in [89] betrachtet. Während dort verstärkt auf die Motivation eingegangen wird, liegt der Schwerpunkt hier in der formalen Ausarbeitung.

In [89] wurde diskutiert, dass jede der interagierenden Komponenten eines verteilten Systems einen bestimmten Grad an Autonomie besitzt. Ist dieser gleich Null, dann spricht man auch von reaktiven Komponenten (vgl. Kapitel 1.1). Neben der Interaktion mit anderen Komponenten

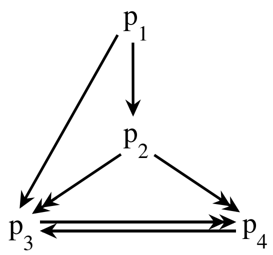


Abbildung 9.5: $G_1(b_1)$; einspitzige Pfeile repräsentieren $\rightarrow_1$,
zweispitzige Pfeile repräsentieren $\rightarrow_2$.

und daraus resultierenden externen Einflüssen kann das Verhalten einer Komponente einem bestimmten internen *organisatorischen Ablaufplan (organizational scheme)* unterliegen. Solch ein Plan legt fest, wie sich die Komponente in bestimmten Situationen verhalten kann oder muss und welches die möglichen Folgesituationen sind. In Anlehnung an organisatorische Abläufe in der (menschlichen) Verwaltungshierarchie lassen sich zwei Situationsarten beschreiben. Zum einen gibt es *Entscheidungssituationen (decision situations)*, die jeweils zu *autonomen Aktionen* führen, welche nicht durch irgendwelche Einflüsse oder Vorschriften hervorgerufen sind. Des Weiteren existieren *Zwangssituationen (procedural situations)*, aus denen sich ausschließlich, aufgrund von organisatorischen Pflichten, *organisatorisch erzwungene Aktionen* ergeben.
Formal kann ein organisatorischer Ablaufplan in einer Komponente eines verteilten Systems durch einen gerichteten Graphen dargestellt werden, dessen Knoten die Zustände/Phasen der Komponente sind, und der zwei Typen von Kanten besitzt (siehe z.B. Abbildung 9.5). Ein Kantentyp spezifiziert (autonome) Zustandsübergänge heraus aus Entscheidungssituationen und ein Kantentyp spezifiziert (organisatorisch erzwungene) Zustandsübergänge heraus aus Zwangssituationen.

Für I–Systeme stellt sich jetzt die Frage, ob man organisatorische Ablaufpläne mit ihnen modellieren kann, d.h. ob man Ereignisse in einem Relevanzbereich b_1 eines I–Systems IS durch eine spezielle Konstruktion (von Nachbarbereichen, Kopplungs- und Erregungsrelation) derart steuern kann, dass zum einen die möglichen Zielphasen und zum anderen die Typen von Phasentransitionen (frei oder erzwungen) vorgegeben werden können. Das Einhalten der Vorgaben muss dann in der Semantik von IS mit Sicht auf b_1 ablesbar sein. Der folgende Satz zeigt, dass die Modellierung eines organisatorischen Ablaufplanes in Form einer lokalen Ereignisstruktur in b_1 möglich ist. Dazu sind nur ein weiterer träger Bereich $\underline{b}_2$, der nur eine Phase mehr als b_1 beinhaltet, und eine geeignete Festlegung der Kopplungs- und Erregungsrelation erforderlich.
Da in diesem Beispiel die Möglichkeit der Ausführung von Phasentransitionen und deren Klassifizierung in frei und erzwungen von Interesse ist, werden die folgenden Betrachtungen auf die Erweiterte Casetrace-Semantik eines I–Systems IS und deren Sicht auf einen Bereich b_1 bezogen.

Satz 9.20 (Lokale Ereignisstruktur). Sei $IS = (P, B, \underline{B}, K, E)$ ein I–System mit $P = \{p_1, p_2, \ldots, p_m, e_0, e_1, e_2, \ldots, e_m\}$, $m \in I\!N$, $B = \{b_1, \underline{b}_2\}$, $\underline{B} = \{\underline{b}_2\}$, $b_1 = \{p_1, p_2, \ldots, p_m\}$, $\underline{b}_2 = \{e_0, e_1, e_2, \ldots, e_m\}$.
Sei $G(b_1) = (b_1, \rightarrow_1, \rightarrow_2)$ eine gerichtete schlingenfreie Graphenstruktur auf b_1 mit zwei Kantenmengen und $\{p \in b_1 \mid \rightarrow_1(p) \neq \emptyset \wedge \rightarrow_2(p) \neq \emptyset\} = \emptyset$.
Es gelten die folgenden strukturellen Voraussetzungen für $j = 1, 2, \ldots, m$:

i) $K(e_j) \setminus \underline{b}_2 = \{p \in b_1 \setminus \{p_j\} \mid (p_j, p) \notin \rightarrow_1 \cup \rightarrow_2\}$

ii) $E^{-1}(e_j) = \{p \in b_1 \mid (p_j, p) \in \rightarrow_1 \cup \rightarrow_2\}$

iii) $E(e_j) = \begin{cases} \{p_j\} \text{ falls } \rightarrow_2(p_j) \neq \emptyset \\ \emptyset \text{ sonst} \end{cases}$

iv) $K(e_0) \setminus \underline{b}_2 = \emptyset$

v) $E^{-1}(e_0) = b_1$

vi) $E(e_0) = \emptyset$

Dann gilt:

a) $\mathcal{ECT}[\![IS]\!]\lfloor_{\{b_1\}} = \{\{p'_0\}\delta_1\{p'_1\}\delta_2\{p'_2\}\delta_3\ldots\infty \mid \langle p'_0, p'_1, p'_2, \ldots\infty\rangle$ ist unendlicher Pfad in $G(b_1)$ und $\forall i \in I\!N : (\delta_i = \{b_1\} \wedge (p'_{i-1}, p'_i) \in\to_1) \vee (\delta_i = \emptyset \wedge (p'_{i-1}, p'_i) \in\to_2)\} \cup \{\{p'_0\}\delta_1\{p'_1\}\delta_2\{p'_2\}\delta_3\ldots\delta_n\{p'_n\} \mid \langle p'_0, p'_1, p'_2, \ldots, p'_n\rangle$ mit $n \in I\!N$ ist endlicher Pfad in $G(b_1)$ und $(\forall i \in \{1, \ldots, n\} : (\delta_i = \{b_1\} \wedge (p'_{i-1}, p'_i) \in\to_1) \vee (\delta_i = \emptyset \wedge (p'_{i-1}, p'_i) \in\to_2)) \wedge (\to_2(p'_n) = \emptyset)\}$

b) $ECG(IS)\lfloor_{\{b_1\}} = (\{\{p\} \mid p \in b_1\}, \{(\{p\}, \{p'\}) \mid (p, p') \in\to_1\}, \{(\{p\}, \{p'\}) \mid (p, p') \in\to_2\})$

Die Aussage a) drückt aus, dass sich die Erweiterte Casetrace-Semantik von IS mit Sicht auf b_1 genau in den Pfaden im organisatorischen Ablaufplan $G(b_1)$ widerspiegelt, wenn zusätzlich die Kantentypen berücksichtigt werden. Die Aussage b) bedeutet, dass der Erweiterte Casegraph von IS mit Sicht auf b_1 unter Beachtung der unterschiedlichen Syntax dem organisatorischen Ablaufplan entspricht.

Beweis.
Zu **a), $\subseteq$:** Betrachtet wird eine beliebige Ausführung Π von $V_I System(IS)$ mit den möglichen Ausführungsalternativen. Das Ziel ist es, alle auftretenden Cases (abgeleitet aus den z-Globalbelegungen) sowie bei Caseübergängen die zuständigen Aktionen (A4 oder A5) zu bestimmen. Aus diesen Informationen lässt sich dann die Erweiterte Casetrace-Semantik $\mathcal{ECT}[\![IS]\!]$ zusammensetzen. Beim Übergang zu $\mathcal{ECT}[\![IS]\!]\lfloor_{\{b_1\}}$ ergeben sich schließlich Phasenabfolgen, die genau den Pfaden in $G(b_1)$ entsprechen. Weiterhin korrespondieren die Bereichsmengen innerhalb von $\mathcal{ECT}[\![IS]\!]\lfloor_{\{b_1\}}$ zu den Kantentypen bei $G(b_1)$.
Zu **a), $\supseteq$:** Zu einem beliebigen Pfad im Graphen $G(b_1)$ lässt sich eine passende Ausführung Π' von $V_I System(IS)$ konstruieren, bei der die nacheinander auftretenden unterschiedlichen Cases mit Sicht auf b_1 mit den Knoten des Pfades in der Abfolge übereinstimmen und bei der die Abarbeitung der Aktionen A4 und A5 mit den Kantentypen bei $G(b_1)$ verträglich ist.
Zu **b):** Es wird der rechnerische Zusammenhang zwischen $ECG(IS)\lfloor_{\{b_1\}}$ und $\mathcal{ECT}[\![IS]\!]\lfloor_{\{b_1\}}$ aufgezeigt, um darauf aufbauend die Satzaussage a) anzuwenden. Die Ausführungen von $V_I System(IS)$ brauchen dadurch nicht noch einmal analysiert zu werden.

Der vollständige Beweis befindet sich im Anhang A.3 (Seite 177 ff.).

Der Beweis macht noch einmal unterschiedliche Beweisstrategien im Umgang mit I–Systemen und mit deren semantischen Beschreibungsformen (Trace-Semantiken, Zustandsgraphen) deutlich. Zum Beweis der Satzaussage a) ist es erforderlich, sich auf die Ausführungen des zugeordneten $\mathrm{V_I}$Systems zu beziehen, d.h. man bewegt sich zum einen auf der formalen, als auch auf der algorithmischen Modellierungsebene (siehe Kapitel 7.1). Bei der $\subseteq$-Richtung müssen *alle* Möglichkeiten einer Ausführung Π von $V_I System(IS)$ betrachtet werden. Bei der $\supseteq$-Richtung reicht es, *eine* konkrete Ausführung Π' von $V_I System(IS)$ anzugeben. Der Beweis der Satzaussage b) findet ausschließlich auf der formalen Modellierungsebene statt. Bewiesene Eigenschaften einer semantischen Beschreibungsform für I-Systeme (hier: der Erweiterten Casetrace-Semantik von IS mit Sicht auf b_1) werden auf eine andere semantische Beschreibungsform (hier: auf den Erweiterten Casegraphen von IS mit Sicht auf b_1) übertragen, ohne dass auf die Ausführungen von $V_I System(IS)$ eingegangen werden muss. □

Die grundlegenden Konstruktionsideen von IS, aus denen sich die Satzvoraussetzungen i)-vi) begründen, zeigen sich an den Ausführungen von $V_I System(IS)$. Denkt man sich in einem ersten Schritt unter Punkt iii) uneingeschränkt $E(e_j) = \emptyset$, dann bewirken die restlichen Punkte die Induktion eines Zustands-/Transitionsgraphens in b_1, d.h. die möglichen Phasentransitionen werden durch $G(b_1)$ vorgegeben, und alle Phasentransitionen werden dabei als „frei“ angenommen. Eine anschauliche Interpretation der Ausführungen von $V_I System(IS)$ lässt sich wie folgt zusammenfassen: Wenn $V_{\underline{b}_2}$ in e_i ist, für $i \in \{1, \ldots, m\}$, dann muss V_{b_1} in p_i oder in einer Nachfolgephase (bzgl. $G(b_1)$) von p_i sein. Ist $V_{\underline{b}_2}$ in e_0, dann kann V_{b_1} in irgendeiner seiner Phasen sein. Wenn $V_{\underline{b}_2}$ in e_i ist, für $i \in \{0, \ldots, m\}$, und V_{b_1} ist in p_j mit $i \neq j$, dann übt V_{b_1} einen Einfluss auf $V_{\underline{b}_2}$ zur Einnahme von e_j aus. V_{b_1} muss so lange in p_j bleiben, wie $V_{\underline{b}_2}$ nicht in e_j ist. Wenn V_{b_1} in p_i und $V_{\underline{b}_2}$ in e_i sind, dann kann V_{b_1} eine Entscheidung zu einer Phasentransition hin zu einer Phase q treffen. Ist q eine Nachfolgephase von p_i, dann wird V_{b_1} schließlich nach q wechseln.

Ist q keine Nachfolgephase von p_i, dann wird dadurch ein Einfluss auf $V_{\underline{b}_2}$ wirksam, der $V_{\underline{b}_2}$ dazu bewegt, eine der Phasen aus $\{e_j \mid p_i$ ist eine Nachfolgephase von $e_j\} \cup \{e_0\}$ einzunehmen. Nach der Phasentransition ist p_i eine erregende Phase, und V_{b_1} wird stabil. Somit können „falsche" Entscheidungen von V_{b_1} nicht zu ungewollten Phasentransitionen in V_{b_1} führen.
Durch die Erweiterung der Erregungsrelation in einem zweiten Schritt, speziell die Präzisierung der Menge $E(e_j)$ unter Punkt iii) der Satzvoraussetungen, wird der Unterscheidung zwischen Entscheidungs- und Zwangssituationen im Rahmen der Modellierung eines organisatorischen Ablaufplans Rechnung getragen. Wenn V_{b_1} in einer Phase p_i ist, die eine Zwangssituation repräsentiert, genau dann wird ein zusätzlicher Einfluss benötigt, aufgrund dessen V_{b_1} gezwungen wird, p_i zu verlassen. Realisiert wird dieser Einfluss durch die Erweiterung der Erregungsrelation um (e_i, p_i). Ist V_{b_1} in p_i, dann wird $V_{\underline{b}_2}$ schließlich in e_i eintreten (wie bisher) und aufgrund der hinzugekommenen Beziehung folgt eine Erregung von p_i durch e_i, worauf V_{b_1} instabil wird und schließlich eine (erzwungene) Phasentransition von p_i zu einer Nachbarphase von p_i durchführt.
Die Satzvoraussetzung $\{p \in b_1 \mid \rightarrow_1(p) \neq \emptyset \wedge \rightarrow_2(p) \neq \emptyset\} = \emptyset$ setzt die Eigenschaft eines organisatorischen Ablaufplans um, dass Entscheidungssituationen ausschließlich zu autonomen, und Zwangssituationen ausschließlich zu organisatorisch erzwungenen Aktionen führen. In der Einleitung zu diesem Abschnitt wurde diese Betrachtungsweise motiviert.

Die Anzahl der Phasen in $\underline{b}_2$ ist genau um 1 höher als die Anzahl der Phasen in b_1, d.h. $|\underline{b}_2| = |b_1| + 1$. Dieses Verhältnis ist unabhängig von der Mächtigkeit der beiden Kantenmengen von $G(b_1)$, oder anders ausgedrückt, von der Komplexität der Verzweigungsstruktur des zu modellierenden organisatorischen Ablaufplans. Eine obere Schranke für die Gesamtanzahl der Elemente der Kopplungs- und Erregungsrelation, also der benötigten elementaren Interaktionsbeziehungen, liegt bei $O(|K| + |E|) = (O(K(e_0) \setminus \underline{b}_2) + O(|E^{-1}(e_0)|) + O(|E(e_0)|)) + \sum_{j=1}^{m} (O(K(e_j) \setminus \underline{b}_2) + O(|E^{-1}(e_j)|) + O(|E(e_j)|)) = O(m + m * m) = O(|b_1| + |b_1|^2)$.

Infolge der Verwendung der Erweiterten Casetrace-Semantik von IS (mit Sicht auf $\{b_1\}$) in der Satzaussage a) und des Erweiterten Casegraphen von IS (mit Sicht auf $\{b_1\}$) in Satzaussage b), besitzen beide Aussagen einen unterschiedlichen Aussagegehalt in Bezug auf die Ausführungen des zugeordneten $\mathrm{V_I}$Systems $V_I System(IS)$.
Berücksichtigt man die Definition 4.16 der Erweiterten Casetrace-Semantik, dann bringt die Aussage a) zum Ausdruck, dass $\langle p'_0, p'_1, p'_2, \ldots \rangle$ genau dann ein Pfad in $G(b_1)$ ist, wenn es eine Ausführung von $V_I System(IS)$ gibt, in der nur die Phasentransitionen $p'_0 \rightarrow p'_1, p'_1 \rightarrow p'_2, p'_2 \rightarrow p'_3, \ldots$ in genau dieser Reihenfolge auftreten. Der Kantentyp einer Kante des Pfades vermittelt dabei die Ursache der korrespondierenden Phasentransition, d.h. den Namen der Aktion (A4 oder A5), bei der die Phasentransition auftritt. Folglich lässt sich die Erweiterte Casetrace-Semantik von IS mit Sicht auf $\{b_1\}$ aus der Graphenstruktur $G(b_1)$ ableiten.
Die Aussage b) ist eine Abschwächung von Aussage a). Gemäß der Definition 6.10 des Erweiterten Casegraphen ist eine Kante (p'_1, p'_2) genau dann in $G(b_1)$ enthalten, wenn es irgendeine Ausführung von $V_I System(IS)$ gibt, in der eine Phasentransition $p'_1 \rightarrow p'_2$ in V_{b_1} vorkommt. Dabei vermittelt wiederum der Kantentyp die zur Phasentransition gehörige Aktion. Folglich lässt sich der Erweiterte Casegraph von IS mit Sicht auf $\{b_1\}$ aus der Graphenstruktur $G(b_1)$ ableiten.
Es ist offensichtlich, dass b) aus a) folgt. Der Schluss von b) auf a) gilt hingegen nicht, da sich die Erweiterte Casetrace-Semantik eines I–Systems im Allgemeinen nicht aus dem Erweiterten Casegraphen ableiten läßt. In Kapitel 6 wurde diese Thematik behandelt. Die Betrachtung von Sichten ändert nichts an dieser Problematik.
In den bisherigen Arbeiten, in denen formale Ansätze zur Modellierung lokaler Ereignisstrukturen bei I–Systemen behandelt werden ([14, 89, 90]), werden Aussagen von der Aussagekraft von Satz 9.20.b) gemacht. Die Verschärfung durch die Betrachtung der Erweiterten Casetrace-Semantik mit Aussage a) erlaubt es, die Korrektheit von IS tiefergehend zu zeigen. Für das allgemeine Verständnis der Konstruktion von IS ist es wünschenswert, dass Pfade in $G(b_1)$, die ja einen Prozess im Rahmen eines organisatorischen Ablaufplans repräsentieren, anhand einer fortlaufenden Anwendung der Aktionen A1-A13 nachvollzogen werden können. Es scheint nicht sinnvoll, dass sich ein Pfad in $G(b_1)$ nur aus der Kombination von zwei oder mehreren unterschiedlichen Ausführungen von $V_I System(IS)$ begründen lässt. Durch den Beweis der Aussage a) wurde gezeigt, dass dem Wunsch nach einer fortlaufenden Anwendung von A1-A13 entsprochen wird,

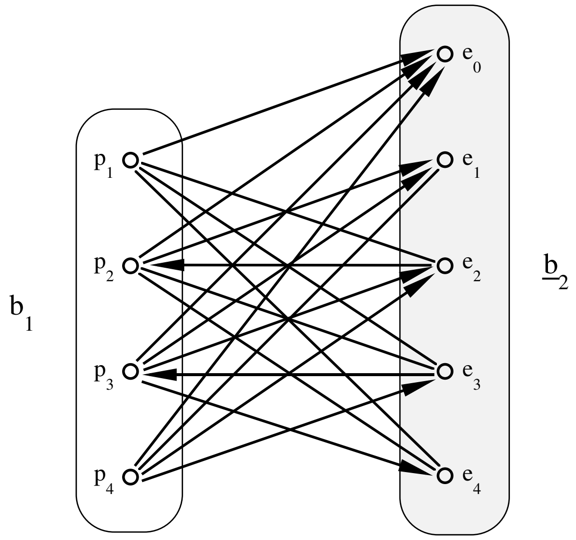


Abbildung 9.6: IS_7; Induktion von $G_1(b_1)$ in b_1

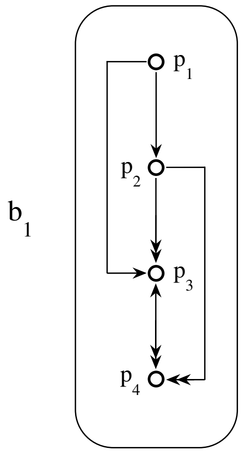


Abbildung 9.7: Alternative Darstellung von IS_7

d.h. dass zu jedem Pfad in $G(b_1)$ eine Ausführung von $V_I System(IS)$ existiert, so dass sich die Knotenfolge des Pfades als Knotenfolge einer Sequenz von Phasentransitionen in der Komponente V_{b_1} wiederfindet. Dabei wird die Zuordnung von Kantentyp ($\rightarrow_1/\rightarrow_2$) zu Phasentransitionsursache (Aktion A4/A5) durchgehend beachtet.

Beispiel 9.21. Gegeben sei der Graph $G_1(b_1)$ aus Abbildung 9.5 mit $\rightarrow_1 = \{(p_1, p_2), (p_1, p_3), (p_4, p_3)\}$ und $\rightarrow_2 = \{(p_2, p_3), (p_2, p_4), (p_3, p_4)\}$. Der Graph repräsentiert einen organisatorischen Ablaufplan mit p_1, p_4 als Entscheidungssituationen und p_2, p_3 als Zwangssituationen. Das I–System IS_7 in Abbildung 9.6 erfüllt die in Satz 9.20 geforderten strukturellen Voraussetzungen. Die semantischen Gleichheiten a) und b) in dem Satz sind somit für IS_7 und $G_1(b_1)$ garantiert. □

Um lokale Ereignisstrukturen, die sich aus organisatorischen Ablaufplänen ableiten, innerhalb eines I–Systems leicht und anschaulich ablesen zu können, wird die bisherige graphische Darstellung der I–Systeme erweitert.

Graphische Darstellung. Liegt bei einem I–System ein Teilsystem aus zwei Bereichen vor, das die Voraussetzungen aus Satz 9.20 erfüllt, wird für dieses Teilsystem nur der autonome Bereich übernommen und die vorgegebene Graphenstruktur in den Bereich eingezeichnet. Entgegengesetzte Pfeile können übereinander gelegt werden. Abbildung 9.7 zeigt die graphische Darstellung des I–Systems IS_7 aus Beispiel 9.21.

Satz 9.20 ist gültig für *jede* schlingenfreie Graphenstruktur $G(b_1)$, die einen organisatorischen Ablaufplan über b_1 repräsentiert. In vielen Anwendungsbeispielen tritt ein *spezieller Tpy* von Graphenstruktur auf, bei dem von jedem Knoten vorausgesetzt werden kann, dass dieser mindestens einen Vorgängerknoten besitzt [14, 85, 89, 91]. Ist diese Einschränkung von $G(b_1)$ gewährleistet, kann die Konstruktion von IS in Satz 9.20 vereinfacht werden, derart, dass die Phase e_0 und alle Elemente der Erregungsrelation, in denen e_0 enthalten ist, entfallen. Die Satzaussagen a) und b) bleiben dabei erhalten.

Satz 9.22 (Spezielle lokale Ereignisstruktur). Sei $IS = (P, B, \underline{B}, K, E)$ ein I–System mit $P = \{p_1, p_2, \dots, p_m, e_1, e_2, \dots, e_m\}$, $m \in I\!N$, $B = \{b_1, \underline{b}_2\}$, $\underline{B} = \{\underline{b}_2\}$, $b_1 = \{p_1, p_2, \dots, p_m\}$, $\underline{b}_2 = \{e_1, e_2, \dots, e_m\}$.
Sei $G(b_1) = (b_1, \rightarrow_1, \rightarrow_2)$ eine gerichtete schlingenfreie Graphenstruktur auf b_1 mit zwei Kantenmengen und $\{p \in b_1 \mid \rightarrow_1(p) \neq \emptyset \wedge \rightarrow_2(p) \neq \emptyset\} = \emptyset$. Jeder Knoten von $G(b_1)$ habe mindestens einen Vorgängerknoten.
Es gelten die folgenden strukturellen Voraussetzungen für $j = 1, 2, \dots, m$:

i) $K(e_j) \setminus \underline{b}_2 = \{p \in b_1 \setminus \{p_j\} \mid (p_j, p) \notin \rightarrow_1 \cup \rightarrow_2\}$

ii) $E^{-1}(e_j) = \{p \in b_1 \mid (p_j, p) \in \rightarrow_1 \cup \rightarrow_2\}$

iii) $E(e_j) = \begin{cases} \{p_j\} \text{ falls } \rightarrow_2(p_j) \neq \emptyset \\ \emptyset \text{ sonst} \end{cases}$

Dann gilt:

a) $\mathcal{ECT}[\![IS]\!]\downarrow_{\{b_1\}} = \{\{p'_0\}\delta_1\{p'_1\}\delta_2\{p'_2\}\delta_3 \dots \infty \mid \langle p'_0, p'_1, p'_2, \dots \infty\rangle$ ist unendlicher Pfad in $G(b_1)$ und $\forall i \in I\!N : (\delta_i = \{b_1\} \wedge (p'_{i-1}, p'_i) \in\rightarrow_1) \vee (\delta_i = \emptyset \wedge (p'_{i-1}, p'_i) \in\rightarrow_2)\} \cup \{\{p'_0\}\delta_1\{p'_1\}\delta_2\{p'_2\}\delta_3 \dots \delta_n\{p'_n\} \mid \langle p'_0, p'_1, p'_2, \dots, p'_n\rangle$ mit $n \in I\!N$ ist endlicher Pfad in $G(b_1)$ und $(\forall i \in \{1, \dots, n\} : (\delta_i = \{b_1\} \wedge (p'_{i-1}, p'_i) \in\rightarrow_1) \vee (\delta_i = \emptyset \wedge (p'_{i-1}, p'_i) \in\rightarrow_2)) \wedge (\rightarrow_2(p'_n) = \emptyset)\}$

b) $ECG(IS)\downarrow_{\{b_1\}} = (\{\{p\} \mid p \in b_1\}, \{(\{p\}, \{p'\}) \mid (p, p') \in\rightarrow_1\}, \{(\{p\}, \{p'\}) \mid (p, p') \in\rightarrow_2\})$

Die Bedeutungen der Satzaussagen (in Worten) entsprechen denen von Satz 20, d.h. die Aussage a) drückt aus, dass sich die Erweiterte Casetrace-Semantik von IS mit Sicht auf b_1 genau in den Pfaden im organisatorischen Ablaufplan $G(b_1)$ widerspiegelt, wenn zusätzlich die Kantentypen berücksichtigt werden. Die Aussage b) bedeutet, dass der Erweiterte Casegraph von IS mit Sicht auf b_1 unter Beachtung der unterschiedlichen Syntax dem organisatorischen Ablaufplan entspricht.

Beweis. Der Beweis von **a)** und **b)** verläuft analog zum Beweis von Satz 9.20. Die Phase e_0 wird dabei aus allen Mengen, in denen sie enthalten ist, herausgenommen. Cases mit e_0 werden nicht betrachtet. Im Rahmen der Analyse bzw. Konstruktion von Ausführungen von $V_ISystem(IS)$ werden Variablenzuweisungen/-anpassungen der Variablen $\_e_{in}(e_0)$, $\_e_{out}(e_0)$, $\_k(e_0)$, $\_z(e_0)$, usw. nicht aufgeführt und erläutert. Indexbereiche werden angepasst.

Der vollständige Beweis befindet sich im Anhang A.3 (Seite 192 ff.).

In dem Beweis wird ebenfalls deutlich gemacht, warum die Satzvoraussetzung „Jeder Knoten von $G(b_1)$ habe mindestens einen Vorgängerknoten." notwendig ist. □

Die Anzahl der Phasen in $\underline{b}_2$ von IS in Satz 9.22 stimmt mit der Anzahl der Phasen in b_1 überein, d.h. $|\underline{b}_2| = |b_1|$. Dieses Verhältnis ist, wie schon bei Satz 9.20, unabhängig von der Mächtigkeit der beiden Kantenmengen von $G(b_1)$. Eine obere Schranke für die Gesamtanzahl der Elemente der Kopplungs- und Erregungsrelation liegt bei

$$O(|K| + |E|) = \sum_{j=1}^{m} (O(K(e_j) \setminus \underline{b}_2) + O(|E^{-1}(e_j)|) + O(|E(e_j)|)) = O(m * m) = O(|b_1|^2).$$

Es ist klar, dass Satz 9.22 insbesondere dann gilt, wenn es sich bei $G(b_1)$ um eine zyklische schlingenfreie Graphenstruktur handelt, d.h. bei der jeder Knoten zu mindestens einem Zyklus (unabhängig vom Kantentyp) gehört. Diese Zykluseigenschaft ist aber kein notwendiges Kriterium.

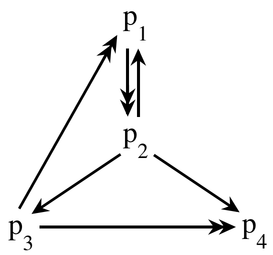


Abbildung 9.8: $G_2(b_1)$

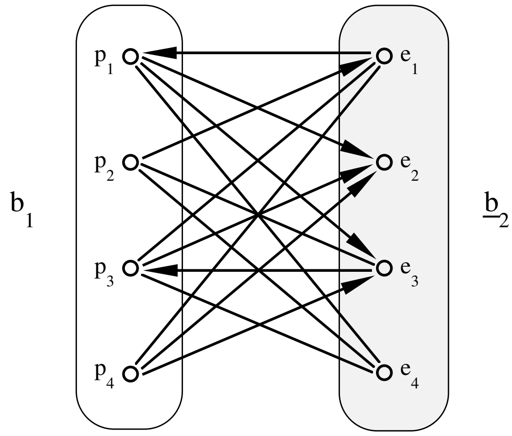


Abbildung 9.9: IS_8; Induktion von $G_2(b_1)$ in b_1

Beispiel 9.23. Man betrachte die (nicht zyklische) Graphenstruktur $G_2(b_1)$ aus Abbildung 9.8 mit $\rightarrow_1 = \{(p_2, p_1), (p_2, p_3), (p_2, p_4)\}$ und $\rightarrow_2 = \{(p_1, p_2), (p_3, p_1), (p_3, p_4)\}$. Jeder Knoten des Graphens hat mindestens einen Vorgängerknoten. Das I–System IS_8 in Abbildung 9.9 erfüllt die in Satz 9.22 geforderten strukturellen Voraussetzungen. Die semantischen Gleichheiten a) und b) in dem Satz sind somit für IS_8 und $G_2(b_1)$ garantiert.
Natürlich kann eine Induktion von $G_2(b_1)$ in b_1 auch über Satz 9.20 erfolgen, der auf jede schlingenfreie Graphenstruktur, die einen organisatorischen Ablaufplan repräsentiert, anwendbar ist. Das entsprechende I–System IS_9 ist in Abbildung 9.10 dargestellt. Es beinhaltet die zusätzliche „überflüssige“ Phase e_0 und die zusätzlichen „überflüssigen“ Elemente der Erregungsrelation (p_i, e_0), $i = 1, \dots, 4$. Als Folge der Sätze 9.20 und 9.22 sind die Erweiterten Casetrace-Semantiken mit Sicht auf $\{b_1\}$ und die Erweiterten Casegraphen mit Sicht auf $\{b_1\}$ von IS_8 und IS_9 identisch, obwohl IS_8 und IS_9 selbst unterschiedlich sind. Diese Beobachtung motiviert die Definition von Äquivalenzen auf I–Systemen. In Kapitel 11 wird dieses Thema behandelt.
Abbildung 9.11 zeigt die alternative Darstellung von sowohl IS_8 als auch von IS_9 mit der in b_1 eingezeichneten Graphenstruktur. Von der Art der Konstruktion wird bei dieser Darstellung abstrahiert. □

Die früheren Arbeiten, in denen formale Ansätze zur Erzeugung lokaler Ereignisstrukturen bei I–Systemen behandelt werden ([14, 90]), geben bei der Konstruktion von IS gemäß Satz 9.22 keine Strukturvoraussetzungen bzgl. einer „Knoten-Vorgängereigenschaft“ für $G(b_1)$ vor. Die Notwendigkeit solch einer Eigenschaft bei $G(b_1)$ für die Korrektheit von IS wurde in dem Beweis von Satz 9.22 deutlich gemacht. Hierbei hat sich die in dieser Arbeit eingeführte und praktizierte Modellierungsmethodik bei I–Systemen ausgezahlt. (In Kapitel 7 wurde diese Methodik beschrieben.) Korrektheitsbeweise, die sich auf die Dynamik eines I–Systems und damit auf die Ausführungen des zugeordneten $\mathrm{V_I}$Systems abstützen, lassen sich klar strukturieren und systematisch durchführen. Dadurch wird die Fehleranfälligkeit und die Gefahr von Unvollständigkeit bei der Beweisführung verringert.

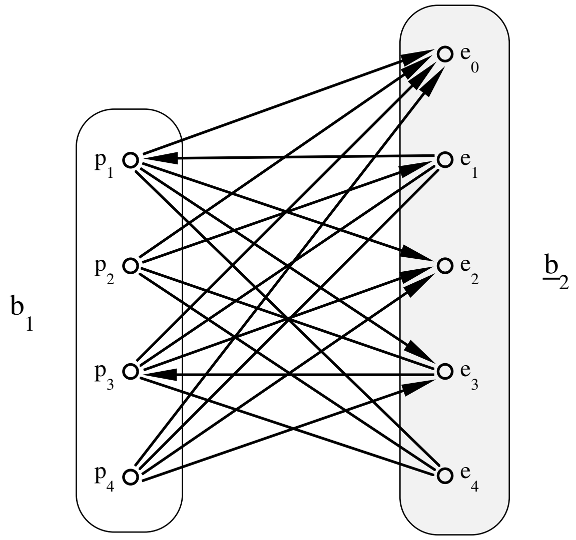


Abbildung 9.10: IS_9; Alternative Induktion von $G_2(b_1)$ in b_1

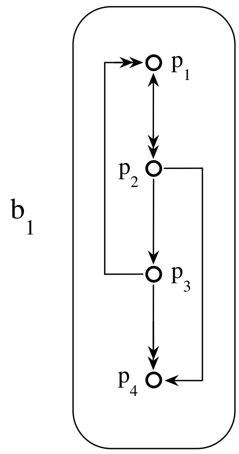


Abbildung 9.11: Alternative Darstellung von IS_8 und von IS_9

Kapitel 10

Strukturbausteine

Modularität ist ein zentrales Konzept zur Bewältigung der Komplexität, die mit dem Entwurf und der Analyse großer verteilter Systeme (wie z.B. das Leitsystem für Fußgänger im Straßenverkehr aus Kapitel 1.1) verbunden ist. Von daher sollte ein formales Modell erlauben, charakteristische Teilstrukturen zu identifizieren und zu analysieren, um danach die Teilergebnisse zur Formulierung von Globalaussagen zusammenzusetzen. Um dabei die Korrektheit der Globalaussagen beweisbar zu machen, muss der Einfluss der Teilstrukturen auf deren unmittelbare Umgebung innerhalb des Gesamtsystems erkennbar sein. In Anlehnung an reale Anwendungen, bei denen die systembildenden Komponenten parametrisiert oft mehrmals verwendet werden (z.B. Programmodule, Templates, Hardwareeinheiten), basiert eine effiziente formale Analyse auf einer „geschickten" Unterteilung in gleichartige Teilstrukturen und auf der Bestimmung derer Eigenschaften.
I–Systeme bieten die Möglichkeit, Eigenschaften für verallgemeinerte Teilstrukturen zu formulieren und zu verifizieren. In diesem Kapitel werden so genannte Strukturbausteine für den inkrementellen Entwurf und die modulare Analyse von I–Systemen exemplarisch vorgestellt und Strategien zum Beweis der Korrektheit der Strukturbausteine präsentiert.
Aktuelle Ansätze zum modularen/komponenten-basierten Design und Analyse verteilter (interaktiver) Systeme auf der Basis formaler Spezifikationen außerhalb von I–Systemen liefern z.B. [1, 15, 20, 48]. Generelle Diskussionen über Beziehungen zwischen kompositionellen Strategien, Vollständigkeit und Modularität sowie eine umfassende Liste weiterer Literaturverweise finden sich in [26].

10.1 Elementare Struktureigenschaften

In diesem Abschnitt werden einige Eigenschaften des Verhaltens eines beliebigen I–Systems IS vorgestellt, in denen der lokale Einfluss der Kopplungs- und der Erregungsrelation auf die Entwicklung der Phasenqualitäten zum Ausdruck kommt. Die Aussagen des folgenden Satzes 10.1 ergänzen die Menge der Aussagen von Satz 4.6. Die beschriebenen strukturellen Eigenschaften sind dabei nach Definition 4.7 als *elementar* anzusehen, da zu deren Beweis die Ausführungen des zugeordneten $\mathrm{V_I}$Systems analysiert werden. Der Satz an Eigenschaften aus Satz 4.6 und Satz 10.1 wird im nächsten Abschnitt verwendet, um in geeigneter Kombination Eigenschaften von dort spezifizierten Strukturbausteinen, als modulare Einheiten bei I–Systemen, effizient zu verifizieren.

Satz 10.1 (Elementare Struktureigenschaften). Sei $IS = (P, B, \underline{B}, K, E)$ ein I–System und $z_0 z_1 z_2 \ldots \in \mathcal{V}[\![IS]\!]$ mit $z_j \in GZustand(IS)$, $j = 0, 1, 2, \ldots$.

a) Sei $p \in P$. Dann gilt für alle $i = 1, 2, \ldots$:

$$(z_i(p) = \mathsf{F} \land z_j(p) = 1, i < j) \Rightarrow (E(p) \neq \emptyset \lor K(p) \setminus b(p) \neq \emptyset \lor (\exists k \in I\!N, i < k < j : z_k(p) = \mathsf{0}))$$

b) Sei $p \in P$. Dann gilt für alle $i = 1, 2, \ldots$:

$$(z_i(p) = \mathsf{F}) \Rightarrow (\exists v \in P \setminus b(p) : ((v, p) \in E \land z_i(v) \neq \mathsf{0}) \lor ((v, p) \in K \land (\exists w \in b(v), \exists j \in I\!N, j \leq i : z_j(w) \in \{v, \mathsf{F}\})))$$

c) Sei $p \in P$. Dann gilt für alle $i = 1, 2, \ldots$:

$$(z_i(p) = \mathsf{F}) \Rightarrow (K(p) \setminus b(p) \neq \emptyset \vee E^{-1}(p) \neq \emptyset)$$

d) Seien $p, q, r, v \in P$ mit $p \neq q \neq r \neq p$, $b(p) = b(q) = b(r) \neq b(v)$, $b(p) \in \underline{B}$. Dann gilt für alle $i = 1, 2, \ldots$:

$$((p, v) \in K \wedge K(q) \setminus b(q) = \emptyset = K(r) \setminus b(r)) \Rightarrow ((z_i(v) = \mathsf{F}) \Rightarrow (K(v) \setminus (b(v) \cup b(p)) \neq \emptyset \vee E^{-1}(v) \neq \emptyset))$$

Beweis. Der Beweis wird in gleicher Weise geführt wie der Beweis von Satz 4.6, d.h. es wird eine beliebige Ausführung Π von $V_I System(IS)$, die $z_0 z_1 z_2 \ldots$ gemäß Definition 4.1 erzeugt, analysiert. Alle Möglichkeiten für Π werden soweit per Fallunterscheidungen untersucht, bis die Gültigkeit der Satzaussagen gezeigt ist. Hierzu werden die Aktionen aus Kapitel 3.2.2 systematisch abgearbeitet.

Der vollständige Beweis befindet sich im Anhang A.4 (Seite 194 ff.). □

Die Aussagen des Satzes lassen sich wie folgt interpretieren:

a) präzisiert den *Übergang von Instabilität zu Stabilität.* Ist eine Komponente des $\mathrm{V_I}$Systems instabil in einer Phase p und zu einem späteren Zeitpunkt stabil in der gleichen Phase, dann liegt mindestens einer der drei folgenden Fälle vor: Erstens, p ist in der Lage, eine erregende Phase zu sein. Die Komponente einer erregenden Phase geht nach Satz 4.6.a schließlich in einen stabilen Zustand über. Zweitens, die Propagierung eines Einflusses (gemäß den Erläuterungen zu Satz 4.6.d) wird zurückgenommen. Dieses setzt Phasen in Nachbarkomponenten voraus, die zu p im wechselseitigen Ausschluss stehen. Drittens, die Phase p wird zwischenzeitlich verlassen und dann wieder (stabil) eingenommen.

b) liefert ein *notwendiges Kriterium für Instabilität.* Instabilität einer Komponente in einer Phase p ist die Auswirkung eines Einflusses einer Nachbarkomponente. Der Einfluss kann in Form einer Erregung von p durch eine Nachbarphase auftreten (siehe Satz 4.6.b) oder als Resultat einer Einflusspropagierung entsprechend den Erläuterungen von Satz 4.6.d.

c) ist eine direkte Folgerung aus b). Instabilität in einer Phase p setzt die Möglichkeit der Erregung von p oder die Existenz mindestens einer zu p wechselseitig ausgeschlossenen Nachbarphase voraus.

d) beschreibt, dass aus einer *trägen* Komponente heraus keine Erregung propagiert werden kann, sofern dort Phasen existieren, die *nicht* in wechselseitigem Ausschluss mit anderen Nachbarphasen stehen. Um Instabilität in v zu erwirken sind zusätzliche Kopplungsbeziehungen zwischen v und Phasen außerhalb von $b(p)$ oder aber Erregungsbeziehungen notwendig.

10.2 Abgeleitete Struktureigenschaften

Das Prinzip der Modularisierung bei der formalen Modellierung mit und der Analyse von I–Systemen basiert auf der Beschreibung von verallgemeinerten Teilstrukturen und Bestimmung derer Eigenschaften. Die dabei betrachteten Teilstrukturen bestehen aus einem oder mehreren in der Regel trägen Bereichen, die untereinander über die Kopplungs- und Erregungsrelation verbunden sind. Zur Systemumgebung hin bestehen festgelegte (parametrisierte) Schnittstellen. Einflüsse und Nicht-Einflüsse der Teilstrukturen auf die Systemumgebung werden als Struktureigenschaften formuliert und bewiesen. Im Gegensatz zu den elementaren Struktureigenschaften von I–Systemen aus Abschnitt 10.1, ist es bei den Beweisen der Eigenschaften der Teilstrukturen nicht notwendig, alle Ausführungen der den I–Systemen zugeordneten $\mathrm{V_I}$Systeme zu analysieren. Es reicht der Bezug auf bekannte (bereits bewiesene) elementare Struktureigenschaften, was die Beweisführungen wesentlich vereinfacht. Die Struktureigenschaften der Teilsysteme werden deshalb als *abgeleitet* bezeichnet.

Definition 10.2 (Abgeleitete Struktureigenschaft). Eine strukturelle Eigenschaft einer Semantik eines I–Systems IS, die mit Hilfe von elementaren Eigenschaften bewiesen werden kann, ohne auf die Ausführungen des zugeordneten $\mathrm{V_I}$Systems $V_I System(IS)$ zurückgreifen zu müssen, bezeichnet man als *abgeleitet*. □

Im Folgenden werden exemplarisch einige Teilstrukturen von I–Systemen vorgestellt und die mit ihnen verbundenen (abgeleiteten) Struktureigenschaften einschließlich deren Beweise präsentiert. Solche Teilstrukturen werden auch als *Strukturbausteine* bezeichnet.

Definition 10.3 (Strukturbaustein). Eine möglicherweise parametrisierte Teilstruktur eines I–Systems IS, deren Einflüsse und Nicht-Einflüsse auf die Systemumgebung bekannt sind, d.h. mit deren Vorkommen eine Menge von spezifischen Struktureigenschaften der Semantiken von IS verknüpft ist, bezeichnet man als *Strukturbaustein*. □

Notation 10.4. Strukturbausteine werden im weiteren als Sätze formuliert. Dabei wird die betrachtete Teilstruktur eines I–Systems als Satzvoraussetzungen an die Komponenten des 5-Tupels, durch das das I–System beschrieben wird, angegeben. Eine graphische Skizze der Teilstruktur veranschaulicht die Vorgaben. Die Struktureigenschaften sind als (zu beweisende) Satzaussagen formuliert.

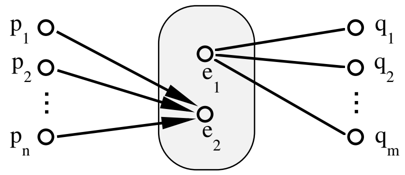
Bei den in den folgenden drei Unterabschnitten vorgestellten Strukturbausteinen werden die aus ihnen resultierenden Struktureigenschaften für das Verhalten $\mathcal{V}[\![\cdot]\!]$, die Casetrace-Semantik $\mathcal{CT}[\![\cdot]\!]$ und die Erweiterte Casetrace-Semantik $\mathcal{ECT}[\![\cdot]\!]$ betrachtet. Anhand dieser drei Semantiken lassen sich die Unterschiede in der Spezifikation von Eigenschaften bei verschiedenen semantischen Grundelementen (globale Aktivitätszustände, Cases, Cases plus Bereichsmengen) aufzeigen. Auf die Interleaving Varianten und Zustandsgraphen wird deshalb nicht weiter eingegangen, zumal sie aus den betrachteten Semantiken leicht abgeleitet werden können (siehe Kapitel 7.2).
Zu jedem Strukturbaustein existiert ein nachfolgendes Korollar, welches die Übersetzung der prädikatenlogisch formulierten Struktureigenschaften in den Wortlaut der Anschauungsebene vornimmt. In Abschnitt 10.3 werden die folgenden Strukturbausteine und Korollare dann verwendet, um anhand eines Beispiels das Prinzip der modularen Analyse bei I–Systemen zu demonstrieren.

10.2.1 Verallgemeinerte Erregung

Der folgende Strukturbaustein (beachte Notation 10.4) verallgemeinert die Quelle und das Ziel der durch die Erregungsrelation eines I–Systems repräsentierten Einflüsse zwischen Systemkomponenten eines verteilten Systems von Phasen auf Mengen von Phasen.

Satz 10.5 (Verallgemeinerte Erregung). Sei $IS = (P, B, \underline{B}, K, E)$ ein I–System mit $p_1, \dots, p_n, e_1, e_2, q_1, \dots, q_m \in P$, pw. versch., $\{e_1, e_2\} \in \underline{B}$, $E(e_2) = \emptyset$, $E^{-1}(e_2) = \{p_1, \dots, p_n\}$, $K(e_2) \setminus \{e_1, e_2\} = \emptyset$, $E(e_1) = \emptyset$, $E^{-1}(e_1) = \emptyset$, $K(e_1) \setminus \{e_1, e_2\} = \{q_1, \dots, q_m\}$ und $n, m \in I\!N$.

Skizze:



Für einen globalen Aktivitätszustand $z \in GZustand(IS)$ sei $M_1^z := \{p \in p_1, \dots p_n \mid z(p) \neq 0\}$ und $M_2^z := \{q \in q_1, \dots q_m \mid z(q) \neq 0\}$. Für einen Case $c \in Case(IS)$ sei $M_1^c := \{p \in p_1, \dots p_n \mid p \in c\}$ und $M_2^c := \{p \in q_1, \dots q_m \mid q \in c\}$.

a) Sei $z_0 z_1 z_2 \dots \in \mathcal{V}[\![IS]\!]$ mit $z_j \in GZustand(IS), j = 0, 1, 2, \dots$. Dann gilt für alle $i = 1, 2, \dots$:

  i) $(M_2^{z_{i-1}} \neq \emptyset) \Rightarrow (M_1^{z_{i-1}} \subseteq M_1^{z_i})$.

ii) $(M_1^{z_{i-1}} \neq \emptyset \wedge M_2^{z_{i-1}} \neq \emptyset) \Rightarrow (\forall q \in M_2^{z_{i-1}} \exists k_q \in I\!N, k_q \geq i-1 : (z_{k_q}(q) \in \{0, \mathsf{F}\} \vee (z_{k_q}(q) = 1 \wedge \exists x \in E(q) : z_{k_q}(x) \neq 0)))$.

iii) $\forall v \in P \backslash \{e_1, e_2, q_1, \ldots, q_m\} : (z_i(v) = \mathsf{F}) \Rightarrow (K(v) \backslash (b(v) \cup \{e_1, e_2\}) \neq \emptyset \vee E^{-1}(v) \backslash \{e_1, e_2\} \neq \emptyset)$.

b) Sei $c_0 c_1 c_2 \ldots \in \mathcal{CT}[\![IS]\!]$ mit $c_j \in Case(IS), j = 0, 1, 2, \ldots$. Dann gilt für alle $i = 1, 2, \ldots$:

$$(M_2^{c_{i-1}} \neq \emptyset) \Rightarrow (M_1^{c_{i-1}} \subseteq M_1^{c_i}).$$

c) Sei $c_0 \delta_1 c_1 \delta_2 c_2 \ldots \in \mathcal{ECT}[\![IS]\!]$ mit $c_j \in Case(IS), \delta_{j+1} \subseteq B, j = 0, 1, 2, \ldots$. Dann gilt für alle $i = 1, 2, \ldots$:

i) $(M_2^{c_{i-1}} \neq \emptyset) \Rightarrow (M_1^{c_{i-1}} \subseteq M_1^{c_i})$.

ii) $\forall v \in P \setminus \{e_1, e_2, q_1, \ldots, q_m\} : (v \in c_{i-1} \setminus c_i \wedge b(v) \notin \delta_i) \Rightarrow (K(v) \setminus (b(v) \cup \{e_1, e_2\}) \neq \emptyset \vee E^{-1}(v) \setminus \{e_1, e_2\} \neq \emptyset)$.

Beweis. Es gelten die Bezeichnungen und Voraussetzungen aus dem Satz. $K^{-1}(e_1)$ und $K^{-1}(e_2)$ sind durch die Angabe von $K(e_1)$ und $K(e_2)$ automatisch festgelegt, da IS ein I–System und damit K eine symmetrische Relation ist. Seien $M_p := \{p_1, \ldots, p_n\}$, $M_e := \{e_1, e_2\}$, $M_q := \{q_1, \ldots, q_m\}$. Sei $i \in \{1, 2, \ldots\}$ beliebig aber fest, jeweils aus dem zugeordneten Wertebereich.

Zu **a.i)**.
$M_2^{z_{i-1}} \neq \emptyset$
$\Rightarrow$ {Definition M_2^{z-1}}
$\exists q \in M_q : z_{i-1}(q) \neq 0$
$\Rightarrow$ {Definition globaler Aktivitätszustand; Beachtung der Kopplungsrelation}
$z_{i-1}(e_1) = 0$
$\Rightarrow$ {Definition globaler/lokaler Aktivitätszustand}
$z_{i-1}(e_2) \neq 0$

Fall 1). $M_1^{z_{i-1}} = \emptyset$.
Aussage a.i) gilt direkt.

Fall 2). $M_1^{z_{i-1}} \neq \emptyset$. Sei $p \in M_1^{z_{i-1}}$ beliebig.
$\Rightarrow$ {Definition $M_1^{z_{i-1}}$}
$z_{i-1}(p) \neq 0$
$\Rightarrow$ {Satz 4.6.a; es gilt $(p, e_2) \in E$}
$z_i(p) \neq 0$
$\Rightarrow$ {Definition $M_1^{z_i}$}
$p \in M_1^{z_i}$
$\Rightarrow$ {p beliebig}
$M_1^{z_{i-1}} \subseteq M_1^{z_i}$
Aussage a.i) gilt.

Zu **a.ii)**.
Seien $p \in M_1^{z_{i-1}}$ beliebig, $q \in M_2^{z_{i-1}}$ beliebig.
$\Rightarrow$ {Definitionen M_1^{z-1}, M_2^{z-1}}
$z_{i-1}(p) \neq 0 \neq z_{i-1}(q)$
$\Rightarrow$ {Definition globaler Aktivitätszustand}
$z_{i-1}(e_2) \neq 0$
$\Rightarrow$ {Satz 4.6.b; es gilt $E(e_2) = \emptyset$}
$\exists k \in I\!N, k \geq i-1 : (z_k(e_2) \in \{0, \mathsf{F}\} \vee (z_k(e_2) = 1 \wedge false))$

Fall 1). $z_k(e_2) = 0$.
$\Rightarrow$ {Definition globaler Aktivitätszustand}
$z_k(e_1) \neq 0$
$\Rightarrow$ {Beachtung der Kopplungsrelation}
$\forall q' \in M_q : z_k(q') = 0$.
Aussage a.ii) gilt mit $k_q := k$.

Fall 2). $z_k(e_2) = \mathsf{F}$.
$\Rightarrow$ {Satz 4.6.d, mehrmals}
$\forall q \in M_2^{z_{i-1}} \exists k_q \in I\!N, k_q \geq k : (z_{k_q}(e_2) \in \{0,1\} \vee z_{k_q}(q) \in \{0,\mathsf{F}\} \vee (z_{k_q}(q) = 1 \wedge \exists x \in E(q) : z_{k_q}(x) \neq 0))$

Fall 2.1). $z_{k_q}(e_2) = 0$.
Analog Fall 1 mit k_q statt k.

Fall 2.2). $z_{k_q}(e_2) = 1$.
$\Rightarrow$ {Satz 4.6.a}
$E(e_2) \neq \emptyset \vee K(e_2) \setminus \{e_1, e_2\} \neq \emptyset \vee (\exists k'_q \in I\!N, k < k'_q < k_q : z_{k'_q}(e_2) = 0)$
$\Rightarrow$ {Die ersten beiden Alternativen sind nach Voraussetzung false}
$\exists k'_q \in I\!N, k < k'_q < k_q : z_{k'_q}(e_2) = 0$.
Analog Fall 1 mit k'_q statt k.

Fall 2.3). $z_{k_q}(q) = 0$.
Aussage a.ii) gilt.

Fall 2.4). $z_{k_q}(q) = \mathsf{F}$.
Aussage a.ii) gilt.

Fall 2.5). $z_{k_q}(q) = 1 \wedge \exists x \in E(q) : z_{k_q}(x) \neq 0$.
Aussage a.ii) gilt.

Zu **a.iii)**.
Als Strukturvoraussetzungen gelten: $K(e_1) \setminus \{e_1, e_2\} = \{q_1, \ldots, q_m\}$, $K(e_2) \setminus \{e_1, e_2\} = \emptyset$, $E(e_1) = \emptyset$, $E(e_2) = \emptyset$.
$\Rightarrow$ {disjunkte Bereiche; symmetrische Kopplungsrelation}
$\forall v \in P \setminus \{e_1, e_2, q_1, \ldots, q_m\} : K(v) \cap \{e_1, e_2\} = \emptyset \wedge E^{-1}(v) \cap \{e_1, e_2\} = \emptyset$. $(*_1)$

Nach Satz 10.1.c gilt: $\forall v \in P \setminus \{e_1, e_2, q_1, \ldots, q_m\} : (z_i(v) = \mathsf{F}) \Rightarrow (K(v) \setminus b(v) \neq \emptyset \vee E^{-1}(v) \neq \emptyset)$
$\Rightarrow$ $\{(*_1)\}$
$\forall v \in P \setminus \{e_1, e_2, q_1, \ldots, q_m\} : (z_i(v) = \mathsf{F}) \Rightarrow (K(v) \setminus (b(v) \cup \{e_1, e_2\}) \neq \emptyset \vee E^{-1}(v) \setminus \{e_1, e_2\} \neq \emptyset)$.
Aussage a.iii) gilt.

Zu **b)**.
Nach Satz 4.14.b existieren $z'_0, z'_1, z'_2, \ldots \in GZustand(IS)$ mit $z'_0 z'_1 z'_2 \ldots \in \mathcal{V}[\![IS]\!]$ und $\lfloor z'_0 z'_1 z'_2 \ldots \rfloor = c_0 c_1 c_2 \ldots$.
$M_2^{c_{i-1}} \neq \emptyset$
$\Rightarrow$ {Definitionen $\lfloor \cdot \rfloor$, $zc(\cdot)$, $M_2^{c_{i-1}}$}
$\exists l \in I\!N : zc(z'_{l-1}) = c_{i-1} \wedge zc(z'_l) = c_i \wedge M_2^{z'_{l-1}} \neq \emptyset$
$\Rightarrow$ {Punkt a.i)}
$M_1^{z'_{l-1}} \subseteq M_1^{z'_l}$
$\Rightarrow$ {Definitionen $zc(\cdot)$, $M_1^{\cdot}$}
$M_1^{zc(z'_{l-1})} \subseteq M_1^{zc(z'_l)}$
$\Rightarrow$ {Wahl von l}
$M_1^{c_{i-1}} \subseteq M_1^{c_i}$.
Aussage b) gilt.

Zu **c.i)**.
$c_0 \delta_1 c_1 \delta_2 c_2 \ldots \in \mathcal{ECT}[\![IS]\!]$
$\Rightarrow$ {Satz 4.20.d}
$c_0 c_1 c_2 \ldots \in \mathcal{CT}[\![IS]\!]$.
Aussage c.i) gilt mit Teil b).

Zu **c.ii)**.
Nach Satz 4.20.b existieren $z'_0, z'_1, z'_2, \ldots \in GZustand(IS)$ mit $z'_0 z'_1 z'_2 \ldots \in \mathcal{V}[\![IS]\!]$ und $\lfloor z'_0 z'_1 z'_2 \ldots \rfloor^{\mathrm{e}} = c_0 \delta_1 c_1 \delta_2 c_2 \ldots$. Sei $v \in P \setminus \{e_1, e_2, q_1, \ldots, q_m\}$ beliebig.

$v \in c_{i-1} \setminus c_i \wedge b(v) \notin \delta_i$

$\Rightarrow$ {Definition $\lfloor\cdot\rfloor^e$ }
$\exists i_1, i_2 \in \mathbb{N}, i_1 < i_2 : (\forall x \in \{i_1, i_1+1, \dots, i_2-1\} : v \in zc(z'_x)) \wedge v \notin zc(z'_{i_2}) \wedge b(v) \notin \{b \in B \mid \forall p \in b : (z'_{i_1}(p) \neq 0 \wedge z'_{i_2}(p) = 0) \Rightarrow z'_{i_2-1}(p) \neq \mathsf{F}\}$
$\Rightarrow$ {Definition $zc(\cdot)$; Umindizierung}
$\exists l \in \mathbb{N} : z'_{l-1}(v) \neq 0 \wedge z'_l(v) = 0 \wedge b(v) \notin \{b \in B \mid \forall p \in b : (z'_{l-1}(p) \neq 0 \wedge z'_l(p) = 0) \Rightarrow z'_{l-1}(p) \neq \mathsf{F}\}$
$\Rightarrow$ {Zusammenfassen}
$\exists l \in \mathbb{N} : z'_{l-1}(v) = \mathsf{F}$
$\Rightarrow$ {Aussage a.iii)}
$K(v) \setminus (b(v) \cup \{e_1, e_2\}) \neq \emptyset \vee E^{-1}(v) \setminus \{e_1, e_2\} \neq \emptyset$.
Aussage c.ii) gilt. □

Der Beweis von Satz 10.5 spielt sich ausschließlich auf der formalen Ebene ab. Durch die vorweggenommene Betrachtung der auftretenden grundlegenden Interaktionen und deren Bedeutung für die Semantiken in den Sätzen 4.6, 4.12, 4.22 und 10.1 sind ausreichend elementare Struktureigenschaften bekannt, um zusammengesetzt die Aussagen des Satzes beweisen zu können. Auf die algorithmische Ebene, d.h. auf die Ausführungen des zugeordneten $\mathrm{V_I}$Systems, braucht nicht mehr zurückgegriffen zu werden. Nach Definition 10.2 beschreiben die einzelnen Unterpunkte des Satzes somit abgeleitete Struktureigenschaften der jeweiligen Semantiken.

Notation 10.6. Der in Satz 10.5 spezifizierte und skizzierte Strukturbaustein wird als $EXC(p_1, \dots, p_n | q_1, \dots, q_m)$ bezeichnet.

Bemerkung 10.7. Die Praktikabilität der I–Systeme beruht nicht zuletzt darauf, dass Formeln, die kausale und temporale Abhängigkeiten bei den Phasenqualitäten ausdrücken (z.B. die Aussagen a)-c) von Satz 10.5), Entsprechungen in der Sprache der Anschauungsebene von Einflüssen, Nicht-Einflüssen, Entscheidungen, möglichen Phasenübergängen oder deren Verhinderung, Zwängen, usw., besitzen. (Die Interpretationen der Phasenqualitäten wurden in Kapitel 3 eingeführt.) Prädikatenlogische Formeln, die Eigenschaften des Verhaltens beschreiben, sind für einen menschlichen Modellierer in vielen Fällen schwer zu lesen (im Gegensatz zu Tools). Um das Verständnis zu erleichtern, bietet es sich an, sich auf der Anschauungsebene zu bewegen, um Systemanforderungen in der dort zur Verfügung stehenden Sprache zu formulieren und zu verifizieren.

Die Übersetzung von Satz 10.5 in den Wortlaut der Anschauungsebene liefert:

Korollar 10.8 (Interpretation von Satz 10.5). Es gelten die Voraussetzungen und Bezeichnungen aus Satz 10.5. Dann gilt:

i) Solange eine Phase q aus $M_q := \{q_1, \dots, q_m\}$ belegt ist, kann keine eingenommene Phase p aus $M_p := \{p_1, \dots, p_n\}$ verlassen werden.

ii) Ist eine der Phasen aus M_p belegt, wird (über den Strukturbaustein $EXC(p_1, \dots, p_n | q_1, \dots, q_m)$) auf jede Komponente, die sich in einer Phase q aus M_q befindet, ein Einfluss zum Verlassen der Phase ausgeübt.

iii) Über den Strukturbaustein $EXC(p_1, \dots, p_n | q_1, \dots, q_m)$ kann kein Einfluss auf andere Komponenten, die sich in einer Phase außerhalb von M_q befinden, ausgeübt werden, deren aktuelle Phase zu verlassen.

Beweis. Bei Verwendung der in Kapitel 3 eingeführten Bedeutungen von Phasenqualitäten ergeben sich die Teile i), ii) und iii) des Korollars direkt aus den Aussagen a.i), a.ii) und a.iii) von Satz 10.5. □

Im Beweis des Korollars wird auf die Aussagen a.i), a.ii) und a.iii) von Satz 10.5 Bezug genommen, die sich auf der formalen Ebene auf das Verhalten von IS beziehen. Für Teil i) des Korollars gibt es ebenfalls Entsprechungen bei Bezug auf die Casetrace-Semantik (siehe Satz 10.5.b)) als auch bei Bezug auf die Erweiterte Casetrace-Semantik (siehe Satz 10.5.c.i)). Von Interesse ist nur die Einnahme oder Nicht-Einnahme einzelner Phasen. Darin liegt der Unterschied zu Teil ii) des Korollars. Neben der Entwicklung der reinen Phasenbelegungen müssen zum Ausdrücken von Einflüssen auf der formalen Ebene die einzelnen Phasenqualitäten (insbesondere F) innerhalb der Semantik verfügbar sein. Das trifft nur auf das Verhalten zu. Teil iii) des Korollars beschreibt die Nicht-Existenz von Einflüssen, oder anders ausgedrückt, die Unmöglichkeit von erzwungenen Phasentransitionen. Beim Verhalten sind erzwungene Phasentransitionen an der Phasenqualität F und bei der Erweiteren Casetrace-Semantik an den Bereichsmengen zwischen den Cases erkennbar. (Letzteres erlaubt Satz 10.5.c.ii)). Die Casetrace-Semantik besitzt keine Möglichkeit, erzwungene Ereignisse in den Traces zu markieren, folglich gibt es zu Teil iii) des Korollars keine Entsprechung bei Teil b) von Satz 10.5.

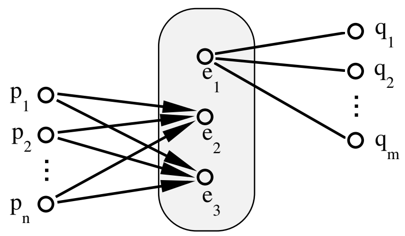
Der Strukturbaustein $EXC(p_1, \ldots, p_n|q_1, \ldots, q_m)$ verallgemeinert (über die beliebige Wahl von $n, m \in I\!N$) die innerhalb der Sätze 4.6.a/.b, 4.12.a und 4.22.a zum Ausdruck gebrachten Auswirkungen zweier eingenommener und in der Erregungsrelation stehenden Phasen. Der gleiche Effekt des Strukturbausteins kann erzielt werden, wenn jedes p aus M_p mit jedem q aus M_q als Paar (p, q) in die Erregungsrelation aufgenommen wird, was bei großer Mächtigkeit der Mengen M_p und M_q dann aber nicht mehr praktikabel ist. In der Reduktion der Anzahl der lokalen Interaktionsbeziehungen zeigt sich neben der modularen Sichtweise ein weiterer Vorteil des Einsatzes von Strukturbausteinen.

10.2.2 Verallgemeinertes Stop

Der folgende Strukturbaustein ist eine Erweiterung des Strukturbausteins von Satz 10.5, mit dem Ziel, die Einflusspropagierung zu unterbinden. Übrig bleibt ausschließlich eine Stop-Funktionalität, bei der gewisse Phasenbelegungen beibehalten werden müssen, solange bestimmte andere Phasen belegt sind. Durch diesen Strukturbaustein sollen aber keine Phasentransitionen erzwungen werden.

Satz 10.9 (Verallgemeinertes Stop). Sei $IS = (P, B, \underline{B}, K, E)$ ein I–System mit $p_1, \ldots, p_n, e_1, e_2, e_3, q_1, \ldots, q_m \in P$, pw. versch., $\{e_1, e_2, e_3\} \in \underline{B}$, $E(e_2) = E(e_3) = \emptyset$, $E^{-1}(e_2) = E^{-1}(e_3) = \{p_1, \ldots, p_n\}$, $K(e_2) \setminus \{e_1, e_2, e_3\} = K(e_3) \setminus \{e_1, e_2, e_3\} = \emptyset$, $E(e_1) = \emptyset$, $E^{-1}(e_1) = \emptyset$, $K(e_1) \setminus \{e_1, e_2, e_3\} = \{q_1, \ldots, q_m\}$ und $n, m \in I\!N$.

Skizze:



Für einen globalen Aktivitätszustand $z \in GZustand(IS)$ sei $M_1^z := \{p \in p_1, \ldots p_n \mid z(p) \neq \mathsf{0}\}$ und $M_2^z := \{q \in q_1, \ldots q_m \mid z(q) \neq \mathsf{0}\}$. Für einen Case $c \in Case(IS)$ sei $M_1^c := \{p \in p_1, \ldots p_n \mid p \in c\}$ und $M_2^c := \{p \in q_1, \ldots q_m \mid q \in c\}$.

a) Sei $z_0 z_1 z_2 \ldots \in \mathcal{V}[\![IS]\!]$ mit $z_j \in GZustand(IS), j = 0, 1, 2, \ldots$. Dann gilt für alle $i = 1, 2, \ldots$:

i) $(M_2^{z_{i-1}} \neq \emptyset) \Rightarrow (M_1^{z_{i-1}} \subseteq M_1^{z_i})$.

ii) $\forall v \in P \setminus \{e_1, e_2, e_3\} : (z_i(v) = \mathsf{F}) \Rightarrow (K(v) \setminus (b(v) \cup \{e_1, e_2, e_3\}) \neq \emptyset \vee E^{-1}(v) \setminus \{e_1, e_2, e_3\} \neq \emptyset)$.

b) Sei $c_0 c_1 c_2 \ldots \in \mathcal{CT}[\![IS]\!]$ mit $c_j \in Case(IS), j = 0, 1, 2, \ldots$. Dann gilt für alle $i = 1, 2, \ldots$:

$$(M_2^{c_{i-1}} \neq \emptyset) \Rightarrow (M_1^{c_{i-1}} \subseteq M_1^{c_i}).$$

c) Sei $c_0\delta_1 c_1\delta_2 c_2\ldots \in \mathcal{ECT}[\![IS]\!]$ mit $c_j \in Case(IS), \delta_{j+1} \subseteq B, j = 0,1,2,\ldots$. Dann gilt für alle $i = 1,2,\ldots$:

i) $(M_2^{c_{i-1}} \neq \emptyset) \Rightarrow (M_1^{c_{i-1}} \subseteq M_1^{c_i})$.

ii) $\forall v \in P \setminus \{e_1,e_2,e_3\} : (v \in c_{i-1} \setminus c_i \wedge b(v) \notin \delta_i) \Rightarrow (K(v) \setminus (b(v) \cup \{e_1,e_2,e_3\}) \neq \emptyset \vee E^{-1}(v) \setminus \{e_1,e_2,e_3\} \neq \emptyset)$.

Beweis. Es gelten die Bezeichnungen und Voraussetzungen aus dem Satz. Seien $M_p := \{p_1,\ldots,p_n\}$, $M_e := \{e_1,e_2,e_3\}$, $M_q := \{q_1,\ldots,q_m\}$. Sei $i \in \{1,2,\ldots\}$ beliebig aber fest, jeweils aus dem zugeordneten Wertebereich.

Zu **a.i)**.
$M_2^{z_{i-1}} \neq \emptyset$
$\Rightarrow$ {Definition $M_2^{z_{i-1}}$}
$\exists q \in M_q : z_{i-1}(q) \neq 0$
$\Rightarrow$ {Beachtung der Kopplungsrelation}
$z_{i-1}(e_1) = 0$
$\Rightarrow$ {Definition globaler Aktivitätszustand}
$z_{i-1}(e_2) \neq 0 \vee z_{i-1}(e_3) \neq 0$

Fall 1). $z_{i-1}(e_2) \neq 0$.

Fall 1.1). $M_1^{z_{i-1}} = \emptyset$.
Aussage a.i) gilt direkt.

Fall 2.1). $M_1^{z_{i-1}} \neq \emptyset$. Sei $p \in M_1^{z_{i-1}}$ beliebig.
$\Rightarrow$ {Definition $M_1^{z_{i-1}}$}
$z_{i-1}(p) \neq 0$
$\Rightarrow$ {Satz 4.6.a; es gilt $(p,e_2) \in E$}
$z_i(p) \neq 0$
$\Rightarrow$ {Definition $M_1^{z_i}$}
$p \in M_1^{z_i}$
$\Rightarrow$ {p beliebig}
$M_1^{z_{i-1}} \subseteq M_1^{z_i}$
Aussage a.i) gilt.

Fall 2). $z_{i-1}(e_3) \neq 0$.

Wegen des symmetrischen Aufbaus analog zu Fall 1 mit e_3 statt e_2.

Zu **a.ii)**.
Als Strukturvoraussetzungen gelten: $K(e_1) \setminus \{e_1,e_2,e_3\} = \{q_1,\ldots,q_m\}$, $K(e_2) \setminus \{e_1,e_2,e_3\} = K(e_3) \setminus \{e_1,e_2,e_3\} = \emptyset$, $E(e_1) = E(e_2) = E(e_3) = \emptyset$.
$\Rightarrow$ {disjunkte Bereiche; symmetrische Kopplungsrelation}
$\forall v \in P \setminus \{e_1,e_2,e_3,q_1,\ldots,q_m\} : K(v) \cap \{e_1,e_2,e_3\} = \emptyset \wedge E^{-1}(v) \cap \{e_1,e_2,e_3\} = \emptyset.$ $(*_1)$

Nach Satz 10.1.c gilt: $\forall v \in P \setminus \{e_1,e_2,e_3,q_1,\ldots,q_m\} : (z_i(v) = \mathsf{F}) \Rightarrow (K(v) \setminus b(v) \neq \emptyset \vee E^{-1}(v) \neq \emptyset)$
$\Rightarrow$ {$(*_1)$}
$\forall v \in P \setminus \{e_1,e_2,e_3,q_1,\ldots,q_m\} : (z_i(v) = \mathsf{F}) \Rightarrow (K(v) \setminus (b(v) \cup \{e_1,e_2,e_3\}) \neq \emptyset \vee E^{-1}(v) \setminus \{e_1,e_2,e_3\} \neq \emptyset)$. $(*_2)$

$M_q \in \underline{B}$ ist träger Bereich.
$\Rightarrow$ {Die Strukturvoraussetzungen ermöglichen die Anwendung von Satz 10.1.d für jedes $q \in M_q$.}
$\forall q \in M_q : (z_i(q) = \mathsf{F}) \Rightarrow (K(q) \setminus (b(q) \cup b(e_1)) \neq \emptyset \vee E^{-1}(q) \neq \emptyset)$
$\Rightarrow$ {Es gilt $b(e_1) = \{e_1,e_2,e_3\}$ und $E^{-1}(q) \cap \{e_1,e_2,e_3\} = \emptyset$.}
$\forall q \in M_q : (z_i(q) = \mathsf{F}) \Rightarrow (K(q) \setminus (b(q) \cup \{e_1,e_2,e_3\}) \neq \emptyset \vee E^{-1}(q) \setminus \{e_1,e_2,e_3\} \neq \emptyset)$. $(*_3)$

$\Rightarrow$ {$(*_2 \wedge *_3)$}
$\forall v \in P \setminus \{e_1,e_2,e_3\} : (z_i(v) = \mathsf{F}) \Rightarrow (K(v) \setminus (b(v) \cup \{e_1,e_2,e_3\}) \neq \emptyset \vee E^{-1}(v) \setminus \{e_1,e_2,e_3\} \neq \emptyset)$.
Aussage a.ii) gilt.

Zu **b)**.
Nach Satz 4.14.b existieren $z'_0, z'_1, z'_2, \ldots \in GZustand(IS)$ mit $z'_0 z'_1 z'_2 \ldots \in \mathcal{V}[\![IS]\!]$ und $\lfloor z'_0 z'_1 z'_2 \ldots \rfloor = c_0 c_1 c_2 \ldots$.
$M_2^{c_{i-1}} \neq \emptyset$
$\Rightarrow$ {Definitionen $\lfloor\cdot\rfloor$, $zc(\cdot)$, $M_2^{\cdot}$}
$\exists l \in I\!N : zc(z'_{l-1}) = c_{i-1} \wedge zc(z'_l) = c_i \wedge M_2^{z'_{l-1}} \neq \emptyset$
$\Rightarrow$ {Punkt a.i)}
$M_1^{z'_{l-1}} \subseteq M_1^{z'_l}$
$\Rightarrow$ {Definitionen $zc(\cdot)$, $M_1^{\cdot}$}
$M_1^{zc(z'_{l-1})} \subseteq M_1^{zc(z'_l)}$
$\Rightarrow$ {Wahl von l}
$M_1^{c_{i-1}} \subseteq M_1^{c_i}$.
Aussage b) gilt.

Zu **c.i)**.
$c_0 \delta_1 c_1 \delta_2 c_2 \ldots \in \mathcal{ECT}[\![IS]\!]$
$\Rightarrow$ {Satz 4.20.d}
$c_0 c_1 c_2 \ldots \in \mathcal{CT}[\![IS]\!]$.
Aussage c.i) gilt mit Punkt b).

Zu **c.ii)**.
Nach Satz 4.20.b existieren $z'_0, z'_1, z'_2, \ldots \in GZustand(IS)$ mit $z'_0 z'_1 z'_2 \ldots \in \mathcal{V}[\![IS]\!]$ und $\lfloor z'_0 z'_1 z'_2 \ldots \rfloor^{\mathsf{e}} = c_0 \delta_1 c_1 \delta_2 c_2 \ldots$. Sei $v \in P \setminus \{e_1, e_2, e_3, q_1, \ldots, q_m\}$ beliebig.
$v \in c_{i-1} \setminus c_i \wedge b(v) \notin \delta_i$
$\Rightarrow$ {Definition $\lfloor\cdot\rfloor^{\mathsf{e}}$ }
$\exists i_1, i_2 \in I\!N, i_1 < i_2 : (\forall x \in \{i_1, i_1+1, \ldots, i_2-1\} : v \in zc(z'_x)) \wedge v \notin zc(z'_{i_2}) \wedge b(v) \notin \{b \in B \mid \forall p \in b : (z'_{i_1}(p) \neq \mathsf{0} \wedge z'_{i_2}(p) = \mathsf{0}) \Rightarrow z'_{i_2-1}(p) \neq \mathsf{F}\}$
$\Rightarrow$ {Definition $zc(\cdot)$; Umindizierung}
$\exists l \in I\!N : z'_{l-1}(v) \neq \mathsf{0} \wedge z'_l(v) = \mathsf{0} \wedge b(v) \notin \{b \in B \mid \forall p \in b : (z'_{l-1}(p) \neq \mathsf{0} \wedge z'_l(p) = \mathsf{0}) \Rightarrow z'_{l-1}(p) \neq \mathsf{F}\}$
$\Rightarrow$ {Zusammenfassen}
$\exists l \in I\!N : z'_{l-1}(v) = \mathsf{F}$
$\Rightarrow$ {Aussage a.ii)}
$K(v) \setminus (b(v) \cup \{e_1, e_2, e_3\}) \neq \emptyset \vee E^{-1}(v) \setminus \{e_1, e_2, e_3\} \neq \emptyset$.
Aussage c.ii) gilt. $\square$

Der Beweis von Satz 10.9 wird analog zum Beweis von Satz 10.5 auf der formalen Ebene abgewickelt. Die benötigten elementaren Struktureigenschaften sind in vorangegangenen Sätzen bereits bewiesen worden. Die Ausführungen des zugeordneten $\mathrm{V_I}$Systems brauchen nicht mehr analysiert zu werden.

Notation 10.10. Der in Satz 10.9 spezifizierte und skizzierte Strukturbaustein wird als $STOP(p_1, \ldots, p_n | q_1, \ldots, q_m)$ bezeichnet.

Die Übersetzung von Satz 10.9 in den Wortlaut der Anschauungsebene (entsprechend Bemerkung 10.7) liefert:

Korollar 10.11 (Interpretation von Satz 10.9). Es gelten die Voraussetzungen und Bezeichnungen aus Satz 10.9. Dann gilt:

i) Solange eine Phase q aus $M_q := \{q_1, \ldots, q_m\}$ belegt ist, kann keine eingenommene Phase p aus $M_p := \{p_1, \ldots, p_n\}$ verlassen werden.

ii) Über den Strukturbaustein $STOP(p_1, \ldots, p_n | q_1, \ldots, q_m)$ kann kein Einfluss auf andere Komponenten ausgeübt werden, deren aktuelle Phase zu verlassen.

Beweis. Bei Verwendung der in Kapitel 3 eingeführten Bedeutungen von Phasenqualitäten ergeben sich die Teile i) und ii) des Korollars direkt aus den Aussagen a.i) und a.ii) von Satz 10.9. □

Im Gegensatz zum Strukturbaustein $EXC(p_1, \dots, p_n|q_1, \dots, q_m)$ besteht beim Strukturbaustein $STOP(p_1, \dots, p_n|q_1, \dots, q_m)$ keine Einflussnahme auf die Komponenten der Phasen aus M_q. Bei $STOP(p_1, \dots, p_n|q_1, \dots, q_m)$ führt eine Instabilität der Komponente von e_2 in e_2 (oder in e_3), als Folge von $M_p \neq \emptyset$, bei einer nicht freien Phase e_1 (d.h. $M_q \neq \emptyset$), zu einer Phasentransition in die freie Phase e_3 (e_2). Da auch dort ein Einfluss über die Erregungsbeziehung wirksam ist, folgt eine Instabilität mit anschließender Phasentransition zurück nach e_2 (e_3). Der Ablauf wiederholt sich, solange e_1 nicht frei ist. Dieser alternierende Wechsel verhindert eine Einflusspropagierung zu den Komponenten der zu e_1 wechselseitig ausgeschlossenen und belegten Phasen aus M_q. Bei $EXC(p_1, \dots, p_n|q_1, \dots, q_m)$ existiert keine dritte freie Phase e_x, die die Einflusspropagierung verhindert.

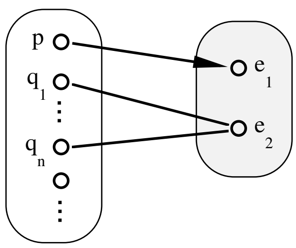
Die Aussage i) von Korollar 10.11 entspricht genau der Aussage i) von Korollar 10.8 und kann auf der formalen Ebene unter Bezug auf das Verhalten, die Casetrace-Semantik und die Erweiterte Casetrace-Semantik ausgedrückt werden (Satz 10.9.a.i), Satz 10.9.b), Satz 10.9.c.i)). Die Aussage ii) von Korollar 10.11 ähnelt der Aussage iii) von Korollar 10.8, erweitert aber die Menge der betroffenen Phasen um M_q. Das Verhalten (siehe Satz 10.9.a.ii)) und die Erweiterte Casetrace-Semantik (siehe Satz 10.9.c.ii) bieten wiederum eine Basis für gleichwertige Formulierungen auf der formalen Ebene. Die Aussagekraft der Casetrace-Semantik reicht hingegen nicht aus, weshalb es keinen Teil ii) bei Satz 10.9.b) gibt.

10.2.3 Ausgeschlossene Phasentransitionen

Der folgende Strukturbaustein dient dem Ausschluss von Phasentransitionen. Die Restriktion bezieht sich dabei auf eine ausgezeichnete Ursprungs-Phase und eine Menge von verbotenen Ziel-Phasen.

Satz 10.12 (Ausgeschlossene Phasentransition). Sei $IS = (P, B, \underline{B}, K, E)$ ein I–System mit $p, q_1, \dots, q_n, e_1, e_2 \in P$, pw. versch., $b(p) = b(q_1) = \dots = b(q_n)$, $\{e_1, e_2\} \in \underline{B}$, $E(e_1) = \emptyset$, $E^{-1}(e_1) = \{p\}$, $K(e_1) \setminus \{e_1, e_2\} = \emptyset$, $E(e_2) = \emptyset$, $E^{-1}(e_2) = \emptyset$, $K(e_2) \setminus \{e_1, e_2\} = \{q_1, \dots, q_n\}$ und $n \in I\!N$.

Skizze:



a) Sei $z_0 z_1 z_2 \ldots \in \mathcal{V}[\![IS]\!]$ mit $z_j \in GZustand(IS), j = 0, 1, 2, \ldots$. Dann gilt für alle $i = 1, 2, \ldots$:

- **i)** $(z_{i-1}(p) \neq 0) \Rightarrow (\forall q \in \{q_1, \dots, q_n\} : z_i(q) = 0)$.
- **ii)** $\forall v \in P \setminus \{e_1, e_2\} : (z_i(v) = \mathsf{F}) \Rightarrow (K(v) \setminus (b(v) \cup \{e_1, e_2\}) \neq \emptyset \vee E^{-1}(v) \setminus \{e_1, e_2\} \neq \emptyset)$.

b) Sei $c_0 c_1 c_2 \ldots \in \mathcal{CT}[\![IS]\!]$ mit $c_j \in Case(IS), j = 0, 1, 2, \ldots$. Dann gilt für alle $i = 1, 2, \ldots$:

$$(p \in c_{i-1}) \Rightarrow (\forall q \in \{q_1, \dots, q_n\} : q \notin c_i).$$

c) Sei $c_0\delta_1c_1\delta_2c_2\ldots \in \mathcal{ECT}[\![IS]\!]$ mit $c_j \in Case(IS), \delta_{j+1} \subseteq B, j = 0,1,2,\ldots$. Dann gilt für alle $i = 1,2,\ldots$:

i) $(p \in c_{i-1}) \Rightarrow (\forall q \in \{q_1,\ldots,q_n\} : q \notin c_i)$.

ii) $\forall v \in P\backslash\{e_1,e_2\} : (v \in c_{i-1}\backslash c_i \wedge b(v) \notin \delta_i) \Rightarrow (K(v)\backslash(b(v)\cup\{e_1,e_2\}) \neq \emptyset \vee E^{-1}(v)\backslash\{e_1,e_2\} \neq \emptyset)$.

Beweis. Es gelten die Bezeichnungen und Voraussetzungen aus dem Satz. Seien $M_e := \{e_1,e_2\}$, $M_q := \{q_1,\ldots,q_n\}$. Sei $i \in \{1,2,\ldots\}$ beliebig aber fest, jeweils aus dem zugeordneten Wertebereich.

Zu **a.i)**.
$z_{i-1}(p) \neq 0$
$\Rightarrow$ {Definition globaler Aktivitätszustand}
$\forall q \in M_q : z_{i-1}(q) = 0$
$\Rightarrow$ {Beachtung der Kopplungsrelation}
$z_{i-1}(e_1) \neq 0 \vee z_{i-1}(e_2) \neq 0$

Fall 1). $z_{i-1}(e_1) \neq 0$.
$\Rightarrow$ {Satz 4.6.a, $(p,e_1) \in E$}
$z_i(p) \neq 0$
$\Rightarrow$ {Definition globaler Aktivitätszustand, $M_q \subseteq b(p)$}
$\forall q \in M_q : z_i(q) = 0$.
Aussage a.i) gilt.

Fall 2). $z_{i-1}(e_2) \neq 0$. Untersuchung aller möglichen Folgezustände.

Fall 2.1). $z_i(p) \neq 0 \wedge (\forall q \in M_q : z_i(q) = 0) \wedge z_i(e_1) = 0 \wedge z_i(e_2) \neq 0$.
Aussage a.i) gilt.

Fall 2.2). $z_i(p) \neq 0 \wedge (\forall q \in M_q : z_i(q) = 0) \wedge z_i(e_1) \neq 0 \wedge z_i(e_2) = 0$.
Aussage a.i) gilt.

Fall 2.3). $z_i(p) = 0 \wedge (\exists q \in M_q : z_i(q) \neq 0) \wedge z_i(e_1) = 0 \wedge z_i(e_2) \neq 0$.
$\Rightarrow$ {$(q,e_2) \in K$}
z_i ist kein globaler Aktivitätszustand.
Fall 2.3) kann nicht eintreten.

Fall 2.4). $z_i(p) = 0 \wedge (\exists q \in M_q : z_i(q) \neq 0) \wedge z_i(e_1) \neq 0 \wedge z_i(e_2) = 0$.
$\Rightarrow$ {Satz 5.1}
$\exists z \in GZustand(IS) : z(p) \neq 0 \wedge (\forall q \in M_q : z(q) = 0) \wedge z(e_1) \neq 0 \wedge z(e_2) = 0$ und $z_0\ldots z_{i-1}.z.z_i\ldots \in \mathcal{V}[\![IS]\!]$
$\Rightarrow$ {Satz 4.6.a mit z und z_i, $(p,e_1) \in E$}
$z_i(p) \neq 0$
$\Rightarrow$
Widerspruch zur Fallvoraussetzung.
Fall 2.4) kann nicht eintreten.

Zu **a.ii)**.
Als Strukturvoraussetzungen gelten: $K(e_1) \setminus \{e_1,e_2\} = \emptyset$, $K(e_2) \setminus \{e_1,e_2\} = \{q_1,\ldots,q_n\}$, $E(e_1) = E(e_2) = \emptyset$.
$\Rightarrow$ {disjunkte Bereiche; symmetrische Kopplungsrelation}
$\forall v \in P \setminus \{e_1,e_2,q_1,\ldots,q_n\} : K(v) \cap \{e_1,e_2\} = \emptyset \wedge E^{-1}(v) \cap \{e_1,e_2\} = \emptyset$. $(*_1)$
Nach Satz 10.1.c gilt: $\forall v \in P \setminus \{e_1,e_2,q_1,\ldots,q_n\} : (z_i(v) = \mathsf{F}) \Rightarrow (K(v) \setminus b(v) \neq \emptyset \vee E^{-1}(v) \neq \emptyset)$
$\Rightarrow$ {$(*_1)$}
$\forall v \in P \setminus \{e_1,e_2,q_1,\ldots,q_n\} : (z_i(v) = \mathsf{F}) \Rightarrow (K(v) \setminus (b(v) \cup \{e_1,e_2\}) \neq \emptyset \vee E^{-1}(v) \setminus \{e_1,e_2\} \neq \emptyset)$. $(*_2)$

Sei $q \in M_q$ beliebig aber fest. Sei $z_i(q) = \mathsf{F}$.
Annahme: $\nexists v \in P \setminus (b(q) \cup \{e_1,e_2\}) : (v,q) \in E \vee (v,q) \in K$.

$z_i(q) = \mathsf{F}$
$\Rightarrow$ {Satz 10.1.b}
$\exists v \in P \setminus b(q) : ((v,q) \in E \wedge z_i(v) \neq \mathsf{0}) \vee ((v,q) \in K \wedge (\exists w \in b(v) \exists j \in I\!N, j \leq i : z_j(w) \in \{v, \mathsf{F}\}))$
$\Rightarrow$ {Annahme}
$\exists v \in \{e_1, e_2\} : ((v,q) \in E \wedge z_i(v) \neq \mathsf{0}) \vee ((v,q) \in K \wedge (\exists w \in b(v) \exists j \in I\!N, j \leq i : z_j(w) \in \{v, \mathsf{F}\}))$
$\Rightarrow$ $\{K(q) \cap M_e = \{e_2\},\ E^{-1}(q) \cap M_e = \emptyset,\ M_e \setminus \{e_2\} = \{e_1\},\ M_e \in \underline{B}\}$
$\exists j \in I\!N, j \leq i : z_j(e_1) = \mathsf{F}$
$\Rightarrow$ {Sätze 10.1.a, 4.6.b; Strukturvoraussetzungen von M_e}
$\exists j \in I\!N, j \leq i : (\forall l \in I\!N, j \leq l \leq i : z_l(e_1) = \mathsf{F})$
$\Rightarrow$ {Satz 10.1.b; $K(e_1) \setminus \{e_1, e_2\} = \emptyset$, $E^{-1}(e_1) = \{p\}$}
$\exists j \in I\!N, j \leq i : (\forall l \in I\!N, j \leq l \leq i : z_l(p) \neq \mathsf{0})$
$\Rightarrow$ {Definition globaler Aktivitätszustand; $b(p) = b(q)$}
$z_i(q) = \mathsf{0}$
$\Rightarrow$ {Am Anfang der Folgerungskette gilt $z_i(q) = \mathsf{F}$}
Widerspruch. Die Annahme ist somit falsch.

Folglich gilt: $\exists v \in P \setminus (b(q) \cup \{e_1, e_2\}) : (v,q) \in E \vee (v,q) \in K$.
$\Rightarrow$ {Umschreiben; K ist symmetrisch}
$K(q) \setminus (b(q) \cup \{e_1, e_2\}) \neq \emptyset \vee E^{-1}(q) \setminus \{e_1, e_2\} \neq \emptyset.$ $(*_3)$

$\Rightarrow$ $\{(*_2) \wedge (\forall q \in M_q : *_3)\}$
$\forall v \in P \setminus \{e_1, e_2\} : (z_i(v) = \mathsf{F}) \Rightarrow (K(v) \setminus (b(v) \cup \{e_1, e_2\}) \neq \emptyset \vee E^{-1}(v) \setminus \{e_1, e_2\} \neq \emptyset).$
Aussage a.ii) gilt.

Zu **b)**.
Nach Satz 4.14.b existieren $z'_0, z'_1, z'_2, \ldots \in GZustand(IS)$ mit $z'_0 z'_1 z'_2 \ldots \in \mathcal{V}[\![IS]\!]$ und $\lfloor z'_0 z'_1 z'_2 \ldots \rfloor = c_0 c_1 c_2 \ldots$.

$p \in c_{i-1}$
$\Rightarrow$ {Definitionen $\lfloor\cdot\rfloor$, $zc(\cdot)$}
$\exists l \in I\!N : zc(z'_{l-1}) = c_{i-1} \wedge zc(z'_l) = c_i \wedge z'_{l-1}(p) \neq \mathsf{0}$
$\Rightarrow$ {Punkt a.i)}
$\forall q \in \{q_1, \ldots, q_n\} : z_l(q) = \mathsf{0}$
$\Rightarrow$ {Definitionen $zc(\cdot)$}
$\forall q \in \{q_1, \ldots, q_n\} : q \notin zc(z_l)$
$\Rightarrow$ {Wahl von l}
$\forall q \in \{q_1, \ldots, q_n\} : q \notin c_i$.
Aussage b) gilt.

Zu **c.i)**.
$c_0 \delta_1 c_1 \delta_2 c_2 \ldots \in \mathcal{ECT}[\![IS]\!]$
$\Rightarrow$ {Satz 4.20.d}
$c_0 c_1 c_2 \ldots \in \mathcal{CT}[\![IS]\!]$.
Aussage c.i) gilt mit Teil b).

Zu **c.ii)**.
Nach Satz 4.20.b existieren $z'_0, z'_1, z'_2, \ldots \in GZustand(IS)$ mit $z'_0 z'_1 z'_2 \ldots \in \mathcal{V}[\![IS]\!]$ und $\lfloor z'_0 z'_1 z'_2 \ldots \rfloor^{\mathrm{e}} = c_0 \delta_1 c_1 \delta_2 c_2 \ldots$. Sei $v \in P \setminus \{e_1, e_2\}$ beliebig.

$v \in c_{i-1} \setminus c_i \wedge b(v) \notin \delta_i$
$\Rightarrow$ {Definition $\lfloor\cdot\rfloor^{\mathrm{e}}$ }
$\exists i_1, i_2 \in I\!N, i_1 < i_2 : (\forall x \in \{i_1, i_1+1, \ldots, i_2 - 1\} : v \in zc(z'_x)) \wedge v \notin zc(z'_{i_2}) \wedge b(v) \notin \{b \in B \mid \forall p \in b : (z'_{i_1}(p) \neq \mathsf{0} \wedge z'_{i_2}(p) = \mathsf{0}) \Rightarrow z'_{i_2-1}(p) \neq \mathsf{F}\}$
$\Rightarrow$ {Definition $zc(\cdot)$; Umindizierung}
$\exists l \in I\!N : z'_{l-1}(v) \neq \mathsf{0} \wedge z'_l(v) = \mathsf{0} \wedge b(v) \notin \{b \in B \mid \forall p \in b : (z'_{l-1}(p) \neq \mathsf{0} \wedge z'_l(p) = \mathsf{0}) \Rightarrow z'_{l-1}(p) \neq \mathsf{F}\}$
$\Rightarrow$ {Zusammenfassen}
$\exists l \in I\!N : z'_{l-1}(v) = \mathsf{F}$
$\Rightarrow$ {Aussage a.ii)}
$K(v) \setminus (b(v) \cup \{e_1, e_2\}) \neq \emptyset \vee E^{-1}(v) \setminus \{e_1, e_2\} \neq \emptyset.$
Aussage c.ii) gilt. $\square$

Der Beweis von Satz 10.12 verwendet elementare Struktureigenschaften der Semantiken und braucht nicht mehr auf die Ausführungen des zugeordneten $\mathrm{V_I}$Systems zurückgreifen. Wie schon bei den Beweisen der abgeleiteten Struktureigenschaften zu den Strukturbausteinen $EXC(p_1, \dots, p_n|q_1, \dots, q_m)$ und $STOP(p_1, \dots, p_n|q_1, \dots, q_m)$ bewegt man sich auf der formalen Ebene und vermeidet die algorithmische.

Notation 10.13. Der in Satz 10.12 spezifizierte und skizzierte Strukturbaustein wird als $NOT(p|q_1, \dots, q_n)$ bezeichnet.

Die Übersetzung von Satz 10.12 in den Wortlaut der Anschauungsebene (entsprechend Bemerkung 10.7) liefert:

Korollar 10.14 (Interpretation von Satz 10.12). Es gelten die Voraussetzungen und Bezeichnungen aus Satz 10.12. Dann gilt:

i) Eine Phasentransition $p \to q$ mit q aus $M_q := \{q_1, \dots, q_n\}$ kann nicht eintreten.

ii) Über den Strukturbaustein $NOT(p|q_1, \dots, q_n)$ kann kein Einfluss auf andere Komponenten (insbesondere die Komponente von p und M_q) ausgeübt werden, deren aktuelle Phase zu verlassen.

Beweis. Bei Verwendung der in Kapitel 3 eingeführten Bedeutungen von Phasenqualitäten ergeben sich die Teile i) und ii) des Korollars direkt aus den Aussagen a.i) und a.ii) von Satz 10.12. □

Die Gültigkeit der Aussage ii) des Korollars für die Komponente der Phasen M_q ist nicht offensichtlich, da Kopplungskanten von e_2 zu jeder Phase aus M_q existieren, über die potentiell Einflüsse propagiert werden können. Als eine Voraussetzung für eine Einflusspropagierung von der Komponente von e_1 und e_2 zur Komponente von M_q müssen e_1 und p zur gleichen Zeit belegt sein. Als weitere Voraussetzung muss zusätzlich eine der Phasen aus M_q belegt sein, damit e_2 nicht frei ist und somit die Propagierung angestoßen wird. Da nun aber die Phasen p und M_q alle zur gleichen Komponente gehören, widersprechen sich die beiden Voraussetzungen und eine Einflusspropagierung kann deshalb nicht auftreten.

Wie schon bei den Korollaren 10.8 und 10.11 ist es auch bei Korollar 10.14 von der Bezugs-Semantik abhängig, für welche Korollaraussagen sich gleichwertige Formulierungen auf der formalen Ebene angeben lassen. So besitzt die Casetrace-Semantik eine zu geringe Ausdruckskraft, um Teil ii) von Korollar 10.14 erfassen zu können.

Durch die mehrfache, unterschiedlich parametrisierte Verwendung des Strukturbausteins $NOT(p|q_1, \dots, q_n)$ ist es möglich, das lokale Verhalten einer Systemkomponente gezielt einzuschränken. Es ergibt sich die Möglichkeit, für einen Bereich eines I–Systems die Folgen der erlaubten und unerlaubten Phasentransitionen vorzugeben (z.B. als Graph), um dann, über die Verwendung des Strukturbausteins, die Vorgaben umzusetzen. In Kapitel 11.4 wird hierzu ein Beispiel gegeben werden, in dem dann auch ein Bezug zu Satz 9.20 (Lokale Ereignisstruktur) hergestellt wird.

10.3 Beispiel: Die Verwendung von Strukturbausteinen bei der Modellierung einer synchronen Kommunikation zwischen autonomen Prozessen

In diesem Abschnitt wird demonstriert, wie mit Hilfe der in den vorherigen Abschnitten vorgestellten Strukturbausteine vorgegebene Systemanforderungen an ein I–System verifiziert werden können. Dabei wird durch die gezielte Identifikation von bekannten Teilstrukturen im Gesamt-I–System ein modularer Analyseansatz verfolgt. Unter Rückgriff auf die (bereits bewiesenen) Eigenschaften der vorkommenden Strukturbausteine lässt sich die Erfüllung der Systemanforderungen direkt nachweisen.

10.3.1 Systembeschreibung und Modellierung

Betrachtet wird eine synchrone Kommunikation zwischen zwei autonomen Prozessen P_1 und P_2 in einem verteilten System über einen Kommunikationskanal α gemäß [18, 85, 91].
Das Kommunikationsschema von P_1 (für P_2 symmetrisch) lässt sich in drei Phasen einteilen und wie folgt darstellen:

$$\overbrace{\text{TIE}(\alpha : P_2)}^{ti_1} \rightarrow \overbrace{\text{Communication}}^{co_1} \rightarrow \overbrace{\text{UNTIE}(\alpha : P_2)}^{un_1}$$

Die erste Phase ti_1 ist eine Registrierungsphase, in der der Kommunikationskanal α zum Prozess P_2 aufgebaut wird. In der darauffolgenden zweiten Phase co_1 findet die synchrone Kommunikation zwischen P_1 und P_2 statt. Nach Beendigung der Kommunikation wird in der abschließenden dritten Phase un_1 der Kommunikationskanal α aufgelöst.

Es gelten dabei folgende Synchronisationsbedingungen zwischen P_1 und P_2 aus Sicht von P_1 (für P_2 symmetrisch):

SB1: Wenn P_1 in ti_1 ist und P_2 ist noch nicht in ti_2 oder co_2, dann muss P_1 warten.

SB2.a: Wenn P_1 in co_1 ist und P_2 ist noch in ti_2, dann kann P_1 nicht fortfahren.

SB2.b: Wenn P_1 in co_1 ist und P_2 ist noch in ti_2, dann wird ein Einfluss auf P_2 ausgeübt, ti_2 zu verlassen.

SB3.a: Wenn P_1 in un_1 ist und P_2 ist noch in co_2, dann hat P_1 zu warten.

SB3.b: Wenn P_1 in un_1 ist und P_2 ist noch in co_2, dann wird ein Einfluss auf P_2 ausgeübt, co_2 zu verlassen.

SB4: Wenn P_1 in ti_1 ist und P_2 ist außerhalb des Kommunikationsschemas, d.h. in einer Phase re_2, dann gibt es keinen Einfluss von P_1 auf P_2, re_2 zu verlassen.

Es sei darauf hingewiesen, dass die obigen Bedingungen SB2.b und SB3.b explizite Forderungen nach Einflüssen enthalten, durch die bestimmte Aktionen (das Verlassen von Phasen) erzwungen werden sollen. Demhingegen werden in SB4 solche Einflüsse explizit untersagt. Auf die besondere Eignung von I–System zur expliziten Modellierung von Einflüssen (deren Existenz oder deren Fehlen) zwischen Komponenten eines verteilten Systems wurde bereits in den vorangegangenen Kapiteln eingegangen. In diesem Zusammenhang sei auch noch einmal auf [89] verwiesen.

Das I–System IS_{10} in Abbildung 10.1 modelliert das spezifizierte Kommunikationsverhalten. Das lokale Kommunikationsschema ist als zyklischer Graph in die Bereiche P_1 und P_2 mit eingezeichnet. Die Realisierung solcher lokalen Ereignisstrukturen wurde in Kapitel 9.7 behandelt. Bei IS_{10} werden die Strukturbausteine aus Abschnitt 10.2 verwendet. $STOP(ti_1, ti_2|re_1, re_2, un_1, un_2)$ ist eine Variante des Verallgemeinerten Stops, $EXC(co_1, co_2|ti_1, ti_2)$ und $EXC(un_1, un_2|co_1, co_2)$ sind Varianten der Verallgemeinerten Erregung. Entsprechend der Definiton 9.1 handelt es sich bei P_1 und P_2 um Relevanzbereiche von IS_{10}, und bei $\underline{VS}$, $\underline{VE}_1$ und $\underline{VE}_2$ um Kontrollbereiche. Abbildung 10.1 zeigt IS_{10} in einem Start-Case $\{re_1, e_2, e_4, e_6, re_2\}$.

10.3.2 Verifikation der Synchronisatonsbedingungen

Satz 10.15 (Synchrone Kommunikation). Die Konstruktion IS_{10} in Abbildung 10.1 erfüllt SB1 bis SB4. Sie ist minimal in dem Sinne, dass durch die angegebenen Verbindungen mit $\underline{VS}$, $\underline{VE}_1$ und $\underline{VE}_2$ keine nicht geforderten Einflüsse auf P_1 und P_2 wirken.

Beweis. Die Korrektheit der Konstruktion ergibt sich in einfacher Weise aus der Vereinigung der Eigenschaften der verwendeten Strukturbausteine.

Zu SB1: Wenn P_2 noch nicht in ti_2 oder co_2 ist, dann muss P_2 in re_2 oder un_2 sein. SB1 folgt nun direkt aus Korollar 10.11.i) für den Strukturbaustein $STOP(ti_1, ti_2|re_1, re_2, un_1, un_2)$, d.h. für $M_q = \{re_1, re_2, un_1, un_2\}$ und $M_p = \{ti_1, ti_2\}$.

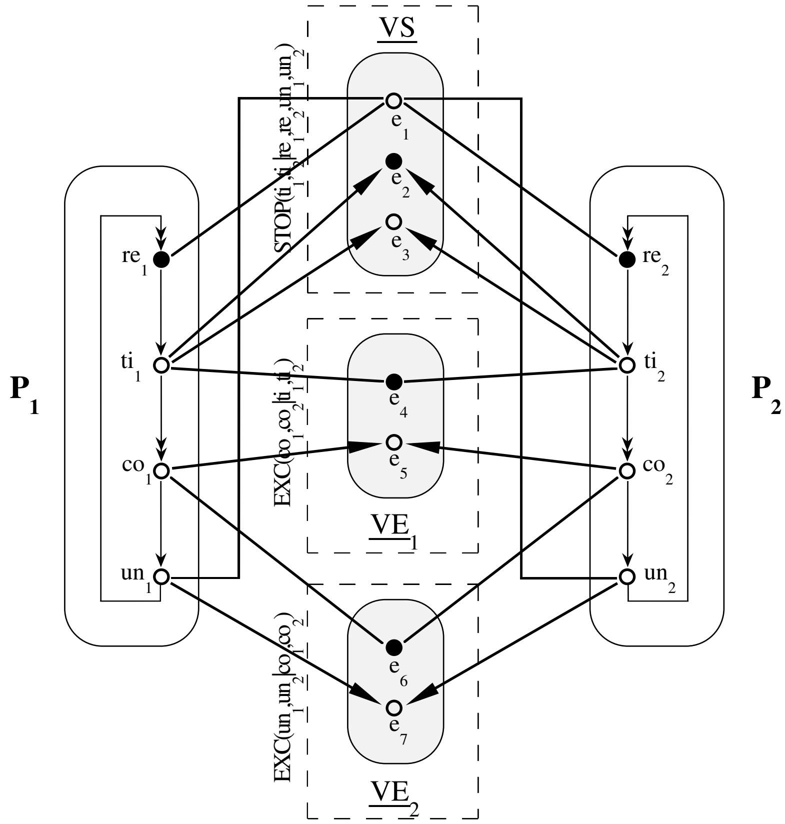


Abbildung 10.1: IS_{10}; Synchrone Kommunikation, realisiert durch 3 Strukturbausteine

Zu SB2.a/SB2.b: SB2.a folgt direkt aus Korollar 10.8.i) und SB2.b folgt direkt aus Korollar 10.8.ii), jeweils für den Strukturbaustein $EXC(co_1, co_2|ti_1, ti_2)$, d.h. für $M_q = \{ti_1, ti_2\}$ und $M_p = \{co_1, co_2\}$.

Zu SB3.a/SB3.b: SB3.a folgt direkt aus Korollar 10.8.i) und SB3.b folgt direkt aus Korollar 10.8.ii), jeweils für den Strukturbaustein $EXC(un_1, un_2|co_1, co_2)$, d.h. für $M_q = \{co_1, co_2\}$ und $M_p = \{un_1, un_2\}$.

Zu SB4: Wegen Korollar 10.11.ii) wird auf P_2 in re_2 kein Einfluss über $STOP(ti_1, ti_2|re_1, re_2, un_1, un_2)$ ausgeübt. Das Korollar 10.8.ii) schließt zudem Einflüsse über $EXC(co_1, co_2|ti_1, ti_2)$ und $EXC(un_1, un_2|co_1, co_2)$ aus. Da es keine weiteren Verbindungen zwischen P_1 und P_2 gibt, gilt SB4.

Zur Minimalität: Die Einflüsse der drei verwendeten Strukturbausteine überlagern sich. Einflüsse gehen durch das Zusammensetzen nicht verloren, da sie durch Nachrichten im zugeordneten $\mathrm{V_I}$System repräsentiert werden (siehe Kapitel 3.2) und wegen Verhaltensaxiom VA2 (siehe Kapitel 3.2.1) dort keine Nachrichten verloren gehen. Es kommen auch keine Nachrichten hinzu, da der Aufbau der Strukturbausteine durch das Zusammensetzen nicht verändert wird. Von daher folgt die Minimalität der Konstruktion IS_{10}, so wie es im Satz formuliert ist, direkt aus den Korollaren 10.8.iii) und 10.11.ii). □

Dieses Beispiel zeigt die praktische Anwendbarkeit der Strukturbausteine. Ein Anwender kann bei der Modellierung die Details über die Semantiken außer Acht lassen. Bei der Formulierung von Systemanforderungen für eine kleine Anzahl von Bereichen und bei deren Umsetzung reicht es, in Worten von Phasen und Einflüssen zu sprechen. Auf dieser Sprachebene sind mit jedem Strukturbaustein eine Menge von Eigenschaften verbunden (siehe z.B. Korollare 10.11, 10.11, 10.11). Alle Eigenschaften sind lokal und betreffen nur Phasen und Übergänge in den involvierten Relevanzbereichen, die mit dem Strukturbaustein verbunden sind, unabhängig davon, wie diese mit anderen Bereichen gekoppelt sind. Die Eigenschaften stellen Verhaltensanforderungen temporaler Synchronisationsanforderungen dar. Die Kenntnis des Gesamtsystems wird nicht vorausgesetzt. Damit werden allerdings die wesentlichen Modellierungsgegebenheiten und Anforderungen verteilter Systeme abgedeckt. Durch die Kombination von Strukturbausteinen werden deren Eigenschaftsmengen vereinigt. Zusätzlich hinzugenommene Strukturbausteine zerstören nicht die Restriktionen, die durch bereits verwendete Strukturbausteine auferlegt werden. Einflüsse/Restriktionen überlagern sich einfach. Die Dynamik/Semantik der I-Systeme ist so definiert worden, dass dieses lokale Überlagerungs-Prinzip umgesetzt wird.

10.4 Inkrementelles Modellieren

Hat man einen Satz von Strukturbausteinen mit spezifischen Struktureigenschaften entwickelt, kann man diesen zum *inkrementellen Entwurf* von I–Systemen verwenden. Es ergibt sich eine „Baukasten"-Modellierungsmethodik, bei der ein am Anfang beliebig freies Phasenspiel in ausgezeichneten Bereichen schrittweise durch das Einbringen passender Strukturbausteine soweit eingeschränkt wird, dass durch die Systemanforderungen gegebene Verhaltensvorgaben erfüllt werden. Neu hinzugenommene Bausteine beeinflussen dabei nur die lokale Systemumgebung der durch sie verbundenen Phasen/Bereiche. Abläufe innerhalb des Restsystems bleiben, ausgenommen von Propagierungseffekten, hingegen unbeeinflusst.
Die Korrektheit der Gesamtkonstruktion ergibt sich einfach aus der Vereinigung der Struktureigenschaften der verwendeten Strukturbausteine, so wie es in Abschnitt 10.3 demonstriert wurde. Um die Korrektheit der Konstruktion durchgehend zu gewährleisten, dürfen eingesetzte Strukturbausteine im Verlauf des Modellierens nicht verändert werden, es sei denn, die Veränderungen werden in ihren Auswirkungen auf das Verhalten des Systems vollständig erfasst.

Der große Vorteil eines inkrementellen Modellierungsverfahrens liegt in der Möglichkeit der „On-the-fly"-Berücksichtigung von unvorhergesehenen zusätzlichen Systemanforderungen, die sich erst im Laufe eines Modellierungsprozesses ergeben. Bei der Modellierung von verteilten Systemen bedingen sich spezielle Probleme durch die Verteiltheit der Kontrolle. Interaktions-Anforderungen zwischen Komponenten sind lokal zu behandeln, d.h. sie betreffen kleine Teilsysteme von interagierenden benachbarten Komponenten, können allerdings (unerwünschte) globale Propagierungseffekte hervorrufen, die in den bisherigen Designphasen nicht absehbar waren. So mögen z.B. erwartete Ereignisse in einer Systemkomponente niemals eintreten, da eine Restriktion irgendwo anders im System die Propagierung von Einflüssen nach sich zieht, die schließlich das erwartete Ereignis unterbinden. Durch den Einsatz eines inkrementellen Modellierungsverfahrens kann auf solche unerwünschten Phänomene reagiert werden, ohne den kompletten Modellierungsprozess neu beginnen zu müssen. Außerhalb der Theorie der I–Systeme werden inkrementelle Modellierungsstrategien z.B. in [45, 59] behandelt.

Das folgende Beispiel demonstriert den inkrementellen Entwurf von I–Systemen. Es wird dort ein nicht-triviales Synchronisationsproblem zwischen (Betriebssystem-)Prozessen durch das schrittweise Einbringen von restringierenden Interaktionsbeziehungen gelöst. Das Beispiel ist in gekürzter Form auch in [89] zu finden.

10.4.1 Beispiel: Realisierung von Prioritäten zwischen verteilten Prozessen

Eine häufige Ursache von Interessenkonflikten in einem verteilten Computersystem liegt in der Notwendigkeit, rare oder kostenintensive Ressourcen zwischen verschiedenen Gruppen von Komponenten zu teilen. Angenommen, wir haben in solch einem Netzwerk zwei Komponenten b_1 und b_2, die zu bestimmten Zeiten rechenintensive zeitkritische Jobs auf eine spezielle high-speed Maschine M übertragen, um sie dort auszuführen. Die Jobs dürfen, wenn sie erst einmal initialisiert worden sind, aufgrund ihrer engen Deadlines nicht unterbrochen werden. Um Konflikte bezüglich des Zugriffs auf M zu vermeiden, werden in vielen Fällen Prioritätsregelungen zwischen den konkurrierenden Prozessen implementiert. Aus organisatorischen Gründen besitze b_2 eine höhere Priorität als b_1.

In diesem Abschnitt werden verschiedene Aspekte von Prioritätenbehandlung im Rahmen des angesprochenen Szenarios diskutiert und mit I–Systemen inkrementell modelliert. Dabei werden b_1 und b_2 jeweils durch einen Bereich repräsentiert. Den lokalen Verhalten von b_1 und b_2 werden schrittweise Restriktionen auferlegt, so dass beim Zugriff auf M die Prioritäten respektiert werden.

10.4.1.1 Lokale Ereignisstrukturen in b_1 und b_2

Der Vorgang des Zugriffs auf M kann für b_j, $j = 1, 2$, durch folgende 4-Phasen Ereignisstruktur beschrieben werden.

$$\begin{tikzcd}
\mathrm{rg}_j \arrow[r, two heads] & \mathrm{cs}_j \arrow[r, two heads] & \mathrm{cl}_j \arrow[dl] \\
& \mathrm{r}_j \arrow[ul] &
\end{tikzcd}$$

In der Registrierungsphase (registration phase) rg_j wird der Zugriff auf M vorbereitet. Es erfolgt die Anmeldung bei M und die Abstimmung mit den Konkurrenz-Bereichen. Auf die Registrierungsphase folgt die Zugriffsphase (critical section phase) cs_j, in der b_j befugt ist, Jobs auf M auszuführen. Nach der Zugriffsphase werden in der Trennungsphase (clearing phase) cl_j die Verbindungen zu M aufgelöst, so dass M für neue Anfragen zur Verfügung steht. Bei den Übergängen $rg_j \to cs_j$ und $cs_j \to cl_j$ handelt es sich um erzwungene Phasentransitionen, die schließlich eintreten werden, sofern keine zusätzlichen externen Einflüsse auferlegt werden, die dies verhindern. In der Restphase (remainder phase) r_j sind alle verbleibenden Aktivitäten von b_j zusammengefasst. In r_j gelangt b_j von cl_j aus, und von r_j aus kann b_j in rg_j eintreten. Das Eintreten in und das Verlassen von r_j werden als autonome Aktionen in b_j angenommen, die von lokalen Kontrollentscheidungen abhängig sind. Bei den Übergängen $cl_j \to r_j$ und $r_j \to rg_j$ handelt es sich deshalb um freie Phasentransitionen.

Die Darstellung und Realisierung von lokalen Ereignisstrukturen wurde ausführlich in Kapitel 9.7 behandelt. Da die obige 4-Phasen Ereignisstruktur zyklisch ist (und somit jeder Knoten mindestens einen Vorgängerknoten besitzt), lässt sich mit der Konstruktion aus Satz 9.22 das gewünschte Verhalten bei den Bereichen b_1 und b_2 erzielen. In Übereinstimmung mit der graphischen Darstellung in Kapitel 9.7 werden im Folgenden die Ereignisstrukturen in die beiden Bereiche eingezeichnet, anstatt die zusätzlichen Kontrollbereiche und Kopplungs- und Erregungsbeziehungen anzugeben.

10.4.1.2 Statische Zugriffspriorität

Wenn eine von mehreren Komponenten geteilte Ressource in einem verteilten System ausschließlich exklusiv benutzbar ist, und somit der Zugriff auf sie auf der Basis von wechselseitigem Ausschluss geregelt werden muss, dann ergibt sich als resultierende Prioritätsanforderung, dass wenn zwei Prozesse beide bereit sind, auf die Ressource zuzugreifen, dann kann der Prozess mit der höheren Priorität voranschreiten und den Zugriff durchführen, wohingegen der andere zu warten hat. Im Kontext des Beispiels lässt sich diese Anforderung wie folgt formulieren:

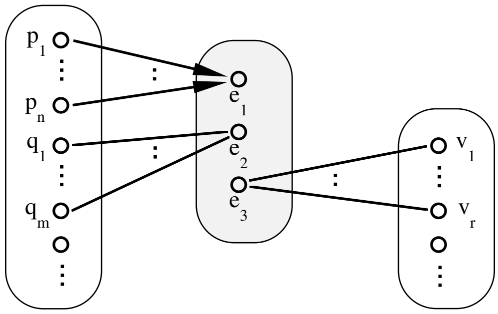
Anforderungen 1.

PR0: Wenn die Komponenten b_1 in rg_1 und b_2 in rg_2 sind, dann kann nur b_2 in cs_2 eintreten, wohingegen b_1 warten muss.

Zur Realisierung dieser Restriktion wird ein neuer Strukturbaustein mit dem folgenden Satz eingeführt. Er ergänzt die Menge der Strukturbausteine aus Abschnitt 10.2, wobei hier nur Struktureigenschaften von Interesse sind, die sich auf das Verhalten $\mathcal{V}[\![\cdot]\!]$ beziehen, bzw. auf deren Entsprechungen auf der Anschauungsebene durch die Interpretation der Phasenqualitäten.

Satz 10.16 (Bedingter Ausschluss von Phasentransitionen). Sei $IS = (P, B, \underline{B}, K, E)$ ein I–System mit $p_1, \ldots, p_n, q_1, \ldots, q_m, e_1, e_2, e_3, v_1, \ldots, v_r \in P$ und $b(p_1) = \ldots = b(p_n) = b(q_1) = \cdots = b(q_m)$, $\{e_1, e_2, e_3\} \in \underline{B}$, $b(v_1) = \ldots = b(v_r)$ $b(p_1) \neq b(v_1) \neq b(e_1) \neq b(p_1)$, für $n, m, r \in I\!N$. Weiterhin gelte $E^{-1}(e_1) = \{p_1, \ldots, p_n\}$, $K(e_2) \setminus b(e_2) = \{q_1, \ldots, q_m\}$, $K(e_3) \setminus b(e_3) = \{v_1, \ldots, v_r\}$ und $E(e_1) = K(e_1) \setminus b(e_1) = E(e_2) = E^{-1}(e_2) = E(e_3) = E^{-1}(e_3) = \emptyset$.

Skizze:



Sei $z_0 z_1 z_2 \ldots \in \mathcal{V}[\![IS]\!]$ mit $z_j \in GZustand(IS), j = 0, 1, 2, \ldots$.
Sei $b_1 := b(p_1)$, $b_2 := b(e_1)$ und $b_3 := b(v_1)$.
Dann gilt für alle $p \in \{p_1, \ldots, p_n\}$, $q \in \{q_1, \ldots, q_m\}$, $v \in \{v_1, \ldots, v_r\}$, $i = 1, 2, \ldots$:

i) *Wenn b_3 in v ist, dann ist keine Phasentransition $p \to q$ in b_1 möglich.*
$(z_{i-1}(v) \neq 0 \wedge z_{i-1}(p) \neq 0) \Rightarrow (z_i(q) = 0)$

ii) *b_1 kann über b_2 keinen Einfluss auf andere Bereiche (insbesondere b_3) ausüben, Phasen (insbesondere v) zu verlassen oder Phasen nicht einzunehmen.*
$\forall u_1, u_2 \in P \setminus (b_2 \cup b_1)$ mit $b(u_1) = b(u_2)$:
$((z_i(u_1) = \mathsf{F}) \Rightarrow (K(u_1) \setminus (b(u_1) \cup b_2) \neq \emptyset \vee E^{-1}(u_1) \setminus b_2 \neq \emptyset))) \wedge$
$(((z_i(u_1) = u_2 \vee (z_i(u_1) = \mathsf{F} \wedge (\forall u_3 \in b(u_1) \setminus \{u_1, u_2\} : u_3 \text{ ist nicht frei in } z_i))) \wedge (\nexists k \geq i : z_k(u_2) = 1)) \Rightarrow (K(u_2) \setminus (b(u_2) \cup b_2) \neq \emptyset \vee E(u_1) \setminus b_2 \neq \emptyset))$

iii) *Wenn b_3 in keiner der Phasen $\{v_1, \ldots, v_r\}$ ist, kann b_3 über b_2 nicht verhindern, dass b_1 eine Phasentransition $p \to q$ durchführt.*
$(z_{i-1}(p) = q \wedge (\forall l \geq i-1 : z_l(v_1) = \ldots = z_l(v_r) = 0)) \Rightarrow ((\exists k \geq i : z_k(q) = 1) \vee (K(q) \setminus (b(q) \cup b_2) \neq \emptyset \vee E(p) \setminus b_2 \neq \emptyset)))$

iv) *b_3 kann über b_2 keinen Einfluss auf andere Bereiche (insbesondere b_1) ausüben, Phasen (insbesondere p oder q) zu verlassen, oder Phasen außerhalb von $\{q_1, \ldots, q_m\}$ nicht einzunehmen.*
$\forall u_1 \in P \setminus (b_2 \cup b_3), u_2 \in P \setminus (b_2 \cup b_3 \cup \{q_1, \ldots, q_m\})$ mit $b(u_1) = b(u_2)$:
$((z_i(u_1) = \mathsf{F}) \Rightarrow (K(u_1) \setminus (b(u_1) \cup b_2) \neq \emptyset \vee E^{-1}(u_1) \setminus b_2 \neq \emptyset))) \wedge$
$(((z_i(u_1) = u_2 \vee (z_i(u_1) = \mathsf{F} \wedge (\forall u_3 \in b(u_1) \setminus \{u_1, u_2\} : u_3 \text{ ist nicht frei in } z_i))) \wedge (\nexists k \geq i : z_k(u_2) = 1)) \Rightarrow (K(u_2) \setminus (b(u_2) \cup b_2) \neq \emptyset \vee E(u_1) \setminus b_2 \neq \emptyset))$ □

Notation 10.17. Der in Satz 10.16 spezifizierte und skizzierte Strukturbaustein wird als $EXCL(v_1, \ldots, v_r | (p_1, \ldots, p_n), (q_1, \ldots, q_m))$ bezeichnet.

Im Rahmen dieses Beipiels wird auf den formalen Beweis von Satz 10.16 verzichtet. Die Beweismethode ist die gleiche wie bei den Strukturbausteinen in Abschnitt 10.2. Die kursiv angegebenen Formulierungen entsprechen den darunterliegenden Formeln und ergeben sich aus den Bedeutungen der Phasenqualitäten, wie sie in Kapitel 3 eingeführt wurden.
In diesem Zusammenhang sei noch einmal auf die Bemerkung 10.7 hingewiesen. Dem Tenor der Bemerkung folgend werden im verbleibenden Teil dieses Beispiel-Abschnittes Systemanforderungen in der auf der Anschauungsebene zur Verfügung stehenden Sprache formuliert und verifiziert.

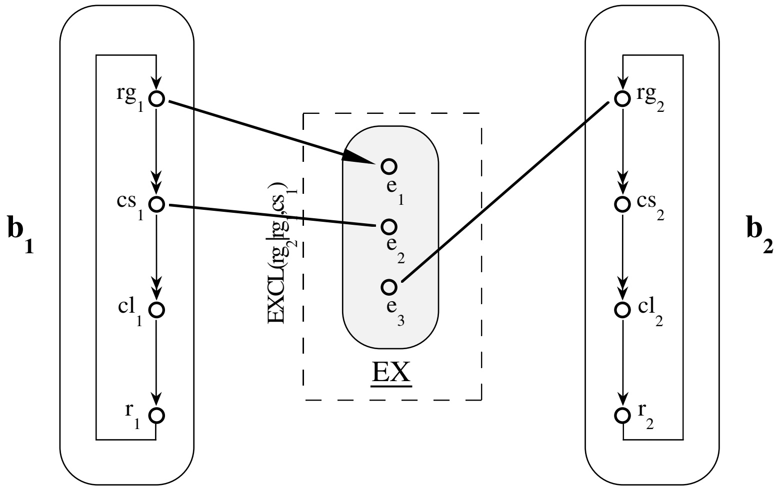


Abbildung 10.2: IS_{11}; Minimale Realisierung von PR0

Zur Realisierung der Anforderung PR0 wird der Strukturbaustein $EXCL(\cdot)$ in der Parametrisierung $EXCL(rg_2|rg_1, cs_1)$ verwendet. Hierbei verbindet ein träger Bereich $\{e_1, e_2, e_3\}$, der im Folgenden mit $\underline{EX}$ bezeichnet wird, die Bereiche b_1 und b_2, wie es in Abbildung 10.2 dargestellt ist.

Satz 10.18 (Statische Zugriffspriorität). Die Konstruktion IS_{11} in Abbildung 10.2 erfüllt PR0. Sie ist minimal in dem Sinne, dass durch die angegebenen Verbindungen mit $\underline{EX}$ keine weiteren Restriktionen b_1 oder b_2 auferlegt werden.

Beweis. Satz 10.18 ist eine direkte Folgerung aus Satz 10.16.i)-iv) mit $n = m = v = 1$ und $p_1 := rg_1$, $q_1 := cs_1$ und $v_1 := rg_2$.
Teil i) von Satz 10.16 gewährleistet den Ausschluss einer Phasentransition $rg_1 \to cs_1$ in b_1, wenn b_2 in rg_2 ist. Teil ii) ermöglicht b_2, eine Phasentransition $rg_2 \to cs_2$ durchzuführen, die aufgrund der lokalen Ereignisstruktur in b_2 schließlich auch eintreten wird. Teil ii) bis iv) beschreiben die Minimalität der Konstruktion. Durch die Konstruktion entstehen keine weiteren Einflüsse, die zusätzliche Phasenübergänge in b_1 oder b_2 erzwingen oder verhindern. □

Der Satz zeigt noch einmal die Praktikabilität von Strukturbausteinen. Aus den Struktureigenschaften der Bausteine lassen sich Verhaltenseigenschaften des modellierten Systems direkt ablesen, ohne dass der Anwender die Axiome und Aktionen zur Dynamik von I–Systemen kennen und analysieren muss. Weiterhin zeigt die Konstruktion in Abbildung 10.2, dass sich mit I–Systemen ohne weiteres *asymmetrische* Interaktionsstrukturen modellieren lassen. (Das I–System IS_{10} in Abbildung 10.1 ist ein Beispiel für eine *symmetrische* Struktur.)

10.4.1.3 Erzwingende Zugriffspriorität

In dem Beispiel-Szenario darf zu jedem Zeitpunkt höchstens eine der beiden Komponenten b_1, b_2 einen Job auf der Machine M ausgelagert haben und dort ausführen. Die Sensibilität der zu verarbeitenden Daten erfordere diese Sicherheitsbestimmung. Somit, wenn b_1 einen Job auf M in Ausführung hat, und b_2 wartet auf eine Ausführungsgelegenheit, dann sollte b_2 aufgrund der höheren Priorität einen Einfluss auf b_1 ausüben (z.B. durch eine spezielle Benachrichtigung), den Job auf M zu beenden und Platz für b_2 zu schaffen. Die Komponente b_1 darf hingegen, wegen ihrer niedrigeren Priorität, solch einen Einfluss nicht auf b_2 ausüben können, d.h. eine Prioritätenumkehr sollte ausgeschlossen sein. In der Ausdrucksweise der I–Systeme ergibt sich für das Beispiel:

Anforderungen 2.

PR1: Die Phasen cs_1 und cs_2 sind wechselseitig ausgeschlossen.

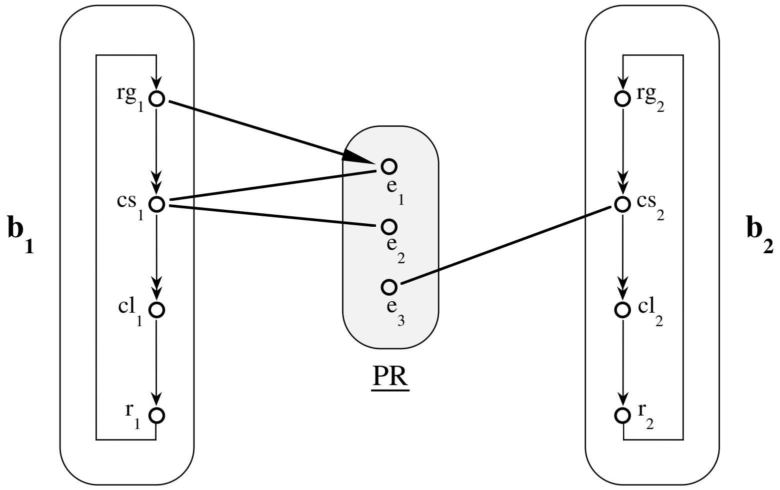


Abbildung 10.3: IS_{12}; Minimale Realisierung von PR1, PR2, PR3

PR2: Wenn b_2 in rg_2 ist, und b_1 ist in cs_1, dann übt b_2 einen Einfluss auf b_1 aus, cs_1 zu verlassen.

PR3: Wenn b_1 in rg_1 ist, und b_2 ist in cs_2, dann übt b_1 *keinen* Einfluss auf b_2 aus, cs_2 zu verlassen.

Realisieren lässt sich diese Form der Priorität durch ein I–System, wie es in Abbildung 10.3 dargestellt ist und bei der ein träger Bereich $\underline{PR}$ die Bereiche b_1 und b_2 wie angegeben verbindet. Die Konstruktion ist eine Variation des Strukturbausteins $EXCL(cs_2|rg_1, cs_1)$ (vergleichbar Abbildung 10.2 mit Kante (cs_2, e_3) anstelle von (rg_2, e_3)) mit einer zusätzlichen Kopplungskante (e_1, cs_1). Diese hinzugenommene Kante repräsentiert einen zusätzlichen Einfluss, der notwendig ist, da Satz 10.16.v) der Anforderung PR2 widerspricht und $EXCL(cs_2|rg_1, cs_1)$ in seiner Urform die Anforderungen 2 folglich nicht erfüllen kann.

Satz 10.19 (Erzwingende Zugriffspriorität). Die Konstruktion IS_{12} in Abbildung 10.3 erfüllt PR1, PR2 und PR3. Sie ist minimal in dem Sinne, dass durch die angegebenen Verbindungen mit $\underline{PR}$ keine weiteren Restriktionen b_1 oder b_2 auferlegt werden.

Beweis. Die Erfüllung von PR1 ist offensichtlich, denn für jeden globalen Aktivitätszustand $z \in GZustand(IS_{12})$ gilt nach Definition 3.3, dass $z(e_1) = 0$ und $z(e_2) = 0$ aus $z(cs_1) \neq 0$ folgen. Da $\underline{PR}$ nur aus drei Phasen besteht, muss $z(e_3) = 1$ gelten, und damit gilt auch $z(cs_2) = 0$.

Zur Überprüfung von PR2 und PR3 wird eine beliebige Ausführung Π von $V_I System(IS_{12})$ betrachtet.

Angenommen, es gelte $z^{\Pi,t}\langle V_{b_1}\rangle(cs_1) \neq 0$ und $z^{\Pi,t}\langle V_{b_2}\rangle(rg_2) \neq 0$ ab einem Zeitpunkt $t^0 \leq t$ der Ausführung Π. Wegen Satz 3.14 muss ebenfalls $z^{\Pi,t}\langle V_{\underline{PR}}\rangle(e_3) \neq 0$ gelten. $(*)$
Die lokale Ereignisstruktur in b_2 erzwingt einen Zeitpunkt $t^1 \geq t$, zu dem bei V_{b_2} die Variablenbelegungen $\_z(rg_2) = \mathsf{F}$, $\_k(cl_2) = true$ und $\_k(r_2) = true$ gelten (gemäß Kapitel 9.7). Wegen $(*)$ gilt zudem $\_k(cs_2) = true$ oder $\_mark(e_3) = true$ (Beobachtung 2 in dem Beweis von Satz 3.14). V_{b_2} führt die Aktion A5 (Fall *not* a) *not* b) und *not* c)) aus, und es erfolgt ein Solicitation-Aufruf A6, infolgedessen V_{b_2} eine $solicit(b_2 \setminus \{rg_2\})$-Nachricht an $V_{\underline{PR}}$ sendet.
Bei $V_{\underline{PR}}$ kommt die Nachricht zu einem Zeitpunkt $t^2 \geq t^1$ an. $V_{\underline{PR}}$ führt A11 und dann A13.iv aus und setzt schließlich $\_z(e_3) := \mathsf{F}$. Wegen $(*)$ gilt $\_k(e_1) = true = \_k(e_2)$. Als nächstes führt $V_{\underline{PR}}$ die Aktion A5 (Fall *not* a) *not* b) und *not* c)) aus, und es erfolgt ein Solicitation-Aufruf A6, infolgedessen $V_{\underline{PR}}$ eine $solicit(\underline{PR} \setminus \{e_3\})$-Nachricht an V_{b_1} sendet.
Bei V_{b_1} kommt die Nachricht zu einem Zeitpunkt $t^3 \geq t^2$ an. V_{b_1} führt A11 und dann A13.iv aus und setzt schließlich $\_z(cs_1) := \mathsf{F}$. Also gilt PR2.

Angenommen, es gelte $z^{\Pi,t}\langle V_{b_1}\rangle(rg_1) \neq 0$ und $z^{\Pi,t}\langle V_{b_2}\rangle(cs_2) \neq 0$ ab einem Zeitpunkt $t^0 \leq t$ der Ausführung Π. Wegen Satz 3.14 muss ebenfalls entweder $z^{\Pi,t}\langle V_{\underline{PR}}\rangle(e_1) \neq 0$ oder $z^{\Pi,t}\langle V_{\underline{PR}}\rangle(e_2) \neq 0$ gelten.

Vernachlässigt man die Kopplungskante (e_1, cs_1), ergibt sich der Strukturbaustein $EXCL(cs_2|rg_1, cs_1)$, und nach Satz 10.16.ii) kann b_1 keinen Einfluss auf b_2 ausüben. Durch die Hinzunahme von (e_1, cs_1) kann sich nur im Fall $z^{\Pi,t}\langle V_{\underline{PR}}\rangle(e_1) \neq 0$ etwas an dieser Situation ändern.

Angenommen, in diesem Fall könnte b_1 einen Einfluss auf b_3 ausüben, cs_2 zu verlassen. Dann müsste es einen Zeitpunkt $t^1 \geq t$ geben, zu dem V_{b_2} eine $solicit(\cdot)$-Nachricht von $V_{\underline{PR}}$ erhält. Um eine $solicit(\cdot)$-Nachricht abzuschicken, muss $V_{\underline{PR}}$ die Aktion A6 ausführen, die ausschließlich in Aktion A4 oder Aktion A5 aufgerufen wird. Da $\underline{PR}$ ein träger Bereich ist, kommt nur A5 in Frage. Der Solicitation-Aufruf erfolgt bei A5 im Fall *not* a) *not* b) und *not* c) der Aktionsbeschreibung, d.h. zu einem Zeitpunkt $t^2 \in [t, t^1[$ muss bei $V_{\underline{PR}}$ insbesondere $M_1 = \emptyset = M_2$ gelten (M_1, M_2 wie in der Aktionsbeschreibung von A5). Dies ist aber nicht möglich, da wegen $z^{\Pi,t^2}\langle V_{b_1}\rangle(cs_1) = 0$ bei $V_{\underline{PR}}$ unter anderem die Variablenbelegung $\_k(e_2) = false$ vorliegt, d.h. es steht eine freie Phase zur Verfügung. Somit kann b_1 keinen Einfluss auf b_3 ausüben, eine Phase zu verlassen, und es gilt PR3.

Die Minimalität der Konstruktion in Abbildung 10.3 unter Vernachlässigung der Kopplungskante (e_1, cs_1) folgt aus der Minimalität des Strukturbausteins $EXCL(cs_2|rg_1, cs_1)$, welche durch Satz 10.16.ii)-iv) gezeigt wurde. Durch die Hinzunahme der Kopplungskante wird nur ein weiterer notwendiger Einfluss auf b_1 in cs_1 realisiert, derart, wie es im vorangegangenen Abschnitt beschrieben worden ist. Dieser Einfluss ist auf b_1 in cs_1 begrenzt, denn die Existenz der Kopplungskante hat bei Ausführung der Aktion A13 (Update von Phasenqualitäten) keinen Einfluss auf das Setzen der $\_z(p)$-Variablen, wenn $p \neq cs_1$ gilt. Daraus ergibt sich die Minimalität der Gesamtkonstruktion. □

10.4.1.4 Reserviertes Zugriffsrecht

Im Rahmen des Beispiel-Szenarios werde nun angenommen, dass b_1 eine Serie von zeitkritischen Jobs auf M ausführen möchte. Jeder dieser Jobs besitzt eine kurzfristige Deadline im Vergleich zu den Jobs, deren Ausführung auf M von b_2 beabsichtigt wird. Nun mag der Fall eintreten, dass der Job-Scheduler von M entscheidet, einen Job von b_1 zu starten, um die Einhaltung der Deadlines zu gewährleisten, anstatt die höhere Priorität von b_2 zu respektieren und einen Job von b_2 mit unkritischer Deadline zuzulassen. Offensichtlich lässt sich dieser problematische Fall, der ein weiteres Beispiel für eine Prioritätenumkehr darstellt, unendlich oft wiederholen, was schließlich zu einem Verhungern von b_2 führt, da b_1 andauernd b_2 vorgezogen wird. Um diesen unerwünschten Effekt zu vermeiden, werden die folgenden Anforderungen formuliert.

Anforderungen 3.

PR4: Die Bereiche b_j, $j = 1, 2$, werden nicht daran gehindert, von r_j aus in rg_j einzutreten.

PR5: Wenn b_2 in rg_2 ist, und b_1 ist in cs_1, dann kann b_1 nur bis maximal rg_1 voranschreiten, solange wie b_2 noch nicht in cs_2 eingetreten ist.

PR6: Nachdem b_2 rg_2 verlassen hat, hat b_1 die Chance, wieder nach cs_1 zu gehen.

Da in diesem Modellierungsschritt gezielt das Eintreten und das Verlassen von rg_1 und rg_2 betrachtet werden, besteht keine Forderung nach wechselseitigem Ausschluss von cs_1 und cs_2.

Die Konstruktionsidee zur Realisierung der erforderlichen Synchronisationsmechanismen zwischen b_1 und b_2 besteht in dem Einsatz eines einzelnen Kontrollbereiches $\underline{CP}$, dessen Verhalten seinerseits durch einen weiteren Kontrollbereich $\underline{EX}'$ gesteuert wird. Die Gesamtkonstruktion IS_{13} ist in Abbildung 10.4 dargestellt. Der Kontrollbereich $\underline{EX}'$ ist Bestandteil von $EXCL(rg_2|(e_{12}), (e_{11}))$, einer Version des Strukturbausteines $EXCL(v_1, \ldots, v_r|(p_1, \ldots, v_n), (q_1, \ldots, q_m))$ aus Satz 10.16. Mit Hilfe der bekannten Struktureigenschaften dieses Bausteines ergibt sich auf einfache Weise die Korrektheit sowie die Minimalität von IS_{13}.

Satz 10.20 (Reserviertes Zugriffsrecht). Die Konstruktion IS_{13} in Abbildung 10.4 erfüllt PR4, PR5 und PR6. Sie ist minimal in dem Sinne, dass durch die angegebenen Verbindungen mit $\underline{CP}$ und $\underline{EX}'$ keine weiteren Restriktionen b_1 oder b_2 auferlegt werden.

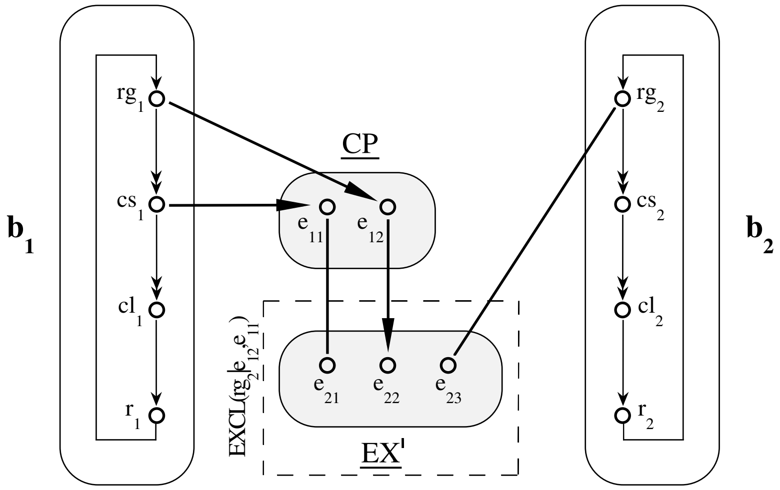


Abbildung 10.4: IS_{13}; Minimale Realisierung von PR4, PR5, PR6

Beweis. Die Phasentransition $r_j \to rg_j$ ist Teil der lokalen Ereignisstruktur von b_j, $j = 1, 2$, und wird somit durch lokale Zwänge nicht verhindert. Die Interaktionsbeziehungen zwischen b_1 und $\underline{CP}$ verhindern ebenfalls nicht das Eintreten von b_1 in rg_1, da sowohl r_1 keine $\mathrm{E_{out}}$-Nachbarphase in $\underline{CP}$ als auch rg_1 keine K-Nachbarphase in $\underline{CP}$ besitzt. Mindestens eines von beidem wäre aber notwendig, um eine Phasentransition $r_1 \to rg_1$, die nur mittels Aktion A4 oder A5 stattfinden kann, unterbinden zu können. Die Interaktionsbeziehung zwischen b_2 und $\underline{EX}'$ kann wegen Satz 10.16.ii) (für $EXCL(rg_2|e_{12}, e_{11})$) nicht das Eintreten von b_2 in rg_2 verhindern. Zusammenfassend gilt PR4.

Zur Überprüfung von PR5 und PR6 wird eine beliebige Ausführung Π von $V_I System(IS_{13})$ betrachtet.

Angenommen, es gelte $z^{\Pi,t^0}\langle V_{b_1}\rangle(cs_1) \neq 0$ und $z^{\Pi,t^0}\langle V_{b_2}\rangle(rg_2) \neq 0$ zu einem Zeitpunkt t^0 der Ausführung Π.
Wenn $V_{\underline{CP}}$ zum Zeitpunkt t^0 in e_{12} ist, dann folgt aus Satz 10.16.i), dass dies weiterhin mindestens solange gilt, wie V_{b_2} in rg_2 bleibt. Wenn $V_{\underline{CP}}$ zum Zeitpunkt t^0 in e_{11} ist, dann übt V_{b_1} in cs_1 einen Einfluss auf $V_{\underline{CP}}$ aus, e_{11} zu verlassen, d.h. es gilt bei $V_{\underline{CP}}$ schließlich die Variablenbelegung $\_z(e_{11}) = \mathsf{F}$. Daraufhin führt $V_{\underline{CP}}$ die Aktion A5 aus mit dem Ergebnis einer Phasentransition $e_{11} \to e_{12}$. Somit gibt es einen Zeitpunkt $t^1 \geq t^0$, bei dem V_{b_1} in cs_1 und $V_{\underline{CP}}$ in e_{12} sind. Letzteres gilt ab dann mindestens solange, wie V_{b_2} noch in rg_2 ist (wegen Satz 10.16.i)). $(*)$
Da V_{b_1} mittels spezieller Nachrichten ($ackinit(\cdot)$ oder $done(\cdot)$) immer über die aktuell eingenommene Phase in $V_{\underline{CP}}$ informiert wird, gilt bei V_{b_1} schließlich $\_e_{out}(cs_1) = false$, d.h. V_{b_1} wird durch $V_{\underline{CP}}$ nicht mehr zum Verbleiben in cs_1 gezwungen. Vorher konnte V_{b_1} diese Phase nicht verlassen, ansonsten hätte sich ein Widerspruch zu Satz 4.6.a ergeben.
Die lokale Ereignisstruktur von b_1 erzwingt nun Phasentransitionen $cs_1 \to cl_1$ und $cl_1 \to r_1$. Von r_1 aus kann V_{b_1} nach rg_1 wechseln (wegen PR4). In rg_1 muss V_{b_1} dann solange stoppen, wie $V_{\underline{CP}}$ in e_{12} ist, und das ist nach $(*)$ mindestens solange der Fall, wie V_{b_2} noch in rg_2 ist. Somit gilt PR5.

Angenommen, es gelte $z^{\Pi,t^2}\langle V_{b_1}\rangle(rg_1) \neq 0$, $z^{\Pi,t^2}\langle V_{\underline{CP}}\rangle(e_{12}) \neq 0$ und $z^{\Pi,t^2}\langle V_{b_2}\rangle(cs_2) \neq 0$ zu einem Zeitpunkt t^2 der Ausführung Π.
Dann übt V_{b_1} in cs_1 einen Einfluss auf $V_{\underline{CP}}$ aus, e_{12} zu verlassen. Aus Satz 10.16.iii) (für $EXCL(rg_2|(e_{12}),(e_{11}))$) folgt, dass $V_{\underline{CP}}$ über die Interaktionsbeziehungen mit $V_{\underline{EX}'}$ nicht an einer Phasentransition $e_{12} \to e_{11}$ gehindert wird. Da weder e_{12} eine $\mathrm{E_{out}}$-Nachbarphase außerhalb von $\underline{EX}'$ noch e_{11} eine K-Nachbarphase außerhalb von $\underline{EX}'$ besitzen, können auch keine Einflüsse von außerhalb von $V_{\underline{EX}'}$ auftreten, die $e_{12} \to e_{11}$ verhindern (vergleichbar der Argumentation bei dem Beweis von PR4). Somit gilt bei $V_{\underline{CP}}$ schließlich $\_z(e_{11}) = \mathsf{F}$ und $V_{\underline{CP}}$ führt die Aktion A5 aus mit dem Ergebnis einer Phasentransition $e_{12} \to e_{11}$.
Da V_{b_1} mittels einer $done(\cdot)$-Nachricht von $V_{\underline{CP}}$ über die erfolgte Phasentransition informiert wird (als Teil von A5), gilt bei V_{b_1} nach Ausführung von A10 schließlich $\_e_{out}(rg_1) = false$, d.h. V_{b_1} wird durch $V_{\underline{CP}}$ nicht mehr zum Verbleiben in rg_1 gezwungen.

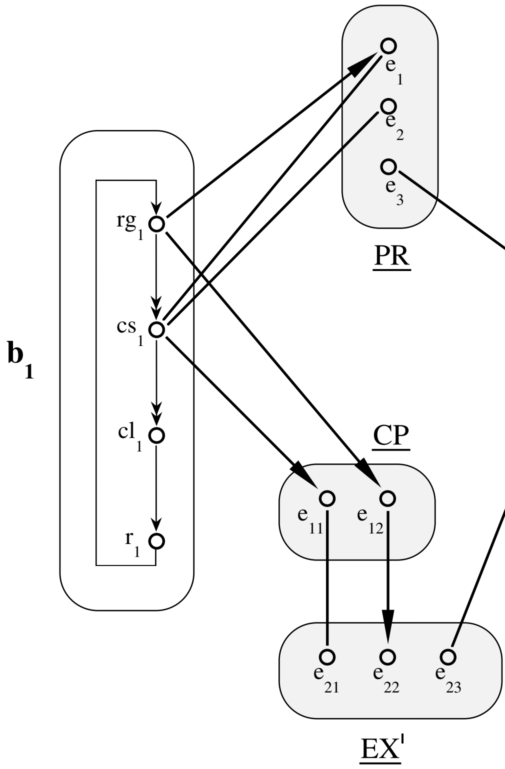


Abbildung 10.5: IS_{14}; Minimale Realisierung von PR1, PR2, PR3, PR4, PR5, PR6

Aufgrund der lokalen Ereignisstruktur von b_1 wird V_{b_1} schließlich eine Phasentransition $rg_1 \rightarrow cs_1$ durchführen, vorausgesetzt, dass V_{b_2} dies nicht durch ein erneutes Eintreten in rg_2 verhindert. Folglich gilt PR6. □

10.4.1.5 Garantierter Zugriff unter Prioritäten

Im Rahmen der inkrementellen Entwicklung einer geeigneten Zugriffsregulierung von b_1 und b_2 auf M, insbesondere unter Beachtung der gegebenen Prioritäten, werde nun gefordert, dass die Zugriffe den Anforderungen 2 *und* 3 genügen.
Um alle diese Anforderungen zu erfüllen, reicht es, die Konstruktionen aus Abbildung 10.3 und Abbildung 10.4 einfach zu kombinieren, da sich deren restriktive Effekte überlagern und da sie aufgrund der bewiesenen Minimalitätseigenschaften keine unerwünschten Seiteneffekte hervorrufen. Die Gesamtkonstruktion ist in Abbildung 10.5 dargestellt.

Satz 10.21 (Garantierter Zugriff unter Prioritäten). Die Konstruktion IS_{14} in Abbildung 10.5 erfüllt PR1 bis PR6. Sie ist minimal in dem Sinne, dass durch die angegebenen Verbindungen mit $\underline{PR}$, $\underline{CP}$ und $\underline{EX}'$ keine weiteren Restriktionen b_1 oder b_2 auferlegt werden.

Beweis. Die Erfüllung von PR1, PR2 und PR3 folgt direkt aus Satz 10.19, die Erfüllung von PR4, PR5 und PR6 aus Satz 10.20. Die restringierenden Einflüsse in den Teilkonstruktionen IS_{12} und IS_{13} überlagern sich bei IS_{14}. Einflüsse gehen durch die Zusammenführung nicht verloren, da sie Nachrichten im zugeordneten V_ISystem entsprechen und wegen Verhaltensaxiom VA2 (siehe Kapitel 3.2.1) dort keine Nachrichten verloren gehen. Es kommen auch keine Nachrichten hinzu, da der Aufbau der Teilkonstruktionen nicht verändert wurde. Die Minimalität der Gesamtkonstruktion folgt deshalb aus der Minimalität der Teilkonstruktionen. □

Die beiden Teilkonstruktionen innerhalb von IS_{14} lassen sich als Realisierungen von unabhängigen Kontrollfunktionen für die Interaktion zwischen b_1 und b_2 betrachten. Die graphische Darstellung in Abbildung 10.5 motiviert die Implementierung dieser Funktionen als Module eines Schedulers für M.

Das I–System IS_{14} lässt sich durch die Einbringung weiterer Einflüsse oder durch die Kombination mit anderen Konstruktionen, die der Erfüllung zusätzlicher Anforderungen dienen (z.B. Anforderungen, die cl_1und cl_2 betreffen), weiter ausbauen. Das vorgestellte lokale Überlagerungsprinzip bei den Einflüssen ist die Basis für die Korrektheit und Minimalität der resultierenden Systeme. In diesem Sinne demonstrieren die einzelnen Abschnitte des behandelten Beispiels die besondere Eignung von I–Systemen zum inkrementellen und modularen Enwurf verteilter Systeme von in der Praxis relevanten Ausmaßen.

Im Rahmen der inkrementellen Modellierung ist es ein Ziel, eine umfassende Sammlung von Strukturbausteinen als „Baukasten" zur Verfügung zu stellen, um alle praxisrelevanten Arten von Interaktionsbeziehungen zwischen Komponenten eines verteilten Systems modellieren zu können. Voraussetzung dazu ist die Entwicklung einer ausreichenden Menge von elementaren Struktureigenschaften, um die abgeleiteten Eigenschaften der Strukturbausteine zu beweisen. Die bisher vorgestellten elementaren Struktureigenschaften und Strukturbausteine sind dabei als eine Basis anzusehen, deren Erweiterung sich aus der Behandlung weiterer Anwendungen ergeben wird.

Kapitel 11

I–System-Transformationen

Der Entwurf von I–Systemen erfolgt modular und inkrementell. In Kapitel 10 wurden entsprechende Techniken vorgestellt, insbesondere der Einsatz von Strukturbausteinen. Durch das schrittweise Einbringen neuer Kontrollbereiche in ein Teilsystem eines I–Systems sowie deren Ein- und Verbindung mittels Kopplungs- und Erregungsrelationen wird das Verhalten in den Relevanzbereichen (Definition 9.1) gesteuert. Wie bereits in Kapitel 10.4 erwähnt wurde, kann dabei die lokale Behandlung von lokalen Systemanforderungen unerwünschte globale Propagierungseffekte hervorrufen, die in den bisherigen Designschritten nicht absehbar waren. Um diesen Effekten entgegen zu wirken, kann die Hinzunahme weiterer Kontrollbereiche und/oder Kopplungs- und Erregungsbeziehungen zwischen Phasen erforderlich sein, was allerdings neue unerwünschte globale Propagierungseffekte nach sich ziehen kann, usw.
Während einer Entwurfsphase finden somit an unterschiedlichen Stellen eines I–Systems lokal Anpassungen statt, sei es durch die Behandlung von unerwünschten Propagierungseffekten oder durch die Realisierung von Systemvorgaben mittels Strukturbausteinen. Aufgrund dieses Lokalitätsprinzips bei der Modellierung kann sich eine „Unordnung“ in der Gesamtstruktur ergeben. Z.B. können durch zwei unabhängige Modellierungsschritte redundante Teilstrukturen (Bereiche, Phasen) entstehen, die den globalen Zustandsraum unnötig (sofern Redundanz nicht von vornherein gefordert wird) aufblähen. In Abschnitt 11.4.3 wird hierzu ein Beispiel gegeben. Die Vergrößerung des Zustandsraumes ist in der Regel mit Mehrkosten (Speicherplatz, Rechenzeit) bei einer Systemanalyse verbunden und sollte deshalb, wenn möglich, vermieden werden. Überflüssige Komponenten im I–System wirken sich ebenfalls nachteilig auf eine spätere modellnahe Implementierung aus, sofern ihrer wegen unnötige Software- oder Hardware-Komponenten eingerichtet werden.
Die Aufgabe besteht jetzt darin, ein I–System, das aus einem Modellierungsprozess hervorgegangen ist, derart umzuformen, dass sich sowohl eine anschließende Analyse- als auch Implementierungsphase effizienter gestalten lassen, und ohne dass wichtige Informationen verloren gehen.

11.1 Ziele

Die Transformation eines I–Systems IS in ein neues I–System IS' verfolgt in der Regel ein oder mehrere Ziele, insbesondere die

- Reduzierung der Anzahl der Kontrollbereiche,
- Reduzierung der Anzahl der Phasen pro Bereich,
- Verkleinerung der Kopplungs- oder/und der Erregungsrelation,
- Herleitung bekannter Teilsysteme.

Gefordert wird dabei, dass sich das Systemverhalten in den Relevanzbereichen von IS, die beim Übergang von IS nach IS' erhalten bleiben, nicht ändert. Diese Forderung gewährleistet, dass Systemanforderungen, die ja immer nur für die Relevanzbereiche formuliert sind (siehe

Kapitel 9.1), genau dann durch IS erfüllt werden, wenn die gleichen Anforderungen auch durch IS' erfüllt werden. Dadurch wird eine erneute Verifikation erspart, vorausgesetzt, dass der Transformationsvorgang als korrekt bewiesen ist.

Das Erreichen der oben genannten Ziele hat unterschiedliche vorteilhafte Auswirkungen auf die Systemanalyse am I–System selbst oder auf eine spätere modellnahe Implementierung auf der Basis des I–Systems.
Eine Reduzierung der Anzahl der Kontrollbereiche hat zur Folge, dass weniger nebenläufig interagierende Systemkomponenten vorhanden sind. Dieses Weniger an Nebenläufigkeit vereinfacht die Analyse des betrachteten I–Systems, insbesondere dann, wenn die zur Verfügung stehenden Strukturbausteine nicht ausreichen und die Ausführungen des zugeordneten $\mathrm{V_I}$System anhand der die Dynamik festlegenden Axiome und Aktionen (aus Kapitel 3.2) untersucht werden müssen. Der globale Zustandsraum verkleinert sich. Für eine modellnahe Implementierung bedeutet eine Reduzierung der Anzahl der Kontrollbereiche, dass auf Softwareebene weniger Softwaremodule bzw. auf Hardwareebene weniger Hardwarekomponenten benötigt werden. Dadurch entfallen unnötige Fehlerquellen, nicht zuletzt dadurch, dass die Installation von mehr oder weniger aufwändigen Synchronisationsmechanismen zwischen den Komponenten eingespart wird. Ereignisstrukturen in den Relevanzbereichen können schneller abgearbeitet werden.
Eine Reduzierung der Anzahl der Phasen pro Bereich ist ebenfalls mit einer Vereinfachung des Analysevorganges verbunden. Bei der Bestimmung der Ausführungen des $\mathrm{V_I}$Systems sind im Rahmen der Aktionsbeschreibungen weniger Phasenqualitäten zu überprüfen und zu aktualisieren. Weniger Phasen bedeuten im Allgemeinen auch weniger zu berücksichtigende Alternativen bei anstehenden erzwungenen Phasentransitionen. Diese Erleichterungen bei der Untersuchung der Dynamik ermöglichen einfachere Beweise elementarer Struktureigenschaften (Definition 4.7). Auf eine spätere Implementierung hat die Reduzierung der Anzahl der Phasen pro Bereich keinen so direkten Einfluss wie die Reduzierung der Anzahl der Kontrollbereiche. Vielmehr gibt es indirekte Auswirkungen, da eine Reduzierung der Phasenanzahl in der Regel mit einer Reduzierung der Interaktionsbeziehungen zwischen Komponenten, sprich einer Verkleinerung der Kopplungs- oder Erregungsrelation beim I–System, verbunden ist.
Die Verkleinerung der Kopplungs- und/oder Erregungsrelation im I–System entspricht einer Verringerung der Interaktionsbeziehungen zwischen den modellierten Systemkomponenten. Sind weniger Phasenpaare in den Relationen enthalten, ist damit ein geringeres Kommunikationsaufkommen im zugeordneten $\mathrm{V_I}$System während der Abarbeitung der Aktionen A1-A13 verbunden. Das erleichtert wiederum die Bestimmung der Ausführungen des $\mathrm{V_I}$Systems und damit die Analyse des I–Systems. Bezogen auf eine modellnahe Implementierung bedeutet eine Verkleinerung der Kopplungs- und/oder Erregungsrelation, dass mit einer niedrigeren Kommunikationslast zu rechnen ist, wodurch (je nach Dimensionierung der Hardware) die Systemperformance im Hinblick auf Schnelligkeit und Fehlerunanfälligkeit verbessert werden kann.
Mit der Herleitung bekannter Teilsysteme möchte man die Wiederverwendbarkeit von Analyseergebnissen ermöglichen. Das Ziel ist es, Teilstrukturen innerhalb des I–Systems zu bilden, deren Einflüsse und Nicht-Einflüsse auf die Systemumgebung bereits bewiesen sind. Insbesondere sollen bekannte Strukturbausteine identifiziert werden können, um Verifikationsverfahren, wie sie in Kapitel 10 vorgestellt wurden, zu ermöglichen. Es ist leicht einzusehen, dass durch die Wiederverwendung von vorliegenden Analyseergebnissen aufwendige Beweise zum Systemverhalten, speziell dann, wenn der Rückgriff auf die Ausführungen des $\mathrm{V_I}$Systems notwendig wird, vereinfacht werden können. Der Vorteil der Wiederverwendbarkeit lässt sich auch auf die Implementierungsphase übertragen, denn der Einsatz existierender Softwaremodule oder Hardwarekomponenten, deren korrekte Funktionalität sich bereits gezeigt hat oder gar bewiesen ist, stellt einen entscheidenden Kosten- und Zeitfaktor bei der Realisierung komplexer verteilter Systeme dar (vgl. [37, 40, 52, 54]). Es sei dabei darauf hingewiesen, dass das Ziel der Herleitung bekannter Teilsysteme im I–System den Zielen der Reduzierung der Anzahl der Kontrollbereiche, der Reduzierung der Anzahl der Phasen pro Bereich und der Verkleinerung der Kopplungs- und/oder Erregungsrelation entgegenwirken kann, derart, dass z.B. die Hinzunahme weiterer Kontrollbereiche notwendig sein kann, um einen bestimmten Strukturbaustein als Teilstruktur zu erhalten. Hier bleibt es dem Designer überlassen, seine Zielsetzungen geeignet abzuwägen.

11.2 Formale Ansätze

Es existieren bisher noch keine Forschungsarbeiten, die sich mit Transformationen von I–Systemen beschäftigen. Allerdings gibt es erste Ansätze und Lösungen für die Vorgängermodelle der I–Systeme, die Lose Gekoppelten Systeme und die Interaktionssysteme (siehe auch Kapitel 1.3 und Kapitel 8). Für beide Modelle wurden Reduktionsverfahren mit dem Ziel entwickelt, die Anzahl der Bereiche zu minimieren.
In [65] ist ein Algorithmus für *Lose Gekoppelte Systeme mit Transitions-Relationen* angegeben, der beschreibt, wie eine beliebige Anzahl von Bereichen zu einem einzelnen Bereich zusammengefasst werden kann, derart, dass der Einfluss dieses einzelnen Bereiches auf das Restsystem genau dem Einfluss der zusammengefassten Bereiche auf dasselbe Restsystem entspricht. Formal ist ein Lose Gekoppeltes System mit Transitions-Relationen ein Tupel bestehend aus einem Lose Gekoppelten System und einer Relation über den Phasen des Lose Gekoppelten Systems. Die Relation legt die möglichen Phasenwechsel in den Bereichen des Lose Gekoppelten Systems fest. Details finden sich in [65]. In Kapitel 8 wurde die Einbettung der Lose Gekoppelten Systeme in die I–Systeme behandelt. Es ist zu erwarten, dass sich das Reduktionsverfahren aus [65] auf die speziellen I–Systeme aus der Menge $ISystem_{LCS}$, d.h. I–Systeme ohne träge Bereiche und mit leerer Erregungsrelation (Definition 8.9) übertragen lässt. Es wird die Aufgabe zukünftiger Forschungsarbeiten sein, dieses auszuarbeiten.
In [13] ist ein Verfahren angegeben, um Bereiche bei so genannten *Strukturierten Interaktionssystemen* zusammenzufassen, ohne den Einfluss auf das Restsystem zu verändern. Strukturierte Interaktionssysteme beinhalten in ihrer formalen Definition gegenüber den Interaktionssystemen im Wesentlichen drei zusätzliche mathematische Komponenten. Es handelt sich hierbei um zwei Relationen über der Phasenmenge, die die möglichen Phasentransitionen in den Bereichen spezifizieren, unterteilt, ob die Phasentransitionen immer unter Zwang stattfinden oder nicht. Des Weiteren gibt es eine Funktion, die der „Beschriftung“ der Phasentransitionen dient. Die Beschriftung besagt, dass in bestimmten Zuständen Einflüsse ignoriert werden, Transitionen nicht erlaubt, oder Kopplungen als wirkungslos anzusehen sind. Zur Formalisierung sei auf [13] verwiesen. Das Reduktionsverfahren basiert stark auf einer geeigneten Anpassung der Transitionsbeschriftungen. Auf „normale“ Interaktionssysteme (ohne die drei zusätzlichen Komponenten) ist das Verfahren in der Regel daher nicht anwendbar. I–Systeme entsprechen in ihrer formalen Struktur (gemäß Definition 2.1) den Interaktionssystemen bis auf eine Komponente. Die Menge der Cases gehört bei den I–Systemen nicht mit zur Struktur. Insbesondere besitzen I–Systeme nicht die drei zusätzlichen Komponenten der Strukturierten Interaktionssysteme. Von daher sind die Reduktionsmethoden aus [13] ohne eine (nicht triviale) Modellerweiterung nicht übertragbar.

In diesem Kapitel wird anhand eines Beispiels gezeigt, wie Transformationen von I–Systemen formal behandelt werden. Es wird dabei auf eine Modellerweiterung (weder in der formalen Struktur noch in den Festlegungen der Dynamik) verzichtet, da durch z.B. zusätzliche Beschriftungen die Einfachheit und Anschaulichkeit des Modells gestört würde. So könnten explizit modellierte Einflüsse nicht relevant sein, wenn wie im Fall der Strukturierten Interaktionssysteme Zusatzbeschriftungen erlauben ließen, explizit dargestellte Einflüsse zu ignorieren. Eine Überlagerung von expliziter Modellierung und entgegenwirkenden Zusatzbeschriftungen ist nur schwer mit der Anschauung zu vereinbaren. Zu beachten ist auch, dass die Hinzunahme neuer Komponenten in der Struktur I–System eine Erweiterung der Axiome und Aktionen, die die Dynamik festlegen, mit sich führt, da zusätzliche Kontrollstrukturen integriert werden müssen. Die Handhabbarkeit des ganzen Modells wird dadurch zusätzlich erschwert. Von daher ist es das vorrangige Bestreben, Transformationsverfahren zu entwickeln, die ohne eine Modellerweiterung auskommen.
Es werden im Folgenden keine als korrekt bewiesenen Transformationsregelsysteme vorgestellt. Es wird vielmehr die Notwendigkeit der bisher eingeführten Formalismen verdeutlicht und dabei der Bezug zu den vorangegangenen Kapiteln hergestellt. Die Entwicklung von vollständigen Regelsystemen, deren Formulierung und Korrektheitsbeweise, werden einen zukünftigen Forschungsschwerpunkt bilden (siehe Kapitel 12.2.5).

11.3 Äquivalenzen auf I–Systemen

In dem vorangegangenen Abschnitt wurde mit dem Begriff Transformation die Aufgabe verbunden, ein I–System IS in ein „handhabbareres“ I–System IS' umzuformen, unter der Vorgabe, dass sich das Systemverhalten in den Relevanzbereichen nicht ändert. Formal bedeutet das, dass festgelegt werden muss, wann IS und IS' als äquivalent betrachtet werden. Die Äquivalenz soll sinnvollerweise nicht über eine Gleichheit der formalen Strukturen definiert werden, sondern über eine Übereinstimmung von auftretenden Ereignisfolgen.

In der Literatur findet sich eine Vielzahl von wissenschaftlichen Beiträgen, die sich mit der Äquivalenz von parallelen und nebenläufigen Prozessen beschäftigen und unterschiedliche Ansätze präsentieren. Als grundlegend sind hierbei die Arbeiten von R. Milner und C.A.R. Hoare anzusehen. Milner hat in [67] die *Observation-Äquivalenz* von CCS-Programmen eingeführt. Die Idee ist, dass zwei Programme P und Q genau dann als äquivalent gelten, wenn für jedes Ergebnis P' eines so genannten s-Experiments auf P ein äquivalentes Ergebnis Q' eines s-Experiments auf Q existiert, und symmetrisch, wenn für jedes Ergebnis Q' eines so genannten s-Experiments auf Q ein äquivalentes Ergebnis P' eines s-Experiments auf P existiert. (Bezogen auf die Terminologie der I–Systeme ist ein s-Experiment vergleichbar mit einer Phasentransitionsfolge.) Es scheint sich bei dem Ansatz um eine zirkuläre Definition zu handeln. Formal wird dieses Problem gelöst durch eine iterative Definition mit einer Sequenz von abnehmend feinen Äquivalenzen. Mit *Bisimulation* als Verfeinerung der iterativ definierten Observation-Äquivalenz für Transitionssysteme beschäftigen sich D. Park [71] und D. J. Walker [82]. In Form einer binären Relation beschreibt die Bisimulation ein wechselseitiges Matching von Transitionen. Sie ist die feinste Äquivalenz auf Transitionsgraphen. Varianten von Bisimulation (schwache, starke, Divergenz sensitive, multilevel, verteilte, branching) präsentiert [5]. Für Transitionsgraphen mit unendlichem Verhalten erweitert [42] die Bisimulation um Fairness-Kriterien. Hoare definiert in [44] eine *Failure-Äquivalenz* für CSP Prozesse. Betrachtet werden hierbei so genannte „Refusals“ eines Prozesses P. Dies sind Ereignisse, die die Umgebung von P als Ausführungsmöglichkeiten anbieten kann, ohne dass das Voranschreiten von P ermöglicht wird. P gerät in einen Deadlock. Die Einführung des Konzeptes der Refusals erlaubt die formale Unterscheidung von deterministischen und nichtdeterministischen Prozessen. „Failures“ eines Prozesses P sind Tupel bestehend aus einer Trace von Ereignissen, die P der Reihe nach ausführen kann und einer Menge von Refusals von P, gültig nach Abarbeitung der Trace. Failures sagen mehr über das Verhalten eines Prozesses aus als Traces (oder Refusals) alleine. Besteht ein Interesse ausschließlich an den Folgen von beobachtbaren Ereignissen im Rahmen der Prozessaktivitäten, führt dieses zur Betrachtung von Trace-Äquivalenzen von Prozessen. [33] liefert eine Definition von *Interleaving Trace-Äquivalenz*, die auf der Gleichheit der Menge der Interleaving Traces von zwei Prozessen beruht. Innerhalb der Interleaving Traces sind nur einzelne Aktionen bei Konfigurationsübergängen (Zustandsübergängen) erlaubt. Dies entspricht den Definitionen der Interleaving Trace-Semantiken bei I–Systemen (Kapitel 5). Die Verallgemeinerung für Nebenläufigkeit wird in [33] *Step Trace Äquivalenz* genannt. Statt einzelner Aktionen bei Konfigurationsübergängen werden Mengen von nebenläufigen Aktionen betrachtet. Hier ergibt sich ein Ansatzpunkt für einen allgemeinen Äquivalenzbegriff bei I–Systemen. Eine Trace-Äquivalenz ist nicht so fein wie eine Bisimulation oder Failure-Äquivalenz und erfordert in der Regel weniger Wissen über das Prozessverhalten. So besitzen z.B. die Prozesse a.(b+c) und (a.b)+(a.c) (in CCS Notation) gleiche Traces, sind aber unterscheidbar durch Bisimulation. Systemeinschränkungen, die die Gleichwertigkeit der Ansätze mit sich führen, untersucht [28]. Die Übertragung der Konzepte Bisimulation und Trace-Äquivalenz von Transitionssystemen auf Petri-Netze liefert E.-R. Olderog [70]. Des Weiteren wird dort die *Readiness-Äquivalenz* als inverser Ansatz zur Failure-Äquivalenz vorgestellt. Ready-Sets als Gegenpart zu den Refusals umfassen stattfindbare Folgeereignisse. Einen alternativen Ansatz, um die Äquivalenz von Prozessen zu definieren, präsentiert [81]. Zwei Prozesse gelten dort als äquivalent, wenn sie die gleiche Menge von Eigenschaften erfüllen, wobei modale Logiken verwendet werden, um die Eigenschaften zu spezifizieren. Im Extremfall der leeren Eigenschaftsmenge gilt folglich die Äquivalenz aller Prozesse. Einen Überblick über die vorangegangenen exemplarisch vorgestellten Äquivalenzen sowie weitere Varianten (z.B. Possible-Futures Äquivalenz, Simulation Äquivalenzen) und deren gegenseitige Beziehungen finden sich in [32]. Ergänzungen liefern [11, 25, 58].

Wegen der per Definition festgelegten Eindeutigkeit der Phasennamen und Disjunktheit der Bereiche bei einem I–System wird durch die Definition einer Trace-Äquivalenz (vergleichbar der Step Trace Äquivalenz aus [33]) ein geeignetes formales Mittel bereitgestellt, um eine Gleichheit des Systemverhaltens von zwei I–Systemen zu dokumentieren. Da man mit I–Systemen einen rein zustandsdiskreten Modellansatz verfolgt, d.h. ohne Beschriftung der Phasentransitionen auskommt, entfällt die Möglichkeit, durch gleichbenannte Phasentransitionen (zu unterschiedlichen Phasen) Nichtdeterminismus (nondeterministic choices) zu erreichen. Äquivalenzen, die bei Nichtdeterminismus feinere Unterscheidungen erlauben (z.B. Failure- oder Bisimulations-Äquivalenzen (siehe [32])), bringen bei I–Systemen keinen Gewinn.
Die folgende Definition von Trace-Äquivalenz bei I–Systemen stützt sich auf die bisher eingeführten Trace-Semantiken Verhalten, Casetrace-Semantik, Erweiterte Casetrace-Semantik sowie deren Interleaving Varianten. Um die geforderte Gleichheit des Systemverhaltens auf bestimmte Bereiche zu begrenzen, werden zur Definition die Projektionsvarianten der Trace-Semantiken (definiert in Kapitel 9) herangezogen.

Definition 11.1 (Trace-Äquivalenz von I–Systemen). Sei IS_1 ein I–System mit Bereichsmenge B_1 und IS_2 ein I–System mit Bereichsmenge B_2. Sei $T \subseteq B_1 \cap B_2$. Für $S \in \{\mathcal{V}, \mathcal{V}^{\mathrm{i}}, \mathcal{CT}, \mathcal{CT}^{\mathrm{i}}, \mathcal{ECT}, \mathcal{ECT}^{\mathrm{i}}\}$ wird die Äquivalenzrelation $\sim_{S,T} \subseteq ISystem \times ISystem$ festgelegt durch:

$$IS_1 \sim_{S,T} IS_2 \quad \Leftrightarrow \quad S[\![IS_1]\!]\lfloor_T = S[\![IS_2]\!]\lfloor_T \qquad \square$$

Laut obiger Definition ist die Trace-Äquivalenz von I–Systemen in Abhängigkeit von einer Trace-Semantik S und einer Menge von Bereichen T definiert. Zwei I–Systeme gelten genau dann als äquivalent bezüglich S und T, wenn deren Systemverhalten, welches durch die Semantik S beschrieben wird, bei Sicht auf die Bereiche aus T identisch sind. Die Aktivitäten in den Restbereichen sowie die Anzahl der Restbereiche sind für die Äquivalenz nicht relevant. dass es sich bei der angegebenen Relation $\sim_{S,T}$ um eine Äquivalenzrelation handelt, ist offensichtlich. Die Reflexivität, Symmetrie und Transitivität ergeben sich direkt aus der geforderten Gleichheit der Semantiken mit Sicht auf T. Gleichheit ist bekanntermaßen eine Äquivalenzrelation.

Analog zur Trace-Äquivalenz kann eine Graph-Äquivalenz von I–Systemen formuliert werden, indem anstelle der Trace-Semantiken die Zustandsgraphen mit Sicht auf T (aus Kapitel 9.6) verglichen werden.

Definition 11.2 (Graph-Äquivalenz von I–Systemen). Sei IS_1 ein I–System mit Bereichsmenge B_1 und IS_2 ein I–System mit Bereichsmenge B_2. Sei $T \subseteq B_1 \cap B_2$. Für $G \in \{VG, CG, ECG\}$ wird die Äquivalenzrelation $\sim_{G,T} \subseteq ISystem \times ISystem$ festgelegt durch:

$$IS_1 \sim_{G,T} IS_2 \quad \Leftrightarrow \quad G(IS_1)\lfloor_T = G(IS_2)\lfloor_T \qquad \square$$

Auf Graph-Äquivalenzen wird in diesem Kapitel nicht weiter eingegangen, da die Trace-Äquivalenzen ausreichen, um die grundlegenden Ideen zur formalen Handhabung von I–System-Transformationen vorzustellen. Die Ansätze lassen sich ohne weiteres auf Graph-Äquivalenzen übertragen.

11.4 Beispiele für I–System-Transformationen

Ausgangspunkt sei das I–System IS_{15} aus Abbildung 11.1. Es besteht aus einem Relevanzbereich b_0 und drei Kontrollbereichen $\underline{b}_1$, $\underline{b}_2$, $\underline{b}_3$. Betrachtet man die vier Bereiche als Repräsentanten interagierender Prozesse eines Betriebssystems, dann haben letztere drei die Aufgabe den Prozessablauf des ersten zu steuern. Die Struktur von IS_{15} resultiert aus der dreimaligen Einbringung des Strukturbausteins $NOT(\cdot\,|\,\cdot)$ aus Kapitel 10.2.3.
Als Semantik, auf die in diesem Beispiel bei der Formulierung von Systemeigenschaften Bezug genommen wird, wird die Casetrace-Semantik und deren Projektion auf Teilsysteme verwendet.

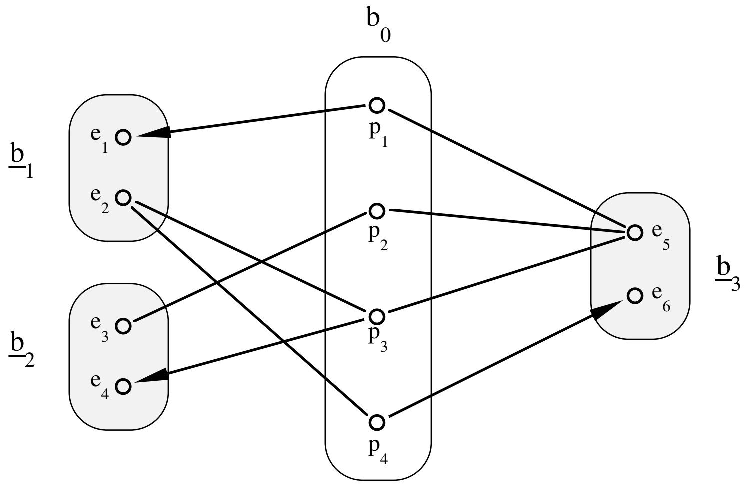


Abbildung 11.1: IS_{15}; Relevanzbereich b_0 mit drei Kontrollbereichen

Die Casetrace-Semantik stellt einen konzeptionell einfachen Vertreter semantischer Beschreibungsformen bei I–Systemen dar und ist geeignet, die Prinzipien der I–System Transformationen anschaulich demonstrieren zu können. Für die Casetrace-Semantik $\mathcal{CT}[\![IS_{15}]\!]$ von IS_{15} lassen sich die folgenden strukturellen Eigenschaften angeben.

Satz 11.3 (Eigenschaften von $\mathcal{CT}[\![IS_{15}]\!]$). Sei IS_{15} das I–System aus Abbildung 11.1 und $c_0c_1c_2 \ldots \in \mathcal{CT}[\![IS_{15}]\!]$ mit $c_j \in Case(IS_{15}), j = 0, 1, 2, \ldots$. Dann gilt für alle $i = 1, 2, \ldots$:

a) $(p_1 \in c_{i-1}) \Rightarrow (\forall q \in \{p_3, p_4\} : q \notin c_i)$

b) $(p_3 \in c_{i-1}) \Rightarrow (p_2 \notin c_i)$

c) $(p_4 \in c_{i-1}) \Rightarrow (\forall q \in \{p_1, p_2, p_3\} : q \notin c_i)$

Beweis. Der Bereich b_0 bildet zusammen mit jedem der Bereiche $\underline{b}_1$, $\underline{b}_2$, $\underline{b}_3$ und den verbindenden Teilen der Kopplungs- und Erregungsrelation einen Strukturbaustein $NOT(p|q_1, \ldots, q_n)$ gemäß Satz 10.12, Notation 10.13.
Aussage a) gilt nach Satz 10.12.b mit $p := p_1$ und $\{q_1, \ldots, q_n\} := \{p_3, p_4\}$.
Aussage b) gilt nach Satz 10.12.b mit $p := p_3$ und $\{q_1, \ldots, q_n\} := \{p_2\}$.
Aussage c) gilt nach Satz 10.12.b mit $p := p_4$ und $\{q_1, \ldots, q_n\} := \{p_1, p_2, p_3\}$. □

11.4.1 Zusammenlegung von Kontrollbereichen

Das Ziel ist nun eine Transformation von IS_{15} in ein bezüglich $\sim_{CT,\{b_0\}}$ äquivalentes I–System IS_{16} mit einer geringeren Anzahl von Kontrollbereichen. Da bei der Transformation die Ereignisstruktur in dem (einzigen) Relevanzbereich b_0 bewahrt werden soll, wird bei der Formalisierung der Übergang zur Casetrace-Semantik mit Sicht auf $\{b_0\}$ vollzogen.
Die Ereignisstruktur in b_0 lässt sich anhand der dort möglichen bzw. unmöglichen Phasentransitionen näher beschreiben.

Satz 11.4 (Ausgeschlossene Phasentransitionen in b_0 bei IS_{15}). Sei IS_{15} das I–System aus Abbildung 11.1 und $\{p'_0\}\{p'_1\}\{p'_2\} \ldots \in \mathcal{CT}[\![IS_{15}]\!]\lfloor_{\{b_0\}}$ mit $p'_j \in b_0, j = 0, 1, 2, \ldots$. Dann gilt für alle $i = 1, 2, \ldots$:

a) $(p'_{i-1} = p_1) \Rightarrow (p'_i \notin \{p_3, p_4\})$

b) $(p'_{i-1} = p_3) \Rightarrow (p'_i \neq p_2)$

c) $(p'_{i-1} = p_4) \Rightarrow (p'_i \notin \{p_1, p_2, p_3\})$

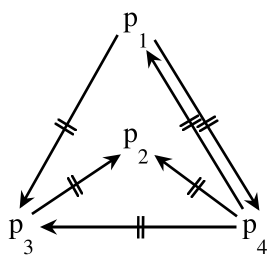


Abbildung 11.2: Ausgeschlossene Phasentransitionen in b_0 bei IS_{15}

Beweis. Unter Berücksichtigung der Berechnung von $\mathcal{CT}[\![IS_{15}]\!]\lfloor_T$ aus $\mathcal{CT}[\![IS_{15}]\!]$ entsprechend Definition 9.8, sind die Aussagen a), b) und c) direkte Folgerungen aus Satz 11.3.a, 11.3.b und 11.3.c, da hier $T = \{b_0\} = \{\{p_1, p_2, p_3, p_4\}\}$. □

Durch den Satz werden in dem Bereich b_0 die in der Abbildung 11.2 dargestellten Phasentransitionen ausgeschlossen. Für jede Phase, von der ausgehend Phasentransitionen ausgeschlossen sind, existiert ein Kontrollbereich in IS_{15} als Teil eines Strukturbausteins $NOT(\cdot\,|\,\cdot)$. p_1 wird von $\underline{b}_1$ „kontrolliert", p_3 von $\underline{b}_2$ und p_4 von $\underline{b}_3$.

Als Beispiel einer Transformation von IS_{15}, mit dem Ziel die Anzahl der Kontrollbereiche herabzusetzen, lassen sich die beiden trägen Bereiche $\underline{b}_1$ und $\underline{b}_2$ zu einem Kontrollbereich $\underline{b}_{12}$ zusammenfassen. $\underline{b}_{12}$ wird durch eine geeignete Anpassung der Kopplungs- und Erregungsrelation mit b_0 verbunden. Der träge Bereich $\underline{b}_3$ und dessen Anbindung an b_0 bleiben unverändert. Das resultierende I–System IS_{16} ist in Abbildung 11.3 dargestellt.

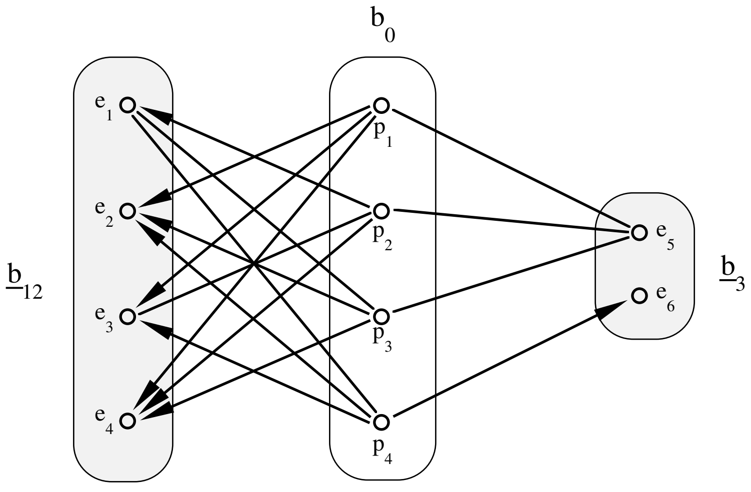
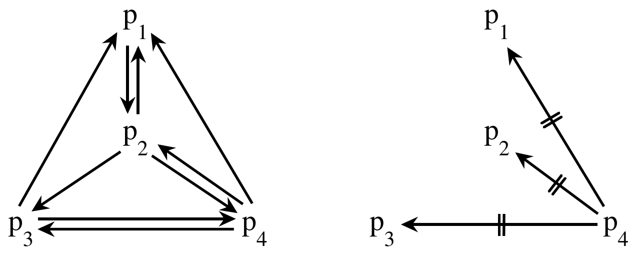


Abbildung 11.3: IS_{16}; Relevanzbereich b_0 mit zwei Kontrollbereichen

Vergleicht man IS_{16} mit IS_{15}, dann zeigt sich, dass trotz einer Reduzierung der Anzahl der Kontrollbereiche die Gesamtanzahl der Phasen in den Kontrollbereichen gleich geblieben ist, allerdings auf Kosten einer Erhöhung der Mächtigkeit der Kopplungs- und Erregungsrelation. Entscheidend ist jedoch, dass die Transformation keinen Einfluss auf das durch die Casetrace-Semantik mit Sicht auf $\{b_0\}$ erfassbare Verhalten in dem Relevanzbereich b_0 hat. Dies bringt der folgende Satz zum Ausdruck.



(a) möglich ohne $\underline{b}_3$ (b) verboten durch $\underline{b}_3$

Abbildung 11.4: Phasentransitionen in b_0 bei IS_{16}

Satz 11.5 (Äquivalenz von IS_{15} und IS_{16}). Seien IS_{15} und IS_{16} die I–Systeme aus den Abbildungen 11.1 und 11.3. Es gilt:

$$IS_{15} \sim_{\mathcal{CT},\{b_0\}} IS_{16}$$

Beweis. Auf die präzise Ausarbeitung des Beweises von Satz 11.5 wird im Rahmen dieses Beispiels verzichtet. Eine „intuitive Korrektheit“ lässt sich wie folgt begründen:
In dem Teilsystem $IS_{16}\lfloor_{\{\underline{b}_{12},b_0\}}$ sind aufgrund der Verbindung (mittels Kopplungs- und Erregungsrelation) mit $\underline{b}_{12}$ in b_0 nur Phasentransitionen erlaubt, die in Abbildung 11.4.a dargestellt sind. Dies folgt aus Satz 9.22.b, wobei hier nur freie Phasentransitionen zu berücksichtigen sind (d.h. in Satz 9.22 gilt $\rightarrow_2 = \emptyset$). Alle in Abbildung 11.4.a nicht eingezeichneten Phasentransitionen können auch nicht auftreten.
Das Teilsystem $IS_{16}\lfloor_{\{b_0,\underline{b}_3\}}$ beinhaltet den Strukturbaustein $NOT(p_4|p_1,p_2,p_3)$, durch den laut Satz 10.12 in b_0 alle von p_4 ausgehenden Phasentransitionen ausgeschlossen werden, dargestellt in Abbildung 11.4.b.
Die Überlagerung der beiden Teilsysteme $IS_{16}\lfloor_{\{\underline{b}_{12},b_0\}}$ und $IS_{16}\lfloor_{\{b_0,\underline{b}_3\}}$ führt dazu, dass in b_0 jede Phasentransition ausgeschlossen ist, die nicht in Abbildung 11.4.a oder aber in Abbildung 11.4.b eingezeichnet ist. Zusammenfassend werden bei IS_{16} alle Transitionen aus Abbildung 11.2 in b_0 verhindert. Dies legt nahe, dass das Verhalten in b_0 bei IS_{16} dem Verhalten in b_0 bei IS_{15} entspricht, woraus sich dann die zu zeigende Äquivalenz der beiden I–Systeme ergibt. □

Bemerkung 11.6. Allgemein ergibt sich die Korrektheit solcher Äquivalenzen aus der Korrektheit der Transformationsregeln, die angewendet wurden. Neben der Beschreibung der syntaktischen Umformungsanweisungen muss für jede neue Transformationsregel bewiesen werden, dass sich durch deren Anwendung das in die Semantik einfließende Verhalten in den Relevanzbereichen des betrachteten I–Systems nicht ändert. Die relevanten Einflüsse der umgeformten Systemumgebung auf die Relevanzbereiche müssen den relevanten Einflüssen der Ursprungsumgebung entsprechen. Wie bereits erwähnt wurde, wird die Entwicklung umfassender Transformationsregeln und damit auch deren Korrektheitsbeweise ein Inhalt weiterführender Forschungsarbeiten sein.

Ausgehend von IS_{16} lassen sich nun in einem weiteren Transformationsschritt auch noch die Kontrollbereiche $\underline{b}_{12}$ und $\underline{b}_3$ zusammenfassen. Das resultierende I–System IS_{17} ist in Abbildung 11.5 dargestellt.
Der Vergleich von IS_{17} mit IS_{16} macht deutlich, dass mit der Einsparung des Kontrollbereiches $\underline{b}_3$ zugleich die Gesamtanzahl der Phasen um zwei reduziert wurde. Es sind keine ergänzenden Phasen notwendig, um den Verlust von $\underline{b}_3$ aufzufangen. Des Weiteren hat die Mächtigkeit der Erregungsrelation abgenommen, da die Erregungsbeziehungen von e_4 zu Kopplungsbeziehungen geworden sind und die Erregungsbeziehung von e_6 entfällt. Die Mächtigkeit der Kopplungsrelation ist insgesamt gleich geblieben. Zusammenfassend hat dieser Transformationsschritt in diesem Beispiel eine deutliche Vereinfachung des Gesamtsystems bewirkt. Sich daraus ergebende mögliche Vorteile für eine Systemanalyse oder Implementierungsphase wurden in Abschnitt 11.1 aufgezeigt.
Der folgende Satz unterlegt, dass auch dieser Transformationsschritt keinen Einfluss auf das durch die Casetrace-Semantik mit Sicht auf $\{b_0\}$ erfassbare Verhalten in dem Bereich b_0 hat.

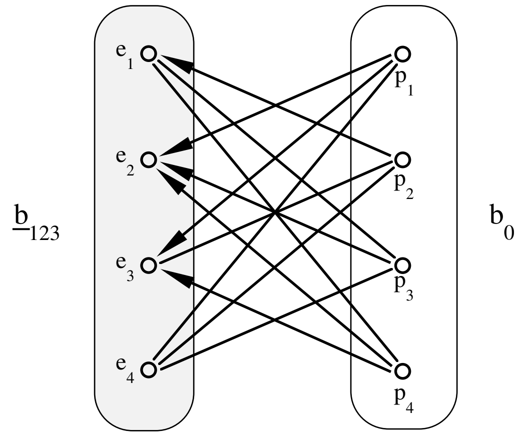


Abbildung 11.5: IS_{17}; Relevanzbereich b_0 mit einem Kontrollbereich

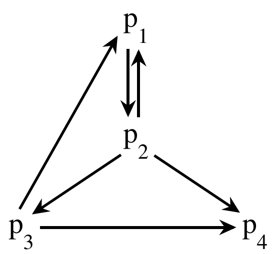


Abbildung 11.6: Mögliche Phasentransitionen in b_0 bei IS_{17}

Satz 11.7 (Äquivalenz von $\boldsymbol{IS_{16}}$ und $\boldsymbol{IS_{17}}$). Seien IS_{16} und IS_{17} die I–Systeme aus den Abbildungen 11.3 und 11.5. Es gilt:

$$IS_{16} \sim_{\mathcal{CT},\{b_0\}} IS_{17}$$

Beweis. Wie schon bei Satz 11.5 wird hier erneut auf eine präzise formale Ausarbeitung des Beweises verzichtet und auf die Bemerkung 11.6 in dem Beweis von Satz 11.5 verwiesen.

Eine „intuitive Korrektheit“ lässt sich wie folgt begründen:
IS_{17} erfüllt die strukturellen Voraussetzungen von Satz 9.22 unter Bezug auf den Graphen in Abbildung 11.6. Es sind hierbei nur freie Phasentransitionen zu berücksichtigen, d.h. es wird $\rightarrow_2 = \emptyset$ vorausgesetzt. Aus der Satzaussage 9.22.b lässt sich ableiten, dass in b_0 genau die Phasentransitionen möglich sind, die als Kanten in dem Graphen enthalten sind. Diese entsprechen nun genau den Transitionen aus Abbildung 11.4.a abzüglich der Transitionen aus Abbildung 11.4.b, also den Phasentransitionen, die das Verhalten in b_0 bei IS_{16} repräsentieren. Es ist somit nahe liegend, dass die zu zeigende Äquivalenz der I–Systeme IS_{17} und IS_{16} gilt. □

11.4.2 Herleitung bekannter Teilstrukturen

Die Kombination der Sätze 11.5 und 11.7 liefert (da $\sim_{\mathcal{CT},\{b_0\}}$ eine Äquivalenzrelation ist) direkt $IS_{15} \sim_{\mathcal{CT},\{b_0\}} IS_{17}$. Somit lassen sich Eigenschaften, die für $\mathcal{CT}[\![IS_{17}]\!]|_{\{b_0\}}$ bekannt (bewiesen) sind, ohne weiteren Beweis auf $\mathcal{CT}[\![IS_{15}]\!]|_{\{b_0\}}$ übertragen, um dadurch das Systemverhalten des Ausgangs-I–Systems IS_{15} genauer zu beschreiben, ohne zusätzlichen Aufwand in dessen Analyse investieren zu müssen. Im Rahmen des Beispiels lässt sich für IS_{17} die folgende Eigenschaft festhalten:

Satz 11.8 (Festgelegte Phasenfolgen in b_0 bei IS_{17}). Sei IS_{17} das I–System aus Abbildung 11.5. Es gilt:

$\mathcal{CT}[\![IS_{17}]\!]\downharpoonright_{\{b_0\}} = \{\{p'_0\}\{p'_1\}\{p'_2\}\ldots\infty \mid \forall j \in I\!N_0 : p'_j \in b_0 \text{ und } \forall i \in I\!N : A_i\} \cup \{\{p'_0\}\{p'_1\}\{p'_2\}\ldots\{p_n\} \mid n \in I\!N, \forall j \in \{0,1,2,\ldots,n\} : p'_j \in b_0 \text{ und } \forall i \in \{1,2,\ldots,n\} : A_i\}$

mit

$A_i \equiv (((p'_{i-1} = p_1) \Rightarrow (p'_i = p_2)) \wedge ((p'_{i-1} = p_2) \Rightarrow (p'_i \in \{p_1,p_3,p_4\})) \wedge ((p'_{i-1} = p_3) \Rightarrow (p'_i \in \{p_1,p_4\})) \wedge ((p'_{i-1} \neq p_4)))$.

Beweis. In Verbindung mit dem Graphen in Abbildung 11.6, als gerichtete schlingenfreie Graphenstruktur $G(b_0) = (b_0, \rightarrow_1, \rightarrow_2)$ auf b_0, mit der abgebildeten Kantenmenge als $\rightarrow_1$ und $\rightarrow_2 = \emptyset$, erfüllt IS_{17} die strukturellen Voraussetzungen von Satz 9.22. Somit gilt dessen Aussage a):

$\mathcal{ECT}[\![IS_{17}]\!]\downharpoonright_{\{b_0\}} = \{\{p'_0\}\delta_1\{p'_1\}\delta_2\{p'_2\}\delta_3\ldots\infty \mid \langle p'_0,p'_1,p'_2,\ldots\infty\rangle$ ist unendlicher Pfad in $G(b_0)$ und $\forall i \in I\!N : (\delta_i = \{b_0\} \wedge (p'_{i-1},p'_i) \in\rightarrow_1) \vee (\delta_i = \emptyset \wedge (p'_{i-1},p'_i) \in\rightarrow_2)\} \cup \{\{p'_0\}\delta_1\{p'_1\}\delta_2\{p'_2\}\delta_3\ldots\delta_n\{p'_n\} \mid \langle p'_0,p'_1,p'_2,\ldots,p'_n\rangle$ mit $n \in I\!N$ ist endlicher Pfad in $G(b_0)$ und $(\forall i \in \{1,\ldots,n\} : (\delta_i = \{b_0\} \wedge (p'_{i-1},p'_i) \in\rightarrow_1) \vee (\delta_i = \emptyset \wedge (p'_{i-1},p'_i) \in\rightarrow_2)) \wedge (\rightarrow_2(p'_n) = \emptyset)\}$.

Beim Übergang von der Erweiterten Casetrace-Semantik zur Casetrace-Semantik entfallen die Bereichsmengen in den Traces (vgl. Satz 4.20.d), also:

$\mathcal{CT}[\![IS_{17}]\!]\downharpoonright_{\{b_0\}} = \{\{p'_0\}\{p'_1\}\{p'_2\}\ldots\infty \mid \langle p'_0,p'_1,p'_2,\ldots\infty\rangle$ ist unendlicher Pfad in $G(b_0)$ und $\forall i \in I\!N : ((p'_{i-1},p'_i) \in\rightarrow_1 \vee (p'_{i-1},p'_i) \in\rightarrow_2)\} \cup \{\{p'_0\}\{p'_1\}\{p'_2\}\ldots\{p'_n\} \mid \langle p'_0,p'_1,p'_2,\ldots,p'_n\rangle$ mit $n \in I\!N$ ist endlicher Pfad in $G(b_0)$ und $(\forall i \in \{1,\ldots,n\} : ((p'_{i-1},p'_i) \in\rightarrow_1 \vee (p'_{i-1},p'_i) \in\rightarrow_2)) \wedge (\rightarrow_2 (p'_n) = \emptyset)\}$.

Es gilt nach Voraussetzung $\rightarrow_2 = \emptyset$, woraus folgt:

$\mathcal{CT}[\![IS_{17}]\!]\downharpoonright_{\{b_0\}} = \{\{p'_0\}\{p'_1\}\{p'_2\}\ldots\infty \mid \langle p'_0,p'_1,p'_2,\ldots\infty\rangle$ ist unendlicher Pfad in $G(b_0)$ und $\forall i \in I\!N : (p'_{i-1},p'_i) \in\rightarrow_1\} \cup \{\{p'_0\}\{p'_1\}\{p'_2\}\ldots\{p'_n\} \mid \langle p'_0,p'_1,p'_2,\ldots,p'_n\rangle$ mit $n \in I\!N$ ist endlicher Pfad in $G(b_0)$ und $\forall i \in \{1,\ldots,n\} : (p'_{i-1},p'_i) \in\rightarrow_1\}$.

Berücksichtigt man die Anordnung der Kanten in dem Graphen, erhält man die Formulierung für $\mathcal{CT}[\![IS_{17}]\!]\downharpoonright_{\{b_0\}}$ aus dem Satz. □

Satz 11.8 (bezogen auf IS_{17}) ist in seiner Aussage schärfer als Satz 11.4 (bezogen auf IS_{15}). Er gibt an, dass Abbildung 11.6 als Kanten des Graphens tatsächlich alle in b_0 möglichen Phasentransitionen beinhaltet und die nicht eingezeichneten Phasentransitionen ausgeschlossen sind. Invers betrachtet bedeutet das, dass Abbildung 11.2 genau die in b_0 ausgeschlossenen Phasentransitionen beinhaltet und die nicht eingezeichneten Phasentransitionen immer möglich sind. Letztere Möglichkeiten lassen sich Satz 11.4 nicht entnehmen. Von daher folgt Satz 11.4 aus Satz 11.8, wo hingegen die Umkehrrichtung nicht gilt.
Die Umformung von IS_{15} hin zu IS_{17} führt demnach dazu, dass mit letzterem ein I–System hergeleitet wird, dessen Ereignisstruktur in b_0 im Detail bekannt ist. Die aufgezeigte Äquivalenz von IS_{15} und IS_{17} erlaubt es nun, dieses Wissen für IS_{15} zu übernehmen, ohne eine tief gehende Analyse der Dynamik (Untersuchung der Ausführungen von $V_I System(IS_{15})$) betreiben zu müssen. Das „einfach“ gewonnene Zusatzwissen über IS_{15} kann schließlich bei der Verifikation von Systemanforderungen eingesetzt werden.

11.4.3 Redundante Teilstrukturen

In einem I–System ist eine Teilstruktur (z.B. Phase, Bereich, Relation, Teilsystem) als redundant anzusehen, wenn sie innerhalb des Gesamtsystems eine (notwendige) Funktionalität repräsentiert, die bereits durch eine andere Teilstruktur abgedeckt wird. Der Wegfall einer redundanten Teilstruktur hat keine Auswirkungen auf das durch die betrachtete Semantik mit Sicht auf die Relevanzbereiche erfassbare Systemverhalten. Allerdings ist der Wegfall mit einer Verkleinerung des globalen Zustandsraumes verbunden, was sich in der Regel vorteilhaft in punkto Laufzeit auf eine Systemanalyse auswirkt, insbesondere dann, wenn die Erreichbarkeit von Systemsituationen

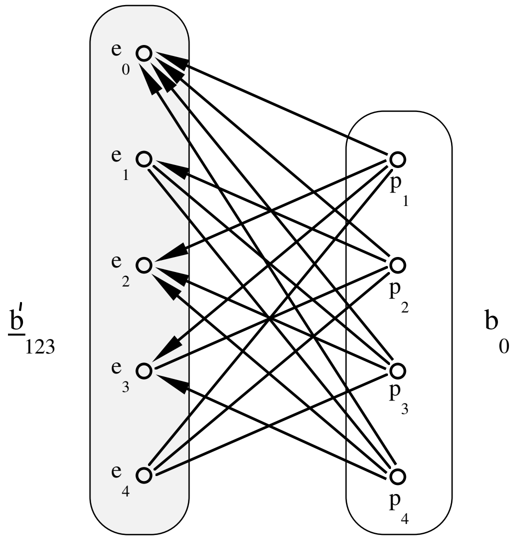


Abbildung 11.7: IS_{18}; redundante Phase e_0

untersucht wird. Ein Ziel von I–System-Transformationen ist es, redundante Teilstrukturen zu beseitigen. Im Folgenden soll anhand zweier Beispiele das Vorliegen solcher redundanter Strukturen demonstriert werden.

Als erstes Beispiel dient das I–System IS_{18} aus Abbildung 11.7. Offensichtlich unterscheidet es sich von IS_{17} nur um die zusätzliche Phase e_0 im trägen Bereich $\underline{b}'_{123}$ und deren Anbindung an das Restsystem mittels Kopplungsrelation (innerhalb $\underline{b}'_{123}$) und Erregungsrelation (mit b_0). Trotz der Unterschiede in der syntaktischen Struktur sind beide I–Systeme bezüglich $\sim_{\mathcal{CT},\{b_0\}}$ semantisch äquivalent.

Satz 11.9 (Äquivalenz von IS_{17} und IS_{18}). Seien IS_{17} und IS_{18} die I–Systeme aus den Abbildungen 11.5 und 11.7. Es gilt:

$$IS_{17} \sim_{\mathcal{CT},\{b_0\}} IS_{18}$$

Beweis. In Verbindung mit dem Graphen aus Abbildung 11.6, als gerichtete schlingenfreie Graphenstruktur $G(b_0) = (b_0, \rightarrow_1, \rightarrow_2)$ auf b_0, mit der abgebildeten Kantenmenge als $\rightarrow_1$ und $\rightarrow_2 = \emptyset$, erfüllt IS_{18} die strukturellen Voraussetzungen von Satz 9.20. Somit gilt dessen Aussage a) mit $\mathcal{ECT}[\![IS_{18}]\!]\lfloor_{\{b_0\}} = \ldots.$
Nun hat $G(b_0)$ die Eigenschaft, dass jeder seiner Knoten mindestens einen Vorgängerknoten besitzt. In Verbindung mit IS_{17} sind somit auch die strukturellen Voraussetzungen von Satz 9.22 erfüllt, und es gilt dort die Aussage a) mit $\mathcal{ECT}[\![IS_{17}]\!]\lfloor_{\{b_0\}} = \ldots.$
Die rechten Seiten der Gleichungen von Satz 9.20.a und Satz 9.22.a sind identisch, also gilt $\mathcal{ECT}[\![IS_{18}]\!]\lfloor_{\{b_0\}} = \mathcal{ECT}[\![IS_{17}]\!]\lfloor_{\{b_0\}}$, und somit erst recht $\mathcal{CT}[\![IS_{18}]\!]\lfloor_{\{b_0\}} = \mathcal{CT}[\![IS_{17}]\!]\lfloor_{\{b_0\}}$. □

Die Äquivalenz von IS_{17} und IS_{18} zeigt, dass für dieses Beispiel die Existenz und Einbindung von e_0 unbedeutsam und deren Funktionalität schon durch die Phasen e_1 bis e_4 und deren Verbindungen mit b_0 bereitgestellt wird. Es ist vorstellbar, dass IS_{18} beispielsweise in einer Modellierungsphase durch eine Konstruktion gemäß Satz 9.20 entstanden ist, ohne dass der Spezialfallcharakter und die dadurch mögliche Konstruktion gemäß Satz 9.22 berücksichtigt wurde. Eine Transformation von IS_{18} in IS_{17} kann die Redundanz beseitigen und damit das System vereinfachen (für die Anschauung, Analyse oder spätere Implementierung). Eine analoge Situation ergibt sich beim Vergleich von IS_9 aus Abbildung 9.10 mit IS_8 aus Abbildung 9.9 (Kapitel 9.7).

Als zweites Beispiel für das Vorliegen einer redundanten Teilstruktur dient das I–System IS_{19} aus Abbildung 11.8. Es unterscheidet sich von IS_{17} um den zusätzlichen Bereich $\underline{b}_3$ und dessen

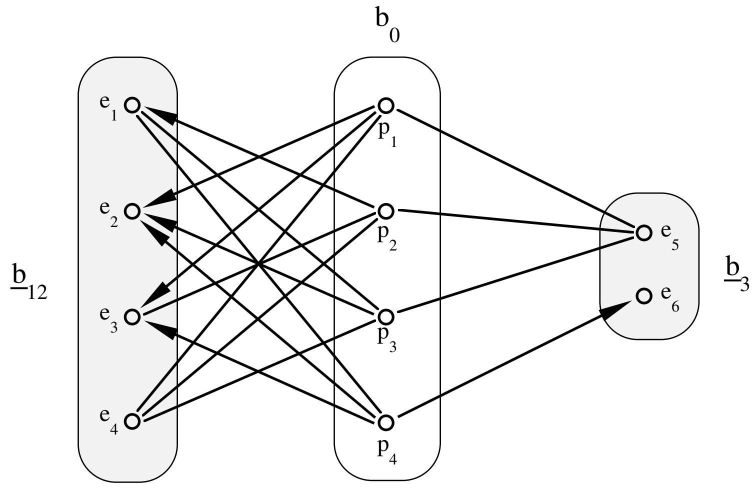


Abbildung 11.8: IS_{19}; redundanter Bereich $\underline{b}_3$

Anbindung an b_0 mittels Kopplungs- und Erregungsrelation. Trotz dieser Erweiterung und der mit ihr verbundenen Einschränkung des Ereignisraumes bei b_0, ist IS_{19} bezüglich $\sim_{\mathcal{CT},\{b_0\}}$ semantisch äquivalent zu IS_{17}.

Satz 11.10 (Äquivalenz von IS_{17} und IS_{19}). Seien IS_{17} und IS_{19} die I–Systeme aus den Abbildungen 11.5 und 11.8. Es gilt:

$$IS_{17} \sim_{\mathcal{CT},\{b_0\}} IS_{19}$$

Beweis. Auf eine formale Ausarbeitung wird an dieser Stelle verzichtet. Eine „intuitive Korrektheit" begründet sich wie folgt:
Aus Satz 11.8 folgt, dass in dem Teilsystem $IS_{19}\lfloor_{\{\underline{b}_{12},b_0\}}$ genau die Phasenabfolgen in b_0 auftreten können, zu denen es einen entsprechenden Pfad in dem Graphen aus Abbildung 11.6 gibt.
Das Teilsystem $IS_{19}\lfloor_{\{b_0,\underline{b}_3\}}$ beinhaltet den Strukturbaustein $NOT(p_4|p_1,p_2,p_3)$, durch den unter Beachtung von Satz 10.12 in b_0 alle von p_4 ausgehenden Phasentransitionen ausgeschlossen werden, dargestellt in Abbildung 11.4.b.
Aus der Disjunktheit der Kantenmengen der Graphen aus Abbildung 11.6 und Abbildung 11.4.b lässt sich schließen, dass in dem Ergebnis der Überlagerung von $IS_{19}\lfloor_{\{\underline{b}_{12},b_0\}}$ und $IS_{19}\lfloor_{\{b_0,\underline{b}_3\}}$, d.h. in IS_{19}, durch den Strukturbaustein $NOT(p_4|p_1,p_2,p_3)$ nur Phasentransitionen ausgeschlossen werden, die allein schon aufgrund des Vorhandenseins von $\underline{b}_{12}$ und dessen Verbindung mit b_0 (mittels Kopplungs- und Erregungsrelation) nicht auftreten können.
Es folgt: $\mathcal{CT}[\![IS_{19}]\!]\lfloor_{\{b_0\}} = \underbrace{\mathcal{CT}[\![IS_{19}\lfloor_{\{\underline{b}_{12},b_0\}}]\!]\lfloor_{\{b_0\}}}_{= \mathcal{CT}[\![IS_{17}]\!]\lfloor_{\{b_0\}}}$. □

Die gezeigte Äquivalenz von IS_{17} und IS_{19} bedeutet für dieses Beispiel, dass die Anbindung des Kontrollbereiches $\underline{b}_3$ (bei IS_{19}) unbedeutsam ist. Durch den Einfluss von $\underline{b}_3$ sollen in b_0 Phasentransitionen verhindert werden, deren Eintreten bereits durch den Einfluss von $\underline{b}_{12}$ ausgeschlossen werden. Es ist vorstellbar, dass IS_{19} im Rahmen eines inkrementellen Modellierungsschrittes entstanden ist, durch die unabhängige Behandlung lokaler Synchronisationsanforderungen und eine damit verbundene Einbringung der Bereiche $\underline{b}_{12}$ und $\underline{b}_3$, ohne die Überschneidung in der Funktionalität wahrzunehmen. Eine Transformation von IS_{19} in IS_{17} beseitigt die Redundanz und vereinfacht das Gesamtsystem.

Abschließend sei darauf hingewiesen, dass die Existenz redundanter Teilstrukturen nicht von vornherein als unerwünschte Systemeigenschaft eingeordnet werden darf. In verteilten Systemen dienen redundante Komponenten der Fehlertoleranz im Hinblick auf Ausfallsicherheit [23, 73]. Somit ist es bei bestimmten Anwendungen sogar gewollt, redundante Teilstrukturen aufzubauen oder deren Existenz am Modell zu beweisen, ohne sie dann über I–System-Transformationen zu beseitigen.

11.4.4 Anmerkungen

Die Durchführbarkeit der in den vorangegangenen Abschnitten vorgestellten oder vergleichbarer I–System-Transformationen in einem größeren Rahmen mit wesentlich komplexeren I–Systemen steht und fällt mit der Formulierung einer umfassenden Menge korrekter Transformationsregeln. Mit dieser nicht-trivialen Thematik werden sich zukünftige Forschungsarbeiten befassen. Es muss sich dann zeigen, welche Art von Transformationsregeln sich entwickeln lassen und inwieweit eine Modellanpassung mittels Zusatzbeschriftungen in Kauf genommen werden muss, trotz der am Anfang dieses Kapitels erwähnten Nachteile.

In den Abschnitten 11.1 und 11.4 wurde beschrieben, dass eine schrittweise Transformation eines I–Systems IS_a in ein I–System IS_b dazu führen kann, dass sich eine Analyse von IS_a effizienter gestalten lässt, indem bekannte Eigenschaften des äquivalenten IS_b mit einbezogen werden. Damit diese Argumentation gilt, muss der Gewinn an Effizienz bei der Analyse die Kosten für die Transformationen aufwiegen. Folglich ist es erforderlich, dass Komplexitätsbetrachtungen die Entwicklung von Transformationsregeln begleiten.

Vor der Durchführung einer I–System-Transformation muss die Anwendbarkeit einer Transformationsregel festgestellt werden, d.h. es muss eine Teilstruktur im Gesamtsystem erkannt werden, die zu einer der vorhandenen Transformationsregeln passt. In dem Beispiel-Abschnitt 11.4 erfolgte dies durch „scharfes Hinsehen“. Ein zukünftiges Ziel wird es sein, diesen Vorgang mittels effizienter Algorithmen zu automatisieren.

In dem gesamten Abschnitt 11.4 bezogen sich die semantischen Äquivalenzen ausschließlich auf die „einfache“ Casetrace-Semantik $\mathcal{CT}[\![\cdot]\!]$ mit Sicht auf einen einzigen Relevanzbereich b_0. Diese Fokussierung ist ausreichend, um die grundlegenden Prinzipien der I–System-Transformationen beispielhaft zu präsentieren. Je nach praktischer Anwendung sind hingegen auch die Erweiterte Casetrace-Semantik $\mathcal{ECT}[\![\cdot]\!]$ und das Verhalten $\mathcal{V}[\![\cdot]\!]$ (oder die entsprechenden Zustandsgraphen) mit Sicht auf mehr als einen Relevanzbereich von Interesse. Es fällt in den Bereich der zukünftigen Forschungsarbeiten, für diese geeignete Transformationsregeln und repräsentative Beispiele zu finden.

Kapitel 12

Schlussbetrachtung

12.1 Fazit

Das Ziel dieser Arbeit war es, basierend auf der Bearbeitung dreier Schwerpunktthemen, die Theorie der I–Systeme entscheidend weiter zu entwickeln (siehe Kapitel 1.2). Bezogen auf die einzelnen Schwerpunkte lassen sich folgende erreichte Ergebnisse herausstellen:

Zum Schwerpunkt „Realisierung der Modellierungsebenen“:

Es ist gelungen, die in der Motivation (Kapitel 1.1) präsentierten Modellierungsebenen als festen Bestandteil in das formale Modell aufzunehmen. Die 4 Modellierungsebenen und deren zentrale Inhalte und Schnittstellen sind in Kapitel 7.1 beschrieben. Zur Verwirklichung der Strukturierung war es notwendig, die bisherigen Begriffe und Definitionen des formalen Modells zu überarbeiten. In diesem Rahmen wurde auch die graphische Repräsentation der I–Systeme verbessert.
Ein Problem war die Formulierung und Integration einer eindeutigen Dynamik. Gelöst wurde es durch die Einführung der V_ISysteme als Ausführungsinstrument für programmiersprachlich spezifizierte verteilte Algorithmen, so genannte Aktionen (Kapitel 3). Die Anschauungsebene liefert zum leichteren Verständnis der Aktionen Interpretationen im Kontext wechselseitiger Einflüsse und deren Auswirkungen. Dieser neuartige Ansatz hat sich als vorteilhaft gegenüber früheren Ansätzen mit umgangssprachlichen, teils zweideutigen Formulierungen von Verhaltensanweisungen herausgestellt, insbesondere bei formalen Beweisführungen, die auf einer Analyse der Dynamik basieren (siehe z.B. Kapitel 9.7).
Im Zuge der Präzisierung der Dynamik wurden auch erstmals Fairness-Anforderungen mit in das formale Modell aufgenommen (Definition 3.10.2). Des Weiteren wurden mit Cases und globalen Aktivitätszuständen zwei unterschiedlich ausdrucksstarke Typen von Systemzuständen auf der formalen Ebene zur Verfügung gestellt (Kapitel 3.1).
Über die Modellierungsebenen lassen sich Systemeigenschaften klassifizieren. So werden elementare Struktureigenschaften (Definition 4.7) auf der formalen Ebene formuliert und auf der algorithmischen Ebene bewiesen, wohingegen abgeleitete Struktureigenschaften (Definition 10.2) ausschließlich auf der formalen Ebene formuliert und bewiesen werden. In Beispielen wurden die unterschiedlichen Beweisstrategien vorgeführt.

Zum Schwerpunkt „Lösung des Semantik-Problems“:

Es erfolgte die Entwicklung von neuartigen Trace-Semantiken und Zustandsgraphen für I–Systeme (Kapitel 4, 5, 6). Diese erlauben in unterschiedlicher Ausdrucksstärke und Darstellungskomplexität die Dokumentation von Schlüsselphänomenen, wie sie in Kapitel 1.1 beschrieben sind, besonders das Wirken von Einflüssen. Da sowohl die Trace-Semantiken als auch die Zustandsgraphen auf der formalen Ebene anzuordnen sind, bestand die Aufgabe, eine saubere Schnittstelle zur Dynamik auf der algorithmischen Ebene zu schaffen. Mit Hilfe der in Kapitel 3 definierten z-Belegung (Definition 3.11) ist dieses gelungen.
Abbildung 7.2 zeigt alle in dieser Arbeit definierten Semantiken und Zustandsgraphen für I–Systeme und die Beziehungen untereinander in Bezug auf deren Ausdrucksstärke. Aus der Abbildung lassen sich zwei wichtige Teilergebnisse dieser Arbeit ablesen: Zum einen wurde

in Kapitel 5 für alle drei Trace-Semantiken (Verhalten, Casetrace-Semantik und Erweiterte Casetrace-Semantik) bewiesen, dass deren Interleaving-Anteil, sofern er vollständig bekannt ist, ausreicht, um sie im Ganzen rekonstruieren zu können. Zum anderen wurde in Kapitel 6 gezeigt, dass sich die Interleaving Trace-Semantiken (und damit auch die normalen Trace-Semantiken) im Allgemeinen *nicht* aus den gleich lautenden Zustandsgraphen zurückgewinnen lassen, d.h. die Informationspotentiale der Darstellungsformen Trace-Semantik und gerichteter Graph sind bei I–Systemen unterschiedlich. Aus dieser neuen Erkenntnis ergeben sich neue Fragestellungen, die in weiterführenden Arbeiten behandelt werden sollen (siehe Abschnitt 12.2.2).
Dass es mindestens eine Klasse von I–Systemen gibt, bei denen Trace-Semantik (speziell Casetrace-Semantik) und gleichnamiger Zustandsgraph (Casegraph) gleichwertig in ihrem Informationsgehalt sind, wurde in Kapitel 8 mit der Menge $ISystem_{LCS}$ (Definition 8.9) gezeigt. In dem Kapitel wurde gleichzeitig bewiesen, dass die beabsichtigte Einbettung der Lose Gekoppelten Systeme (siehe Kapitel 8.1) in die Menge der I–Systeme, die syntaktisch und graphisch offensichtlich ist, auch semantisch vollzogen worden ist. Bisher gab es diesbezüglich keinen formalen Nachweis.
Für alle eingeführten Trace-Semantiken wurde gezeigt, dass sie nicht präfix-abgeschlossen sind. Diese Eigenschaft ist eine Grundlage zur Analyse von Systemverhalten auf Fortschritt (vgl. [75]). Ergänzend lassen sich je nach Semantik auftretende Systemereignisse begrifflich unterscheiden, z.B. bei der Erweiterten Casetrace-Semantik freie von erzwungenen Phasentransitionen (siehe Definition 4.18). Es wurde darauf geachtet, dass die anschauliche Interpretation der Dynamik bei der Begriffsbildung auf der formalen Ebene berücksichtigt werden.
Über die Struktur der Semantiken, in Abhängigkeit von der (syntaktischen) Konstruktion der I–Systeme, wurden in Kapitel 4 elementare Kern-Eigenschaften von I–Systemen in Form von Charakterisierungssätzen ausgedrückt und bewiesen. Die Interpretation der Sätze zeigte, dass das vorgestellte formale Modell I–System mit der Motivation verträglich ist.
Die Eignung der eingeführten Semantiken zur Systemanalyse und -verifikation hat sich bei der Bearbeitung des nächsten Schwerpunktes erwiesen.

Zum Schwerpunkt „Durchführung von modularem Entwurf/Analyse und I–System-Transformationen“:

Als Entwurfs- bzw. Analysemodule wurden in Kapitel 10 so genannte Strukturbausteine präsentiert. Bei ihnen handelt es sich um parametrisierte Teilstrukturen von I–Systemen, deren Interaktionsbeziehungen zur Systemumgebung und damit Einflüsse und Nicht-Einflüsse auf das Gesamtsystemverhalten bekannt sind. Die (Nicht-)Einflüsse lassen sich formal als Aussagen über die Struktur der eingeführten Semantiken formulieren und mit Hilfe elementarer Struktureigenschaften beweisen. Eine modulare Systemanalyse/-verifikation eines I–Systems nur unter Verwendung von Strukturbausteinen findet ausschließlich auf der formalen Ebene statt und erspart eine (umfangreiche) Analyse der Ausführungen des zugeordneten $\mathrm{V_I}$Systems. Deshalb muss es ein Ziel sein, einen umfassenden Satz an Strukturbausteinen zu entwerfen, mit dem sich alle praxisrelevanten Interaktionsbeziehungen überdecken lassen. In dieser Arbeit wurde neben den notwendigen Begriffsbildungen mit ersten konkreten Beispielen eine Ausgangsbasis geschaffen.
Am Beispiel eines synchronen Kommunikationsmechanismus (Kapitel 10.3) wurde der Einsatz von Strukturbausteinen zur modularen Verifikation von Synchronisationsanforderungen vorgeführt. Die Korrektheit des Modells ergab sich einfach durch die Vereinigung der Struktureigenschaften der verwendeten Bausteine. In einem weiteren Beispiel (Kapitel 10.4.1) wurde die besondere Eignung von I–Systemen zum inkrementellen Entwurf komplexer verteilter Systeme demonstriert. Ein nicht-triviales Synchronisationsproblem zwischen (Betriebssystem-)Prozessen wurde durch das schrittweise Einbringen von restringierenden Interaktionsbeziehungen gelöst.
Es wurden wichtige Grundlagen für die Arbeit mit I–System-Transformationen geschaffen. Hierzu gehört eine Zusammenstellung und Begründung von Transformationszielen, wie z.B. die Minimierung der Anzahl der Kontrollbereiche eines I–Systems bei gleichem Verhalten in den Relevanzbereichen zur Vereinfachung der Systemanalyse. Die Unterscheidung von Relevanz- und Kontrollbereichen wurde in Kapitel 9.1 diskutiert.
Im Rahmen der Korrektheit von I–System-Transformationsregeln stellt sich die Frage nach einer Gleichwertigkeit von unterschiedlichen I–Systemen. Kapitel 11.3 liefert hierzu Definitionen für Äquivalenzen auf I–Systemen. Diese beruhen auf Projektionen (genannt Sichten, eingeführt in Kapitel 9) der Trace-Semantiken und Zustandsgraphen auf eine ausgezeichnete Teilmenge der

Bereiche. Anhand eines längeren Beispiels wurde die Anwendung von Transformationsregeln exemplarisch vorgeführt und deren Zweckmäßigkeit verdeutlicht. Im Gegensatz zu früheren Ansätzen ist keine Erweiterung der grundlegenden formalen Strukturen notwendig.
Ein nennenswertes Nebenergebnis von Kapitel 9 ist eine Verallgemeinerung des in früheren Arbeiten verwendeten Modellierungsansatzes zur Modellierung so genannter organisatorischer Ablaufpläne in Form von lokalen Ereignisstrukturen in einem Bereich eines I–Systems, inklusive der Modellierung interner Zwänge (Kapitel 9.7). Die Einschränkung, dass jeder Knoten der zugrunde liegenden Graphenstruktur (die den organisatorischen Ablaufplan repräsentiert) mindestens einen Vorgängerknoten besitzen muss, entfällt. Weiterhin wurde die Korrektheit der Modellierung für die Erweiterte Casetrace-Semantik gezeigt. Bisher bezog sich die Korrektheit auf den (ausdrucksschwächeren) Casegraphen. Bei der Bearbeitung des Beispiels hat sich die neue Art der Spezifikation der Dynamik als Vorteil herausgestellt, indem der Spezialfallcharakter des bisherigen Ansatzes deutlich wurde.

Als Fazit ist somit festzuhalten, dass die erfolgreiche Bearbeitung der Schwerpunkte wichtige Beiträge zur Theorie der I–Systeme erbracht hat, d.h. das gesetzte Ziel dieser Arbeit wurde erreicht. Des Weiteren haben sich neue Fragestellungen und Aufgaben ergeben, die Ansatzpunkte und Motivation für weiterführende Arbeiten sind.

12.2 Weiterführende Arbeiten

Diese Arbeit dient als Grundlage für eine Vielzahl aufbauender Forschungsarbeiten, die das Ziel verfolgen, die Theorie der I–Systeme weiter zu entwickeln, sowie den praktischen Einsatz der I–Systeme als Modellierungs-, Analyse-, und Verifikationswerkzeug im Anwendungsgebiet der verteilten Systeme voranzutreiben. Es lassen sich grob die folgenden Aufgabenfelder abgrenzen:

12.2.1 Alternativen bei Dynamik und Semantik

Bedingt durch die spezifischen Modellierungsziele ist die Spezifizierung der Dynamik von I–Systemen komplexer als die von anderen formalen Modellen (z.B. Petri-Netzen). In Kapitel 3 wurde diese Thematik behandelt, und Lösungsansätze zur Formalisierung wurden präsentiert. Letztendlich beruht die Dynamik auf 3 Axiomen und 13 algorithmisch notierten Aktionen, die die Ausführungen der V_ISysteme bestimmen. Jede Aktion ist für sich betrachtet kompakt und durch anschauliche Interpretationen begreifbar gehalten. Trotzdem ist die Handhabung aller 13 Aktionen gewöhnungsbedürftig und bedarf einer gewissen Einarbeitungszeit. Hier besteht nun die Aufgabe, wenn möglich Vereinfachungen oder Hilfestellungen zu erarbeiten, ohne die Modellierungsmöglichkeiten einzuschränken.
Als semantische Beschreibungsformen wurden in dieser Arbeit bestimmte Trace-Semantiken und Zustandsgraphen betrachtet. Es ist zu untersuchen, welche Vorteile die Einführung weiterer, auch anders gearteter Semantiken (z.B. partielle Ordnungen) mit sich bringen kann. Im folgenden Abschnitt 12.2.2 wird ein in Frage kommender Punkt angesprochen.
Im Zuge einer syntaktischen Vereinfachung von zu formulierenden Struktureigenschaften oder Systemanforderungen, die sich auf die Traces der Trace-Semantiken oder die Pfade in den Zustandsgraphen beziehen, sollte der Einsatz geeigneter temporaler Logiken diskutiert werden.

12.2.2 Gleichwertigkeit von Trace-Semantiken und Zustandsgraphen

Ein wichtiges Resultat in Kapitel 6 ist die Erkenntnis, dass sich Trace-Semantiken und Zustandsgraphen von I–Systemen im Allgemeinen in ihrem Informationspotential bzgl. der Erfassung eines Systemverhaltens unterscheiden. Die Trace-Semantiken geben exakter die Aktivität der zugeordneten V_ISysteme wieder, wohingegen die weniger ausdrucksstarken Zustandsgraphen eine endliche und als Graphen anschaulichere Darstellungsform besitzen. Für eine umfassende Systemanalyse ist es wünschenswert, diesen Unterschied nicht berücksichtigen zu müssen, um dann zwischen Trace-Semantiken und deren Darstellung als Graphen je nach Bedarf wechseln zu können. Dies gilt insbesondere für den Einsatz von (Visualisierungs-)Tools. Es ergeben sich die folgenden Fragestellungen:

a) Lassen sich für das vorliegende formale Modell Kriterien formulieren, aus denen sich die Gleichheit in der Ausdruckskraft von Trace-Semantiken und Zustandsgraphen für bestimmte I–Systeme ableiten lässt?

b) Lässt sich das bestehende formale Modell so anpassen, dass in dem angepassten Modell die Gleichheit direkt für alle I–Systeme gilt?

Bezüglich Fragestellung a) ist mit der Menge $ISystem_{LCS}$ in Kapitel 8.3 bereits eine erste Teilmenge von $ISysten$ gefunden worden, für deren Elemente eine Äquivalenz gezeigt wurde.

Bezüglich Fragestellung b) gibt es folgende zwei erste Ansätze, die vermutlich eine Äquivalenz mit sich bringen. Eine formale Umsetzung und die Überprüfung der Vermutung sind die nächsten anstehenden Aufgaben.

Ansatz 1 verfolgt eine Lösungsstrategie auf der 2. Modellierungsebene (gemäß Abbildung 7.1). Durch die Einführung von neuen Semantiken soll eine feinere Granularität bei der Dokumentierung von Systemabläufen erreicht werden. Eine Ursache für die Diskrepanz zwischen Trace-Semantiken und Zustandsgraphen liegt in der (bewussten) Abstraktion von allen laufenden Nachrichten zwischen Komponenten/Bereichen auf der formalen Ebene. Es ist denkbar, diese vollständige Abstraktion zu lockern, um speziell das Wirken von Einflüssen eingehender zu erfassen. Durch die Auswertung der lokalen $\_s(\cdot)$-Variablen kann das Schicken und Empfangen von *solicit*- und *cancel*-Nachrichten (interpretiert als Aufbau und Abbau von propagierten Zwängen) in den Komponenten eines V_ISystems beobachtet werden. Mittels der s-Belegung (Definition 3.11) ist es möglich, diese Information auf die formale Ebene zu übertragen und für die Konstruktion neuer Semantiken, ggf. über neue globale Systemzustände oder Zusatzinformationen in den Traces, zu verwenden. Daraus ergeben sich dann neue Knoten und/oder Kantenmengen für die entsprechenden Zustandsgraphen.

Ansatz 2 verfolgt eine Lösungsstrategie auf der 3. Modellierungsebene. Das Ziel ist es, die Spezifizierung der Dynamik der I–Systeme so abzuändern, dass, trotz Abstraktion von allen laufenden Nachrichten, auf der formalen Ebene die Nennung eines relevanten globalen Systemzustandes (relevanter globaler Aktivitätszustand oder Case) ausreicht, um dessen Folgezustände eindeutig (graphisch) angeben zu können, und ohne eine Abhängigkeit von Vorgängerzuständen berücksichtigen zu müssen. Die grundlegende Idee ist, zur Festlegung des Verhaltens der V_ISysteme einen speziellen „Referenzalgorithmus“ für verteiltes Ressourcen-Management, z.B. [92], einzusetzen und anzupassen. Als exklusive Ressourcen werden dabei die Kopplungsrelationen betrachtet. Der Algorithmus garantiert die Einhaltung des geforderten wechselseitigen Ausschlusses von in der Kopplungsrelation stehenden Nachbarphasen. Des Weiteren garantiert er Deadlockfreiheit im V_ISystem, und durch eine Erweiterung des Algorithmus kann außerdem Verhungern ausgeschlossen werden. Aufgrund dieser Eigenschaften werden explizite Fairness-Anforderungen, wie sie in Definition 3.10 einfließen, unnötig. Erarbeitet werden muss noch eine geeignete Integration der Erregungsrelation und das Setzen der lokalen $\_z(\cdot)$-Variablen mit Phasenqualitäten. Die Definition der Trace-Semantiken und Zustandsgraphen kann dann wie bisher erfolgen.
Die Vermutung der Gleichheit im Informationsgehalt von Trace-Semantiken und Graphen bei diesem Ansatz begründet sich in der (voraussichtlichen) Umgehung von Kommunikationssituationen, in denen zwischen zwei Komponenten zwei Nachrichten aufeinander folgend unterwegs sind, und bei denen die erste eine Änderung der lokalen $\_z(\cdot)$-Variablen in der Zielkomponente bewirkt und die zweite diese Änderung wieder rückgängig macht. Das Beispiel im Beweis von Satz 6.5 basiert auf solch einer Situation mit *solicit*- und *cancel*-Nachrichten, eine Ursache für die Nicht-Gleichheit beim I–System IS_2. Zu untersuchen bleibt, welchen Einfluss die verschiedenen Synchronisationsnachrichten im Referenzalgorithmus auf die Ausführungen der V_ISysteme und damit auf die Semantiken der I–Systeme haben.
Bei einer Formalisierung des Ansatzes sind die vorhandenen Modellierungsebenen zu beachten. So werden z.B. im Algorithmus aus [92] die Ressourcen von eigenständigen Prozessen, den Resource-Managern, verwaltet, und es ergibt sich die Aufgabe der Realisierung dieser Resource-Manager im Kontext der V_ISysteme auf der algorithmischen Ebene. Hinzu kommt die Schaffung geeigneter

Schnittstellen zur Anschauungsebene, um das Verständnis der Systemabläufe für den Anwender zu erleichtern.

12.2.3 Eigenschaften von I–Systemen

Das Ziel ist zum einen die Übertragung von bekannten, immer wieder verwendeten Begriffen für Eigenschaften formaler Modelle (z.B. aus dem Gebiet der Petri-Netze) auf das Modell der I–Systeme. In Frage kommende Begriffe sind z.B. Fairness, Lebendigkeit, Erreichbarkeit. Zum anderen sollen neue relevante Eigenschaften, die spezifisch im Modell der I–Systeme auftreten, definiert werden. Hier sind solche Begriffe denkbar wie z.B. Stabilität, Entscheidungsfreiheit, Zwangsinvarianz, die die Besonderheit der I–Systeme, Einflüsse explizit zu modellieren, ausnutzen. Für alle Eigenschaften sollen Analyseverfahren entwickelt werden. Komplexitätsbetrachtungen sind durchzuführen. In diesem Zusammenhang stellt sich die Frage nach geeigneten Datenstrukturen zur Repräsentation von I–Systemen.

12.2.4 Strukturbausteine

In Kapitel 10 wurden spezielle Strukturbausteine präsentiert, durch die eine modulare Analyse bzw. ein inkrementelles modulares Design von I–Systemen unterstützt wird. Die Beispiele wurden dabei so gewählt, dass sie die in dieser Arbeit im Blickpunkt stehenden Interaktionsbeziehungen zwischen Systemkomponenten erfassen. Die Beispiele reichen allerdings nicht aus, um alle denkbaren Interaktionsbeziehungen abzudecken. Je umfangreicher ein I–System über Strukturbausteine abgedeckt wird, desto weniger müssen ergänzende elementare Strukturaussagen formuliert und bewiesen werden. Das Ziel muss es deshalb sein, einen umfassenden Satz an Strukturbausteinen zu entwickeln, der als „Baukasten“ dient, mit dem sich alle praxisrelevanten Interaktionsbeziehungen modellieren lassen.

12.2.5 Transformationsregeln

Der Einsatz von I–System-Transformationen wurde ausführlich in Kapitel 11 motiviert. Dort wurden auch die bezüglich dieser Thematik anstehenden Arbeiten benannt. Im Mittelpunkt steht die Entwicklung von allgemeinen Transformations-Regelsystemen, einschließlich derer Korrektheitsbeweise. Dabei ist zu klären, ob eine Modellanpassung/-erweiterung notwendig und sinnvoll ist (vgl. Strukturierte Interaktionssysteme).
Als eines der Ziele von I–System-Transformationen wurde in Kapitel 11.1 die Reduzierung der Anzahl der Kontrollbereiche zur Vereinfachung der Systemanalyse herausgestellt. Während für Lose Gekoppelte Systeme das Reduktionsproblem mit Reduktion auf einen einzigen Bereich gelöst ist (siehe [65]), ist es für I–Systeme wegen der komplexeren Dynamik und ausdrucksstärkeren Semantiken noch ein offenes Problem.

12.2.6 Erweiterungen

Für jedes formale Modell stellt sich die Frage nach sinnvollen Erweiterungen, um die Modellierungsmöglichkeiten auszubauen. In diesem Sinne sollen für I–Systeme bewährte Modellierungsaspekte anderer formaler Modelle (z.B. bei Hierarchischen Petri Netzen [29] oder Modecharts [41]) diskutiert und je nach Eignung auf I–Systeme übertragen werden. Im Blickpunkt stehen insbesondere Erweiterungen zur hierarchischen Modellierung und zur Erfassung von Realzeit-Randbedingungen.

12.2.7 Tools

Die Anwendbarkeit von formalen Modellen bei praktischen komplexen Anwendungen ist nicht zuletzt abhängig von der Verfügbarkeit von Tools, die den Modellierungs- und Analysevorgang durchgehend unterstützen. Für I–Systeme sind ebenfalls solche Tools zu entwickeln. Diese beinhalten z.B. graphische Editoren, Visualisierungsmodule, Simulatoren, Analyse- und Verifikationskomponenten, Datenbanken (z.B. für Strukturbausteine), Transformationstools, Hilfe-Funktionen, WWW-Anbindung. Im Rahmen der diversen softwaretechnischen Fragen (z.B. Programmiersprache, Programmierumgebung, Hardwareplattform, Entwurfsprozess, Testumgebung) ist zu klären, inwieweit existierende Tools integriert werden können. So ist es z.B. nahe liegend, existierende Model-Checker (siehe [9, 22]) für Verifikationszwecke einzusetzen. Die in dieser Arbeit definierten Trace-Semantiken bieten sich hierzu als potentielle Schnittstelle an.

12.2.8 Anwendungsgebiete

In den bisherigen Arbeiten (siehe Kapitel 1.3) wurde deutlich, dass sich I–Systeme als formales Modell für unterschiedliche Anwendungsszenarien einsetzen lassen. So finden sich z.B. Beispiele aus dem Software- und Hardwareentwurf als auch zur Beschreibung organisatorischer Ablaufstrukturen. Von einer konkreten Anwendung wurde bei dem Modellentwurf in dieser Arbeit bewusst abstrahiert, um die Allgemeingültigkeit der Formalismen zu verfolgen, d.h. es lag der Schwerpunkt in der Ausarbeitung der Modellierungsebenen 2, 3 und 4 aus Abbildung 7.1 und der Schaffung einer Schnittstelle zur Anwendungsebene 1. Zukünftige Arbeiten werden sich verstärkt mit der Ebene 1 beschäftigen. Dabei sollen bestehende Anwendungsszenarien (z.B. das Leitsystems für Fußgänger im Straßenverkehr aus Kapitel 1.1 ausgearbeitet, neue Anwendungsfelder gefunden und jeweils die Anbindungen an die Ebene 2 präzisiert werden.

Anhang A

Beweise auf der algorithmischen Modellierungsebene

Um die Lesbarkeit der Arbeit zu erleichtern sind lange Beweise von Sätzen in diesen Anhang verlagert worden. Den Beweisen ist gemeinsam, dass sie, oder wesentliche Teile von ihnen, auf der algorithmischen Modellierungsebene (siehe Kapitel 7.1) geführt werden. Einleitende Beweisideen und Anmerkungen zu den Beweisen finden sich weiterhin in den zugehörigen Kapiteln.

A.1 Beweise aus Kapitel 4

Beweis von Satz 4.6 (Charakterisierung des Verhaltens)

Es gelten die Bezeichnungen und Voraussetzungen aus dem Satz. Sei eine beliebige Ausführung Π von $V_I System(IS)$ gegeben, die gemäß Definition 4.1 $z_0 z_1 z_2 \ldots$ erzeugt. (Unter Bezugnahme auf Π wird im Beweis die Notation 3.12 verwendet.) Seien $t_0, t_1, t_2, \ldots$ die durch die Definition gegebenen Globalzeitpunkte und $i \in \{1, 2, \ldots\}$ beliebig aber fest, jeweils aus dem zugeordnetem Wertebereich.

Alle Möglichkeiten für das Aussehen der Ausführung Π werden soweit per Fallunterscheidungen untersucht, bis die Gültigkeit der Satzaussagen gezeigt ist. Hierzu werden die Aktionen aus Kapitel 3.2.2 systematisch abgearbeitet. Die Kriterien (1) und (2) aus Definition 3.10 (Ausführung) kommen im Beweis zum Tragen.

Zu **a)** und **b)**. Sei $(p, v) \in E$. Zu zeigen: Aus $z^{t_{i-1}}\langle V_{b(p)}\rangle(p) \neq \mathsf{0}$ und $z^{t_{i-1}}\langle V_{b(v)}\rangle(v) \neq \mathsf{0}$ folgt:

i) $z^{t_i}\langle V_{b(p)}\rangle(p) \neq \mathsf{0}$

ii) $\exists t', t' \geq t_{i-1} : z^{t'}\langle V_{b(p)}\rangle(p) = \mathsf{1}$

iii) $\exists t'', t'' \geq t_{i-1} : (z^{t''}\langle V_{b(v)}\rangle(v) \in \{\mathsf{0}, \mathsf{F}\}) \vee (z^{t''}\langle V_{b(v)}\rangle(v) = \mathsf{1} \wedge \exists x \in E(v) : z^{t''}\langle V_{b(x)}\rangle(x) \neq \mathsf{0})$

Teil a) gilt dann mit $k := max\{j \in I\!N_0 \mid t_j \leq t'\}$.
Teil b) gilt dann mit $k := max\{j \in I\!N_0 \mid t_j \leq t''\}$.

Zu i). Annahme: $z^{t_i}\langle V_{b(p)}\rangle(p) = \mathsf{0}$.
$\Rightarrow$
Bei $V_{b(p)}$ hat zum Globalzeitpunkt t_i eine Phasentransition $p \to q$ mit $q \in b(p)$ stattgefunden. Phasentransitionen erfolgen gemäß Aktionsbeschreibung A4 oder A5. Entsprechend den Vorbedingungen gilt somit $z^{t_{i-1}}\langle V_{b(p)}\rangle(p) \in \{q, \mathsf{F}\}$. Die Initialisierung ist bei $V_{b(p)}$ schon abgeschlossen.
Wegen $z^{t_{i-1}}\langle V_{b(v)}\rangle(v) \neq \mathsf{0}$ und den Beobachtungen 1 und 2 aus dem Beweis zu Satz 3.14 gilt: $in^{t_i}\langle V_{b(p)}\rangle(v) = true$ oder $mark^{t_i}\langle V_{b(p)}\rangle(v) = true$.

Fall i.1). Es gilt $in^{t_i}\langle V_{b(p)}\rangle(v) = true$ und $mark^{t_i}\langle V_{b(p)}\rangle(v) = false$.
$\_in(\cdot)$ wird gesetzt in A1 und A10. In beiden Aktionen erfolgt am Ende ein Update-Aufruf (A13).

Wegen $v \in E(p)$ gilt $\_e_{out}(p)$, und A13.i, A13.ii oder A13.v findet Anwendung, $\_z(p)$ wird auf 1 gesetzt. Danach kann, solange $\_in(v)$ bei $V_{b(p)}$ wahr ist, keine Aktion stattfinden, die zu $\_z(p) \neq 1$ führt. Wegen $\_e_{out}(p)$ führt jeder Update-Aufruf zur Ausführung von A13.v, $\_z(p) = 1$ bleibt erhalten. Der einzige Fall einer Aktion, $\_z(p) = 1$ ohne Update Aufruf zu verändern, besteht bei A3. Doch auch hier muss zur Ausführung von $\_z(p) := q$ die Bedingung *not* $\_e_{out}(p)$ erfüllt sein.
$\Rightarrow$
$z^{t_i}\langle V_{b(p)}\rangle(p) = 1 \quad \Rightarrow \quad$ Widerspruch!

Fall i.2). Es gilt $mark^{t_i}\langle V_{b(p)}\rangle(v) = true$.

Zum Zeitpunkt t_i gilt bei $V_{b(p)}$, bezogen auf die Aktionsbeschreibungen von A4 und A5, der Fall *not* b), da v nach Voraussetzung eine $\mathrm{E_{out}}$-Nachbarphase von p ist.
$\Rightarrow$
Es erfolgt *keine* Phasentransition $p \to q \quad \Rightarrow \quad$ Widerspruch!

Beide Fälle führen zu einem Widerspruch. Damit ist die Annahme falsch und es gilt i).

Zu ii). Fallunterscheidung:

Fall ii.1). $\forall t \in [t_0, t_{i-1}] : z^t\langle V_{b(v)}\rangle(v) \neq 0$.

Es gilt $z^{t_0}\langle V_{b(v)}\rangle(v) = 1$. Als erstes wird bei allen Komponenten die Initialisierung (A1) ausgeführt (VA1). $V_{b(v)}$ erhält von $V_{b(p)}$ die Nachricht *reqinit* und antwortet gemäß A2 mit *ackinit*(v). Nach Erhalt dieser Antwort wird bei $V_{b(p)}$ zu einem Zeitpunkt t^1 $\_in(v) := true$ ausgeführt, danach ein Update-Aufruf (A13), dem Fall ii.1.1, ii.1.2 oder ii.1.3 folgt.

Fall ii.1.1). $z^{t^1}\langle V_{b(p)}\rangle(p) \neq 0$.

Beim Update erfolgt zu einem Zeitpunkt t^2 $\_z(p) := 1$ wegen $v \in E(p)$. Danach kann, analog zu Fall i.1), solange $\_in(v) = true$ bei $V_{b(p)}$ gilt, keine Aktion stattfinden, die zu $\_z(p) \neq 1$ führt. $\_in(v) = true$ bleibt solange gültig, bis $V_{b(p)}$ eine Nachricht *done*$(v \to w)$ mit $w \in b(v)$ erhält (A10), als Information, dass bei $V_{b(v)}$ eine Phasentransition stattgefunden hat. Eine Phasentransition kann für diesen Fall bei $V_{b(v)}$ nur nach t_{i-1} erfolgen.
$\Rightarrow$
Mit $t' := max\{t_{i-1}, t^2\}$ gilt Teil ii).

Fall ii.1.2). $z^{t^1}\langle V_{b(p)}\rangle(p) = 0$ und $t^1 \leq t_{i-1}$.

Wegen $z^{t_{i-1}}\langle V_{b(p)}\rangle(p) \neq 0$ muss bei $V_{b(p)}$ im Zeitraum $[t^1, t_{i-1}]$ eine Phasentransition $q \to p$ mit $q \in b(p)$ erfolgen (A4 oder A5). Nach der Transition wird innerhalb der gleichen Aktion ein Update-Aufruf durchgeführt (A13). Es wird zwischenzeitlich nicht auf andere eingehende Nachrichten reagiert (VA2). Weiterhin gilt $\_in(v) = true$ mindestens bis t_{i-1} (analog zu Fall ii.1.1), das Update wird deshalb $\_z(p) := 1$ durchführen zu einem Zeitpunkt t^2.
$\Rightarrow$
Mit $t' := max\{t_{i-1}, t^2\}$ gilt Teil ii).

Fall ii.1.3). $z^{t^1}\langle V_{b(p)}\rangle(p) = 0$ und $t^1 > t_{i-1}$.

Wegen $z^{t_{i-1}}\langle V_{b(p)}\rangle(p) \neq 0$ muss bei $V_{b(p)}$ im Zeitraum $[t_{i-1}, t^1]$ eine Phasentransition $p \to q$ mit $q \in b(p)$ erfolgen (A4 oder A5). Das ist aber nicht möglich, da die Initialisierung bei $V_{b(p)}$ noch nicht abgeschlossen ist. A4/A5 kann nur nach A1 stattfinden (VA1).
$\Rightarrow$
Widerspruch, dieser Fall kann somit nicht eintreten.

Fall ii.2). $\exists t \in [t_0, t_{i-1}] : z^t\langle V_{b(v)}\rangle(v) = 0$.

Wegen $z^{t_{i-1}}\langle V_{b(v)}\rangle(v) \neq 0$ existiert ein Zeitpunkt $t_x \in [t_0, t_{i-1}]$, zu dem eine für den Zeitraum $[t_0, t_{i-1}]$ zeitlich letzte Phasentransition $w \to v$ mit $w \in b(v)$ bei $V_{b(v)}$ stattfindet. Nach Ausführung der Phasentransition schickt $V_{b(v)}$ die Nachricht *done*$(w \to v)$ unter anderem an $V_{b(p)}$ (A4 oder A5). $V_{b(p)}$ reagiert zu einem Zeitpunkt $t^1, t^1 > t_x$ mit $\_in(v) := true$ (A10) und nachfolgendem

Update-Aufruf (A13), gefolgt von Fall ii.2.1, ii.2.2 oder ii.2.3.

Fall ii.2.1). $z^{t^1}\langle V_{b(p)}\rangle(p) \neq 0$.
Analog ii.1.1).

Fall ii.2.2). $z^{t^1}\langle V_{b(p)}\rangle(p) = 0$ und $t^1 \leq t_{i-1}$.
Analog ii.1.2).

Fall ii.2.3). $z^{t^1}\langle V_{b(p)}\rangle(p) = 0$ und $t^1 > t_{i-1}$.
Wegen $z^{t_{i-1}}\langle V_{b(p)}\rangle(p) \neq 0$ muss bei $V_{b(p)}$ im Zeitraum $[t_{i-1}, t^1]$ eine Phasentransition $p \to q$ mit $q \in b(p)$ erfolgen (A4 oder A5). Die Durchführung der Phasentransition setzt bei $V_{b(p)}$ das Senden einer Nachricht $reqmark(M)$ nach $V_{b(v)}$ sowie den Empfang der Nachricht $ackmark$ von $V_{b(v)}$ voraus (A4/A5). Zum Zeitpunkt t_{i-1} ist bei $V_{b(v)}$ die Phasentransition $w \to v$ ausgeführt. Vor der Reaktion auf $reqmark(M)$, und damit vor Senden von $ackmark$, ist demnach die Nachricht $done(w \to v)$ von $V_{b(v)}$ nach $V_{b(p)}$ geschickt worden. Wegen Axiom VA2, die Nachrichten werden in der Reihenfolge empfangen, in der sie gesendet werden, erfolgt bei $V_{b(p)}$ die Registrierung von $ackmark$ nach der Reaktion auf $done(w \to v)$ (A10). Somit findet die Phasentransition $p \to q$ nach dem Zeitpunkt t^1 statt.
$\Rightarrow$
Widerspruch, dieser Fall kann somit nicht eintreten.

Damit ist diese Fallunterscheidung abgeschlossen. Für jeden Fall wurde gezeigt, dass, sofern er überhaupt eintreten kann, die Aussage ii) erfüllt ist.

Zu iii). Fallunterscheidung:

Fall iii.1). $\forall t \in [t_0, t_{i-1}] : z^t\langle V_{b(p)}\rangle(p) \neq 0$.
Es gilt $z^{t_0}\langle V_{b(p)}\rangle(p) = 1$. Als erstes wird bei allen Komponenten die Initialisierung (A1) ausgeführt (VA1). $V_{b(p)}$ erhält von $V_{b(v)}$ die Nachricht $reqinit$ und antwortet gemäß A2 mit $ackinit(p)$. Nach Erhalt dieser Antwort wird bei $V_{b(v)}$ $\_in(p) := true$ ausgeführt, danach ein Update-Aufruf (A13) mit Update zu einem Zeitpunkt t^1 (A10, A13).

Fall iii.1.1). $t^1 \leq t_{i-1}$ und $\forall t \in [t^1, t_{i-1}] : z^t\langle V_{b(v)}\rangle(v) \neq 0$.
Sei t^2 der Zeitpunkt des letzten Updates A13, mit aktuell eingenommener Phase v, im Zeitraum $[t^1, t_{i-1}]$. Dieser Zeitpunkt existiert, da mit t^1 mindestens ein Kandidat vorliegt.

Fall iii.1.1.1). $\forall x \in E(v) : in^{t^2}\langle V_{b(v)}\rangle(x) = false$.
Das Update bei t^2 liefert $\_z(v) := \mathsf{F}$ bei $V_{b(v)}$ (A13.iii/iv/v), denn es gilt $not\ \_e_{out}(v)$ und $\_e_{in}(v)$. Da es sich um das letzte Update vor/bei t_{i-1} handelt, bleiben die Belegungen mindestens bis t_{i-1} erhalten.
$\Rightarrow$
Mit $t'' := t_{i-1}$ gilt Teil iii).

Fall iii.1.1.2). $\exists x \in E(v) : in^{t^2}\langle V_{b(v)}\rangle(x) = true$.
Das Update bei t^2 liefert $\_z(v) := 1$ bei $V_{b(v)}$ (A13.i/ii/v), denn es gilt $\_e_{out}(v)$. Da es sich um das letzte Update vor/bei t_{i-1} handelt, bleiben die Belegungen mindestens bis t_{i-1} erhalten.

Fall iii.1.1.2.1). $\exists y \in E(v) : z^{t_{i-1}}\langle V_{b(y)}\rangle(y) \neq 0$.
$\Rightarrow$
Mit $t'' := t_{i-1}$ gilt Teil iii).

Fall iii.1.1.2.2). $\forall y \in E(v) : z^{t_{i-1}}\langle V_{b(y)}\rangle(y) = 0$.
Bei mindestens einer der $\mathrm{E_{out}}$-Nachbarkomponenten von $V_{b(v)}$ hat eine Phasentransition (A4/A5)

stattgefunden. Folglich erhält $V_{b(v)}$ eine $done(x' \to z')$-Nachricht, mit $z' \in b(x')$, von jeder Komponente $V_{b(x')}$ mit $x' \in E(v)$ und $in^{t^2}\langle V_{b(v)}\rangle(x') = true$. Die Reaktionen bei $V_{b(v)}$ gemäß A10 führen jeweils zu einem Update-Aufruf (A13), bei aktueller Phase v, nach t_{i-1}. Wegen $\_e_{out}(v)$ bleibt v mindestens bis zum letzten dieser Updates aktuell und damit auch p (wegen i)). Der Zeitpunkt des letzten dieser Updates sei t^3.

Fall iii.1.1.2.2.1). $\exists x'' \in E(v), \exists t^4 \in [t^2, t^3] : z^{t^4}\langle V_{b(x'')}\rangle(x'') \neq \mathsf{0}$.

$\Rightarrow$
Mit $t'' := t^3$ gilt Teil iii).

Fall iii.1.1.2.2.2). $\forall x'' \in E(v), \forall t^4 \in [t^2, t^3] : z^{t^4}\langle V_{b(x'')}\rangle(x'') = \mathsf{0}$.

Das Update zum Zeitpunkt t^3 führt bei $V_{b(v)}$ zu $\_z(v) := \mathsf{F}$ (A13.iii/iv/v), denn es gilt nun *not* $\_e_{out}(v)$ und $\_e_{in}(v)$.
$\Rightarrow$
Mit $t'' := t^3$ gilt Teil iii).

Fall iii.1.2). $t^1 \leq t_{i-1}$ und $\exists t \in [t^1, t_{i-1}] : z^t\langle V_{b(v)}\rangle(v) = \mathsf{0}$.

Wegen $z^{t_{i-1}}\langle V_{b(v)}\rangle(v) \neq \mathsf{0}$ existiert ein Zeitpunkt $t_x \in [t^1, t_{i-1}]$, zu dem eine für den Zeitraum $[t^1, t_{i-1}]$ zeitlich letzte Phasentransition $w \to v$ mit $w \in b(v)$ bei $V_{b(v)}$ stattfindet (A4/A5). Nach der Transition wird innerhalb der gleichen Aktion ein Update-Aufruf (A13) mit v als aktuell eingenommener Phase durchgeführt. Das Update erfolgt zu einem Zeitpunkt t^2.

Fall iii.1.2.1). $t^2 \leq t_{i-1}$.

Analog Fall iii.1.1) mit Unterfällen, mit t^2 statt t^1, t^3 statt t^2, usw..

Fall iii.1.2.2). $t^2 > t_{i-1}$.

Das Update ist bei $V_{b(v)}$ das erste Update nach t_{i-1}. Folglich gilt für alle $t \in [t_{i-1}, t^2]$: $z^t\langle V_{b(v)}\rangle(v) \neq \mathsf{0}$ und $z^t\langle V_{b(p)}\rangle(p) \neq \mathsf{0}$ wegen i).

Fall iii.1.2.2.1). $\forall x \in E(v) : in^{t^2}\langle V_{b(v)}\rangle(x) = false$.

Das Update liefert $\_z(v) := \mathsf{F}$ bei $V_{b(v)}$ (A13.iii/iv), denn es gilt *not* $\_e_{out}(v)$ und $\_e_{in}(v)$.
$\Rightarrow$
Mit $t'' := t^2$ gilt Teil iii).

Fall iii.1.2.2.2). $\exists x \in E(v) : in^{t^2}\langle V_{b(v)}\rangle(x) = true$.

Das Update liefert $\_z(v) := 1$ bei $V_{b(v)}$ (A13.i/ii/v), denn es gilt $\_e_{out}(v)$.

Fall iii.1.2.2.2.1). $\exists y \in E(v) : z^{t^2}\langle V_{b(y)}\rangle(y) \neq \mathsf{0}$.

Analog Fall iii.1.1.2.1) mit t^2 statt t_{i-1}.

Fall iii.1.2.2.2.2). $\forall y \in E(v) : z^{t^2}\langle V_{b(y)}\rangle(y) = \mathsf{0}$.

Analog Fall iii.1.1.2.2) mit Unterfällen, mit t^2 statt t_{i-1}.

Fall iii.1.3). $t^1 > t_{i-1}$.

Erst ab t^1 kann bei $V_{b(v)}$ eine Phasentransition auftreten, frühstens dann ist dort die Initialisierung abgeschlossen (A1). Folglich gilt für alle $t \in [t_{i-1}, t^1]$: $z^t\langle V_{b(v)}\rangle(v) \neq \mathsf{0}$ und $z^t\langle V_{b(p)}\rangle(p) \neq \mathsf{0}$ wegen i).

Fall iii.1.3.1). $\forall x \in E(v) : in^{t^1}\langle V_{b(v)}\rangle(x) = false$.

Analog Fall iii.1.2.2.1) mit t^1 statt t^2.

Fall iii.1.3.2). $\exists x \in E(v) : in^{t^1}\langle V_{b(v)}\rangle(x) = true$.

Analog Fall iii.1.2.2.2) mit Unterfällen, mit t^1 statt t^2, t^2 statt t^3, usw..

Fall iii.2). $\exists t' \in [t_0, t_{i-1}] : z^{t'}\langle V_{b(p)}\rangle(p) = 0$.

Wegen $z^{t_{i-1}}\langle V_{b(p)}\rangle(p) \neq 0$ existiert ein Zeitpunkt $t_x \in [t_0, t_{i-1}]$, zu dem eine für den Zeitraum $[t_0, t_{i-1}]$ zeitlich letzte Phasentransition $q \rightarrow p$ mit $q \in b(p)$ bei $V_{b(p)}$ stattfindet. Nach Ausführung der Phasentransition schickt $V_{b(p)}$ die Nachricht $done(q \rightarrow p)$ unter anderem an $V_{b(v)}$ (A4/A5). $V_{b(v)}$ reagiert mit $\_in(p) := true$ (A10) und nachfolgendem Update-Aufruf (A13). Das Update erfolgt zu einem Zeitpunkt $t^1, t^1 > t_x$.

Fall iii.2.1). $t^1 \leq t_{i-1}$ und $\forall t \in [t^1, t_{i-1}] : z^t\langle V_{b(v)}\rangle(v) \neq 0$.

Analog Fall iii.1.1).

Fall iii.2.2). $t^1 \leq t_{i-1}$ und $\exists t \in [t^1, t_{i-1}] : z^t\langle V_{b(v)}\rangle(v) = 0$.

Analog Fall iii.1.2).

Fall iii.2.3). $t^1 > t_{i-1}$ und $\forall t \in [t_{i-1}, t^1] : z^t\langle V_{b(v)}\rangle(v) \neq 0$.

Für alle t gilt auch $z^t\langle V_{b(p)}\rangle(p) \neq 0$ wegen i).

Fall iii.2.3.1). $\forall x \in E(v) : in^{t^1}\langle V_{b(v)}\rangle(x) = false$.

Analog Fall iii.1.2.2.1) mit t^1 statt t^2.

Fall iii.2.3.2). $\exists x \in E(v) : in^{t^1}\langle V_{b(v)}\rangle(x) = true$.

Analog Fall iii.1.2.2.2) mit Unterfällen, mit t^1 statt t^2, t^2 statt t^3, usw..

Fall iii.2.4). $t^1 > t_{i-1}$ und $\exists t \in [t_{i-1}, t^1] : z^t\langle V_{b(v)}\rangle(v) = 0$.

Wegen Teil i) existiert ein Zeitpunkt $t^2 \in [t_{i-1}, t^1]$, für den gilt: $z^{t^2}\langle V_{b(v)}\rangle(v) = 0$ und $z^{t^2}\langle V_{b(p)}\rangle(p) \neq 0$
$\Rightarrow$
Mit $t'' := t^2$ gilt Teil iii).

Damit ist die Fallunterscheidung abgeschlossen. Für jeden Fall wurde gezeigt, dass die Aussage iii) erfüllt ist.

Zu **c)**. Sei $(p, w) \in K$ und $(q, v) \in K$. Zu zeigen:

c') Aus $z^{t_{i-1}}\langle V_{b(p)}\rangle(p) \neq 0$ und $z^{t_{i-1}}\langle V_{b(v)}\rangle(v) \neq 0$ folgt: $z^{t_i}\langle V_{b(p)}\rangle(q) = 0$ oder $z^{t_i}\langle V_{b(v)}\rangle(w) = 0$.

Da während der Ausführung von $V_I System(IS)$ als z-Globalbelegungen nur globale Aktivitätszustände auftreten (Satz 3.14), muss nach Definition 3.3.(2) $p \neq q$ und $v \neq w$ gelten.

Annahme: $z^{t_i}\langle V_{b(q)}\rangle(p) \neq 0$ und $z^{t_i}\langle V_{b(w)}\rangle(w) \neq 0$.

Bei $V_{b(p)}$ findet zum Zeitpunkt t_i eine Phasentransition $p \rightarrow q$ statt. Phasentransitionen erfolgen gemäß Aktionsbeschreibung A4 oder A5. Entsprechend den Aktionsbeschreibungen (Fall c) bei A4/A5) muss während der Phasentransition *not* $\_mark(v)$ bei $V_{b(p)}$ gelten, denn v ist nach Voraussetzung eine K-Nachbarphase von q. Gleichzeitig muss, entsprechend Beobachtung 1 aus dem Beweis zu Satz 3.14, $\_mark(p)$ und $\_mark(q)$ bei $V_{b(v)}$ gelten.

Für $V_{b(v)}$ liegt die symmetrische Situation vor. Zum Zeitpunkt t_i der Phasentransition $v \rightarrow w$ gilt *not* $\_mark(p)$ bei $V_{b(v)}$, sowie $\_mark(v)$ und $\_mark(w)$ bei $V_{b(p)}$.

Mit *not* $\_mark(v)$ und $\_mark(v)$ bei $V_{b(p)}$, sowie mit *not* $\_mark(p)$ und $\_mark(p)$ bei $V_{b(v)}$, jeweils zum Zeitpunkt t_i, liegt ein Widerspruch vor. Die Annahme ist demnach falsch, und es gilt Teil c') und damit c).

Zu **d)**. Sei $(q, v) \in K$. Zu zeigen:

d') Aus $z^{t_{i-1}}\langle V_{b(p)}\rangle(p) \in \{q, \mathsf{F}\}$ und $z^{t_{i-1}}\langle V_{b(v)}\rangle(v) \neq 0$ folgt: $\exists t', t' \geq t_{i-1} : (z^{t'}\langle V_{b(p)}\rangle(p) \in \{0, 1\}) \vee (z^{t'}\langle V_{b(v)}\rangle(v) \in \{0, \mathsf{F}\}) \vee (z^{t'}\langle V_{b(v)}\rangle(v) = 1 \wedge \exists x \in E(v) : z^{t'}\langle V_{b(x)}\rangle(x) \neq 0)$.

Teil d) gilt dann mit $k := max\{j \in I\!N_0 \mid t_j \in \{t_0, t_1, t_2, \dots\}, t_j \leq t'\}$.

Fall 1). $z^{t_{i-1}}\langle V_{b(p)}\rangle(p) = \mathsf{F}$.

Die Initialisierung (A1) bei $V_{b(p)}$ ist abgeschlossen, und es gibt dort einen für den Zeitraum $[t_0, t_{i-1}]$ letzten Ausführungsbeginn einer Aktion A, bei der die Zuweisung $\_z(p) := \mathsf{F}$ zu einem Zeitpunkt $t^1, t^1 \leq t_{i-1}$ erfolgt. Der Update-Aufruf A13 wird dabei der aufrufenden Aktion zugeordnet. Nach Beendigung von A wird als nächstes auf eingegangene Nachrichten reagiert (VA2, A2, A8-A12). Die Reaktionen sind abgeschlossen zu einem Zeitpunkt t^2.

Fall 1.1). $z^{t^2}\langle V_{b(p)}\rangle(p) \in \{0, q'\}$ mit $q' \in b(p)$.

$z^{t^2}\langle V_{b(p)}\rangle(p) = 0$ kann nur durch eine Phasentransition innerhalb von A4 oder A5 erreicht werden, aber nicht durch die Reaktion auf eine eingetroffene Nachricht. $z^{t^2}\langle V_{b(p)}\rangle(p) = q'$ setzt die Ausführung von A3 voraus und kann ebenfalls nicht das Ergebnis einer Reaktion auf eine eingetroffene Nachricht nach A sein.

$\Rightarrow$

Fall 1.1) kann nicht auftreten.

Fall 1.2). $z^{t^2}\langle V_{b(p)}\rangle(p) = 1$.

Da A die letzte Aktion vor/bei t_{i-1} ist, die zu $\_z(p) = \mathsf{F}$ führt, gilt $t^2 > t_{i-1}$.

$\Rightarrow$

Die Aussage d') gilt mit $t' := t^2$.

Fall 1.3). $z^{t^2}\langle V_{b(p)}\rangle(p) = \mathsf{F}$.

Bei $V_{b(p)}$ wird als nächste Aktion A5 durchgeführt. Entsprechend der Aktionsbeschreibung können unterschiedliche Ereignisse eintreten.

Fall 1.3.1). Bezogen auf die Aktionsbeschreibung von A5 gilt a).

Während des Empfangs der *ackmark* Nachrichten wird zwischenzeitlich auf eine eingetroffene Nachricht reagiert, was zu $\_z(p) := 1$ zu einem Zeitpunkt $t^3, t^3 > t^2$ führt.

$\Rightarrow$

Die Aussage d') gilt mit $t' := t^3$.

Fall 1.3.2). Bezogen auf die Aktionsbeschreibung von A5 gilt b).

Die Aktion A5 wird beendet und als nächstes wieder auf eingetroffene Nachrichten reagiert (VA2, A2, A8-A12). Die Reaktionen sind abgeschlossen zu einem Zeitpunkt t^3.

Fall 1.3.2.1). $z^{t^3}\langle V_{b(p)}\rangle(p) \in \{0, q'\}$ mit $q' \in b(p)$.

Analog 1.1) mit A5 statt A und t^3 statt t^2.

Fall 1.3.2.2). $z^{t^3}\langle V_{b(p)}\rangle(p) = 1$.

$\Rightarrow$

Die Aussage d') gilt mit $t' := t^3$.

Fall 1.3.2.3). $z^{t^3}\langle V_{b(p)}\rangle(p) = \mathsf{F}$.

Analog 1.3) mit Unterfällen, mit t^3 statt t^2, t^4 statt t^3, usw.. Die erneute Ausführung von A5 ist dabei garantiert durch die Reaktion auf noch ausstehende Nachrichten. Wegen $\_mark(v')$ für eine Nachbarphase v' von p wird $V_{b(p)}$ schließlich eine $done(\cdot)$- oder *break*-Nachricht von $V_{b(v')}$ erhalten und gemäß A9 oder A10 reagieren. Führt dieser Fall erneut zu Fall 1.3.2.3, werden die Zeiten entsprechend angepasst. Eine unendliche Iteration tritt wegen des Kriteriums (2) aus

Definition 3.10 bei der zugrunde liegenden Ausführung Π nicht auf.

Fall 1.3.3). Bezogen auf die Aktionsbeschreibung von A5 gilt *not* a) und *not* b) und c).

Eine Phasentransition $p \to q'$, mit $q' \in b(p)$, findet bei $V_{b(p)}$ statt zu einem Zeitpunkt t^3. Es wird $\_z(p) := 0$ und $\_z(q') := 1$ gleichzeitig ausgeführt.
$\Rightarrow$
Die Aussage d') gilt mit $t' := t^3$.

Fall 1.3.4). Bezogen auf die Aktionsbeschreibung von A5 gilt *not* a) und *not* b) und *not* c).

Nach dem Senden von *break*-Nachrichten an alle Nachbarkomponenten erfolgt bei $V_{b(p)}$ ein Solicitation-Aufruf bzgl. $b(p)\backslash\{p\}$ (A6). Als Folge davon erhält $V_{b(v)}$ die Nachricht $solicit(b(p)\backslash\{p\})$ und reagiert gemäß A11. Wegen $q \in b(p) \setminus \{p\}$ erfolgt $\_s(q) := true$ bei $V_{b(v)}$ und innerhalb der gleichen Aktion auch ein Update-Aufruf (A13). Das Update findet statt zu einem Zeitpunkt t^3.

Fall 1.3.4.1). $t^3 \leq t_{i-1}$ und $\forall t \in [t^3, t_{i-1}] : z^t\langle V_{b(v)}\rangle(v) \neq 0$ und $s^t\langle V_{b(v)}\rangle(q) = true$.

Sei t^4 der Zeitpunkt des letzten Updates eines Update-Aufrufes (A13) mit aktueller Phase v im Zeitraum $[t^3, t_{i-1}]$. Dieser Zeitpunkt existiert, da mit t^3 mindestens ein Kandidat vorliegt.

Fall 1.3.4.1.1). $\forall x \in E(v) : in^{t^4}\langle V_{b(v)}\rangle(x) = false$.

Das letzte Update liefert $\_z(v) := \mathsf{F}$ bei $V_{b(v)}$ (A13.iii/iv/v), denn es gilt *not* $\_e_{out}(v)$ und $\_s(q)$. q ist K-Nachbarphase von v nach Voraussetzung. Da es sich um das letzte Update vor/bei t_{i-1} handelt, bleiben die Belegungen mindestens bis t_{i-1} erhalten.
$\Rightarrow$
Mit $t' := t_{i-1}$ gilt Aussage d').

Fall 1.3.4.1.2). $\exists x \in E(v) : in^{t^4}\langle V_{b(v)}\rangle(x) = true$.

Das letzte Update liefert $\_z(v) := 1$ bei $V_{b(v)}$ (A13.i/ii/v), denn es gilt $\_e_{out}(v)$. Da es sich um das letzte Update vor/bei t_{i-1} handelt, bleiben die Belegungen mindestens bis t_{i-1} erhalten.

Fall 1.3.4.1.2.1). $\exists y \in E(v) : z^{t_{i-1}}\langle V_{b(y)}\rangle(y) \neq 0$.

$\Rightarrow$
Mit $t' := t_{i-1}$ gilt Aussage d').

Fall 1.3.4.1.2.2). $\forall y \in E(v) : z^{t_{i-1}}\langle V_{b(y)}\rangle(y) = 0$.

Bei mindestens einer der $\mathrm{E_{out}}$-Nachbarkomponenten von $V_{b(v)}$ hat eine Phasentransition (A4/A5) stattgefunden. Folglich erhält $V_{b(v)}$ eine $done(x' \to z')$-Nachricht, mit $z' \in b(x')$, von jeder Komponente $V_{b(x')}$ mit $x' \in E(v)$ und $in^{t^4}\langle V_{b(v)}\rangle(x') = true$. Die Reaktionen bei $V_{b(v)}$ gemäß A10 führen jeweils zu einem Update-Aufruf (A13) mit aktueller Phase v, nach t_{i-1}. Wegen $\_e_{out}(v)$ bleibt v mindestens bis zum letzten dieser Updates aktuell. Der Zeitpunkt des letzten dieser Updates sei t^5.

Fall 1.3.4.1.2.2.1). $\exists x'' \in E(v), \exists t^6 \in [t^4, t^5] : z^{t^6}\langle V_{b(x'')}\rangle(x'') \neq 0$.

$\Rightarrow$
Mit $t' := t^6$ gilt Aussage d').

Fall 1.3.4.1.2.2.2). $\forall x'' \in E(v), \forall t^6 \in [t^4, t^5] : z^{t^6}\langle V_{b(x'')}\rangle(x'') = 0 \wedge s^{t^5}\langle V_{b(v)}\rangle(q) = true$.

Das Update zum Zeitpunkt t^5 führt bei $V_{b(v)}$ zu $\_z(v) := \mathsf{F}$ (A13.iii/iv/v), denn es gilt nun *not* $\_e_{out}(v)$ und $\_s(q)$. Dabei ist q K-Nachbarphase von v nach Voraussetzung.
$\Rightarrow$
Mit $t' := t^5$ gilt Aussage d').

Fall 1.3.4.1.2.2.3). $\forall x'' \in E(v), \forall t^6 \in [t^4, t^5] : z^{t^6}\langle V_{b(x'')}\rangle(x'') = 0 \wedge s^{t^5}\langle V_{b(v)}\rangle(q) = false$.

Im Zeitraum $[t_{i-1}, t^5]$ findet bei $V_{b(v)}$ eine Aktion statt, mit $\_s(q) := false$ als eine der Anweisungen. In Frage kommen hierzu nur die Aktionen A10 und A12. Im Fall von A10 wird auf eine

$done(p \rightarrow \cdot)$-Nachricht von $V_{b(p)}$ reagiert, dort hat eine Phasentransition zu einem Zeitpunkt t^7, $t^7 \in [t_{i-1}, t^5]$ stattgefunden, und es gilt dann $z^{t^7}\langle V_{b(p)}\rangle(p) = 0$. Im Fall von A12 wird auf eine $cancel(M)$-Nachricht, mit $q \in M$, von $V_{b(p)}$ reagiert. Das Senden dieser Nachricht erfolgt bei $V_{b(p)}$ durch einen Cancellation-Aufruf (A7), als Aufruf ausschließlich während Update-Aktion A13.i/ii. Nach dem Aufruf wird, gemäß A13.i/ii, noch $\_z(p) := 1$ ausgeführt, sofern p zu diesem Zeitpunkt die aktuelle Phase ist. Folglich gibt es einen Zeitpunkt t^7, $t^7 \in [t_{i-1}, t^5]$ mit $z^{t^7}\langle V_{b(p)}\rangle(p) \in \{0, 1\}$.
$\Rightarrow$
Mit $t' := t^7$ gilt Aussage d').

Fall 1.3.4.2). $t^3 \leq t_{i-1}$ und $\exists t \in [t^3, t_{i-1}] : z^t\langle V_{b(v)}\rangle(v) = 0$ und $\forall \bar{t} \in [t^3, t_{i-1}] : s^{\bar{t}}\langle V_{b(v)}\rangle(q) = true$.

Wegen $z^{t_{i-1}}\langle V_{b(v)}\rangle(v) \neq 0$ existiert ein Zeitpunkt $t_x \in [t^3, t_{i-1}]$, zu dem eine für den Zeitraum $[t^3, t_{i-1}]$ zeitlich letzte Phasentransition $w \rightarrow v$ mit $w \in b(v)$ bei $V_{b(v)}$ stattfindet (A4,A5). Nach Ausführung der Phasentransition wird innerhalb der gleichen Aktion ein Update-Aufruf (A13) mit aktueller Phase v durchgeführt. Das Update erfolgt zu einem Zeitpunkt t^4.

Fall 1.3.4.2.1). $t^4 \leq t_{i-1}$

Analog Fall 1.3.4.1) mit Unterfällen, mit t^4 statt t^3, t^5 statt t^4, usw..

Fall 1.3.4.2.2). $t^4 > t_{i-1}$

Das Update ist bei $V_{b(v)}$ das erste Update nach t_{i-1}. Folglich gilt für alle $t \in [t_{i-1}, t^4]$: $z^t\langle V_{b(v)}\rangle(v) \neq 0$ und $s^t\langle V_{b(v)}\rangle(q) = true$.

Fall 1.3.4.2.2.1). $\forall x \in E(v) : in^{t^4}\langle V_{b(v)}\rangle(x) = false$.

Das Update liefert $\_z(v) := \mathsf{F}$ bei $V_{b(v)}$ (A13.iii/iv/v), denn es gilt $not\ \_e_{out}(v)$ und $\_e_{in}(v)$.
$\Rightarrow$
Mit $t' := t^4$ gilt Teil d').

Fall 1.3.4.2.2.2). $\exists x \in E(v) : in^{t^4}\langle V_{b(v)}\rangle(x) = true$.

Das Update liefert $\_z(v) := 1$ bei $V_{b(v)}$ (A13.i/A13.ii), denn es gilt $\_e_{out}(v)$.

Fall 1.3.4.2.2.2.1). $\exists y \in E(v) : z^{t^4}\langle V_{b(y)}\rangle(y) \neq 0$.

Analog Fall 1.3.4.1.2.1) mit t^4 statt t_{i-1}.

Fall 1.3.4.2.2.2.2). $\forall y \in E(v) : z^{t^4}\langle V_{b(y)}\rangle(y) = 0$.

Analog Fall 1.3.4.1.2.2) mit Unterfällen, mit t^4 statt t_{i-1}.

Fall 1.3.4.3). $t^3 \leq t_{i-1}$ und $\exists t \in [t^3, t_{i-1}] : s^t\langle V_{b(v)}\rangle(q) = false$.

Im Zeitraum $[t^3, t_{i-1}]$ findet bei $V_{b(v)}$ eine Aktion statt, mit $\_s(q) := false$ als eine der Anweisungen. In Frage kommen hierzu nur die Aktionen A10 und A12.

Fall 1.3.4.3.1). Aktion A10 findet statt.

Es wird auf eine $done(\cdot)$-Nachricht von $V_{b(p)}$ reagiert, dort hat eine Phasentransition im Zeitraum $[t^2, t_{i-1}]$ stattgefunden. Direkt nach der Phasentransition gilt $\_z(p) \in \{0, 1\}$ bei $V_{b(p)}$. Wegen $z^{t_{i-1}}\langle V_{b(p)}\rangle(p) = \mathsf{F}$ muss vor t_{i-1} noch eine Aktion bei $V_{b(p)}$ erfolgen, bei der die Zuweisung $\_z(p) := \mathsf{F}$ erfolgt. Das ist ein Widerspruch zu der Festlegung von t^1.
$\Rightarrow$
Dieser Fall kann nicht eintreten.

Fall 1.3.4.3.2). Aktion A12 findet statt.

$V_{b(v)}$ reagiert auf eine $cancel(M)$-Nachricht, mit $q \in M$, von $V_{b(p)}$. Das Senden dieser Nachricht erfolgt bei $V_{b(p)}$ durch einen Cancellation-Aufruf (A7), als Aufruf ausschließlich während Update-Aktion A13.i/ii. Nach dem Aufruf wird, gemäß A13.i/ii, noch $\_z(x) := 1$ ausgeführt, wobei x die aktuell eingenommene Phase bei $V_{b(p)}$ ist. Der Zeitpunkt der Ausführung sei t^4.

Fall 1.3.4.3.2.1). $z^{t^4}\langle V_{b(p)}\rangle(p) = 0$.

Wegen $z^{t_{i-1}}\langle V_{b(p)}\rangle(p) \neq 0$ muss bei $V_{b(p)}$ im Zeitraum $[t^2, t_{i-1}]$ eine Phasentransition stattfinden. Analog Fall 1.3.4.3.1 führt das zu einem Widerspruch zu der Festlegung von t^1.
$\Rightarrow$
Dieser Fall kann nicht eintreten.

Fall 1.3.4.3.2.2). $t^4 \leq t_{i-1}$ und $z^{t^4}\langle V_{b(p)}\rangle(p) \neq 0$.
Wegen $z^{t_{i-1}}\langle V_{b(p)}\rangle(p) = \mathsf{F}$ muss vor t_{i-1} noch eine Aktion bei $V_{b(p)}$ erfolgen, bei der die Zuweisung $\_z(p) := \mathsf{F}$ erfolgt. Das ist ein Widerspruch zu der Festlegung von t^1.
$\Rightarrow$
Dieser Fall kann nicht eintreten.

Fall 1.3.4.3.2.3). $t^4 > t_{i-1}$ und $z^{t^4}\langle V_{b(p)}\rangle(p) \neq 0$.
$\Rightarrow$
Mit $t' := t^4$ gilt Aussage d').

Fall 1.3.4.4). $t^3 > t_{i-1}$ und $z^{t^3}\langle V_{b(v)}\rangle(v) = 0$.
$\Rightarrow$
Mit $t' := t^4$ gilt Teil d').

Fall 1.3.4.5). $t^3 > t_{i-1}$ und $z^{t^3}\langle V_{b(v)}\rangle(v) \neq 0$.

Da ein Unterfall von 1.3.4) behandelt wird und sich $V_{b(v)}$ demnach noch innerhalb Aktion A11 befindet, gilt $s^{t^3}\langle V_{b(v)}\rangle(q) = true$.

Fall 1.3.4.5.1). $\forall x \in E(v) : in^{t^3}\langle V_{b(v)}\rangle(x) = false$.

Analog Fall 1.3.4.2.2.1) mit t^3 statt t^4.

Fall 1.3.4.5.2). $\exists x \in E(v) : in^{t^4}\langle V_{b(v)}\rangle(x) = true$.

Analog Fall 1.3.4.2.2.2) mit Unterfällen, mit t^3 statt t^4, t^4 statt t^5, usw..

Fall 2). $z^{t_{i-1}}\langle V_{b(p)}\rangle(p) = q$.

Die Initialisierung (A1) bei $V_{b(p)}$ ist abgeschlossen, und es gibt dort einen für den Zeitraum $[t_0, t_{i-1}]$ letzten Ausführungsbeginn einer Aktion A, bei der die Zuweisung $\_z(p) := q$ zu einem Zeitpunkt $t^1, t^1 \leq t_{i-1}$ erfolgt. Bei A kann es sich nur um eine A3-Ausführung handeln. Nach Beendigung von A wird als nächstes auf eingegangene Nachrichten reagiert (VA2, A2, A8-A12). Die Reaktionen sind abgeschlossen zu einem Zeitpunkt t^2.

Fall 2.1). $z^{t^2}\langle V_{b(p)}\rangle(p) \in \{\mathsf{0}, q'\}$ mit $q' \in b(p) \setminus \{q\}$.

Analog Fall 1.1) mit $q' \in b(p) \setminus \{q\}$ statt $q' \in b(p)$.

Fall 2.2). $z^{t^2}\langle V_{b(p)}\rangle(p) = 1$.

Analog Fall 1.2) mit $\_z(p) = q$ statt $\_z(p) = \mathsf{F}$.

Fall 2.3). $z^{t^2}\langle V_{b(p)}\rangle(p) = \mathsf{F}$.

Analog Fall 1.3). Da A die letzte Aktion vor/bei t_{i-1} ist, die zu $\_z(p) = q$ führt, gilt $t^2 > t_{i-1}$ und damit auch $t^3, t^4, \dots > \mathsf{0}$. Folglich reduziert sich die Anzahl der in Frage kommenden Unterfälle.

Fall 2.4). $z^{t^2}\langle V_{b(p)}\rangle(p) = q$.

Bei $V_{b(p)}$ wird als nächste Aktion A4 durchgeführt. Entsprechend der Aktionsbeschreibung können unterschiedliche Ereignisse eintreten.

Fall 2.4.1). Bezogen auf die Aktionsbeschreibung von A4 gilt a).

Während des Empfangs der *ackmark* Nachrichten wurde zwischenzeitlich auf eine eingetroffene Nachricht reagiert, was zu $\_z(p) := 1$ zu einem Zeitpunkt $t^3, t^3 > t^2$ führt.
$\Rightarrow$
Die Aussage d') gilt mit $t' := t^3$.

Fall 2.4.2). Bezogen auf die Aktionsbeschreibung von A4 gilt b).

Die Aktion wird beendet und als nächstes wieder auf eingetroffene Nachrichten reagiert (VA2, A2, A8-A12). Die Reaktionen sind abgeschlossen zu einem Zeitpunkt t^3.

Fall 2.4.2.1). $z^{t^3}\langle V_{b(p)}\rangle(p) \in \{0, q'\}$ mit $q' \in b(p) \setminus \{q\}$.

Analog 2.1) mit A4 statt A und t^3 statt t^2.

Fall 2.4.2.2). $z^{t^3}\langle V_{b(p)}\rangle(p) = 1$.

$\Rightarrow$
Die Aussage d') gilt mit $t' := t^3$.

Fall 2.4.2.3). $z^{t^3}\langle V_{b(p)}\rangle(p) = \mathsf{F}$.

Analog 2.3) mit Unterfällen und t^3 statt t^2, t^4 statt t^3, usw.

Fall 2.4.2.4). $z^{t^3}\langle V_{b(p)}\rangle(p) = q$.

Analog 2.4) mit Unterfällen, mit t^3 statt t^2, t^4 statt t^3, usw.. Die erneute Ausführung von A4 ist dabei garantiert durch die Reaktion auf noch ausstehende Nachrichten. Wegen $\_mark(v')$ für eine Nachbarphase v' von p wird $V_{b(p)}$ schließlich eine $done(\cdot)$- oder *break*-Nachricht von $V_{b(v')}$ erhalten und gemäß A9 oder A10 reagieren. Führt dieser Fall erneut zu Fall 2.4.2.3, werden die Zeiten entsprechend angepasst. Eine unendliche Iteration tritt wegen des Kriteriums (2) aus Definition 3.10 bei der zugrunde liegenden Ausführung Π nicht auf.

Fall 2.4.3). Bezogen auf die Aktionsbeschreibung von A4 gilt *not* a) und *not* b) und c).

Die Phasentransition $p \to q$ findet bei $V_{b(p)}$ statt zu einem Zeitpunkt t^3. Es wird $\_z(p) := 0$ und $\_z(q) := 1$ gleichzeitig ausgeführt.
$\Rightarrow$
Die Aussage d') gilt mit $t' := t^3$.

Fall 2.4.4). Bezogen auf die Aktionsbeschreibung von A5 gilt *not* a) und *not* b) und *not* c).

Nach dem Senden von *break*-Nachrichten an alle Nachbarkomponenten, erfolgt bei $V_{b(p)}$ ein Solicitation-Aufruf bzgl. $\{q\}$ (A6). Als Folge davon erhält $V_{b(v)}$ die Nachricht $solicit(\{q\})$ und reagiert gemäß A11. Es erfolgt $\_s(q) := true$ bei $V_{b(v)}$ und innerhalb der gleichen Aktion auch ein Update-Aufruf (A13) mit Update zu einem Zeitpunkt t^3. Die Unterfälle treten dann analog zu 1.3.4.1) bis 1.3.4.5) auf.

Damit ist die Fallunterscheidung zu d) abgeschlossen. Für die auftretenden Fälle wurde gezeigt, dass die Aussage d') gilt, und folglich gilt auch d).

Es sind nun alle Teilaussagen a)-d) von Satz 4.6 bewiesen worden. Der Beweis des Satzes ist abgeschlossen. $\square$

A.2 Beweise aus Kapitel 8

Beweis von Satz 8.6 (Gleichheit von $CG(LCS)$ und $CG(IS)$)

Es gelten die Voraussetzungen aus dem Satz. $\underline{B}$ bezeichne die Menge der trägen Bereiche und E die Erregungsrelation von IS ($\underline{B} = E = \emptyset$).

Sei $CG(\overbrace{LCS}^{(P,B,C,K)}) = (C, \rightarrow)$

$\Rightarrow$ {Definition 8.1.3}
$C = \{c \subseteq P \mid (\forall b \in B : |c \cap b| = 1) \wedge (\forall p_1, p_2 \in c : (p_1, p_2) \notin K)\}$ $(*_1)$

$\Rightarrow$ {Definitionen 8.4 und 8.3.a}
$\rightarrow = \{(c_1, c_2) \in C \times C \mid |c_1 \setminus c_2| = |c_2 \setminus c_1| = 1\}.$ $(*_2)$

Sei $CG(\overbrace{IS}^{(P,B,\emptyset,K,\emptyset)}) = (C', \rightarrow')$

$\Rightarrow$ {Definitionen 6.6 und 3.1}
$C' = Case(IS) = \{c \subseteq P \mid (\forall b \in B : |c \cap b| = 1) \wedge (\forall p_1, p_2 \in c : (p_1, p_2) \notin K)\}$ $(*_3)$

$\Rightarrow$ {Definition 6.6}
$\rightarrow' = \{(c_1, c_2) \in C' \times C' \mid \exists ctr_1, ctr_2 \in Case(IS)^* : ctr_1.c_1.c_2.ctr_2 \in \mathcal{CT}^{\mathrm{i}}[\![IS]\!]\}$

$\Rightarrow$ {Definition 5.13}
$\rightarrow' = \{(c_1, c_2) \in C' \times C' \mid \exists ctr_1, ctr_2 \in Case(IS)^*, \exists c'_0, c'_1, c'_2, \ldots \in Case(IS) : ctr_1.c_1.c_2.ctr_2 = c'_0, c'_1, c'_2, \ldots$ und $c'_0 c'_1 c'_2 \ldots \in \mathcal{CT}[\![IS]\!]$ mit $\forall i = 1, 2, \ldots : |c'_i \setminus c'_{i-1}| = 1\}$ $(*_4)$

$\Rightarrow$ {Definition 4.8}
$\rightarrow' = \{(c_1, c_2) \in C' \times C' \mid \exists ctr_1, ctr_2 \in Case(IS)^*, \exists c'_0 \in Case(IS), \exists$ Ausführung Π von $V_I System(IS)$ und eine max. Folge $t_0, t_1, t_2, \ldots$ von Globalzeitpunkten der Ausführung:
$ctr_1.c_1.c_2.ctr_2 = zc(z^{\Pi,t_0}\langle V_I System(IS)\rangle).zc(z^{\Pi,t_1}\langle V_I System(IS)\rangle).zc(z^{\Pi,t_2}\langle V_I System(IS)\rangle). \ldots,$
t_0 ist Startzeitpunkt, $z^{\Pi,t_0}\langle V_I System(IS)\rangle = cz(c'_0)$, $\forall i = 1, 2, \ldots :$
$(t_{i-1} < t_i,\ zc(z^{\Pi,t_{i-1}}\langle V_I System(IS)\rangle) \neq zc(z^{\Pi,t_i}\langle V_I System(IS)\rangle),\ (\forall t, t_{i-1} \leq t < t_i : zc(z^{\Pi,t}\langle V_I System(IS)\rangle) = zc(z^{\Pi,t_{i-1}}\langle V_I System(IS)\rangle)),\ |zc(z^{\Pi,t_i}\langle V_I System(IS)\rangle) \setminus zc(z^{\Pi,t_{i-1}}\langle V_I System(IS)\rangle)| = 1)\ \}$

$\Rightarrow$ {Zusammenfassen}
$\rightarrow' = \{(c_1, c_2) \in C' \times C' \mid \exists$ Ausführung Π von $V_I System(IS)$ und eine max. Folge $t_0, t_1, t_2, \ldots$ von Globalzeitpunkten der Ausführung, $\exists k \in \mathbb{N}$, $\exists c'_0 \in Case(IS)$:
t_0 ist Startzeitpunkt, $z^{\Pi,t_0}\langle V_I System(IS)\rangle = cz(c'_0)$, $\forall i = 1, 2, \ldots :$
$(t_{i-1} < t_i,\ |zc(z^{\Pi,t_i}\langle V_I System(IS)\rangle) \setminus zc(z^{\Pi,t_{i-1}}\langle V_I System(IS)\rangle)| = 1,\ (\forall t, t_{i-1} \leq t < t_i : zc(z^{\Pi,t}\langle V_I System(IS)\rangle) = zc(z^{\Pi,t_{i-1}}\langle V_I System(IS)\rangle))),\ c_1 = zc(z^{\Pi,t_{k-1}}\langle V_I System(IS)\rangle),$
$c_2 = zc(z^{\Pi,t_k}\langle V_I System(IS)\rangle)\ \}$ $(*_5)$

Zu zeigen 1: $C = C'$

C
$= \{$wegen $(*_1)\}$
$\{c \subseteq P \mid (\forall b \in B : |c \cap b| = 1) \wedge (\forall p_1, p_2 \in c : (p_1, p_2) \notin K)\}$
$= \{$wegen $(*_3)\}$
C'.

Zu zeigen 2: $\rightarrow \subseteq \rightarrow'$

Sei $(c_1, c_2) \in \rightarrow$.

$\Rightarrow$ {wegen $(*_2)$}
$((c_1, c_2) \in C \times C) \wedge (|c_1 \setminus c_2| = |c_2 \setminus c_1| = 1)$
$\Rightarrow$ {direkte Folgerung}
$\exists p, q \in P : \{p\} = c_1 \setminus c_2 \wedge \{q\} = c_2 \setminus c_1$

Setze $\{p\} := c_1 \setminus c_2$ und $\{q\} := c_2 \setminus c_1$.

$\Rightarrow$ {Definition 8.1.3.b}
$b(p) = b(q)$ $(*_6)$
$\Rightarrow$ {Definition 8.1.3.b/c}
$(v \in c_1 \setminus \{p\}) \Rightarrow ((v \notin b(p) \wedge v \in c_2 \setminus \{q\}) \Rightarrow ((q,v) \notin K))$ $(*_7)$

Gemäß $(*_5)$ reicht es zu zeigen: $\exists$ Ausführung Π von $V_I System(IS)$ und eine max. Folge $t_0, t_1, t_2, \ldots$ von Globalzeitpunkten der Ausführung, $\exists k \in \mathbb{N}$, $\exists c_0' \in Case(IS)$: t_0 ist Startzeitpunkt, $z^{\Pi,t_0}\langle V_I System(IS)\rangle = cz(c_0')$, $\forall i = 1,2,\ldots : (t_{i-1} < t_i$, $|zc(z^{\Pi,t_i}\langle V_I System(IS)\rangle) \setminus zc(z^{\Pi,t_{i-1}}\langle V_I System(IS)\rangle)| = 1$, $(\forall t, t_{i-1} \leq t < t_i : zc(z^{\Pi,t}\langle V_I System(IS)\rangle) = zc(z^{\Pi,t_{i-1}}\langle V_I System(IS)\rangle)))$, $c_1 = zc(z^{\Pi,t_{k-1}}\langle V_I System(IS)\rangle)$, $c_2 = zc(z^{\Pi,t_k}\langle V_I System(IS)\rangle)$ $(*_8)$

Konstruktion der Ausführung Π:

Startzeitpunkt t^0:
Als Startbelegung setze $\_z(x) := 1$ bei $V_{b(x)}$ für alle $x \in c_1$. Alle booleschen Variablen aller Komponenten von $V_I System(IS)$ seien $false$. Damit sind die Vorbedingungen erfüllt, um bei allen Komponenten die Aktion A1 auszuführen.

$$\Rightarrow z^{\Pi,t^0}\langle V_I System(IS)\rangle(x) = \begin{cases} 1 \text{ falls } x \in c_1 \\ 0 \text{ sonst} \end{cases}$$

$\Rightarrow zc(z^{\Pi,t^0}\langle V_I System(IS)\rangle) = c_1$.

Zeitraum $[t^0, t^1]$:
Alle Komponenten führen die Initialisierungsaktionen A1 und (gegebenenfalls mehrfach) A2 aus. Der Empfang der $ackinit(\cdot)$-Nachrichten führt bei jeder Komponente zu einer Anpassung der $\_in(\cdot)$-Variablen, was Einfluss auf die booleschen $\_e_{in}(\cdot)$-, $\_e_{out}(\cdot)$- und $\_k(\cdot)$-Belegungen hat. Allerdings gilt auch weiterhin $\_e_{in}(v) = false$ für alle $v \in P$ wegen der Satzvoraussetzung $E = \emptyset$. Die $\_mark(\cdot)$ und $\_s(\cdot)$-Variablen bleiben bei ihrer Anfangsbelegung $false$. Innerhalb von A1 kommt es bei jeder Komponente am Ende zu einem Update-Aufrufe A13.v, der zu keiner Veränderung der $\_z(\cdot)$-Variablen führt. Sei t^1 der Globalzeitpunkt, zu dem die letzte Komponente A1 (und damit alle auch A2) beendet hat.

$$\Rightarrow \forall t \in [t^0, t^1] : z^{\Pi,t}\langle V_I System(IS)\rangle(x) = \begin{cases} 1 \text{ falls } x \in c_1 \\ 0 \text{ sonst} \end{cases}$$

$\Rightarrow \forall t \in [t^0, t^1] : zc(z^{\Pi,t}\langle V_I System(IS)\rangle) = c_1$.

Zeitraum $]t^1, t^2]$:
Nach t^1 führt $V_{b(p)}$ als nächste Aktion A3 aus. Dies ist möglich, da $b(p)$ nach Satzvoraussetzung autonom ist $((\underline{B} = \emptyset) \Rightarrow (b(p) \in B \setminus \underline{B}))$. Wegen $E = \emptyset$ existiert keine $\mathrm{E_{out}}$-Nachbarphase von p und $V_{b(p)}$ kann $\_z(p) := q$ (beachte $(*_6)$) ausführen. Dies erfolgt zu einem Zeitpunkt t^2. Bei den anderen Komponenten finden keine Aktionen statt.

$$\Rightarrow \forall t \in]t^1, t^2[: z^{\Pi,t}\langle V_I System(IS)\rangle(x) = \begin{cases} 1 \text{ falls } x \in c_1 \\ 0 \text{ sonst} \end{cases}$$

$$z^{\Pi,t^2}\langle V_I System(IS)\rangle(x) = \begin{cases} q \text{ falls } x = p \\ 1 \text{ falls } x \in c_1 \setminus \{p\} \\ 0 \text{ sonst} \end{cases}$$

$\Rightarrow \forall t \in]t^1, t^2] : zc(z^{\Pi,t}\langle V_I System(IS)\rangle) = c_1$.

Zeitraum $]t^2, t^3]$:
Nach t^2 führt $V_{b(p)}$ als nächste Aktion A4 aus. Die Vorbedingung ist erfüllt. $V_{b(p)}$ sendet an alle $V_{b'}$, b' ist Nachbarbereich von $\{p,q\}$, die Nachricht $reqmark(\{p,q\})$. Alle $V_{b'}$ empfangen die Nachricht und setzen jeweils $\_mark(p) := true$ und $\_mark(q) := true$. Danach senden sie $ackmark$ zurück (A8). Weitere Variablen werden bei $V_{b'}$ nicht verändert. $V_{b(p)}$ empfängt alle $ackmark$-Nachrichten (Fortsetzung A4). Direkt danach gilt bei $V_{b(p)}$ $\_z(p) = q$, da sich die Variable inzwischen nicht geändert hat. Weiterhin gilt $\_mark(v) = false$ für alle $\mathrm{E_{out}}$-Nachbarphasen v von p, da $E = \emptyset$ nach Satzvoraussetzung, und es gilt $\_k(q) = false$ wegen $(*_7)$, und es gilt $\_mark(v') = false$ für jede K-Nachbarphase v' von q, da bisher keine $reqmark(M)$-Nachricht mit $v' \in M$ bearbeitet wurde (A8), was einzig zu einer $true$-Belegung führen kann. Gemäß der Aktionsbeschreibung von A4 tritt also bei $V_{b(p)}$ der Fall „*not* a) und *not* b) und c)“ ein. Die Phasentransition $p \rightarrow q$ (d.h. $\_z(p) := 0$, $\_z(q) := 1$ gleichzeitig) wird zu einem Zeitpunkt t^3 ausgeführt. Bei allen anderen

Komponenten finden keine weiteren Aktionen statt.

$$\Rightarrow \forall t \in]t^2, t^3[: z^{\Pi,t}\langle V_I System(IS)\rangle(x) = \begin{cases} q \text{ falls } x = p \\ 1 \text{ falls } x \in c_1 \setminus \{p\} \\ 0 \text{ sonst} \end{cases}$$

$$z^{\Pi,t^3}\langle V_I System(IS)\rangle(x) = \begin{cases} 1 \text{ falls } x \in c_2 \\ 0 \text{ sonst} \end{cases}$$

$$\Rightarrow \forall t \in]t^2, t^3[: zc(z^{\Pi,t}\langle V_I System(IS)\rangle) = c_1,$$
$$zc(z^{\Pi,t^3}\langle V_I System(IS)\rangle) = c_2.$$

Zeitraum $]t^3, \infty[$:
Nach der Phasentransition befindet sich $V_{b(p)}$ noch bei der Abarbeitung von A4. Als nächstes erfolgt $\_s(q) := false$ und dann ein Update-Aufruf (A13). Die aktuellen Variablenbelegungen bei $V_{b(p)}$ ($\_z(q) = 1$ und $\_e_{in}(q) = false = \_s(v)$ für alle $v \in P$) führen zur Ausführung von A13.v, die $\_z(\cdot)$-Belegung bleibt unverändert. Nach dem Update sendet $V_{b(p)}$ an alle $V_{b'}$, b' ist Nachbarbereich von $\{p, q\}$, die Nachricht $done(\{p, q\})$. Alle $V_{b'}$ empfangen die Nachricht (A10) und setzen jeweils $\_mark(p) := false$ und $\_mark(q) := false$. Die $\_s(v')$-Belegung wird für alle $v' \in b'$ auf $false$ gesetzt, was sie allerdings auch vorher schon war. Jedes $V_{b'}$ führt als nächstes aus A10 heraus den Update-Aufruf (A13) aus. Die aktuellen Variablenbelegungen bei $V_{b'}$ ($\_z(p') = 1$, $\_e_{in}(p') = false$ (wegen $E = \emptyset$) für die momentan eingenommene Phase $p' \in b'$ und $\_s(v) = false$ für alle $v \in P$ (da eine $\_s(\cdot) := true$-Zuweisung bisher nicht stattgefunden hat)) führen zur Ausführung von A13.v, die $\_z(\cdot)$-Belegung bleibt unverändert. Die Abarbeitung von A13 und der aufrufenden Aktion A10 ist bei $V_{b'}$ dann abgeschlossen. Mit dem Senden der $done(\{p, q\})$-Nachrichten hat $V_{b(p)}$ die Ausführung von A4 abgeschlossen. Bei jeder Komponente V_b von $V_I System(IS)$ gilt jetzt $\_z(x) = 1$ für die aktuell eingenommene Phase $x \in b$, $\_e_{in}(v) = \_e_{out}(v) = false$ für jede Phase $v \in b$, $\_mark(v') = \_s(v') = false$ für jede Phase $v' \in P$ und $\_in(v'') = true$ gdw. v'' ist Nachbarphase von b und $\_z(v'') = 1$ bei $V_{b(v'')}$. Jede Komponente V_b führt von nun an nur noch wiederholt die Aktion A3 aus, ohne sich allerdings jemals für eine Zuweisung $\_z(x) := y$ mit $y \in b \setminus \{x\}$ zu entscheiden. Diese wiederholte A3-Ausführung ist möglich, da b nach Satzvoraussetzung autonom ist $((\underline{B} = \emptyset) \Rightarrow (b \in B \setminus \underline{B}))$ und keine Nachrichten zwischen Komponenten mehr unterwegs sind, die bearbeitet werden müssen. Somit treten keine Veränderungen mehr bei irgendwelchen lokalen Variablen (insbesondere $\_z(\cdot)$) auf.

$$\Rightarrow \forall t \in]t^3, \infty[: z^{\Pi,t}\langle V_I System(IS)\rangle(x) = \begin{cases} 1 \text{ falls } x \in c_2 \\ 0 \text{ sonst} \end{cases}$$

$$\Rightarrow \forall t \in]t^3, \infty[: zc(z^{\Pi,t}\langle V_I System(IS)\rangle) = c_2.$$

Die oben angegebene Ausführung Π von $V_I System(IS)$ erfüllt mit $t_0 := t^0$, $t_1 := t^3$, $k := 1$, $c'_0 := c_1$ und t_0, t_1 als max. Folge von Globalzeitpunkten die unter $(*_8)$ geforderten Kriterien. Die Existenz ist somit konstruktiv gezeigt, und folglich gilt $\rightarrow \subseteq \rightarrow'$.

Zu zeigen 3: $\rightarrow \supseteq \rightarrow'$

Sei $(c_1, c_2) \in \rightarrow'$.

$\Rightarrow$ {wegen $(*_4)$}
$((c_1, c_2) \in C' \times C') \wedge (\exists ctr_1, ctr_2 \in Case(IS)^*, \exists c'_0, c'_1, c'_2, \ldots \in Case(IS) : ctr_1.c_1.c_2.ctr_2 = c'_0, c'_1, c'_2, \ldots$ und $c'_0 c'_1 c'_2 \ldots \in \mathcal{CT}[\![IS]\!]$ mit $\forall i = 1, 2, \ldots : |c'_i \setminus c'_{i-1}| = 1)$
$\Rightarrow$ {Abschwächung}
$((c_1, c_2) \in C' \times C') \wedge (\exists c'_0, c'_1, c'_2, \ldots \in Case(IS), \exists k \in I\!N : c'_{k-1} = c_1, c'_k = c_2$ und $|c'_k \setminus c'_{k-1}| = 1)$
$\Rightarrow$ {Zusammenfassen}
$((c_1, c_2) \in C' \times C') \wedge (|c_2 \setminus c_1| = 1)$
$\Rightarrow$ {$C' = C$, $|c_1| = |c_2|$ wegen $(*_3)$}
$((c_1, c_2) \in C \times C) \wedge (|c_2 \setminus c_1| = 1) \wedge (|c_1 \setminus c_2| = 1)$
$\Rightarrow$ {mit $(*_2)$}
$(c_1, c_2) \in \rightarrow$.

Zusammenfassend wurde im Beweis gezeigt: $C = C'$, $\rightarrow \subseteq \rightarrow'$, $\rightarrow \supseteq \rightarrow'$ und damit $\rightarrow = \rightarrow'$. Wegen $(C, \rightarrow) = CG(LCS)$ und $(C', \rightarrow') = CG(IS)$ gilt Satz 8.6. $\square$

Beweis von Satz 8.8 (Gleichheit von $CG(LCS)$ und $\mathcal{CT}^{\mathrm{i}}[\![IS]\!]$)

Es gelten die Voraussetzungen aus dem Satz. $\underline{B}$ bezeichne die Menge der trägen Bereiche und E die Erregungsrelation von IS ($\underline{B} = E = \emptyset$).

Sei $CG(LCS) = (C, \rightarrow)$

$\Rightarrow$ {Definition 8.1.3, Beweis Satz 8.6}
$C = \{c \subseteq P \mid (\forall b \in B : |c \cap b| = 1) \wedge (\forall p_1, p_2 \in c : (p_1, p_2) \notin K)\} = Case(IS)$ $(*_1)$

$\Rightarrow$ {Definitionen 8.4, 8.3.a}
$\rightarrow = \{(c_1, c_2) \in C \times C \mid |c_1 \setminus c_2| = |c_2 \setminus c_1| = 1\}.$ $(*_2)$

Die Definition 5.13 von $\mathcal{CT}^{\mathrm{i}}[\![IS]\!]$ liefert:
$\mathcal{CT}^{\mathrm{i}}[\![IS]\!] = \{c_0c_1c_2 \ldots \in \mathcal{CT}[\![IS]\!] \mid \forall i = 1, 2, \ldots : |c_i \setminus c_{i-1}| = 1\}$

$\Rightarrow$ {Definition 4.8}
$\mathcal{CT}^{\mathrm{i}}[\![IS]\!] = \{c_0c_1c_2 \ldots \in Case(IS)^* \mid \exists$ Ausführung Π von $V_ISystem(IS)$ und eine max. Folge $t_0, t_1, t_2, \ldots$ von Globalzeitpunkten der Ausführung: $\forall j = 0, 1, 2, \ldots : c_j = zc(z^{\Pi, t_j}\langle V_ISystem(IS)\rangle)$, t_0 ist Startzeitpunkt, $z^{\Pi, t_0}\langle V_ISystem(IS)\rangle = cz(c_0)$, $\forall i = 1, 2, \ldots : (t_{i-1} < t_i,\ zc(z^{\Pi, t_{i-1}}\langle V_ISystem(IS)\rangle) \neq zc(z^{\Pi, t_i}\langle V_ISystem(IS)\rangle)$, $(\forall t,\ t_{i-1} \leq t < t_i : zc(z^{\Pi, t}\langle V_ISystem(IS)\rangle) = zc(z^{\Pi, t_{i-1}}\langle V_ISystem(IS)\rangle))$, $|c_i \setminus c_{i-1}| = 1)\}$

$\Rightarrow$ {Zusammenfassen)}
$\mathcal{CT}^{\mathrm{i}}[\![IS]\!] = \{c_0c_1c_2 \ldots \in Case(IS)^* \mid \exists$ Ausführung Π von $V_ISystem(IS)$ und eine max. Folge $t_0, t_1, t_2, \ldots$ von Globalzeitpunkten der Ausführung: $\forall j = 0, 1, 2, \ldots : c_j = zc(z^{\Pi, t_j}\langle V_ISystem(IS)\rangle)$, t_0 ist Startzeitpunkt, $z^{\Pi, t_0}\langle V_ISystem(IS)\rangle = cz(c_0)$, $\forall i = 1, 2, \ldots : (t_{i-1} < t_i,\ |zc(z^{\Pi, t_i}\langle V_ISystem(IS)\rangle) \setminus zc(z^{\Pi, t_{i-1}}\langle V_ISystem(IS)\rangle)| = 1$, $(\forall t,\ t_{i-1} \leq t < t_i : zc(z^{\Pi, t}\langle V_ISystem(IS)\rangle) = zc(z^{\Pi, t_{i-1}}\langle V_ISystem(IS)\rangle))))\}$ $(*_3)$

Zu zeigen 1: $\{c_0c_1c_2 \ldots \mid \langle c_0, c_1, c_2, \ldots\rangle$ ist Pfad in CG(LCS)$\} \subseteq \mathcal{CT}^{\mathrm{i}}[\![IS]\!]$

Sei $\overbrace{\langle c_0, c_1, c_2, \ldots\rangle}^{pfd}$ Pfad in $CG(LCS)$.

$\Rightarrow$ {Pfadunterteilung}
$\forall i = 1, 2, \ldots : \langle c_{i-1}, c_i\rangle$ ist Pfad in $CG(LCS)$
$\Rightarrow$ {Pfaddefinition}
$\forall i = 1, 2, \ldots : (c_{i-1}, c_i) \in \rightarrow$
$\Rightarrow$ {wegen $(*_2)$}
$\forall i = 1, 2, \ldots : ((c_{i-1}, c_i) \in Case(IS) \times Case(IS)) \wedge (|c_{i-1} \setminus c_i| = |c_i \setminus c_{i-1}| = 1)$
$\Rightarrow$ {direkte Folgerung}
$\forall i = 1, 2, \ldots \exists p_i, q_i \in P : \{p_i\} = c_{i-1} \setminus c_i \wedge \{q_i\} = c_i \setminus c_{i-1}$

Setze $\{p_i\} := c_{i-1} \setminus c_i$ und $\{q_i\} := c_i \setminus c_{i-1}$.

$\Rightarrow$ {Definition 8.1.3.b}
$\forall i = 1, 2, \ldots : b(p_i) = b(q_i)$ $(*_4)$
$\Rightarrow$ {Definition 8.1.3.b/c}
$\forall i = 1, 2, \ldots : (v \in c_{i-1} \setminus \{p_i\}) \Rightarrow ((v \notin b(p_i) \wedge v \in c_i \setminus \{q_i\}) \Rightarrow ((q_i, v) \notin K))$ $(*_5)$

Gemäß $(*_3)$ reicht zu zeigen: $\exists$ Ausführung Π von $V_ISystem(IS)$ und eine max. Folge $t_0, t_1, t_2, \ldots$ von Globalzeitpunkten der Ausführung: $\forall j = 0, 1, 2, \ldots : c_j = zc(z^{\Pi, t_j}\langle V_ISystem(IS)\rangle)$, t_0 ist Startzeitpunkt, $z^{\Pi, t_0}\langle V_ISystem(IS)\rangle = cz(c_0)$, $\forall i = 1, 2, \ldots : (t_{i-1} < t_i$, $|zc(z^{\Pi, t_i}\langle V_ISystem(IS)\rangle) \setminus zc(z^{\Pi, t_{i-1}}\langle V_ISystem(IS)\rangle)| = 1$, $(\forall t, t_{i-1} \leq t < t_i : zc(z^{\Pi, t}\langle V_ISystem(IS)\rangle) = zc(z^{\Pi, t_{i-1}}\langle V_ISystem(IS)\rangle)))$ $(*_6)$

Konstruktion der Ausführung Π:
Sei $l := |pfd| + 1$ die Länge des vorgegebenen Pfades plus 1, d.h. $l \in I\!N$ falls pfd eine endliche Länge und $l := \infty$ falls pfd eine unendliche Länge besitzt.

Startzeitpunkt t^0:
Als Startbelegung setze $\_z(x) := 1$ bei $V_{b(x)}$ für alle $x \in c_0$. Alle booleschen Variablen aller

Komponenten von $V_I System(IS)$ seien $false$. Damit sind die Vorbedingungen erfüllt, um bei allen Komponenten die Aktion A1 auszuführen.

$$\Rightarrow z^{\Pi,t^0}\langle V_I System(IS)\rangle(x) = \begin{cases} 1 \text{ falls } x \in c_0 \\ 0 \text{ sonst} \end{cases}$$

$$\Rightarrow zc(z^{\Pi,t^0}\langle V_I System(IS)\rangle) = c_0.$$

Zeitraum $[t^0, t_1^1]$:
Alle Komponenten führen die Initialisierungsaktionen A1 und (gegebenenfalls mehrfach) A2 aus. Der Empfang der $ackinit(\cdot)$-Nachrichten führt bei jeder Komponente zu einer Anpassung der $\_in(\cdot)$-Variablen, was Einfluss auf die booleschen $\_e_{in}(\cdot)$-, $\_e_{out}(\cdot)$- und $\_k(\cdot)$-Belegungen hat. Allerdings gilt auch weiterhin $\_e_{in}(v) = false$ für alle $v \in P$ wegen der Satzvoraussetzung $E = \emptyset$. Die $\_mark(\cdot)$ und $\_s(\cdot)$-Variablen bleiben bei ihrer Anfangsbelegung $false$. Innerhalb von A1 kommt es bei jeder Komponente am Ende zu einem Update-Aufrufe A13.v, der zu keiner Veränderung der $\_z(\cdot)$-Variablen führt. Sei t_1^1 der Globalzeitpunkt, zu dem die letzte Komponente A1 (und damit alle auch A2) beendet hat. Bei jeder Komponente V_b von $V_I System(IS)$ gilt jetzt $\_z(x) = 1$ für die aktuell eingenommene Phase $x \in b$, $\_e_{in}(v) = \_e_{out}(v) = false$ für jede Phase $v \in b$, $\_mark(v') = \_s(v') = false$ für jede Phase $v' \in P$ und $\_in(v'') = true$ gdw. v'' ist Nachbarphase von b und $\_z(v'') = 1$ bei $V_{b(v'')}$.

$$\Rightarrow \forall t \in [t^0, t_1^1] : z^{\Pi,t}\langle V_I System(IS)\rangle(x) = \begin{cases} 1 \text{ falls } x \in c_0 \\ 0 \text{ sonst} \end{cases}$$

$$\Rightarrow \forall t \in [t^0, t_1^1] : zc(z^{\Pi,t}\langle V_I System(IS)\rangle) = c_0.$$

Zeitraum $]t_k^1, t_k^2]$ für $k = 1, 2, \ldots, l$:
Nach t_k^1 führt $V_{b(p_k)}$ als nächste Aktion A3 aus. Dies ist möglich, da $b(p_k)$ nach Satzvoraussetzung autonom ist $((\underline{B} = \emptyset) \Rightarrow (b(p_k) \in B \setminus \underline{B}))$. Wegen $E = \emptyset$ existiert keine $\mathrm{E_{out}}$-Nachbarphase von p_k und $V_{b(p_k)}$ kann $\_z(p_k) := q_k$ (beachte $(*_4)$) ausführen. Dies erfolgt zu einem Zeitpunkt t_k^2. Bei den anderen Komponenten finden keine Aktionen statt.

$$\Rightarrow \forall k \in \{1, 2, \ldots, l\}, \forall t \in]t_k^1, t_k^2[: z^{\Pi,t}\langle V_I System(IS)\rangle(x) = \begin{cases} 1 \text{ falls } x \in c_{k-1} \\ 0 \text{ sonst} \end{cases}$$

$$\forall k \in \{1, 2, \ldots, l\} : z^{\Pi,t_k^2}\langle V_I System(IS)\rangle(x) = \begin{cases} q_k \text{ falls } x = p_k \\ 1 \text{ falls } x \in c_{k-1} \setminus \{p_k\} \\ 0 \text{ sonst} \end{cases}$$

$$\Rightarrow \forall k \in \{1, 2, \ldots, l\}, \forall t \in]t_k^1, t_k^2] : zc(z^{\Pi,t}\langle V_I System(IS)\rangle) = c_{k-1}.$$

Zeitraum $]t_k^2, t_k^3]$ für $k = 1, 2, \ldots, l$:
Nach t_k^2 führt $V_{b(p_k)}$ als nächste Aktion A4 aus. Die Vorbedingung ist erfüllt. $V_{b(p_k)}$ sendet an alle $V_{b'}$, b' ist Nachbarbereich von $\{p_k, q_k\}$, die Nachricht $reqmark(\{p_k, q_k\})$. Alle $V_{b'}$ empfangen die Nachricht und setzen jeweils $\_mark(p_k) := true$ und $\_mark(q_k) := true$. Danach senden sie $ackmark$ zurück (A8). Weitere Variablen werden bei $V_{b'}$ nicht verändert. $V_{b(p_k)}$ empfängt alle $ackmark$-Nachrichten (Fortsetzung A4). Direkt danach gilt bei $V_{b(p_k)}$ $\_z(p_k) = q_k$, da sich die Variable inzwischen nicht geändert hat. Weiterhin gilt $\_mark(v) = false$ für alle E_{out}-Nachbarphasen v von p_k, da $E = \emptyset$ nach Satzvoraussetzung, und es gilt $\_k(q_k) = false$ wegen $(*_5)$, und es gilt $\_mark(v') = false$ für jede K-Nachbarphase v' von q_k, da entweder bisher keine $reqmark(M)$-Nachricht mit $v' \in M$ bearbeitet wurde (A8), was einzig zu einer $true$-Belegung führen kann, oder die Bearbeitung einer $done(M)$-Nachricht die $false$-Belegung bereits wieder hergestellt hat (A10). Gemäß der Aktionsbeschreibung von A4 tritt also bei $V_{b(p_k)}$ der Fall „*not* a) und *not* b) und c)“ ein. Die Phasentransition $p_k \to q_k$ (d.h. $\_z(p_k) := 0$, $\_z(q_k) := 1$ gleichzeitig) wird zu einem Zeitpunkt t_k^3 ausgeführt. Bei allen anderen Komponenten finden keine weiteren Aktionen statt.

$$\Rightarrow \forall k \in \{1, 2, \ldots, l\}, \forall t \in]t_k^2, t_k^3[: z^{\Pi,t}\langle V_I System(IS)\rangle(x) = \begin{cases} q_k \text{ falls } x = p_k \\ 1 \text{ falls } x \in c_{k-1} \setminus \{p_k\} \\ 0 \text{ sonst} \end{cases}$$

$$\forall k \in \{1, 2, \ldots, l\} : z^{\Pi,t_k^3}\langle V_I System(IS)\rangle(x) = \begin{cases} 1 \text{ falls } x \in c_k \\ 0 \text{ sonst} \end{cases}$$

$$\Rightarrow \forall k \in \{1, 2, \ldots, l\}, \forall t \in]t_k^2, t_k^3[: zc(z^{\Pi,t}\langle V_I System(IS)\rangle) = c_{k-1},$$
$$\forall k \in \{1, 2, \ldots, l\} : zc(z^{\Pi,t_k^3}\langle V_I System(IS)\rangle) = c_k.$$

Zeitraum $]t_k^3, t_{k+1}^1]$ für $k = 1, 2, \ldots, l$:
Nach der Phasentransition befindet sich $V_{b(p_k)}$ noch bei der Abarbeitung von A4. Als nächstes erfolgt $\_s(q_k) := false$ und dann ein Update-Aufruf (A13). Die aktuellen Variablenbelegungen

bei $V_{b(p_k)}$ ($\_z(q_k) = 1$ und $\_e_{in}(q_k) = false = \_s(v)$ für alle $v \in P$) führen zur Ausführung von A13.v, die $\_z(\cdot)$-Belegung bleibt unverändert. Nach dem Update sendet $V_{b(p_k)}$ an alle $V_{b'}$, b' ist Nachbarbereich von $\{p_k, q_k\}$, die Nachricht $done(\{p_k, q_k\})$. Alle $V_{b'}$ empfangen die Nachricht (A10) und setzen jeweils $\_mark(p_k) := false$ und $\_mark(q_k) := false$. Die $\_s(v')$-Belegung wird für alle $v' \in b'$ auf $false$ gesetzt, was sie allerdings auch vorher schon war. Jedes $V_{b'}$ führt als nächstes aus A10 heraus den Update-Aufruf (A13) aus. Die aktuellen Variablenbelegungen bei $V_{b'}$ ($\_z(p') = 1$, $\_e_{in}(p') = false$ (wegen $E = \emptyset$) für die momentan eingenommene Phase $p' \in b'$ und $\_s(v) = false$ für alle $v \in P$ (da eine $\_s(\cdot) := true$-Zuweisung bisher nicht stattgefunden hat)) führen zur Ausführung von A13.v, die $\_z(\cdot)$-Belegung bleibt unverändert. Die Abarbeitung von A13 und der aufrufenden Aktion A10 ist bei $V_{b'}$ dann abgeschlossen. Sei t^1_{k+1} der Globalzeitpunkt, zu dem die letzte Komponente $V_{b'}$ A10 abgeschlossen hat. Mit dem Senden der $done(\{p_k, q_k\})$-Nachrichten hat $V_{b(p_k)}$ die Ausführung von A4 abgeschlossen. Bei jeder Komponente V_b von $V_I System(IS)$ gilt zum Zeitpunkt t^1_{k+1} $\_z(x) = 1$ für die aktuell eingenommene Phase $x \in b$, $\_e_{in}(v) = \_e_{out}(v) = false$ für jede Phase $v \in b$, $\_mark(v') = \_s(v') = false$ für jede Phase $v' \in P$ und $\_in(v'') = true$ gdw. v'' ist Nachbarphase von b und $\_z(v'') = 1$ bei $V_{b(v'')}$.

$\Rightarrow \forall k \in \{1, 2, \ldots, l\}, \forall t \in]t^3_k, t^1_{k+1}] : z^{\Pi,t}\langle V_I System(IS)\rangle(x) = \begin{cases} 1 \text{ falls } x \in c_k \\ 0 \text{ sonst} \end{cases}$

$\Rightarrow \forall k \in \{1, 2, \ldots, l\}, \forall t \in]t^3_k, t^1_{k+1}] : zc(z^{\Pi,t}\langle V_I System(IS)\rangle) = c_k$.

Zeitraum $]t^1_{l+1}, \infty]$ bei $l \in I\!N$:
Nach t^1_{l+1} führt Jede Komponente V_b, $b \in B$, von $V_I System(IS)$ nur noch wiederholt die Aktion A3 aus, ohne sich allerdings jemals für eine Zuweisung $\_z(x) := y$ mit $y \in b \setminus \{x\}$ zu entscheiden. Diese wiederholte A3-Ausführung ist möglich, da b nach Satzvoraussetzung autonom ist $((\underline{B} = \emptyset) \Rightarrow (b \in B \setminus \underline{B}))$ und keine Nachrichten zwischen Komponenten mehr unterwegs sind, die bearbeitet werden müssen. Somit treten keine Veränderungen mehr bei irgendwelchen lokalen Variablen (insbesondere $\_z(\cdot)$) auf.

$\Rightarrow \forall t \in]t^1_{l+1}, \infty[: z^{\Pi,t}\langle V_I System(IS)\rangle(x) = \begin{cases} 1 \text{ falls } x \in c_l \\ 0 \text{ sonst} \end{cases}$

$\Rightarrow \forall t \in]t^1_{l+1}, \infty[: zc(z^{\Pi,t}\langle V_I System(IS)\rangle) = c_l$.

Die oben angegebene Ausführung von $V_I System(IS)$ erfüllt mit $t_0 := t^0$, $t_i := t^3_i$ für $i = 1, 2, \ldots, l$ die unter $(*_6)$ geforderten Kriterien. Im Fall eines endlich langen Ausgangspfades ($l \in I\!N$) bildet $t_0, t_1, t_2, \ldots, t_l$ die max. Folge von Globalzeitpunkten. Im Fall eines unendlich langen Ausgangspfades ($l = \infty$) bildet $t_0, t_1, t_2, \ldots, \infty$ die max. Folge von Globalzeitpunkten. Die Existenz ist somit konstruktiv gezeigt, und folglich gilt: $\{c_0 c_1 c_2 \ldots \mid \langle c_0, c_1, c_2, \ldots\rangle$ ist Pfad in CG(LCS)$\} \subseteq \mathcal{CT}^{\mathrm{i}}[\![IS]\!]$.

Zu zeigen 2: $\{c_0 c_1 c_2 \ldots \mid \langle c_0, c_1, c_2, \ldots\rangle$ ist Pfad in CG(LCS)$\} \supseteq \mathcal{CT}^{\mathrm{i}}[\![IS]\!]$

Sei $c_0 c_1 c_2 \ldots \in \mathcal{CT}^{\mathrm{i}}[\![IS]\!]$ mit $c_i \in Case(IS)$, $i = 0, 1, 2, \ldots$.

Zu zeigen: Es existiert ein Pfad $\langle c_0, c_1, c_2, \ldots\rangle$ in $CG(LCS)$.

$c_0 c_1 c_2 \ldots \in \mathcal{CT}^{\mathrm{i}}[\![IS]\!]$
$\Rightarrow$ {Zuordnung}
$\forall i = 1, 2, \ldots : (c_{i-1}, c_i) \in \{(c_1, c_2) \in Case(IS) \times Case(IS) \mid \exists ctr_1, ctr_2 \in Case(IS)^* : ctr_1.c_1.c_2.ctr_2 \in \mathcal{CT}^{\mathrm{i}}[\![IS]\!]\}$

Sei $CG(IS) = (C', \rightarrow')$ der Casegraph von IS.

$\Rightarrow$ {Definition $CG(IS)$}
$\forall i = 1, 2, \ldots : (c_{i-1}, c_i) \in \rightarrow'$
$\Rightarrow$ {Pfadfestlegungen}
$\forall i = 1, 2, \ldots : \langle c_{i-1}, c_i\rangle$ ist Pfad in $CG(IS)$
$\Rightarrow$ {Zusammenfügen der Teilpfade}
$\langle c_0, c_1, c_2, \ldots\rangle$ ist Pfad in $CG(IS)$
$\Rightarrow$ {Satz 8.6}
$\langle c_0, c_1, c_2, \ldots\rangle$ ist Pfad in $CG(LCS)$.

Zusammenfassend wurde im Beweis gezeigt: ($\{c_0 c_1 c_2 \ldots \mid \langle c_0, c_1, c_2, \ldots\rangle$ ist Pfad in CG(LCS)$\} \subseteq \mathcal{CT}^{\mathrm{i}}[\![IS]\!]$) und ($\{c_0 c_1 c_2 \ldots \mid \langle c_0, c_1, c_2, \ldots\rangle$ ist Pfad in CG(LCS)$\} \supseteq \mathcal{CT}^{\mathrm{i}}[\![IS]\!]$). Somit gilt Satz 8.8.

□

A.3 Beweise aus Kapitel 9

Beweis von Satz 9.20 (Lokale Ereignisstruktur)

Es gelten die Bezeichnungen und Voraussetzungen aus dem Satz. Setze $\rightarrow := \rightarrow_1 \cup \rightarrow_2$. Für $p \in b_1$ sei $Suc(p) := \{p' \in b_1 \mid (p,p') \in \rightarrow\}$ die Menge der Nachfolgeknoten und $Pre(p) := \{p' \in b_1 \mid (p',p) \in \rightarrow\}$ die Menge der Vorgängerknoten von p im Graphen $G(b_1)$.

Zu a), $\subseteq$.

Die Erweiterte Casetrace-Semantik von IS ist definiert (Definition 4.16) als:

$\mathcal{ECT}[\![IS]\!] = \{zc(z^{\Pi,t_0}\langle V_I System(IS)\rangle).\delta_1.zc(z^{\Pi,t_1}\langle V_I System(IS)\rangle).\delta_2.zc(z^{\Pi,t_2}\langle V_I System(IS)\rangle) \ldots \in Case(IS).(\mathcal{P}(B).Case(IS))^* \mid \Pi$ ist eine Ausführung von $V_I System(IS)$ und $t_0, t_1, t_2, \ldots$ eine max. Folge von Globalzeitpunkten der Ausführung mit: t_0 ist Startzeitpunkt, $z^{\Pi,t_0}\langle V_I System(IS)\rangle = cz(c_0)$, $c_0 \in Case(IS)$, $\forall i = 1,2,\ldots : (t_{i-1} < t_i,\ zc(z^{\Pi,t_{i-1}}\langle V_I System(IS)\rangle) \neq zc(z^{\Pi,t_i}\langle V_I System(IS)\rangle),\ \forall t, t_{i-1} \leq t < t_i : zc(z^{\Pi,t}\langle V_I System(IS)\rangle) = zc(z^{\Pi,t_{i-1}}\langle V_I System(IS)\rangle),\ \delta_i = \{b(p) \in AB(IS) \mid p \in zc(z^{\Pi,t_i}\langle V_I System(IS)\rangle) \setminus zc(z^{\Pi,t_{i-1}}\langle V_I System(IS)\rangle)$ und $V_{b(p)}$ befindet sich zum Zeitpunkt t_i in Aktion A4.$\}\ )\}$ $(*_1)$

Dabei ist die Menge der Cases gegeben durch:

$Case(IS)$
$= \{$Definition Case$\}$
$\{c \subseteq P \mid \forall b \in B : |c \cap b| = 1 \wedge \forall p_1, p_2 \in c : (p_1,p_2) \notin K\}$
$= \{$Voraussetzungen für P und $B\}$
$\{\{e_j, p_i\} \mid (e_j, p_i) \notin K, j \in \{0,1,\ldots,m\}, i \in \{1,\ldots,m\}\}$
$= \{$Voraussetzung i)$\}$
$\{\{e_j, p_j\} \mid j \in \{1,\ldots,m\}\} \cup \{\{e_j, p_i\} \mid p_j \in Pre(p_i), i,j \in \{1,\ldots,m\}\} \cup \{\{e_0, p_i\} \mid i \in \{1,\ldots,m\}\}$ $(*_2)$

Betrachte im Folgenden eine beliebige Ausführung Π von $V_I System(IS)$ gemäß $(*_1)$ mit den möglichen Ausführungsalternativen. Das Ziel ist es, alle auftretenden Cases sowie bei Caseübergängen die zuständigen Aktionen (A4 oder A5) zu bestimmen. Aus diesen Informationen lässt sich dann die Erweiterte Casetrace-Semantik von IS zusammensetzen. Mittels der Sicht auf b_1 wird letztendlich dieser Teil der Satzaussage bewiesen.

Sei $c_0 \in Case(IS)$ beliebig.

Startzeitpunkt t^0: Als Startbelegung gilt $\_z(x) = 1$ bei $V_{b(x)}$ für beide $x \in c_0$. Alle booleschen Variablen der beiden Komponenten von $V_I System(IS)$, d.h. von V_{b_1} und $V_{\underline{b}_2}$, sind $false$. Damit sind die Vorbedingungen erfüllt, um bei beiden Komponenten die Aktion A1 auszuführen. Die Festlegungen von $Case(IS)$ (siehe $(*_2)$) führen zu einer Unterscheidung von zwei Startmöglichkeiten.

Fall 1). $\exists k_0 \in \{1,\ldots,m\} : z^{\Pi,t^0}\langle V_{b_1}\rangle(p_{k_0}) = 1 \wedge z^{\Pi,t^0}\langle V_{\underline{b}_2}\rangle(e_{k_0}) = 1$

$\Rightarrow zc(z^{\Pi,t^0}\langle V_I System(IS)\rangle) = \{p_{k_0}, e_{k_0}\}$.

Beide Komponenten führen die Initialisierungsaktionen A1 und A2 aus und senden sich gegenseitig $reqinit$ und $ackinit(\cdot)$ Nachrichten. V_{b_1} empfängt $ackinit(e_{k_0})$ von $V_{\underline{b}_2}$ und setzt $\_in(e_{k_0}) := true$. $V_{\underline{b}_2}$ empfängt $ackinit(p_{k_0})$ von V_{b_1} und setzt $\_in(p_{k_0}) := true$. Die lokalen Variablen in beiden Komponenten werden wie folgt automatisch angepasst, für $j = 1,\ldots,m$:

Fall 1.1). $(e_{k_0}, p_{k_0}) \in E$

Bei V_{b_1}:
$(\_e_{in}(p_j) = true$ gdw. $(e_{k_0}, p_j) \in E$ gdw. $j = k_0)$. Die erste Äquivalenz gilt aufgrund der Definition von $\_e_{in}(\cdot)$, die zweite wegen der Satzvoraussetzung iii) und der Fallvoraussetzung.

($\_e_{out}(p_j) = true$ gdw. $(p_j, e_{k_0}) \in E$ gdw. $p_j \in Suc(p_{k_0})$). Die erste Äquivalenz gilt aufgrund der Definition von $\_e_{out}(\cdot)$, die zweite wegen der Satzvoraussetzung ii).
($\_k(p_j) = true$ gdw. $(p_j, e_{k_0}) \in K$ gdw. $p_j \notin Suc(p_{k_0}) \cup \{p_{k_0}\}$). Die erste Äquivalenz gilt aufgrund der Definition von $\_k(\cdot)$, die zweite wegen der Satzvoraussetzung i).
Die $\_mark(\cdot)$ und $\_s(\cdot)$-Variablen bleiben bei ihrer Anfangsbelegung $false$.
Innerhalb von A1 kommt es bei V_{b_1} am Ende zu einem Update-Aufruf A13.iv, der zu $\_z(p_{k_0}) := \mathsf{F}$ zu einem Globalzeitpunkt t^1 führt.

Bei $V_{\underline{b}_2}$:
($\_e_{in}(e_j) = true$ gdw. $(p_{k_0}, e_j) \in E$ gdw. $p_{k_0} \in Suc(p_j)$). Die erste Äquivalenz gilt aufgrund der Definition von $\_e_{in}(\cdot)$, die zweite wegen der Satzvoraussetzung ii).
($\_e_{out}(e_j) = true$ gdw. $(e_j, p_{k_0}) \in E$ gdw. $j = k_0$). Die erste Äquivalenz gilt aufgrund der Definition von $\_e_{out}(\cdot)$, die zweite wegen der Satzvoraussetzung iii) und der Fallvoraussetzung.
($\_k(e_j) = true$ gdw. $(p_{k_0}, e_j) \in K$ gdw. $p_{k_0} \notin Suc(p_j) \cup \{p_j\}$). Die erste Äquivalenz gilt aufgrund der Definition von $\_k(\cdot)$, die zweite wegen der Satzvoraussetzung i).
($\_e_{in}(e_0) = true$ gdw. $(p_{k_0}, e_0) \in E$ gdw. $true$). Die erste Äquivalenz gilt aufgrund der Definition von $\_e_{in}(\cdot)$, die zweite wegen der Satzvoraussetzung v).
($\_e_{out}(e_0) = true$ gdw. $(e_0, p_{k_0}) \in E$ gdw. $false$). Die erste Äquivalenz gilt aufgrund der Definition von $\_e_{out}(\cdot)$, die zweite wegen der Satzvoraussetzung vi).
($\_k(e_0) = true$ gdw. $(p_{k_0}, e_0) \in K$ gdw. $false$). Die erste Äquivalenz gilt aufgrund der Definition von $\_k(\cdot)$, die zweite wegen der Satzvoraussetzung iv).
Die $\_mark(\cdot)$ und $\_s(\cdot)$-Variablen bleiben bei ihrer Anfangsbelegung $false$.
Innerhalb von A1 kommt es bei $V_{\underline{b}_2}$ am Ende zu einem Update-Aufruf A13.v, der zu $\_z(e_{k_0}) := 1$ zu einem Globalzeitpunkt t^2 führt.

Nach t^1 führt V_{b_1} als nächstes die Aktion A5 aus. Nach Senden von $reqmark(b_1)$ an und Empfang von $ackmark$ von $V_{\underline{b}_2}$ gilt, bezogen auf die Aktionsbeschreibung von A5, unter Beachtung der aktuellen Variablenbelegungen bei V_{b_1}: $M_1 = M_2 = \{q' \in b_1 \setminus \{p_{k_0}\} \mid q' \in Suc(p_{k_0}) \cup \{p_{k_0}\} = Suc(p_{k_0})$. Die zweite Gleichheit gilt, da $G(b_1)$ nach Voraussetzung schlingenfrei ist. $M_1 = M_2 = \emptyset$ ist nicht möglich, da nach Fallvoraussetzung $(e_{k_0}, p_{k_0}) \in E$ gilt und damit Satzvoraussetung iii) $\rightarrow_2(p_j) \neq \emptyset$ festlegt. Es folgt $M_1 = M_2 \neq \emptyset$ und Fall „*not* a) und *not* b) und c)“ von A5 tritt ein. V_{b_1} führt zu einem Globalzeitpunkt t^3, $t^3 > t^1$, eine Phasentransition $p_{k_0} \rightarrow q$ aus, mit $q \in M_1 = Suc(p_{k_0})$. Es gilt $q = p_{k_1}$ für ein $k_1 \in \{1, \ldots, m\} \setminus \{k_0\}$. Beim anschließenden Update-Aufruf tritt wegen $\_e_{out}(p_{k_1}) = true$ der Fall A13.v ein, der die aktuellen Variablenbelegungen erhält. A5-abschließend sendet V_{b_1} an $V_{\underline{b}_2}$ die Nachricht $done(p_{k_0} \rightarrow p_{k_1})$.

Da $\underline{b}_2$ nach Voraussetzung nicht autonom ist ($\underline{b}_2 \in \underline{B}$) und $\_z(e_{k_0}) = 1$ gilt, gibt es direkt nach t^2 keine Vorbedingung einer Aktion A1-A13, die erfüllt ist, und $V_{\underline{b}_2}$ ist auf den Empfang von Nachrichten von V_{b_1} angewiesen, bevor neue Variablenbelegungen hervorgerufen werden können. Die erste Nachricht, die $V_{\underline{b}_2}$ nach t^2 erhält, ist $reqmark(b_1)$, abgeschickt von V_{b_1} nach t^1. $V_{\underline{b}_2}$ setzt $\_mark(v) := true$ für alle $v \in b_1$ und schickt $ackmark$ zurück(A8). Die nächste Nachricht, die $V_{\underline{b}_2}$ erhält, ist $done(p_{k_0} \rightarrow p_{k_1})$, zeitlich nach t^3.

$\Rightarrow \forall t \in]t^0, t^3[: zc(z^{\Pi,t}\langle V_I System(IS)\rangle) = \{p_{k_0}, e_{k_0}\}$

$zc(z^{\Pi,t^3}\langle V_I System(IS)\rangle) = \{p_{k_1}, e_{k_0}\}$, $p_{k_1} \in Suc(p_{k_0})$

V_{b_1} befindet sich zum Zeitpunkt t^3 in Aktion A5.

$V_{\underline{b}_2}$ setzt in A10 $\_in(p_{k_0}) := false$ und $\_in(p_{k_1}) := true$. Die lokalen Variablen werden wie folgt automatisch angepasst, für $j = 1, \ldots, m$:
($\_e_{in}(e_j) = true$ gdw. $(p_{k_1}, e_j) \in E$ gdw. $p_{k_1} \in Suc(p_j)$). Die erste Äquivalenz gilt aufgrund der Definition von $\_e_{in}(\cdot)$, die zweite wegen der Satzvoraussetzung ii).
($\_e_{out}(e_j) = true$ gdw. $(e_j, p_{k_1}) \in E$ gdw. $j = k_1$). Die erste Äquivalenz gilt aufgrund der Definition von $\_e_{out}(\cdot)$, die zweite wegen der Satzvoraussetzung iii) und der Fallvoraussetzung.
($\_k(e_j) = true$ gdw. $(p_{k_1}, e_j) \in K$ gdw. $p_{k_1} \notin Suc(p_j) \cup \{p_j\}$). Die erste Äquivalenz gilt aufgrund der Definition von $\_k(\cdot)$, die zweite wegen der Satzvoraussetzung i).
($\_e_{in}(e_0) = true$ gdw. $(p_{k_1}, e_0) \in E$ gdw. $true$). Die erste Äquivalenz gilt aufgrund der Definition von $\_e_{in}(\cdot)$, die zweite wegen der Satzvoraussetzung v).
($\_e_{out}(e_0) = true$ gdw. $(e_0, p_{k_1}) \in E$ gdw. $false$). Die erste Äquivalenz gilt aufgrund der Definition

von $\_e_{out}(\cdot)$, die zweite wegen der Satzvoraussetzung vi).
($\_k(e_0) = true$ gdw. $(p_{k_1}, e_0) \in K$ gdw. $false$). Die erste Äquivalenz gilt aufgrund der Definition von $\_k(\cdot)$, die zweite wegen der Satzvoraussetzung iv).
Alle $\_mark(\cdot)$ und $\_s(\cdot)$-Variablen bleiben oder werden auf $false$ gesetzt.
Der A10-abschließende Update-Aufruf A13.iv bewirkt (da $p_{k_1} \in Suc(p_{k_0})$ und damit $\_e_{in}(e_{k_0}) = true$) $\_z(e_{k_0}) := \mathsf{F}$ bei $V_{\underline{b}_2}$ zu einem Globalzeitpunkt t^4, $t^4 > t^3$.

Nach t^4 führt $V_{\underline{b}_2}$ als nächstes die Aktion A5 aus. Nach Senden von $reqmark(\underline{b}_2)$ an und Empfang von $ackmark$ von V_{b_1} gilt, bezogen auf die Aktionsbeschreibung von A5, unter Beachtung der aktuellen Variablenbelegungen bei $V_{\underline{b}_2}$: $M_1 = \{e_j \in \underline{b}_2 \setminus \{e_{k_0}\} \mid p_{k_1} \in Suc(p_j) \cup \{p_j\} \wedge p_{k_1} \notin Suc(p_j) \wedge j \in \{1, \ldots, m\}\} = \{e_{k_1}\}$, $M_2 = \{e_j \in \underline{b}_2 \setminus \{e_{k_0}\} \mid (j = 0) \vee (p_{k_1} \in Suc(p_j) \cup \{p_j\} \wedge j \in \{1, \ldots, m\})\} = \{e_j \mid p_j \in Pre(p_{k_1}) \wedge j \in \{1, \ldots, m\}\} \cup \{e_0, e_{k_1}\}$. Wegen $e_{k_1} \in M_1 \subseteq M_2$ gilt $M_1 \neq \emptyset \neq M_2$ und Fall „*not* a) und *not* b) und c)“ von A5 tritt ein. $V_{\underline{b}_2}$ führt zu einem Globalzeitpunkt t^5, $t^5 > t^4$, eine Phasentransition $e_{k_0} \to q$ aus, mit $q \in M_1 = \{e_{k_1}\}$. Also $q = e_{k_1}$. Beim anschließenden Update-Aufruf tritt wegen $\_e_{in}(e_{k_1}) = false$ ($G(b_1)$ ist nach Voraussetzung schlingenfrei.) und $\_s(v) = false$ für alle $v \in P$ der Fall A13.v ein, der die aktuellen Variablenbelegungen erhält. A5-abschließend sendet $V_{\underline{b}_2}$ an V_{b_1} die Nachricht $done(e_{k_0} \to e_{k_1})$.

Direkt nach t^3 gilt bei V_{b_1} $\_e_{out}(p_{k_1}) = true$ (wegen $\_in(e_{k_0}) = true \wedge (p_{k_1}, e_{k_0}) \in E$). Die Vorbedingungen aller Aktionen A1-A13 sind entweder nicht erfüllt oder, im Fall A3, es sind nur Alternativen bei der Ausführung möglich, die keine Veränderung der Variablenbelegungen bewirken. V_{b_1} ist somit auf den Empfang von Nachrichten von $V_{\underline{b}_2}$ angewiesen, bevor neue Variablenbelegungen hervorgerufen werden können.
Die erste Nachricht, die V_{b_1} nach t^3 erhält, ist $reqmark(\underline{b}_2)$, abgeschickt von $V_{\underline{b}_2}$ nach t^4. V_{b_1} setzt $\_mark(v) := true$ für alle $v \in \underline{b}_2$ und schickt $ackmark$ zurück (A8). Die nächste Nachricht, die V_{b_1} erhält, ist $done(e_{k_0} \to e_{k_1})$, zeitlich nach t^5.

$$\Rightarrow \forall t \in]t^3, t^5[: zc(z^{\Pi,t}\langle V_I System(IS)\rangle) = \{p_{k_1}, e_{k_0}\}$$
$$zc(z^{\Pi,t^5}\langle V_I System(IS)\rangle) = \{p_{k_1}, e_{k_1}\}$$
$V_{\underline{b}_2}$ befindet sich zum Zeitpunkt t^5 in Aktion A5.

V_{b_1} setzt in A10 $\_in(e_{k_0}) := false$ und $\_in(e_{k_1}) := true$. Die lokalen Variablen von V_{b_1} werden wie folgt automatisch angepasst, für $j = 1, \ldots, m$:

Fall 1.1.1). $(e_{k_1}, p_{k_1}) \in E$

($\_e_{in}(p_j) = true$ gdw. $(e_{k_1}, p_j) \in E$ gdw. $j = k_1$). Die erste Äquivalenz gilt aufgrund der Definition von $\_e_{in}(\cdot)$, die zweite wegen der Satzvoraussetzung iii) und der Fallvoraussetzung.
($\_e_{out}(p_j) = true$ gdw. $(p_j, e_{k_1}) \in E$ gdw. $p_j \in Suc(p_{k_1})$). Die erste Äquivalenz gilt aufgrund der Definition von $\_e_{out}(\cdot)$, die zweite wegen der Satzvoraussetzung ii).
($\_k(p_j) = true$ gdw. $(p_j, e_{k_1}) \in K$ gdw. $p_j \notin Suc(p_{k_1}) \cup \{p_{k_1}\}$). Die erste Äquivalenz gilt aufgrund der Definition von $\_k(\cdot)$, die zweite wegen der Satzvoraussetzung i).
Die $\_mark(\cdot)$ und $\_s(\cdot)$-Variablen bleiben bei ihrer Belegung $false$.
Innerhalb von A10 kommt es bei V_{b_1} am Ende zu einem Update-Aufruf A13.iv, der zu $\_z(p_{k_1}) := \mathsf{F}$ zu einem Globalzeitpunkt t^6 führt.

Nach t^6 führt V_{b_1} als nächstes die Aktion A5 aus. Nach Senden von $reqmark(b_1)$ an und Empfang von $ackmark$ von $V_{\underline{b}_2}$, gilt, bezogen auf die Aktionsbeschreibung von A5, unter Beachtung der aktuellen Variablenbelegungen bei V_{b_1}: $M_1 = M_2 = \{q' \in b_1 \setminus \{p_{k_1}\} \mid q' \in Suc(p_{k_1}) \cup \{p_{k_1}\} = Suc(p_{k_1})$. Die zweite Gleichheit gilt, da $G(b_1)$ nach Voraussetzung schlingenfrei ist. $M_1 = M_2 = \emptyset$ ist nicht möglich, da nach Fallvoraussetzung $(e_{k_1}, p_{k_1}) \in E$ gilt und damit Satzvoraussetung iii) $\to_2(p_j) \neq \emptyset$ festlegt. Es folgt $M_1 = M_2 \neq \emptyset$ und Fall „*not* a) und *not* b) und c)“ von A5 tritt ein. V_{b_1} führt zu einem Globalzeitpunkt t^7, $t^7 > t^6$, eine Phasentransition $p_{k_1} \to q$ aus, mit $q \in M_1 = Suc(p_{k_1})$. Es gilt $q = p_{k_2}$ für ein $k_2 \in \{1, \ldots, m\} \setminus \{k_1\}$. Beim anschließenden Update-Aufruf tritt wegen $\_e_{out}(p_{k_2}) = true$ der Fall A13.v ein, der die aktuellen Variablenbelegungen erhält. A5-abschließend sendet V_{b_1} an $V_{\underline{b}_2}$ die Nachricht $done(p_{k_1} \to p_{k_2})$.

Da $\underline{b}_2$ nach Voraussetzung nicht autonom ist ($\underline{b}_2 \in \underline{B}$) und $\_z(e_{k_1}) = 1$ gilt, gibt es direkt nach t^5 keine Vorbedingung einer Aktion A1-A13, die erfüllt ist, und $V_{\underline{b}_2}$ ist auf den Empfang von

Nachrichten von V_{b_1} angewiesen, bevor neue Variablenbelegungen hervorgerufen werden können. Die erste Nachricht, die $V_{\underline{b}_2}$ nach t^5 erhält, ist $reqmark(b_1)$, abgeschickt von V_{b_1} nach t^6. $V_{\underline{b}_2}$ setzt $\_mark(v) := true$ für alle $v \in b_1$ und schickt $ackmark$ zurück(A8). Die nächste Nachricht, die $V_{\underline{b}_2}$ erhält, ist $done(p_{k_1} \to p_{k_2})$, zeitlich nach t^7.

$\Rightarrow \forall t \in]t^5, t^7[: zc(z^{\Pi,t}\langle V_I System(IS)\rangle) = \{p_{k_1}, e_{k_1}\}$

$zc(z^{\Pi,t^7}\langle V_I System(IS)\rangle) = \{p_{k_2}, e_{k_1}\}$, $p_{k_2} \in Suc(p_{k_1})$

V_{b_1} befindet sich zum Zeitpunkt t^7 in Aktion A5.

$V_{\underline{b}_2}$ setzt in A10 $\_in(p_{k_1}) := false$ und $\_in(p_{k_2}) := true$. Die lokalen Variablen werden wie folgt automatisch angepasst, für $j = 1, \ldots, m$:
($\_e_{in}(e_j) = true$ gdw. $(p_{k_2}, e_j) \in E$ gdw. $p_{k_2} \in Suc(p_j)$). Die erste Äquivalenz gilt aufgrund der Definition von $\_e_{in}(\cdot)$, die zweite wegen der Satzvoraussetzung ii).
($\_e_{out}(e_j) = true$ gdw. $(e_j, p_{k_2}) \in E$ gdw. $j = k_2 \wedge (e_{k_2}, p_{k_2}) \in E$). Die erste Äquivalenz gilt aufgrund der Definition von $\_e_{out}(\cdot)$, die zweite wegen der Satzvoraussetzung iii).
($\_k(e_j) = true$ gdw. $(p_{k_2}, e_j) \in K$ gdw. $p_{k_2} \notin Suc(p_j) \cup \{p_j\}$). Die erste Äquivalenz gilt aufgrund der Definition von $\_k(\cdot)$, die zweite wegen der Satzvoraussetzung i).
($\_e_{in}(e_0) = true$ gdw. $(p_{k_2}, e_0) \in E$ gdw. $true$). Die erste Äquivalenz gilt aufgrund der Definition von $\_e_{in}(\cdot)$, die zweite wegen der Satzvoraussetzung v).
($\_e_{out}(e_0) = true$ gdw. $(e_0, p_{k_2}) \in E$ gdw. $false$). Die erste Äquivalenz gilt aufgrund der Definition von $\_e_{out}(\cdot)$, die zweite wegen der Satzvoraussetzung vi).
($\_k(e_0) = true$ gdw. $(p_{k_2}, e_0) \in K$ gdw. $false$). Die erste Äquivalenz gilt aufgrund der Definition von $\_k(\cdot)$, die zweite wegen der Satzvoraussetzung iv).
Alle $\_mark(\cdot)$ und $\_s(\cdot)$-Variablen bleiben oder werden auf $false$ gesetzt.
Der A10-abschließende Update-Aufruf A13.iv bewirkt (da $p_{k_2} \in Suc(p_{k_1})$ und damit $\_e_{in}(e_{k_1}) = true$) $\_z(e_{k_1}) := \mathsf{F}$ bei $V_{\underline{b}_2}$ zu einem Globalzeitpunkt t^8, $t^8 > t^7$.

Nach t^8 führt $V_{\underline{b}_2}$ als nächstes die Aktion A5 aus. Nach Senden von $reqmark(\underline{b}_2)$ an und Empfang von $ackmark$ von V_{b_1}, gilt, bezogen auf die Aktionsbeschreibung von A5, unter Beachtung der aktuellen Variablenbelegungen bei $V_{\underline{b}_2}$: $M_1 = \{e_j \in \underline{b}_2 \setminus \{e_{k_1}\} \mid p_{k_2} \in Suc(p_j) \cup \{p_j\} \wedge p_{k_2} \notin Suc(p_j) \wedge j \in \{1, \ldots, m\}\} = \{e_{k_2}\}$, $M_2 = \{e_j \in \underline{b}_2 \setminus \{e_{k_1}\} \mid (j = 0) \vee (p_{k_2} \in Suc(p_j) \cup \{p_j\} \wedge j \in \{1, \ldots, m\})\} = \{e_j \mid p_j \in Pre(p_{k_2}) \wedge j \in \{1, \ldots, m\}\} \cup \{e_0, e_{k_2}\}$. Wegen $e_{k_2} \in M_1 \subseteq M_2$ gilt $M_1 \neq \emptyset \neq M_2$ und Fall „*not* a) und *not* b) und c)“ von A5 tritt ein. $V_{\underline{b}_2}$ führt zu einem Globalzeitpunkt t^9, $t^9 > t^8$, eine Phasentransition $e_{k_1} \to q$ aus, mit $q \in M_1 = \{e_{k_2}\}$. Also $q = e_{k_2}$. Beim anschließenden Update-Aufruf tritt wegen $\_e_{in}(e_{k_2}) = false$ ($G(b_1)$ ist nach Voraussetzung schlingenfrei.) und $\_s(v) = false$ für alle $v \in P$ der Fall A13.v ein, der die aktuellen Variablenbelegungen erhält. A5-abschließend sendet $V_{\underline{b}_2}$ an V_{b_1} die Nachricht $done(e_{k_1} \to e_{k_2})$.

Direkt nach t^7 gilt bei V_{b_1} $\_e_{out}(p_{k_2}) = true$ (wegen $\_in(e_{k_1}) = true \wedge (p_{k_2}, e_{k_1}) \in E$). Die Vorbedingungen aller Aktionen A1-A13 sind entweder nicht erfüllt oder, im Fall A3, es sind nur Alternativen bei der Ausführung möglich, die keine Veränderung der Variablenbelegungen bewirken. V_{b_1} ist somit auf den Empfang von Nachrichten von $V_{\underline{b}_2}$ angewiesen, bevor neue Variablenbelegungen hervorgerufen werden können.
Die erste Nachricht, die V_{b_1} nach t^7 erhält, ist $reqmark(\underline{b}_2)$, abgeschickt von $V_{\underline{b}_2}$ nach t^8. V_{b_1} setzt $\_mark(v) := true$ für alle $v \in \underline{b}_2$ und schickt $ackmark$ zurück (A8). Die nächste Nachricht, die V_{b_1} erhält, ist $done(e_{k_1} \to e_{k_2})$, zeitlich nach t^9.

$\Rightarrow \forall t \in]t^7, t^9[: zc(z^{\Pi,t}\langle V_I System(IS)\rangle) = \{p_{k_2}, e_{k_1}\}$

$zc(z^{\Pi,t^9}\langle V_I System(IS)\rangle) = \{p_{k_2}, e_{k_2}\}$

$V_{\underline{b}_2}$ befindet sich zum Zeitpunkt t^9 in Aktion A5.

V_{b_1} setzt in A10 $\_in(e_{k_1}) := false$ und $\_in(e_{k_2}) := true$. Die lokalen Variablen von V_{b_1} werden wie folgt automatisch angepasst, für $j = 1, \ldots, m$:

Fall 1.1.1.1). $(e_{k_2}, p_{k_2}) \in E$

Analog Fall 1.1.1) mit k_{i+1} statt k_i und t^{i+4} statt t^i für alle $i \in I\!N$.

Fall 1.1.1.2). $(e_{k_2}, p_{k_2}) \notin E$

Analog Fall 1.1.2) mit k_{i+1} statt k_i und t^{i+4} statt t^i für alle $i \in I\!N$.

Fall 1.1.2). $(e_{k_1}, p_{k_1}) \notin E$

($\_e_{in}(p_j) = true$ gdw. $(e_{k_1}, p_j) \in E$ gdw. $false$). Die erste Äquivalenz gilt aufgrund der Definition von $\_e_{in}(\cdot)$, die zweite wegen der Satzvoraussetzung iii) und der Fallvoraussetzung.
($\_e_{out}(p_j) = true$ gdw. $(p_j, e_{k_1}) \in E$ gdw. $p_j \in Suc(p_{k_1})$). Die erste Äquivalenz gilt aufgrund der Definition von $\_e_{out}(\cdot)$, die zweite wegen der Satzvoraussetzung ii).
($\_k(p_j) = true$ gdw. $(p_j, e_{k_1}) \in K$ gdw. $p_j \notin Suc(p_{k_1}) \cup \{p_{k_1}\}$). Die erste Äquivalenz gilt aufgrund der Definition von $\_k(\cdot)$, die zweite wegen der Satzvoraussetzung i).
Die $\_mark(\cdot)$ und $\_s(\cdot)$-Variablen bleiben bei ihrer Belegung $false$.
Innerhalb von A10 kommt es bei V_{b_1} am Ende zu einem Update-Aufruf A13.v, der zu $\_z(p_{k_1}) := 1$ zu einem Globalzeitpunkt t^6 führt.

Nach t^6 führt V_{b_1} als nächstes die Aktion A3 aus. Die Vorbedingung ist erfüllt, da b_1 nach Satzvoraussetzung autonom ist ($b_1 \in B \setminus \underline{B}$). Wegen $\_e_{out}(p_{k_1}) = false$ ($G(b_1)$ ist schlingenfrei) und $\_mark(v) = false$ für alle $v \in P$ kann V_{b_1} die Zuweisung $\_z(p_{k_1}) := q$ mit $q \in b_1 \setminus \{p_{k_1}\}$ ausführen. Findet die Zuweisung nicht statt, wird als nächstes wiederum die Aktion A3 ausgeführt, da sich an den Variablenbelegungen in der Zwischenzeit nichts ändert.

Fall 1.1.2.1). $\exists t, t > t^6, \exists q \in b_1 \setminus \{p_{k_1}\} : z^{\Pi,t}\langle V_{b_1}\rangle(p_{k_1}) = q$

Als erste Zuweisung nach t^6 findet bei V_{b_1} zu einem Globalzeitpunkt t^7 die Zuweisung $\_z(p_{k_1}) := q$ mit $q \in b_1 \setminus \{p_{k_1}\}$ statt (Aktion A3).

Fall 1.1.2.1.1). $q \in Suc(p_{k_1})$

Nach t^7 führt V_{b_1} als nächstes die Aktion A4 aus. Nach Senden von $reqmark(\{p_{k_1}, q\})$ an und Empfang von *ackmark* von $V_{\underline{b}_2}$ tritt, bezogen auf die Aktionsbeschreibung von A5, unter Beachtung der aktuellen Variablenbelegungen bei V_{b_1}, der Fall „*not* a) und *not* b) und c)“ ein. V_{b_1} führt zu einem Globalzeitpunkt t^8, $t^8 > t^7$, die Phasentransition $p_{k_1} \to q$ aus. Es gilt $q = p_{k_2}$ für ein $k_2 \in \{1, \ldots, m\} \setminus \{k_1\}$. Beim anschließenden Update-Aufruf tritt wegen $\_e_{out}(p_{k_2}) = true$ der Fall A13.v ein, der die aktuellen Variablenbelegungen erhält. A4-abschließend sendet V_{b_1} an $V_{\underline{b}_2}$ die Nachricht $done(p_{k_1} \to p_{k_2})$.

Da $\underline{b}_2$ nach Voraussetzung nicht autonom ist und $\_z(e_{k_1}) = 1$ gilt, gibt es direkt nach t^5 keine Vorbedingung einer Aktion A1-A13, die erfüllt ist, und $V_{\underline{b}_2}$ ist auf den Empfang von Nachrichten von V_{b_1} angewiesen, bevor neue Variablenbelegungen hervorgerufen werden können.
Die erste Nachricht, die $V_{\underline{b}_2}$ nach t^5 erhält, ist $reqmark(\{p_{k_1}, q\})$, abgeschickt von V_{b_1} nach t^7. $V_{\underline{b}_2}$ setzt $\_mark(p_{k_1}) := true$, $\_mark(q) := true$ und schickt *ackmark* zurück (A8). Die nächste Nachricht, die $V_{\underline{b}_2}$ erhält, ist $done(p_{k_1} \to p_{k_2})$, zeitlich nach t^8.

$\Rightarrow \forall t \in]t^5, t^8[: zc(z^{\Pi,t}\langle V_I System(IS)\rangle) = \{p_{k_1}, e_{k_1}\}$

$zc(z^{\Pi,t^8}\langle V_I System(IS)\rangle) = \{p_{k_2}, e_{k_1}\}$, $p_{k_2} \in Suc(p_{k_1})$

V_{b_1} befindet sich zum Zeitpunkt t^8 in Aktion A4.

$V_{\underline{b}_2}$ setzt in A10 $\_in(p_{k_1}) := false$ und $\_in(p_{k_2}) := true$. Die lokalen Variablen werden wie folgt automatisch angepasst, für $j = 1, \ldots, m$:
($\_e_{in}(e_j) = true$ gdw. $(p_{k_2}, e_j) \in E$ gdw. $p_{k_2} \in Suc(p_j)$). Die erste Äquivalenz gilt aufgrund der Definition von $\_e_{in}(\cdot)$, die zweite wegen der Satzvoraussetzung ii).
($\_e_{out}(e_j) = true$ gdw. $(e_j, p_{k_2}) \in E$ gdw. $j = k_2 \wedge (e_{k_2}, p_{k_2}) \in E$). Die erste Äquivalenz gilt aufgrund der Definition von $\_e_{out}(\cdot)$, die zweite wegen der Satzvoraussetzung iii).
($\_k(e_j) = true$ gdw. $(p_{k_2}, e_j) \in K$ gdw. $p_{k_2} \notin Suc(p_j) \cup \{p_j\}$). Die erste Äquivalenz gilt aufgrund der Definition von $\_k(\cdot)$, die zweite wegen der Satzvoraussetzung i).
($\_e_{in}(e_0) = true$ gdw. $(p_{k_2}, e_0) \in E$ gdw. $true$). Die erste Äquivalenz gilt aufgrund der Definition von $\_e_{in}(\cdot)$, die zweite wegen der Satzvoraussetzung v).
($\_e_{out}(e_0) = true$ gdw. $(e_0, p_{k_2}) \in E$ gdw. $false$). Die erste Äquivalenz gilt aufgrund der Definition von $\_e_{out}(\cdot)$, die zweite wegen der Satzvoraussetzung vi).

($\_k(e_0) = true$ gdw. $(p_{k_2}, e_0) \in K$ gdw. $false$). Die erste Äquivalenz gilt aufgrund der Definition von $\_k(\cdot)$, die zweite wegen der Satzvoraussetzung iv).
Alle $\_mark(\cdot)$ und $\_s(\cdot)$-Variablen bleiben oder werden auf $false$ gesetzt.
Der A10-abschließende Update-Aufruf A13.iv bewirkt (da $p_{k_2} \in Suc(p_{k_1})$ und damit $\_e_{in}(e_{k_1}) = true$) $\_z(e_{k_1}) := \mathsf{F}$ bei $V_{\underline{b}_2}$ zu einem Globalzeitpunkt t^9, $t^9 > t^8$.

Nach t^9 führt $V_{\underline{b}_2}$ als nächstes die Aktion A5 aus. Nach Senden von $reqmark(\underline{b}_2)$ an und Empfang von $ackmark$ von V_{b_1} gilt, bezogen auf die Aktionsbeschreibung von A5, unter Beachtung der aktuellen Variablenbelegungen bei $V_{\underline{b}_2}$: $M_1 = \{e_j \in \underline{b}_2 \setminus \{e_{k_1}\} \mid p_{k_2} \in Suc(p_j) \cup \{p_j\} \wedge p_{k_2} \notin Suc(p_j) \wedge j \in \{1, \ldots, m\}\} = \{e_{k_2}\}$, $M_2 = \{e_j \in \underline{b}_2 \setminus \{e_{k_1}\} \mid (j = 0) \vee (p_{k_2} \in Suc(p_j) \cup \{p_j\} \wedge j \in \{1, \ldots, m\})\} = \{e_j \mid p_j \in Pre(p_{k_2}) \wedge j \in \{1, \ldots, m\}\} \cup \{e_0, e_{k_2}\}$. Wegen $e_{k_2} \in M_1 \subseteq M_2$ gilt $M_1 \neq \emptyset \neq M_2$ und Fall „*not* a) und *not* b) und c)“ von A5 tritt ein. $V_{\underline{b}_2}$ führt zu einem Globalzeitpunkt t^{10}, $t^{10} > t^9$, eine Phasentransition $e_{k_1} \to q$ aus, mit $q \in M_1 = \{e_{k_2}\}$. Also $q = e_{k_2}$. Beim anschließenden Update-Aufruf tritt wegen $\_e_{in}(e_{k_2}) = false$ ($G(b_1)$ ist nach Voraussetzung schlingenfrei.) und $\_s(v) = false$ für alle $v \in P$ der Fall A13.v ein, der die aktuellen Variablenbelegungen erhält. A5-abschließend sendet $V_{\underline{b}_2}$ an V_{b_1} die Nachricht $done(e_{k_1} \to e_{k_2})$.

Direkt nach t^8 gilt bei V_{b_1} $\_e_{out}(p_{k_2}) = true$ (wegen $\_in(e_{k_1}) = true \wedge (p_{k_2}, e_{k_1}) \in E$). Die Vorbedingungen aller Aktionen A1-A13 sind entweder nicht erfüllt oder, im Fall A3, es sind nur Alternativen bei der Ausführung möglich, die keine Veränderung der Variablenbelegungen bewirken. V_{b_1} ist somit auf den Empfang von Nachrichten von $V_{\underline{b}_2}$ angewiesen, bevor neue Variablenbelegungen hervorgerufen werden können.
Die erste Nachricht, die V_{b_1} nach t^8 erhält, ist $reqmark(\underline{b}_2)$, abgeschickt von $V_{\underline{b}_2}$ nach t^9. V_{b_1} setzt $\_mark(v) := true$ für alle $v \in \underline{b}_2$ und schickt $ackmark$ zurück (A8). Die nächste Nachricht, die V_{b_1} erhält, ist $done(e_{k_1} \to e_{k_2})$, zeitlich nach t^{10}.

$\Rightarrow \forall t \in]t^8, t^{10}[: zc(z^{\Pi,t}\langle V_I System(IS)\rangle) = \{p_{k_2}, e_{k_1}\}$

$zc(z^{\Pi,t^{10}}\langle V_I System(IS)\rangle) = \{p_{k_2}, e_{k_2}\}$

$V_{\underline{b}_2}$ befindet sich zum Zeitpunkt t^{10} in Aktion A5.

V_{b_1} setzt in A10 $\_in(e_{k_1}) := false$ und $\_in(e_{k_2}) := true$. Die lokalen Variablen von V_{b_1} werden wie folgt automatisch angepasst, für $j = 1, \ldots, m$:

Fall 1.1.2.1.1.1). $(e_{k_2}, p_{k_2}) \in E$

Analog Fall 1.1.1) mit k_{i+1} statt k_i und t^{i+5} statt t^i für alle $i \in \mathbb{N}$.

Fall 1.1.2.1.1.2). $(e_{k_2}, p_{k_2}) \notin E$

Analog Fall 1.1.2) mit k_{i+1} statt k_i und t^{i+5} statt t^i für alle $i \in \mathbb{N}$.

Fall 1.1.2.1.2). $q \notin Suc(p_{k_1})$

Nach t^7 führt V_{b_1} als nächstes die Aktion A4 aus. Nach Senden von $reqmark(\{p_{k_1}, q\})$ an und Empfang von $ackmark$ von $V_{\underline{b}_2}$ gilt, bezogen auf die Aktionsbeschreibung von A5, unter Beachtung der aktuellen Variablenbelegungen, $\_k(q) = true$. Alle $\_mark(\cdot)$-Variablen sind $false$. Bei V_{b_1} tritt folglich der Fall „*not* a) und *not* b) und *not* c)“ ein. V_{b_1} sendet an $V_{\underline{b}_2}$ die Nachricht $break$. A4-abschließend erfolgt ein Solicitation-Aufruf bzgl. $\{q\}$, Aktion A6. Gemäß A6 sendet V_{b_1} an $V_{\underline{b}_2}$ die Nachricht $solicit(\{q\})$ und setzt $\_s(q) := true$ zu einem Globalzeitpunkt t^8. Danach bleiben die Variablenbelegungen bei V_{b_1} solange konstant, bis eine Nachricht von $V_{\underline{b}_2}$ eintrifft und bearbeitet wird. In der Zwischenzeit findet bei V_{b_1} keine Aktion statt, da keine Vorbedingung erfüllt ist.

Da $\underline{b}_2$ nach Voraussetzung nicht autonom ist und $\_z(e_{k_1}) = 1$ gilt, gibt es direkt nach t^5 keine Vorbedingung einer Aktion A1-A13, die erfüllt ist, und $V_{\underline{b}_2}$ ist auf den Empfang von Nachrichten von V_{b_1} angewiesen, bevor neue Variablenbelegungen hervorgerufen werden können. Die erste Nachricht, die $V_{\underline{b}_2}$ nach t^5 erhält, ist $reqmark(\{p_{k_1}, q\})$, abgeschickt von V_{b_1} nach t^7. $V_{\underline{b}_2}$ setzt $\_mark(p_{k_1}) := true$, $\_mark(q) := true$ und schickt $ackmark$ zurück (A8). Die nächste Nachricht, die $V_{\underline{b}_2}$ erhält, ist $break$, worauf A9 ausgeführt wird mit $\_mark(v) := false$ für alle $v \in b_1$. Somit sind wieder alle $\_mark(\cdot)$-Variablen $false$. Als nächstes bearbeitet $V_{\underline{b}_2}$ die Nachricht $solicit(\{q\})$ von V_{b_1} mittels Aktion A11 und setzt $\_s(q) := true$. Unter Berücksichtigung der aktuellen

Variablenbelegung bei $V_{\underline{b}_2}$ gilt $\_e_{out}(e_{k_1}) = false$, da nach übergeordneter Fallvoraussetzung $(e_{k_1}, p_{k_1}) \notin E$ gilt. Somit resultiert der A11-abschließende Update-Aufruf A13.iv in $\_z(e_{k_1}) := \mathsf{F}$ zu einem Globalzeitpunkt $t^9, t^9 > t^8$.

Nach t^9 führt $V_{\underline{b}_2}$ als nächstes die Aktion A5 aus. Nach Senden von $reqmark(\underline{b}_2)$ an und Empfang von $ackmark$ von V_{b_1} gilt, bezogen auf die Aktionsbeschreibung von A5, unter Beachtung der aktuellen Variablenbelegungen bei $V_{\underline{b}_2}$: $M_1 = \{e_j \in \underline{b}_2 \setminus \{e_{k_1}\} \mid p_{k_1} \in Suc(p_j) \cup \{p_j\} \wedge p_{k_1} \notin Suc(p_j) \wedge j \in \{1,\ldots,m\}\} = \emptyset$, $M_2 = \{e_j \in \underline{b}_2 \setminus \{e_{k_1}\} \mid (j = 0) \vee (p_{k_1} \in Suc(p_j) \cup \{p_j\} \wedge j \in \{1,\ldots,m\})\} = \{e_j \mid p_j \in Pre(p_{k_1}) \wedge j \in \{1,\ldots,m\}\} \cup \{e_0\}$. Wegen $e_0 \in M_2$ gilt $M_2 \neq \emptyset$ und Fall „*not* a) und *not* b) und c)" von A5 tritt ein. $V_{\underline{b}_2}$ führt zu einem Globalzeitpunkt t^{10}, $t^{10} > t^9$, eine Phasentransition $e_{k_1} \to q'$ aus, mit $q' \in M_2$, da $M_1 = \emptyset$. Sei $q' = e_{k_2}$, $k_2 \in \{0, 1, \ldots, m\} \setminus \{k_1\}$. Beim anschließenden Update-Aufruf tritt wegen $\_e_{in}(e_{k_2}) = true$ $(e_{k_2} = e_0 \vee p_{k_1} \in Suc(p_{k_2}))$ der Fall A13.iv ein mit $\_z(e_{k_2}) := \mathsf{F}$ bei $V_{\underline{b}_2}$ zu einem Globalzeitpunkt t^{11}, $t^{11} > t^{10}$. A5-abschließend sendet $V_{\underline{b}_2}$ an V_{b_1} die Nachricht $done(e_{k_1} \to e_{k_2})$.

Die erste Nachricht, die V_{b_1} nach t^8 erhält, ist $reqmark(\underline{b}_2)$, abgeschickt von $V_{\underline{b}_2}$ nach t^9. V_{b_1} setzt $\_mark(v) := true$ für alle $v \in \underline{b}_2$ und schickt $ackmark$ zurück (A8). Die nächste Nachricht, die V_{b_1} erhält, ist $done(e_{k_1} \to e_{k_2})$, zeitlich nach t^{10}.

$\Rightarrow \forall t \in]t^5, t^{10}[: zc(z^{\Pi,t}\langle V_I System(IS)\rangle) = \{p_{k_1}, e_{k_1}\}$

$zc(z^{\Pi,t^{10}}\langle V_I System(IS)\rangle) = \{p_{k_1}, e_{k_2}\}$

$V_{\underline{b}_2}$ befindet sich zum Zeitpunkt t^{10} in Aktion A5.

V_{b_1} setzt in A10 $\_in(e_{k_1}) := false$, $\_in(e_{k_2}) := true$, $\_s(v') := false$ und $\_mark(v') := false$ für alle $v' \in \underline{b}_2$. Die lokalen Variablen von V_{b_1} werden wie folgt automatisch angepasst, für $j = 1, \ldots, m$:
$(\_e_{in}(p_j) = true$ gdw. $(e_{k_2}, p_j) \in E$ gdw. $j = k_2 \wedge (e_{k_2}, p_{k_2}) \in E)$. Die erste Äquivalenz gilt aufgrund der Definition von $\_e_{in}(\cdot)$, die zweite wegen Satzvoraussetzung iii).
$(\_e_{out}(p_j) = true$ gdw. $(p_j, e_{k_2}) \in E$ gdw. $(e_{k_2} = e_0 \vee p_j \in Suc(p_{k_2})))$. Die erste Äquivalenz gilt aufgrund der Definition von $\_e_{out}(\cdot)$, die zweite wegen der Satzvoraussetzungen v) und ii).
$(\_k(p_j) = true$ gdw. $(p_j, e_{k_2}) \in K$ gdw. $p_j \notin Suc(p_{k_2}) \cup \{p_{k_2}\})$. Die erste Äquivalenz gilt aufgrund der Definition von $\_k(\cdot)$, die zweite wegen der Satzvoraussetzung i).
Die $\_mark(\cdot)$ und $\_s(\cdot)$-Variablen bleiben bei ihrer Belegung $false$ bis auf $\_s(q) = true$.
Innerhalb von A10 kommt es bei V_{b_1} am Ende zu einem Update-Aufruf A13. Wegen $(e_{k_2} = e_0 \vee p_{k_1} \in Suc(p_{k_2}))$ gilt $\_e_{out}(p_{k_1}) = true$ und Fall A13.ii tritt ein. Es erfolgt ein Cancellation-Aufruf A7, infolgedessen V_{b_1} an $V_{\underline{b}_2}$ die Nachricht $cancel(\{q\})$ schickt und $\_s(q) := false$ setzt. A13.ii-abschließend erfolgt $\_z(p_{k_1}) := 1$ bei V_{b_1} zu einem Globalzeitpunkt t^{12}.

Nach t^{11} führt $V_{\underline{b}_2}$ als nächstes die Aktion A5 aus und sendet $reqmark(\underline{b}_2)$ an V_{b_1} und erwartet $ackmark$ als Bestätigung. Nach t^{11} und vor Empfang von $ackmark$ von V_{b_1} bearbeitet $V_{\underline{b}_2}$ außerdem noch die eingehende Nachricht $cancel(\{q\})$ gemäß A12. Nach $\_s(q) := false$ erfolgt ein Update-Aufruf A13, Fall v (es gilt $\_e_{in}(k_2) = true$), ohne Veränderung der Variablenbelegungen. Nach Empfang von $ackmark$ von V_{b_1} gilt, bezogen auf die Aktionsbeschreibung von A5, unter Beachtung der aktuellen Variablenbelegungen bei $V_{\underline{b}_2}$: $M_1 = \{e_j \in \underline{b}_2 \setminus \{e_{k_2}\} \mid p_{k_1} \in Suc(p_j) \cup \{p_j\} \wedge p_{k_1} \notin Suc(p_j) \wedge j \in \{1,\ldots,m\}\} = \{e_{k_1}\}$, $M_2 = \{e_j \in \underline{b}_2 \setminus \{e_{k_2}\} \mid (j = 0) \vee (p_{k_1} \in Suc(p_j) \cup \{p_j\} \wedge j \in \{1,\ldots,m\})\} = \{e_j \mid p_j \in Pre(p_{k_1}) \wedge j \in \{1,\ldots,m\}\} \cup \{e_0, e_{k_1}\} \setminus \{e_{k_2}\}$. Wegen $e_{k_1} \in M_1 \subseteq M_2$ gilt $M_1 \neq \emptyset \neq M_2$ und Fall „*not* a) und *not* b) und c)" von A5 tritt ein. $V_{\underline{b}_2}$ führt zu einem Globalzeitpunkt t^{13}, $t^{13} > t^{11}$, eine Phasentransition $e_{k_2} \to q$ aus, mit $q \in M_1 = \{e_{k_1}\}$. Also $q = e_{k_1}$. Beim anschließenden Update-Aufruf tritt wegen $\_e_{in}(e_{k_1}) = false$ ($G(b_1)$ ist nach Voraussetzung schlingenfrei.) und $\_s(v) = false$ für alle $v \in P$ der Fall A13.v ein, der die aktuellen Variablenbelegungen erhält. A5-abschließend sendet $V_{\underline{b}_2}$ an V_{b_1} die Nachricht $done(e_{k_2} \to e_{k_1})$.

Direkt nach t^{12} gilt bei V_{b_1} $\_e_{out}(p_{k_1}) = true$ (wegen $\_in(e_{k_2}) = true \wedge (p_{k_1}, e_{k_2}) \in E$). Die Vorbedingungen aller Aktionen A1-A13 sind entweder nicht erfüllt oder, im Fall A3, es sind nur Alternativen bei der Ausführung möglich, die keine Veränderung der Variablenbelegungen bewirken. V_{b_1} ist somit auf den Empfang von Nachrichten von $V_{\underline{b}_2}$ angewiesen, bevor neue Variablenbelegungen hervorgerufen werden können.

Die erste Nachricht, die V_{b_1} nach t^{12} erhält, ist $reqmark(\underline{b}_2)$, abgeschickt von $V_{\underline{b}_2}$ nach t^{11}. V_{b_1} setzt $\_mark(v) := true$ für alle $v \in \underline{b}_2$ und schickt $ackmark$ zurück (A8). Die nächste Nachricht, die V_{b_1} erhält, ist $done(e_{k_2} \to e_{k_1})$, zeitlich nach t^{13}.

$\Rightarrow \forall t \in]t^{12}, t^{13}[: zc(z^{\Pi,t}\langle V_I System(IS)\rangle) = \{p_{k_1}, e_{k_2}\}$

$zc(z^{\Pi,t^{13}}\langle V_I System(IS)\rangle) = \{p_{k_1}, e_{k_1}\}$

$V_{\underline{b}_2}$ befindet sich zum Zeitpunkt t^{13} in Aktion A5.

V_{b_1} setzt in A10 $\_in(e_{k_2}) := false$ und $\_in(e_{k_1}) := true$. Die lokalen Variablen von V_{b_1} werden wie folgt automatisch angepasst, für $j = 1, \ldots, m$:

Fall 1.1.2.1.2.1). $(e_{k_1}, p_{k_1}) \in E$

Analog Fall 1.1.1) mit t^{i+8} statt t^i für alle $i \in I\!N$.

Fall 1.1.2.1.2.2). $(e_{k_1}, p_{k_1}) \notin E$

Analog Fall 1.1.2) mit t^{i+8} statt t^i für alle $i \in I\!N$.

Fall 1.1.2.2). $\nexists t, t > t^6, \nexists q \in b_1 \setminus \{p_{k_1}\} : z^{\Pi,t}\langle V_{b_1}\rangle(p_{k_1}) = q$

Nach t^6 führt V_{b_1} nur noch wiederholt die Aktion A3 aus, ohne sich jemals für eine Zuweisung $\_z(p_{k_1}) := q$ mit $q \in b_1 \setminus \{p_{k_1}\}$ zu entscheiden. Diese wiederholte A3-Ausführung ist möglich, da b_1 nach Satzvoraussetzung autonom ist und keine Nachrichten zwischen den beiden Komponenten mehr unterwegs sind, die bearbeitet werden müssen. Somit treten keine Veränderungen mehr bei irgendwelchen lokalen Variablen (insbesondere $\_z(\cdot)$) auf.

Da $\underline{b}_2$ nach Voraussetzung nicht autonom ist und $\_z(e_{k_1}) = 1$ gilt, gibt es direkt nach t^5 keine Vorbedingung einer Aktion A1-A13, die erfüllt ist, und $V_{\underline{b}_2}$ ist auf den Empfang von Nachrichten von V_{b_1} angewiesen, bevor neue Variablenbelegungen hervorgerufen werden können. Da V_{b_1} nur noch A3 ausführt, werden keine Nachrichten mehr versendet bzw. empfangen und die Belegungen bleiben somit bei $V_{\underline{b}_2}$ ab t^5 konstant.

$\Rightarrow \forall t \in]t^5, \infty[: zc(z^{\Pi,t}\langle V_I System(IS)\rangle) = \{p_{k_1}, e_{k_1}\}$

Fall 1.2). $(e_{k_0}, p_{k_0}) \notin E$

Bei V_{b_1}:
($\_e_{in}(p_j) = true$ gdw. $(e_{k_0}, p_j) \in E$ gdw. $false$). Die erste Äquivalenz gilt aufgrund der Definition von $\_e_{in}(\cdot)$, die zweite wegen der Satzvoraussetzung iii) und der Fallvoraussetzung.
($\_e_{out}(p_j) = true$ gdw. $(p_j, e_{k_0}) \in E$ gdw. $p_j \in Suc(p_{k_0})$). Die erste Äquivalenz gilt aufgrund der Definition von $\_e_{out}(\cdot)$, die zweite wegen der Satzvoraussetzung ii).
($\_k(p_j) = true$ gdw. $(p_j, e_{k_0}) \in K$ gdw. $p_j \notin Suc(p_{k_0}) \cup \{p_{k_0}\}$). Die erste Äquivalenz gilt aufgrund der Definition von $\_k(\cdot)$, die zweite wegen der Satzvoraussetzung i).
Die $\_mark(\cdot)$ und $\_s(\cdot)$-Variablen bleiben bei ihrer Anfangsbelegung $false$.
Innerhalb von A1 kommt es bei V_{b_1} am Ende zu einem Update-Aufruf A13.v, der zu $\_z(p_{k_0}) := 1$ zu einem Globalzeitpunkt t^1 führt.

Bei $V_{\underline{b}_2}$:
($\_e_{in}(e_j) = true$ gdw. $(p_{k_0}, e_j) \in E$ gdw. $p_{k_0} \in Suc(p_j)$). Die erste Äquivalenz gilt aufgrund der Definition von $\_e_{in}(\cdot)$, die zweite wegen der Satzvoraussetzung ii).
($\_e_{out}(e_j) = true$ gdw. $(e_j, p_{k_0}) \in E$ gdw. $false$). Die erste Äquivalenz gilt aufgrund der Definition von $\_e_{out}(\cdot)$, die zweite wegen der Satzvoraussetzungen iii) und vi) und der Fallvoraussetzung.
($\_k(e_j) = true$ gdw. $(p_{k_0}, e_j) \in K$ gdw. $p_{k_0} \notin Suc(p_j) \cup \{p_j\}$). Die erste Äquivalenz gilt aufgrund der Definition von $\_k(\cdot)$, die zweite wegen der Satzvoraussetzung i).
($\_e_{in}(e_0) = true$ gdw. $(p_{k_0}, e_0) \in E$ gdw. $true$). Die erste Äquivalenz gilt aufgrund der Definition von $\_e_{in}(\cdot)$, die zweite wegen der Satzvoraussetzung v).
($\_e_{out}(e_0) = true$ gdw. $(e_0, p_{k_0}) \in E$ gdw. $false$). Die erste Äquivalenz gilt aufgrund der Definition von $\_e_{out}(\cdot)$, die zweite wegen der Satzvoraussetzung vi).

($\_k(e_0) = true$ gdw. $(p_{k_0}, e_0) \in K$ gdw. $false$). Die erste Äquivalenz gilt aufgrund der Definition von $\_k(\cdot)$, die zweite wegen der Satzvoraussetzung iv).
Die $\_mark(\cdot)$ und $\_s(\cdot)$-Variablen bleiben bei ihrer Anfangsbelegung $false$.
Innerhalb von A1 kommt es bei $V_{\underline{b}_2}$ am Ende zu einem Update-Aufruf A13.v, der zu $\_z(e_{k_0}) := 1$ zu einem Globalzeitpunkt t^2 führt.

Bei $V_{\underline{b}_2}$ kann eine Phasentransition (Aktion A4 oder A5) - sofern überhaupt eine eintritt - nur nach A1 und damit nach t^2 eintreten (Verhaltensaxiom VA1). Da b_1 der einzige Nachbarbereich von $\underline{b}_2$ ist, wird nach Empfang von $ackinit(p_{k_0})$ von V_{b_1} bei $V_{\underline{b}_2}$ innerhalb von A1 auf keine weiteren Nachrichten mehr reagiert (Verhaltensaxiom VA2), auch nicht auf ein $reqmark(\cdot)$. Bei V_{b_1} kann deshalb eine Phasentransition - sofern überhaupt eine eintritt - auch erst nach t^2 eintreten, da V_{b_1} dazu eine $ackmark$ Bestätigung von $V_{\underline{b}_2}$ benötigt.

$\Rightarrow \forall t \in]t^0, t^2] : zc(z^{\Pi,t}\langle V_I System(IS)\rangle) = \{p_{k_0}, e_{k_0}\}$

Nach t^1 führt V_{b_1} als nächstes die Aktion A3 aus, Aktionsbeginn ist ein Globalzeitpunkt t^3. Die Vorbedingung ist erfüllt, da b_1 nach Satzvoraussetzung autonom ist ($b_1 \in B \setminus \underline{B}$). Wegen $\_e_{out}(p_{k_1}) = false$ ($G(b_1)$ ist schlingenfrei) und $\_mark(v) = false$ für alle $v \in P$ kann V_{b_1} die Zuweisung $\_z(p_{k_1}) := q$ mit $q \in b_1 \setminus \{p_{k_1}\}$ ausführen. Findet die Zuweisung nicht statt, wird als nächstes wiederum die Aktion A3 ausgeführt, da sich an den Variablenbelegungen in der Zwischenzeit nichts ändert.

Fall 1.2.1). $\exists t, t > t^3, \exists q \in b_1 \setminus \{p_{k_0}\} : z^{\Pi,t}\langle V_{b_1}\rangle(p_{k_0}) = q$
Analog Fall 1.1.2.1) mit k_{i-1} statt k_i und t^{i-3} statt t^i für alle $i \in I\!N$.

Fall 1.2.2). $\nexists t, t > t^3, \nexists q \in b_1 \setminus \{p_{k_0}\} : z^{\Pi,t}\langle V_{b_1}\rangle(p_{k_0}) = q$
Analog Fall 1.1.2.2) mit k_{i-1} statt k_i und t^{i-3} statt t^i für alle $i \in I\!N$.

Fall 2). $\exists k_0 \in \{0, 1, \ldots, m\}, \exists k_1 \in \{1, \ldots, m\}, k_0 \neq k_1 : z^{\Pi,t^0}\langle V_{b_1}\rangle(p_{k_1}) = 1 \wedge z^{\Pi,t^0}\langle V_{\underline{b}_2}\rangle(e_{k_0}) = 1 \wedge (p_{k_0} \in Pre(p_{k_1}) \vee e_{k_0} = e_0)$

$\Rightarrow zc(z^{\Pi,t^0}\langle V_I System(IS)\rangle) = \{p_{k_1}, e_{k_0}\}$.

Beide Komponenten führen die Initialisierungsaktionen A1 und A2 aus und senden sich gegenseitig $reqinit$ und $ackinit(\cdot)$ Nachrichten. V_{b_1} empfängt $ackinit(e_{k_0})$ von $V_{\underline{b}_2}$ und setzt $\_in(e_{k_0}) := true$. $V_{\underline{b}_2}$ empfängt $ackinit(p_{k_1})$ von V_{b_1} und setzt $\_in(p_{k_1}) := true$. Die lokalen Variablen in beiden Komponenten werden wie folgt automatisch angepasst, für $j = 1, \ldots, m$:

Bei V_{b_1}:
($\_e_{in}(p_j) = true$ gdw. $(e_{k_0}, p_j) \in E$ gdw. $j = k_0 \wedge (e_{k_0}, p_{k_0}) \in E$). Die erste Äquivalenz gilt aufgrund der Definition von $\_e_{in}(\cdot)$, die zweite wegen der Satzvoraussetzung iii).
($\_e_{out}(p_j) = true$ gdw. $(p_j, e_{k_0}) \in E$ gdw. $p_j \in Suc(p_{k_0}) \vee e_{k_0} = e_0$). Die erste Äquivalenz gilt aufgrund der Definition von $\_e_{out}(\cdot)$, die zweite wegen den Satzvoraussetzungen ii) und v).
($\_k(p_j) = true$ gdw. $(p_j, e_{k_0}) \in K$ gdw. $p_j \notin Suc(p_{k_0}) \cup \{p_{k_0}\}$). Die erste Äquivalenz gilt aufgrund der Definition von $\_k(\cdot)$, die zweite wegen den Satzvoraussetzungen i) und iv).
Die $\_mark(\cdot)$ und $\_s(\cdot)$-Variablen bleiben bei ihrer Anfangsbelegung $false$.
Innerhalb von A1 kommt es bei V_{b_1} am Ende zu einem Update-Aufruf A13.v, der zu $\_z(p_{k_1}) := 1$ zu einem Globalzeitpunkt t^1 führt.

Bei $V_{\underline{b}_2}$:
($\_e_{in}(e_j) = true$ gdw. $(p_{k_1}, e_j) \in E$ gdw. $p_{k_1} \in Suc(p_j)$). Die erste Äquivalenz gilt aufgrund der Definition von $\_e_{in}(\cdot)$, die zweite wegen der Satzvoraussetzung ii).
($\_e_{out}(e_j) = true$ gdw. $(e_j, p_{k_1}) \in E$ gdw. $j = k_1 \wedge (e_{k_1}, p_{k_1}) \in E$). Die erste Äquivalenz gilt aufgrund der Definition von $\_e_{out}(\cdot)$, die zweite wegen der Satzvoraussetzung iii).
($\_k(e_j) = true$ gdw. $(p_{k_1}, e_j) \in K$ gdw. $p_{k_1} \notin Suc(p_j) \cup \{p_j\}$). Die erste Äquivalenz gilt aufgrund der Definition von $\_k(\cdot)$, die zweite wegen der Satzvoraussetzung i).
($\_e_{in}(e_0) = true$ gdw. $(p_{k_1}, e_0) \in E$ gdw. $true$). Die erste Äquivalenz gilt aufgrund der Definition von $\_e_{in}(\cdot)$, die zweite wegen der Satzvoraussetzung v).

($\_e_{out}(e_0) = true$ gdw. $(e_0, p_{k_1}) \in E$ gdw. $false$). Die erste Äquivalenz gilt aufgrund der Definition von $\_e_{out}(\cdot)$, die zweite wegen der Satzvoraussetzung vi).
($\_k(e_0) = true$ gdw. $(p_{k_1}, e_0) \in K$ gdw. $false$). Die erste Äquivalenz gilt aufgrund der Definition von $\_k(\cdot)$, die zweite wegen der Satzvoraussetzung iv).
Die $\_mark(\cdot)$ und $\_s(\cdot)$-Variablen bleiben bei ihrer Anfangsbelegung $false$.
Innerhalb von A1 kommt es bei $V_{\underline{b}_2}$ am Ende zu einem Update-Aufruf A13.iv, der zu $\_z(e_{k_0}) := \mathsf{F}$ (beachte $k_0 \neq k_1$) zu einem Globalzeitpunkt t^2 führt.

Nach t^2 führt $V_{\underline{b}_2}$ als nächstes die Aktion A5 aus. Nach Senden von $reqmark(\underline{b}_2)$ an und Empfang von $ackmark$ von V_{b_1}, gilt, bezogen auf die Aktionsbeschreibung von A5, unter Beachtung der aktuellen Variablenbelegungen bei $V_{\underline{b}_2}$: $M_1 = \{e_j \in \underline{b}_2 \setminus \{e_{k_0}\} \mid p_{k_1} \in Suc(p_j) \cup \{p_j\} \wedge p_{k_1} \notin Suc(p_j) \wedge j \in \{1, \ldots, m\}\} = \{e_{k_1}\}$, $M_2 = \{e_j \in \underline{b}_2 \setminus \{e_{k_0}\} \mid (j = 0) \vee (p_{k_1} \in Suc(p_j) \cup \{p_j\} \wedge j \in \{1, \ldots, m\})\} = \{e_j \mid p_j \in Pre(p_{k_1}) \wedge j \in \{1, \ldots, m\}\} \cup \{e_0, e_{k_1}\} \setminus \{e_{k_0}\}$. Wegen $e_{k_1} \in M_1 \subseteq M_2$ gilt $M_1 \neq \emptyset \neq M_2$ und Fall „*not* a) und *not* b) und c)“ von A5 tritt ein. $V_{\underline{b}_2}$ führt zu einem Globalzeitpunkt t^3, $t^3 > t^2$ eine Phasentransition $e_{k_0} \to q$ aus, mit $q \in M_1 = \{e_{k_1}\}$. Also $q = e_{k_1}$. Beim anschließenden Update-Aufruf tritt wegen $\_e_{in}(e_{k_1}) = false$ ($G(b_1)$ ist nach Voraussetzung schlingenfrei.) und $\_s(v) = false$ für alle $v \in P$ der Fall A13.v ein, der die aktuellen Variablenbelegungen erhält. A5-abschließend sendet $V_{\underline{b}_2}$ an V_{b_1} die Nachricht $done(e_{k_0} \to e_{k_1})$.

Direkt nach t^1 gilt bei V_{b_1} $\_e_{out}(p_{k_1}) = true$ (wegen $p_{k_1} \in Suc(p_{k_0}) \vee e_{k_0} = e_0$). Die Vorbedingungen aller Aktionen A1-A13 sind entweder nicht erfüllt oder, im Fall A3, es sind nur Alternativen bei der Ausführung möglich, die keine Veränderung der Variablenbelegungen bewirken. V_{b_1} ist somit auf den Empfang von Nachrichten von $V_{\underline{b}_2}$ angewiesen, bevor neue Variablenbelegungen hervorgerufen werden können.
Die erste Nachricht, die V_{b_1} nach t^1 erhält, ist $reqmark(\underline{b}_2)$, abgeschickt von $V_{\underline{b}_2}$ nach t^2. V_{b_1} setzt $\_mark(v) := true$ für alle $v \in \underline{b}_2$ und schickt $ackmark$ zurück (A8). Die nächste Nachricht, die V_{b_1} erhält, ist $done(e_{k_0} \to e_{k_1})$, zeitlich nach t^3.

$\Rightarrow \forall t \in]t^0, t^3[: zc(z^{\Pi,t}\langle V_I System(IS)\rangle) = \{p_{k_1}, e_{k_0}\}$

$zc(z^{\Pi,t^3}\langle V_I System(IS)\rangle) = \{p_{k_1}, e_{k_1}\}$

$V_{\underline{b}_2}$ befindet sich zum Zeitpunkt t^3 in Aktion A5.

V_{b_1} setzt in A10 $\_in(e_{k_0}) := false$ und $\_in(e_{k_1}) := true$. Die lokalen Variablen von V_{b_1} werden wie folgt automatisch angepasst, für $j = 1, \ldots, m$:

Fall 2.1). $(e_{k_1}, p_{k_1}) \in E$

Analog Fall 1.1.1) mit t^{i-2} statt t^i für alle $i \in I\!N$.

Fall 2.2). $(e_{k_1}, p_{k_1}) \notin E$

Analog Fall 1.1.2) mit t^{i-2} statt t^i für alle $i \in I\!N$.

Da die Ausführung Π beliebig gewählt wurde und die Fallunterscheidungen alle möglichen Ausführungsalternativen abdecken, folgt mit $t_0 := t^0$ und $t_i :=$„Zeitpunkt der i-ten Phasentransition in $V_I System(IS)$“ in $(*_1)$ aus den explizit aufgeführten Folgerungen in den einzelnen Fällen:

$\mathcal{ECT}[\![IS]\!] \subseteq$
$\{\{p_{k_0}, e_{k'_0}\}\delta_1\{p_{k_1}, e_{k'_1}\}\delta_2\{p_{k_2}, e_{k'_2}\}\delta_3 \ldots \infty \mid \forall i \in I\!N : (p_{k_i} \in Suc(p_{k_{i-1}}) \wedge \delta_i = \{b_1\} \wedge (e_{k_{i-1}}, p_{k_{i-1}}) \notin E \wedge e_{k'_{i-1}} = e_{k'_i}) \vee (p_{k_i} \in Suc(p_{k_{i-1}}) \wedge \delta_i = \emptyset \wedge (e_{k_{i-1}}, p_{k_{i-1}}) \in E \wedge e_{k'_{i-1}} = e_{k'_i}) \vee (p_{k_{i-1}} = p_{k_i} \wedge e_{k'_{i-1}} \neq e_{k'_i} \wedge \delta_i = \emptyset)$, mit $k_0, k_1, k_1, \ldots \in \{1, \ldots, m\}$, $k'_0, k'_1, k'_2, \ldots \in \{0, 1, \ldots, m\}\} \cup$
$\{\{p_{k_0}, e_{k'_0}\}\delta_1\{p_{k_1}, e_{k'_1}\}\delta_2\{p_{k_2}, e_{k'_2}\}\delta_3 \ldots \delta_n\{p_{k_n}, e_{k'_n}\} \mid (\forall i \in \{1, \ldots, n\} : (p_{k_i} \in Suc(p_{k_{i-1}}) \wedge \delta_i = \{b_1\} \wedge (e_{k_{i-1}}, p_{k_{i-1}}) \notin E \wedge e_{k'_{i-1}} = e_{k'_i}) \vee (p_{k_i} \in Suc(p_{k_{i-1}}) \wedge \delta_i = \emptyset \wedge (e_{k_{i-1}}, p_{k_{i-1}}) \in E \wedge e_{k'_{i-1}} = e_{k'_i}) \vee (p_{k_{i-1}} = p_{k_i} \wedge e_{k'_{i-1}} \neq e_{k'_i} \wedge \delta_i = \emptyset)) \wedge ((e_{k_n}, p_{k_n}) \notin E)$, mit $k_0, k_1, k_2, \ldots \in \{1, \ldots, m\}$, $k'_0, k'_1, k'_2, \ldots \in \{0, 1, \ldots, m\}$, $n \in I\!N\}$ $(*_3)$

$\Rightarrow$ {Betrachtung der Sicht von $\mathcal{ECT}[\![IS]\!]$ auf $\{b_1\}$}

$\mathcal{ECT}[\![IS]\!]\downharpoonleft_{\{b_1\}} \subseteq$
$\{\{p_{k_0}\}\delta_1\{p_{k_1}\}\delta_2\{p_{k_2}\}\delta_3\ldots\infty \mid \forall i \in I\!N : (p_{k_i} \in Suc(p_{k_{i-1}}) \wedge \delta_i = \{b_1\} \wedge (e_{k_{i-1}}, p_{k_{i-1}}) \notin E) \vee (p_{k_i} \in Suc(p_{k_{i-1}}) \wedge \delta_i = \emptyset \wedge (e_{k_{i-1}}, p_{k_{i-1}}) \in E)$, mit $k_0, k_1, k_2, \ldots \in \{1, \ldots, m\}\} \cup$
$\{\{p_{k_0}\}\delta_1\{p_{k_1}\}\delta_2\{p_{k_2}\}\delta_3\ldots\delta_n\{p_{k_n}\} \mid (\forall i \in \{1, \ldots, n\} : (p_{k_i} \in Suc(p_{k_{i-1}}) \wedge \delta_i = \{b_1\} \wedge (e_{k_{i-1}}, p_{k_{i-1}}) \notin E) \vee (p_{k_i} \in Suc(p_{k_{i-1}}) \wedge \delta_i = \emptyset \wedge (e_{k_{i-1}}, p_{k_{i-1}}) \in E)) \wedge ((e_{k_n}, p_{k_n}) \notin E)$, mit $k_0, k_1, k_2, \ldots \in \{1, \ldots, m\}$, $n \in I\!N\}$ $(*_4)$

Es gelten folgende Implikationen:

$(e_{k_{i-1}}, p_{k_{i-1}}) \notin E \wedge p_{k_i} \in Suc(p_{k_{i-1}})$
$\Rightarrow$ {Satzvoraussetzung iii)}
$\rightarrow_2(p_{k_{i-1}}) = \emptyset \wedge p_{k_i} \in Suc(p_{k_{i-1}})$
$\Rightarrow$ {Definition von $Suc(\cdot)$}
$\rightarrow_2(p_{k_{i-1}}) = \emptyset \wedge (\rightarrow_2(p_{k_{i-1}}) \neq \emptyset \vee \rightarrow_1(p_{k_{i-1}}) \neq \emptyset)$
$\Rightarrow$ {Zusammenfassen}
$\rightarrow_1(p_{k_{i-1}}) \neq \emptyset$.

$(e_{k_{i-1}}, p_{k_{i-1}}) \in E \wedge p_{k_i} \in Suc(p_{k_{i-1}})$
$\Rightarrow$ {Satzvoraussetzung iii)}
$\rightarrow_2(p_{k_{i-1}}) \neq \emptyset$.

$(e_{k_n}, p_{k_n}) \notin E$
$\Rightarrow$ {Satzvoraussetzung iii)}
$\rightarrow_2(p_{k_n}) = \emptyset$.

Wendet man die Implikationen auf die Mengenvereinigung $(*_4)$ an, ergibt sich:

$\mathcal{ECT}[\![IS]\!]\downharpoonleft_{\{b_1\}} \subseteq$
$\{\{p_{k_0}\}\delta_1\{p_{k_1}\}\delta_2\{p_{k_2}\}\delta_3\ldots\infty \mid \forall i \in I\!N : (p_{k_i} \in Suc(p_{k_{i-1}}) \wedge \delta_i = \{b_1\} \wedge \rightarrow_1(p_{k_{i-1}}) \neq \emptyset) \vee (p_{k_i} \in Suc(p_{k_{i-1}}) \wedge \delta_i = \emptyset \wedge \rightarrow_2(p_{k_{i-1}}) \neq \emptyset)$, mit $k_0, k_1, k_2, \ldots \in \{1, \ldots, m\}\} \cup$
$\{\{p_{k_0}\}\delta_1\{p_{k_1}\}\delta_2\{p_{k_2}\}\delta_3\ldots\delta_n\{p_{k_n}\} \mid (\forall i \in \{1, \ldots, n\} : (p_{k_i} \in Suc(p_{k_{i-1}}) \wedge \delta_i = \{b_1\} \wedge \rightarrow_1(p_{k_{i-1}}) \neq \emptyset) \vee (p_{k_i} \in Suc(p_{k_{i-1}}) \wedge \delta_i = \emptyset \wedge \rightarrow_2(p_{k_{i-1}}) \neq \emptyset)) \wedge (\rightarrow_2(p_{k_n}) = \emptyset)$, mit $k_0, k_1, k_2, \ldots \in \{1, \ldots, m\}$, $n \in I\!N\}$ $(*_5)$

Es gilt die Satzvoraussetzung $\{p \in b_1 \mid \rightarrow_1(p) \neq \emptyset \wedge \rightarrow_2(p) \neq \emptyset\} = \emptyset$, und wegen $p_{k_i} \in Suc(p_{k_{i-1}})$ gilt $(p_{k_{i-1}}, p_{k_i}) \in \rightarrow_1 \cup \rightarrow_2$. Unter diesen beiden Voraussetzungen folgt $(p_{k_{i-1}}, p_{k_i}) \in \rightarrow_1$ aus $\rightarrow_1(p_{k_{i-1}}) \neq \emptyset$, und es folgt $(p_{k_{i-1}}, p_{k_i}) \in \rightarrow_2$ aus $\rightarrow_2(p_{k_{i-1}}) \neq \emptyset$. Wendet man die beiden Folgerungen auf die Mengenvereinigung $(*_5)$ an, ergibt sich:

$\mathcal{ECT}[\![IS]\!]\downharpoonleft_{\{b_1\}} \subseteq$
$\{\{p_{k_0}\}\delta_1\{p_{k_1}\}\delta_2\{p_{k_2}\}\delta_3\ldots\infty \mid \forall i \in I\!N : (p_{k_i} \in Suc(p_{k_{i-1}}) \wedge \delta_i = \{b_1\} \wedge (p_{k_{i-1}}, p_{k_i}) \in \rightarrow_1) \vee (p_{k_i} \in Suc(p_{k_{i-1}}) \wedge \delta_i = \emptyset \wedge (p_{k_{i-1}}, p_{k_i}) \in \rightarrow_2)$, mit $k_0, k_1, k_2, \ldots \in \{1, \ldots, m\}\} \cup$
$\{\{p_{k_0}\}\delta_1\{p_{k_1}\}\delta_2\{p_{k_2}\}\delta_3\ldots\delta_n\{p_{k_n}\} \mid (\forall i \in \{1, \ldots, n\} : (p_{k_i} \in Suc(p_{k_{i-1}}) \wedge \delta_i = \{b_1\} \wedge (p_{k_{i-1}}, p_{k_i}) \in \rightarrow_1) \vee (p_{k_i} \in Suc(p_{k_{i-1}}) \wedge \delta_i = \emptyset \wedge (p_{k_{i-1}}, p_{k_i}) \in \rightarrow_2)) \wedge (\rightarrow_2(p_{k_n}) = \emptyset)$, mit $k_0, k_1, k_2, \ldots \in \{1, \ldots, m\}$, $n \in I\!N\}$

$\Rightarrow$ {Zusammenhängende Kanten ergeben Pfade im Graphen}

$\mathcal{ECT}[\![IS]\!]\downharpoonleft_{\{b_1\}} \subseteq$
$\{\{p_{k_0}\}\delta_1\{p_{k_1}\}\delta_2\{p_{k_2}\}\delta_3\ldots\infty \mid \langle p_{k_0}, p_{k_1}, p_{k_2}, \ldots\infty\rangle$ ist unendlicher Pfad in $G(b_1)$ und $\forall i \in I\!N : (\delta_i = \{b_1\} \wedge (p_{k_{i-1}}, p_{k_i}) \in \rightarrow_1) \vee (\delta_i = \emptyset \wedge (p_{k_{i-1}}, p_{k_i}) \in \rightarrow_2)$, mit $k_0, k_1, k_2, \ldots \in \{1, \ldots, m\}\} \cup$
$\{\{p_{k_0}\}\delta_1\{p_{k_1}\}\delta_2\{p_{k_2}\}\delta_3\ldots\delta_n\{p_{k_n}\} \mid \langle p_{k_0}, p_{k_1}, p_{k_2}, \ldots p_{k_n}\rangle$ ist endlicher Pfad in $G(b_1)$ und $(\forall i \in \{1, \ldots, n\} : (\delta_i = \{b_1\} \wedge (p_{k_{i-1}}, p_{k_i}) \in \rightarrow_1) \vee (\delta_i = \emptyset \wedge (p_{k_{i-1}}, p_{k_i}) \in \rightarrow_2)) \wedge (\rightarrow_2(p_{k_n}) = \emptyset)$, mit $k_0, k_1, k_2, \ldots \in \{1, \ldots, m\}$, $n \in I\!N\}$

$\Rightarrow$ {Umindizierung $p_{k_x} \rightarrow p'_x$; Knoten von $G(b_1)$ sind definitionsbedingt Elemente aus b_1}

$\mathcal{ECT}[\![IS]\!]\downharpoonleft_{\{b_1\}} \subseteq$
$\{\{p'_0\}\delta_1\{p'_1\}\delta_2\{p'_2\}\delta_3\ldots\infty \mid \langle p'_0, p'_1, p'_2, \ldots\infty\rangle$ ist unendlicher Pfad in $G(b_1)$ und $\forall i \in I\!N : (\delta_i = \{b_1\} \wedge (p'_{i-1}, p'_i) \in \rightarrow_1) \vee (\delta_i = \emptyset \wedge (p'_{i-1}, p'_i) \in \rightarrow_2)\} \cup$

$\{\{p'_0\}\delta_1\{p'_1\}\delta_2\{p'_2\}\delta_3\ldots\delta_n\{p'_n\} \mid \langle p'_0, p'_1, p'_2, \ldots p'_n\rangle$ ist endlicher Pfad in $G(b_1)$ und $(\forall i \in \{1,\ldots,n\} : (\delta_i = \{b_1\} \wedge (p'_{i-1}, p'_i) \in \rightarrow_1) \vee (\delta_i = \emptyset \wedge (p'_{i-1}, p'_i) \in \rightarrow_2)) \wedge (\rightarrow_2(p'_n) = \emptyset)$, mit $n \in I\!N\}$

Somit gilt die $\subseteq$-Richtung von Teil a).

Zu a), $\supseteq$.

Das Ziel ist es, zu einem beliebigen Pfad im Graphen $G(b_1)$ eine passende Ausführung von $V_I System(IS)$ zu konstruieren, bei der die auftretenden Cases mit Sicht auf b_1 mit den Knoten des Pfades in der Abfolge übereinstimmen. Zusätzlich muss die Ausführung der Aktionen A4 und A5 bei $V_I System(IS)$ mit den Kantentypen bei $G(b_1)$ vereinbar sein.

Sei also $ectr = \{p'_0\}\delta_1\{p'_1\}\delta_2\{p'_2\}\ldots$ beliebig aber fest mit: $\langle p'_0, p'_1, p'_2, \ldots\rangle$ ist endlicher oder unendlicher Pfad in $G(b_1)$ und $\forall i = 1, 2, \ldots : (\delta_i = \{b_1\} \wedge (p'_{i-1}, p'_i) \in \rightarrow_1) \vee (\delta_i = \emptyset \wedge (p'_{i-1}, p'_i) \in \rightarrow_2)\}$.

Setze $l := \lfloor |ectr|/2 \rfloor$ falls $|ectr| \in I\!N$, also die Länge von $ectr$, endlich ist. Setze $l := \infty$ falls $|ectr| \notin I\!N$.

Falls $l \in I\!N$, dann gelte zusätzlich $\rightarrow_2(p'_l) = \emptyset$.

Zu zeigen: Es existiert eine Ausführung $\Pi_{\overline{ectr}}$ von $V_I System(IS)$, die gemäß $(*_1)$ eine Trace $\overline{ectr} = \bar{c}_0\bar{\delta}_1\bar{c}_1\bar{\delta}_2, \ldots$ erzeugt, für die $\overline{ectr}\downarrow_{\{b_1\}} = ectr$ gilt.

Die Konstruktion der Ausführung $\Pi_{\overline{ectr}}$ wird angegeben unter Bezugnahme auf die allgemeine Ausführung Π und den damit verbundenen Fallunterscheidungen aus dem $\subseteq$-Teil des Beweises. Die genaue Beschreibung der Aktivität der beiden Komponenten findet sich im $\subseteq$-Teil. Im folgenden sind alle relevanten Fälle aufgeführt mit den Folgerungen bezüglich der auftretenden Cases und den daraus resultierenden Zuweisungen zu den Elementen der Trace $\overline{ectr}$.

Setze diesbezüglich $c_0 := \{p_{k_0}, e_{k_0}\}$ mit $p_{k_0} = p'_0$, $k_0 \in \{1,\ldots,m\}$. Laut $(*_2)$ gilt $c_0 \in Case(IS)$.

Startzeitpunkt t^0: Als Startbelegung setze $\_z(x) = 1$ bei $V_{b(x)}$ für beide $x \in c_0$. Alle booleschen Variablen der beiden Komponenten von $V_I System(IS)$, d.h. von V_{b_1} und $V_{\underline{b}_2}$, seien $false$. Damit sind die Vorbedingungen erfüllt, um bei beiden Komponenten die Aktion A1 auszuführen.

Fall 1).

$\Rightarrow zc(z^{\Pi_{\overline{ectr}}, t^0}\langle V_I System(IS)\rangle) = \{p_{k_0}, e_{k_0}\}$.

Setze $\bar{c}_0 := \{p'_0, e_{k_0}\}$ mit $p'_0 = p_{k_0}$.

Fall 1.1). $(e_{k_0}, p_{k_0}) \in E$

$\Rightarrow \forall t \in]t^0, t^3[: zc(z^{\Pi_{\overline{ectr}}, t}\langle V_I System(IS)\rangle) = \{p_{k_0}, e_{k_0}\}$,

$zc(z^{\Pi_{\overline{ectr}}, t^3}\langle V_I System(IS)\rangle) = \{p_{k_1}, e_{k_0}\}$, $p_{k_1} \in Suc(p_{k_0})$,

V_{b_1} befindet sich zum Zeitpunkt t^3 in Aktion A5,

$\forall t \in]t^3, t^5[: zc(z^{\Pi_{\overline{ectr}}, t}\langle V_I System(IS)\rangle) = \{p_{k_1}, e_{k_0}\}$,

$zc(z^{\Pi_{\overline{ectr}}, t^5}\langle V_I System(IS)\rangle) = \{p_{k_1}, e_{k_1}\}$

$V_{\underline{b}_2}$ befindet sich zum Zeitpunkt t^5 in Aktion A5.

Setze $\bar{\delta}_1 := \emptyset$, $\bar{c}_1 := \{p'_1, e_{k_0}\}$, $\bar{\delta}_2 := \emptyset$, $\bar{c}_2 := \{p'_1, e_{k_1}\}$ mit $p'_1 = p_{k_1}$.

Fall 1.1.1). $(e_{k_1}, p_{k_1}) \in E$

$\Rightarrow \forall t \in]t^5, t^7[: zc(z^{\Pi_{\overline{ectr}}, t}\langle V_I System(IS)\rangle) = \{p_{k_1}, e_{k_1}\}$

$zc(z^{\Pi_{\overline{ectr}}, t^7}\langle V_I System(IS)\rangle) = \{p_{k_2}, e_{k_1}\}$, $p_{k_2} \in Suc(p_{k_1})$

V_{b_1} befindet sich zum Zeitpunkt t^7 in Aktion A5.

$\forall t \in]t^7, t^9[: zc(z^{\Pi_{\overline{ectr}},t}\langle V_I System(IS)\rangle) = \{p_{k_2}, e_{k_1}\}$

$zc(z^{\Pi_{\overline{ectr}},t^9}\langle V_I System(IS)\rangle) = \{p_{k_2}, e_{k_2}\}$

$V_{\underline{b}_2}$ befindet sich zum Zeitpunkt t^9 in Aktion A5.

Setze $\bar{\delta}_3 := \emptyset$, $\bar{c}_3 := \{p'_2, e_{k_1}\}$, $\bar{\delta}_4 := \emptyset$, $\bar{c}_4 := \{p'_2, e_{k_2}\}$ mit $p'_2 = p_{k_2}$.

Fall 1.1.1.1). $(e_{k_2}, p_{k_2}) \in E$

Analog Fall 1.1.1) mit k_{i+1} statt k_i, t^{i+4} statt t^i, δ_{i+2} statt δ_i, $\bar{c}_{i+2}$ statt $\bar{c}_i$ und p'_{i+1} statt p'_i für alle $i \in I\!N$.

Fall 1.1.1.2). $(e_{k_2}, p_{k_2}) \notin E$

Analog Fall 1.1.2) mit k_{i+1} statt k_i, t^{i+4} statt t^i, δ_{i+2} statt δ_i, $\bar{c}_{i+2}$ statt $\bar{c}_i$ und p'_{i+1} statt p'_i für alle $i \in I\!N$.

Fall 1.1.2). $(e_{k_1}, p_{k_1}) \notin E$

Fall 1.1.2.1). $\exists t, t > t^6$, $\exists q \in b_1 \setminus \{p_{k_1}\} : z^{\Pi_{\overline{ectr}},t}\langle V_{b_1}\rangle(p_{k_1}) = q$

Bei diesem Fall gelte $q = p'_2$. Bei V_{b_1} findet demnach die Zuweisung $\_z(p_{k_1}) := p'_2$ statt. (Beachte: $p'_2 \in b_1 \setminus \{p_{k_1}\}$.)

Fall 1.1.2.1.1). $q \in Suc(p_{k_1})$

Dieser Unterfall tritt nun immer ein, da $q = p'_2$, $p_{k_1} = p'_1$ und $(p'_1, p'_2) \in \rightarrow_1 \vee (p'_1, p'_2) \in \rightarrow_2$ gilt.

$\Rightarrow \forall t \in]t^5, t^8[: zc(z^{\Pi_{\overline{ectr}},t}\langle V_I System(IS)\rangle) = \{p_{k_1}, e_{k_1}\}$

$zc(z^{\Pi_{\overline{ectr}},t^8}\langle V_I System(IS)\rangle) = \{p_{k_2}, e_{k_1}\}$, $p_{k_2} \in Suc(p_{k_1})$

V_{b_1} befindet sich zum Zeitpunkt t^8 in Aktion A4.

$\forall t \in]t^8, t^{10}[: zc(z^{\Pi_{\overline{ectr}},t}\langle V_I System(IS)\rangle) = \{p_{k_2}, e_{k_1}\}$

$zc(z^{\Pi_{\overline{ectr}},t^{10}}\langle V_I System(IS)\rangle) = \{p_{k_2}, e_{k_2}\}$

$V_{\underline{b}_2}$ befindet sich zum Zeitpunkt t^{10} in Aktion A5.

Setze $\bar{\delta}_3 := \{b_1\}$, $\bar{c}_3 := \{p'_2, e_{k_1}\}$, $\bar{\delta}_4 := \emptyset$, $\bar{c}_4 := \{p'_2, e_{k_2}\}$ mit $p'_2 = p_{k_2}$.

Fall 1.1.2.1.1.1). $(e_{k_2}, p_{k_2}) \in E$

Analog Fall 1.1.1) mit k_{i+1} statt k_i, t^{i+5} statt t^i, δ_{i+2} statt δ_i, $\bar{c}_{i+2}$ statt $\bar{c}_i$ und p'_{i+1} statt p'_i für alle $i \in I\!N$.

Fall 1.1.2.1.1.2). $(e_{k_2}, p_{k_2}) \notin E$

Analog Fall 1.1.2) mit k_{i+1} statt k_i, t^{i+5} statt t^i, δ_{i+2} statt δ_i, $\bar{c}_{i+2}$ statt $\bar{c}_i$ und p'_{i+1} statt p'_i für alle $i \in I\!N$.

Fall 1.1.2.2). $\nexists t, t > t^6$, $\nexists q \in b_1 \setminus \{p_{k_1}\} : z^{\Pi_{\overline{ectr}},t}\langle V_{b_1}\rangle(p_{k_1}) = q$

$\Rightarrow \forall t \in]t^5, \infty[: zc(z^{\Pi_{\overline{ectr}},t}\langle V_I System(IS)\rangle) = \{p_{k_1}, e_{k_1}\}$

Es treten keine weiteren Fälle und somit keine weiteren Phasentransitionen mehr auf.

Fall 1.2). $(e_{k_0}, p_{k_0}) \notin E$

$\Rightarrow \forall t \in]t^0, t^2] : zc(z^{\Pi_{\overline{ectr}},t}\langle V_I System(IS)\rangle) = \{p_{k_0}, e_{k_0}\}$

Fall 1.2.1). $\exists t, t > t^3$, $\exists q \in b_1 \setminus \{p_{k_0}\} : z^{\Pi_{\overline{ectr}},t}\langle V_{b_1}\rangle(p_{k_0}) = q$

Analog Fall 1.1.2.1) mit k_{i-1} statt k_i, t^{i-3} statt t^i, δ_{i-2} statt δ_i, $\bar{c}_{i-2}$ statt $\bar{c}_i$ und p'_{i-1} statt p'_i für alle $i \in I\!N$.

Fall 1.2.2). $\nexists t, t > t^3$, $\nexists q \in b_1 \setminus \{p_{k_0}\} : z^{\Pi_{\overline{ectr}}, t}\langle V_{b_1}\rangle(p_{k_0}) = q$
Analog Fall 1.1.2.2) mit k_{i-1} statt k_i und t^{i-3} statt t^i für alle $i \in I\!N$.

Alle nicht aufgeführten Fälle brauchen bei der Ausführung $\Pi_{\overline{ectr}}$ nicht betrachtet zu werden.

Die oben angegebene Ausführung $\Pi_{\overline{ectr}}$ von $V_I System(IS)$ erfüllt mit $t_0 := t^0$, $t_i :=$„Zeitpunkt der i-ten Phasentransition in $V_I System(IS)$" die unter $(*_1)$ geforderten Kriterien. Im Fall eines endlich langen Pfades ($l \in I\!N$) bildet $t_0, t_1, t_2, \ldots, t_l$ die max. Folge von Globalzeitpunkten und die Ausführung endet mit Fall 1.2.2, 1.1.2.2 oder einem Analogon davon. Im Fall eines unendlich langen Pfades ($l = \infty$) bildet $t_0, t_1, t_2, \ldots, \infty$ die max. Folge von Globalzeitpunkten. Fall 1.2.2 oder 1.1.2.2 tritt dann in keiner Parametrisierung auf.

Folglich gilt: $\bar{c}_0\bar{\delta}_1\bar{c}_1\bar{\delta}_2, \bar{c}_2 \ldots = \overline{ectr} \in \mathcal{ECT}[\![IS]\!]$. Die Fallunterscheidungen garantieren dabei (wegen $p_{k_x} = p'_x$):

$$\forall i = 1, 2, \ldots, l : (p'_i \in Suc(p'_{i-1}) \wedge (\overbrace{(\bar{\delta}_i = \{b_1\} \wedge (e_{k_{i-1}}, p'_{i-1}) \notin E)}^{\text{Fall 1.1.2.1.1}} \vee \overbrace{(\bar{\delta}_i = \emptyset \wedge (e_{k_{i-1}}, p'_{i-1}) \in E)}^{\text{Fälle 1.1, 1.1.1}})) \wedge \underbrace{(l \in I\!N \Rightarrow (e_{k_l}, p'_l) \notin E)}_{\text{Fälle 1.1.2, 1.1.2.2}}. \qquad (*_6)$$

Es gelten folgende Implikationen (wegen $p'_x = p_{k_x}$):

$(e_{k_{i-1}}, p'_{i-1}) \notin E \wedge p'_i \in Suc(p'_{i-1})$
$\Rightarrow$ {Satzvoraussetzung iii)}
$\rightarrow_2(p'_{i-1}) = \emptyset \wedge p'_i \in Suc(p'_{i-1})$
$\Rightarrow$ {Definition von $Suc(\cdot)$}
$\rightarrow_2(p'_{i-1}) = \emptyset \wedge (\rightarrow_2(p'_{i-1}) \neq \emptyset \vee \rightarrow_1(p'_{i-1}) \neq \emptyset)$
$\Rightarrow$ {Zusammenfassen}
$\rightarrow_1(p'_{i-1}) \neq \emptyset$.

$(e_{k_{i-1}}, p'_{i-1}) \in E \wedge p'_i \in Suc(p'_{i-1})$
$\Rightarrow$ {Satzvoraussetzung iii)}
$\rightarrow_2(p'_{i-1}) \neq \emptyset$.

$(e_{k_l}, p'_l) \notin E$
$\Rightarrow$ {Satzvoraussetzung iii)}
$\rightarrow_2(p'_l) = \emptyset$.

Wendet man die Implikationen auf $(*_6)$ an, ergibt sich:
$\forall i = 1, 2, \ldots, l : (p'_i \in Suc(p'_{i-1}) \wedge ((\bar{\delta}_i = \{b_1\} \wedge \rightarrow_1(p'_{i-1}) \neq \emptyset) \vee (\bar{\delta}_i = \emptyset \wedge \rightarrow_2(p'_{i-1}) \neq \emptyset))) \wedge (l \in I\!N \Rightarrow \rightarrow_2(p'_l) = \emptyset)$. $(*_7)$

Es gilt die Satzvoraussetzung $\{p \in b_1 \mid \rightarrow_1(p) \neq \emptyset \wedge \rightarrow_2(p) \neq \emptyset\} = \emptyset$, und wegen $p'_i \in Suc(p'_{i-1})$ gilt $(p'_{i-1}, p'_i) \in \rightarrow_1 \cup \rightarrow_2$. Unter diesen beiden Voraussetzungen folgt $(p'_{i-1}, p'_i) \in \rightarrow_1$ aus $\rightarrow_1(p'_{i-1}) \neq \emptyset$, und es folgt $(p'_{i-1}, p'_i) \in \rightarrow_2$ aus $\rightarrow_2(p'_{i-1}) \neq \emptyset$. Wendet man die beiden Folgerungen auf $(*_7)$ an, ergibt sich:
$\forall i = 1, 2, \ldots, l : (p'_i \in Suc(p'_{i-1}) \wedge ((\bar{\delta}_i = \{b_1\} \wedge (p'_{i-1}, p'_i) \in \rightarrow_1) \vee (\bar{\delta}_i = \emptyset \wedge (p'_{i-1}, p'_i) \in \rightarrow_2))) \wedge (l \in I\!N \Rightarrow \rightarrow_2(p'_l) = \emptyset)$.

$\Rightarrow$ {Zusammenhängende Kanten ergeben Pfade im Graphen}
$\langle p'_0, p'_1, p'_2, \ldots \rangle$ ist endlicher oder unendlicher Pfad in $G(b_1)$ und $\forall i = 1, 2, \ldots, l : ((\delta_i = \{b_1\} \wedge (p'_{i-1}, p'_i) \in \rightarrow_1) \vee (\delta_i = \emptyset \wedge (p'_{i-1}, p'_i) \in \rightarrow_2)) \wedge (l \in I\!N \Rightarrow \rightarrow_2(p'_l) = \emptyset)$. $(*_8)$

Nach Definition 9.11.a gilt $\overline{ectr}\lfloor_{\{b_1\}} = \bar{c}_0\lfloor_{\{b_1\}} .(\bar{\delta}_{i_1} \cap \{b_1\}).\bar{c}_{i_1}\lfloor_{\{b_1\}} .(\bar{\delta}_{i_2} \cap \{b_1\}).\bar{c}_{i_2}\lfloor_{\{b_1\}} \ldots$ mit $i_1, i_2, \ldots \in I\!N$, $i_1 < i_2 < \ldots$ und $(j \in \{i_1, i_2, \ldots\}$ gdw. $\bar{c}_j\lfloor_{\{b_1\}} \neq \bar{c}_{j-1}\lfloor_{\{b_1\}})$. $(*_9)$

$\Rightarrow$ { $G(b_1)$ ist schlingenfrei $(p'_x \notin Suc(p'_x))$; $(*_8) \wedge (*_9)$}
$\overline{ectr}\lfloor_{\{b_1\}} = \{p'_0\}\delta_1\{p'_1\}\delta_2\{p'_2\} \ldots$ mit: $\langle p'_0, p'_1, p'_2, \ldots \rangle$ ist endlicher oder unendlicher Pfad in $G(b_1)$ und $\forall i = 1, 2, \ldots, l : ((\delta_i = \{b_1\} \wedge (p'_{i-1}, p'_i) \in \rightarrow_1) \vee (\delta_i = \emptyset \wedge (p'_{i-1}, p'_i) \in \rightarrow_2)) \wedge (l \in I\!N \Rightarrow \rightarrow_2(p'_l) = \emptyset)\}$

$\Rightarrow$ {Voraussetzungen von $ectr$}
$\overline{ectr}\lfloor_{\{b_1\}} = ectr$

Die Existenz der Ausführung $\Pi_{\overline{ectr}}$ mit den geforderten Eigenschaften ist somit konstruktiv gezeigt, und folglich gilt die $\supseteq$-Richtung von Teil a).

Zusammenfassend gelten die $\subseteq$- und die $\supseteq$-Richtung und damit Satz 9.20.a).

Zu b).

Das Ziel ist es, sich Teilergebnisse im Beweis der Satzaussage a) zunutze zu machen, um nicht noch einmal alle Ausführungen von $V_I System(IS)$ untersuchen zu müssen. Damit dies möglich ist, muss die Verbindung zwischen dem Erweiterten Casegraphen von IS mit Sicht auf b_1 und der Erweiterten Casetrace-Semantik von IS mit Sicht auf b_1 aufgezeigt werden.

Der Erweiterte Casegraph von IS mit Sicht auf $\{b_1\}$ ist definiert (Definition 9.17.c) als:

$ECG(IS)\lfloor_{\{b_1\}} = (C, \rightarrowtail_1, \rightarrowtail_2)$ mit: $(*_{10})$

(1) $C = Case(IS)\lfloor_{\{b_1\}}$

(2) $\rightarrowtail_1 \subseteq C \times C$ mit $(c_1, c_2) \in \rightarrowtail_1$ gdw.
$\exists ectr_1 \in \left(Case(IS)\lfloor_{\{b_1\}} .\mathcal{P}(\{b_1\})\right)^*, \exists ectr_2 \in \left(\mathcal{P}(\{b_1\}).Case(IS)\lfloor_{\{b_1\}}\right)^*, \exists \delta \subseteq \{b_1\}$:
$ectr_1.c_1.\delta.c_2.ctr_2 \in \mathcal{ECT}^{\mathrm{i}}[\![IS]\!]\lfloor_{\{b_1\}}$ und $\delta \neq \emptyset$

(3) $\rightarrowtail_2 \subseteq C \times C$ mit $(c_1, c_2) \in \rightarrowtail_2$ gdw.
$\exists ectr_1 \in \left(Case(IS)\lfloor_{\{b_1\}} .\mathcal{P}(\{b_1\})\right)^*, ectr_2 \in \left(\mathcal{P}(\{b_1\}).Case(IS)\lfloor_{\{b_1\}}\right)^*, \delta \subseteq \{b_1\}$:
$ectr_1.c_1.\delta.c_2.ctr_2 \in \mathcal{ECT}^{\mathrm{i}}[\![IS]\!]\lfloor_{\{b_1\}}$ und $\delta \neq \emptyset$

Gemäß $(*_3)$ gilt $\mathcal{ECT}[\![IS]\!] \subseteq M_1 \cup M_2$ mit

$M_1 = \{\{p_{k_0}, e_{k'_0}\}\delta_1\{p_{k_1}, e_{k'_1}\}\delta_2\{p_{k_2}, e_{k'_2}\}\delta_3 \ldots \infty \mid \forall i \in I\!N : (p_{k_i} \in Suc(p_{k_{i-1}}) \wedge \delta_i = \{b_1\} \wedge (e_{k_{i-1}}, p_{k_{i-1}}) \notin E \wedge e_{k'_{i-1}} = e_{k'_i}) \vee (p_{k_i} \in Suc(p_{k_{i-1}}) \wedge \delta_i = \emptyset \wedge (e_{k_{i-1}}, p_{k_{i-1}}) \in E \wedge e_{k'_{i-1}} = e_{k'_i}) \vee (p_{k_{i-1}} = p_{k_i} \wedge e_{k'_{i-1}} \neq e_{k'_i} \wedge \delta_i = \emptyset)$, mit $k_0, k_1, k_1, \ldots \in \{1, \ldots, m\}$, $k'_0, k'_1, k'_2, \ldots \in \{0, 1, \ldots, m\}\}$,

$M_2 = \{\{p_{k_0}, e_{k'_0}\}\delta_1\{p_{k_1}, e_{k'_1}\}\delta_2\{p_{k_2}, e_{k'_2}\}\delta_3 \ldots \delta_n\{p_{k_n}, e_{k'_n}\} \mid (\forall i \in \{1, \ldots, n\} : (p_{k_i} \in Suc(p_{k_{i-1}}) \wedge \delta_i = \{b_1\} \wedge (e_{k_{i-1}}, p_{k_{i-1}}) \notin E \wedge e_{k'_{i-1}} = e_{k'_i}) \vee (p_{k_i} \in Suc(p_{k_{i-1}}) \wedge \delta_i = \emptyset \wedge (e_{k_{i-1}}, p_{k_{i-1}}) \in E \wedge e_{k'_{i-1}} = e_{k'_i}) \vee (p_{k_{i-1}} = p_{k_i} \wedge e_{k'_{i-1}} \neq e_{k'_i} \wedge \delta_i = \emptyset)) \wedge ((e_{k_n}, p_{k_n}) \notin E)$, mit $k_0, k_1, k_2, \ldots \in \{1, \ldots, m\}$, $k'_0, k'_1, k'_2, \ldots \in \{0, 1, \ldots, m\}$, $n \in I\!N\}$.

Bezieht man sich auf die vorangegangene Spezifizierung von M_1 und M_2, dann gilt für jede Trace in einer dieser beiden Mengen: $p_{k_i} \in Suc(p_{k_{i-1}}) \wedge e_{k'_{i-1}} = e_{k'_i}$ oder aber $p_{k_{i-1}} = p_{k_i} \wedge e_{k'_{i-1}} \neq e_{k'_i}$. Da $G(b_1)$ schlingenfrei ist, folgt daraus (für endliche und unendliche Traces):

$(\{p_{k_0}, e_{k'_0}\}\delta_1\{p_{k_1}, e_{k'_1}\}\delta_2\{p_{k_2}, e_{k'_2}\} \ldots \in M_1 \cup M_2) \Rightarrow (\forall i = 1, 2, \ldots : |\{p_{k_i}, e_{k'_i}\} \setminus \{p_{k_{i-1}}, e_{k'_{i-1}}\}| = 1)$, mit $k_0, k_1, k_2, \ldots \in \{1, \ldots, m\}$ und $k'_0, k'_1, k'_2, \ldots \in \{0, 1, \ldots, m\}$
$\Rightarrow$ {wegen $(*_3)$}
$(\{p_{k_0}, e_{k'_0}\}\delta_1\{p_{k_1}, e_{k'_1}\}\delta_2\{p_{k_2}, e_{k'_2}\} \ldots \in \mathcal{ECT}[\![IS]\!]) \Rightarrow (\forall i = 1, 2, \ldots : |\{p_{k_i}, e_{k'_i}\} \setminus \{p_{k_{i-1}}, e_{k'_{i-1}}\}| = 1)$, mit $k_0, k_1, k_2, \ldots \in \{1, \ldots, m\}$ und $k'_0, k'_1, k'_2, \ldots \in \{0, 1, \ldots, m\}$
$\Rightarrow$ {Definition 5.7}
$\mathcal{ECT}[\![IS]\!] = \mathcal{ECT}^{\mathrm{i}}[\![IS]\!]$.

Aufgrund der Gleichheit von Erweiterter Casetrace-Semantik und Erweiterter Interleaving Casetrace-Semantik und wegen $\delta \subseteq \{b_1\}$, lässt sich $(*_{10})$ wie folgt umformulieren.

$ECG(IS)\lfloor_{\{b_1\}} = (C, \rightarrowtail_1, \rightarrowtail_2)$ mit:

(1) $C = Case(IS)\lfloor_{\{b_1\}}$

(2) $\rightarrowtail_1 \subseteq C \times C$ mit $(c_1, c_2) \in \rightarrowtail_1$ gdw.
$\exists ectr_1 \in \left(Case(IS)\lfloor_{\{b_1\}} .\{\emptyset, \{b_1\}\}\right)^*, \exists ectr_2 \in \left(\{\emptyset, \{b_1\}\}.Case(IS)\lfloor_{\{b_1\}}\right)^*, \exists \delta \in \{\emptyset, \{b_1\}\}$: $ectr_1.c_1.\delta.c_2.ctr_2 \in \mathcal{ECT}[\![IS]\!]\lfloor_{\{b_1\}}$ und $\delta = \{b_1\}$

(3) $\rightarrowtail_2 \subseteq C \times C$ mit $(c_1, c_2) \in \rightarrowtail_2$ gdw.
$\exists ectr_1 \in \left(Case(IS)\lfloor_{\{b_1\}} .\{\emptyset, \{b_1\}\}\right)^*, \exists ectr_2 \in \left(\{\emptyset, \{b_1\}\}.Case(IS)\lfloor_{\{b_1\}}\right)^*, \exists \delta \in \{\emptyset, \{b_1\}\}$: $ectr_1.c_1.\delta.c_2.ctr_2 \in \mathcal{ECT}[\![IS]\!]\lfloor_{\{b_1\}}$ und $\delta = \emptyset$

Die Spezifizierung von $Case(IS)$ erfolgte in $(*_2)$. Teil a) des Satzes 9.20 beschreibt die Struktur von $\mathcal{ECT}[\![IS]\!]\lfloor_{\{b_1\}}$. Die ausschließliche Existenzforderung von $ectr_2$, ohne Festlegung einer Länge (auch die leere Trace ist zugelassen), ermöglicht die gemeinsame Betrachtung von endlichen und unendlichen Traces in (2) und (3). Er ergibt sich:

$ECG(IS)\lfloor_{\{b_1\}} = (C, \rightarrowtail_1, \rightarrowtail_2)$ mit:

(1) $C = (\{\{e_j, p_j\} \mid j \in \{1, \ldots, m\}\} \cup \{\{e_j, p_i\} \mid p_j \in Pre(p_i), i, j \in \{1, \ldots, m\}\} \cup \{\{e_0, p_i\} \mid i \in \{1, \ldots, m\}\})\lfloor_{\{b_1\}} = \{\{p_j\} \mid j \in \{1, \ldots, m\}\} = \{\{p\} \mid p \in b_1\}$

(2) $\rightarrowtail_1 \subseteq C \times C$ mit $(c_1, c_2) \in \rightarrowtail_1$ gdw.
$\exists ectr_1 \in \left(Case(IS)\lfloor_{\{b_1\}} .\{\emptyset, \{b_1\}\}\right)^*, \exists ectr_2 \in \left(\{\emptyset, \{b_1\}\}.Case(IS)\lfloor_{\{b_1\}}\right)^*, \exists \delta \in \{\emptyset, \{b_1\}\}$: $ectr_1.c_1.\delta.c_2.ctr_2 = \{p'_0\}\delta_1\{p'_1\}\delta_2\{p'_2\}\ldots$ und $\langle p'_0, p'_1, p'_2, \ldots\rangle$ ist endlicher oder unendlicher Pfad in $G(b_1)$ und $\forall i \in I\!N : (\delta_i = \{b_1\} \wedge (p'_{i-1}, p'_i) \in \rightarrow_1) \vee (\delta_i = \emptyset \wedge (p'_{i-1}, p'_i) \in \rightarrow_2)$ und $\delta = \{b_1\}$

(3) $\rightarrowtail_2 \subseteq C \times C$ mit $(c_1, c_2) \in \rightarrowtail_2$ gdw.
$\exists ectr_1 \in \left(Case(IS)\lfloor_{\{b_1\}} .\{\emptyset, \{b_1\}\}\right)^*, \exists ectr_2 \in \left(\{\emptyset, \{b_1\}\}.Case(IS)\lfloor_{\{b_1\}}\right)^*, \exists \delta \in \{\emptyset, \{b_1\}\}$: $ectr_1.c_1.\delta.c_2.ctr_2 = \{p'_0\}\delta_1\{p'_1\}\delta_2\{p'_2\}\ldots$ und $\langle p'_0, p'_1, p'_2, \ldots\rangle$ ist endlicher oder unendlicher Pfad in $G(b_1)$ und $\forall i \in I\!N : (\delta_i = \{b_1\} \wedge (p'_{i-1}, p'_i) \in \rightarrow_1) \vee (\delta_i = \emptyset \wedge (p'_{i-1}, p'_i) \in \rightarrow_2)$ und $\delta = \emptyset$

Die Punkte (2) und (3) lassen sich nun vereinfachen. Es muss ein $s \in I\!N$ existieren mit $c_1 = \{p'_{s-1}\}$, $c_2 = \{p'_s\}$. Setze $p := p'_{s-1}$ und $p' := p'_s$. Aus $(p, p') \in \rightarrow_1 \cup \rightarrow_2$ folgt die Existenz eines Pfades mit Teilpfad $\langle p, p'\rangle$ in $G(b_1)$ und damit die Existenz von $ectr_1$ und $ectr_2$. δ lässt sich aus der Zugehörigkeit von (p, p') zu entweder $\rightarrow_1$ oder $\rightarrow_2$ schließen. Im ersten Fall ergibt sich $\delta = \{b_1\}$, im zweiten $\delta = \emptyset$. Zusammenfassend:

$ECG(IS)\lfloor_{\{b_1\}} = (C, \rightarrowtail_1, \rightarrowtail_2)$ mit:

(1) $C = \{\{p\} \mid p \in b_1\}$

(2) $\rightarrowtail_1 \subseteq C \times C$ mit $(c_1, c_2) \in \rightarrowtail_1$ gdw.
$\exists p, p' \in b_1 : c_1 = \{p\}$, $c_2 = \{p'\}$ und $(p, p') \in \rightarrow_1$

(3) $\rightarrowtail_2 \subseteq C \times C$ mit $(c_1, c_2) \in \rightarrowtail_2$ gdw.
$\exists p, p' \in b_1 : c_1 = \{p\}$, $c_2 = \{p'\}$ und $(p, p') \in \rightarrow_2$

Somit gilt $ECG(IS)\lfloor_{\{b_1\}} = (\{\{p\} \mid p \in b_1\}, \{(\{p\}, \{p'\}) \mid (p, p') \in \rightarrow_1\}, \{(\{p\}, \{p'\}) \mid (p, p') \in \rightarrow_2\})$. Dies entspricht der Aussage b) des Satzes 9.20. $\square$

Beweis von Satz 9.22 (Spezielle lokale Ereignisstruktur)

Der Beweis von **a)** und **b)** verläuft analog zum Beweis von Satz 9.20. Die Phase e_0 wird dabei aus allen Mengen, in denen sie enthalten ist, herausgenommen. Cases mit e_0 werden nicht betrachtet. Variablenzuweisungen/-anpassungen der Variablen $\_e_{in}(e_0)$, $\_e_{out}(e_0)$, $\_k(e_0)$, $\_z(e_0)$, usw. werden nicht aufgeführt und erläutert. Die Indexbereiche werden entsprechend angepasst.

Beispiele aus Fall 1.1:
Streiche „$(\_e_{in}(e_0) = true$ gdw. $(p_{k_1}, e_0) \in E$ gdw. $true)$. Die erste Äquivalenz gilt aufgrund der Definition von $\_e_{in}(\cdot)$, die zweite wegen der Satzvoraussetzung v)."
Ersetze „$M_2 = \{e_j \in \underline{b}_2 \setminus \{e_{k_0}\} \mid (j = 0) \vee (p_{k_1} \in Suc(p_j) \cup \{p_j\} \wedge j \in \{1, \ldots, m\})\} = \{e_j \mid p_j \in Pre(p_{k_1}) \wedge j \in \{1, \ldots, m\}\} \cup \{e_0, e_{k_1}\}$" durch „$M_2 = \{e_j \in \underline{b}_2 \setminus \{e_{k_0}\} \mid p_{k_1} \in Suc(p_j) \cup \{p_j\} \wedge j \in \{1, \ldots, m\}\} = \{e_j \mid p_j \in Pre(p_{k_1}) \wedge j \in \{1, \ldots, m\}\} \cup \{e_{k_1}\}$".

Beispiel: Neuformulierung Fall 2:
$\exists k_0, k_1 \in \{1,\dots,m\}, k_0 \neq k_1 : z^{\Pi,t^0}\langle V_{b_1}\rangle(p_{k_1}) = 1 \wedge z^{\Pi,t^0}\langle V_{\underline{b}_2}\rangle(e_{k_0}) = 1 \wedge p_{k_0} \in Pre(p_{k_1})$

Beispiel: Neuformulierung von $(*_3)$ in Beweisteil a) und von M_1 und M_2 in Beweisteil b): Ersetze jedesmal „$\{0,1,\dots,m\}$“ durch „$\{1,\dots,m\}$“.

Insbesondere ergibt sich aus $(*_2)$:
$Case(IS) = \{\{e_j, p_j\} \mid j \in \{1,\dots,m\}\} \cup \{\{e_j, p_i\} \mid p_j \in Pre(p_i), i, j \in \{1,\dots,m\}\}.$

Ergänzung (Notwendigkeit der Satzvoraussetung „Jeder Knoten von $G(b_1)$ habe mindestens einen Vorgängerknoten.“).

Aus der (bezüglichen Satz 9.20 zusätzlichen) Satzvoraussetzung „Jeder Knoten von $G(b_1)$ habe mindestens einen Vorgängerknoten.“ folgt: $\forall p \in b_1 : Pre(p) \neq \emptyset$. $(*)$

Die Eigenschaft $(*)$ ist notwendig, um den Unterfall 1.1.2.1.2 bei der Ausführung Π aufrecht zu erhalten. Laut Fallbeschreibung führt $V_{\underline{b}_2}$ nach t^9 als nächstes die Aktion A5 aus. Nun gilt wie vorher $M_1 = \{e_j \in \underline{b}_2 \setminus \{e_{k_1}\} \mid p_{k_1} \in Suc(p_j) \cup \{p_j\} \wedge p_{k_1} \notin Suc(p_j) \wedge j \in \{1,\dots,m\}\} = \emptyset$, allerdings $M_2 = \{e_j \in \underline{b}_2 \setminus \{e_{k_1}\} \mid p_{k_1} \in Suc(p_j) \cup \{p_j\} \wedge j \in \{1,\dots,m\}\} = \{e_j \mid p_j \in Pre(p_{k_1}) \wedge j \in \{1,\dots,m\}\}$. Die Vorgängereigenschaft $(*)$ garantiert $Pre(p_{k_1}) \neq \emptyset$ und somit $M_2 \neq \emptyset$. Der Fall „*not* a) und *not* b) und c)“ von A5 tritt ein. Das weitere Verhalten verläuft wieder analog.

Gelte hingegen *not* $(*)$, insbesondere $Pre(p_{k_1}) = \emptyset$, dann ist eine Fortführung der Ausführung Π, es ergibt sich eine Ausführung Π', im Fall 1.1.2.1.2 wie folgt möglich:

Wegen nun $M_2 = \emptyset$ tritt der Fall „*not* a) und *not* b) und *not* c)“ von A5 bei $V_{\underline{b}_2}$ ein. $V_{\underline{b}_2}$ sendet an V_{b_1} die Nachricht *break*. A5-abschließend erfolgt ein Solicitation-Aufruf bzgl. $\underline{b}_2 \setminus \{e_{k_1}\}$, Aktion A6. Gemäß A6 sendet $V_{\underline{b}_2}$ an V_{b_1} die Nachricht $solicit(\underline{b}_2 \setminus \{e_{k_1}\})$ und setzt $\_s(q) := true$, für alle $q \in \underline{b}_2 \setminus \{e_{k_1}\}$. A6 ist abgeschlossen zu einem Globalzeitpunkt t^{10}. Danach bleiben die Variablenbelegungen bei $V_{\underline{b}_2}$ solange konstant, bis eine Nachricht von V_{b_1} eintrifft und bearbeitet wird. In der Zwischenzeit findet bei $V_{\underline{b}_2}$ keine Aktion statt, da keine Vorbedingung erfüllt ist.

Die erste Nachricht, die V_{b_1} nach t^8 erhält, ist $reqmark(\underline{b}_2)$, abgeschickt von $V_{\underline{b}_2}$ nach t^9. V_{b_1} setzt $\_mark(v) := true$, für alle $v \in \underline{b}_2$ und schickt *ackmark* zurück (A8). Die nächste Nachricht, die V_{b_1} erhält, ist *break*, worauf A9 ausgeführt wird mit $\_mark(v) := false$ für alle $v \in \underline{b}_2$. Somit sind wieder alle $\_mark(\cdot)$-Variablen $false$. Als nächstes bearbeitet V_{b_1} die Nachricht $solicit(\underline{b}_2 \setminus \{e_{k_1}\})$ von V_{b_1} mittels Aktion A11 und setzt $\_s(v) := true$, für alle $v \in \underline{b}_2 \setminus \{e_{k_1}\}$. Unter Berücksichtigung der aktuellen Variablenbelegung bei V_{b_1} gilt $\_e_{out}(p_{k_1}) = false$. Somit resultiert der A11-abschließende Update-Aufruf A13.iv in $\_z(p_{k_1}) := \mathsf{F}$ zu einem Globalzeitpunkt $t^{11}, t^{11} > t^8$.

Nach t^{11} führt V_{b_1} als nächstes die Aktion A5 aus. Nach Senden von $reqmark(b_1)$ an und Empfang von *ackmark* von $V_{\underline{b}_2}$, gilt, bezogen auf die Aktionsbeschreibung von A5, unter Beachtung der aktuellen Variablenbelegungen bei V_{b_1}: $M_1 = M_2 = \{q' \in b_1 \setminus \{p_{k_1}\} \mid q' \in Suc(p_{k_1}) \cup \{p_{k_1}\} = Suc(p_{k_1})$. Die zweite Gleichheit gilt, da $G(b_1)$ nach Voraussetzung schlingenfrei ist. Angenommen, es gilt $Suc(p_{k_1}) \neq \emptyset$, dann folgt $M_1 = M_2 \neq \emptyset$ und Fall „*not* a) und *not* b) und c)“ von A5 tritt ein. V_{b_1} führt zu einem Globalzeitpunkt t^{12} eine Phasentransition $p_{k_1} \to q$ aus, mit $q \in M_1 = Suc(p_{k_1})$. Es gilt $q = p_{k_2}$ für ein $k_2 \in \{1,\dots,m\} \setminus \{k_1\}$. Beim anschließenden Update-Aufruf tritt wegen $\_e_{out}(p_{k_1}) = true$ der Fall A13.v ein, der die aktuellen Variablenbelegungen erhält. A5-abschließend sendet V_{b_1} an $V_{\underline{b}_2}$ die Nachricht $done(p_{k_1} \to p_{k_2})$.

Ab t^{10} ist $V_{\underline{b}_2}$ auf den Empfang von Nachrichten von V_{b_1} angewiesen, bevor neue Variablenbelegungen hervorgerufen werden können. Die erste Nachricht, die $V_{\underline{b}_2}$ nach t^{10} erhält, ist $reqmark(b_1)$, abgeschickt von V_{b_1} nach t^{11}. $V_{\underline{b}_2}$ setzt $\_mark(v) := true$ für alle $v \in b_1$ und schickt *ackmark* zurück(A8). Die nächste Nachricht, die $V_{\underline{b}_2}$ erhält, ist $done(p_{k_0} \to p_{k_1})$, zeitlich nach t^{12}.

$\Rightarrow \forall t \in]t^5, t^{12}[: zc(z^{\Pi',t}\langle V_I System(IS)\rangle) = \{p_{k_1}, e_{k_1}\}$

$zc(z^{\Pi',t^{12}}\langle V_I System(IS)\rangle) = \{p_{k_2}, e_{k_1}\}$, $p_{k_2} \in Suc(p_{k_1})$

V_{b_1} befindet sich zum Zeitpunkt t^{12} in Aktion A5.

Die Präzisierung des weiteren Verlaufes der Ausführung Π' ist für die folgenden Betrachtungen nicht notwendig.

Mit Π' als Ausführung und $t_0 := t^0$, $t_i :=$„Zeitpunkt der i-ten Phasentransition in $V_I System(IS)$“ folgt aus $(*_1)$ aus dem Beweis von Satz 9.20:

$\exists \overline{ectr}_1 \in (Case(IS).\mathcal{P}(B))^*, \exists \overline{ectr}_2 \in (\mathcal{P}(B).Case(IS))^*, \exists i \in I\!N :$
$\overline{ectr}_1.\underbrace{\{p_{k_{i-1}}, e_{k_{i-1}}\}.\delta.\{p_{k_i}, e_{k_{i-1}}\}}_{str}.\overline{ectr}_2 \in \mathcal{ECT}[\![IS]\!]$ mit $\delta = \emptyset \wedge (e_{k_{i-1}}, p_{k_{i-1}}) \notin E.$

Die Subtrace str resultiert dabei aus der oben angegebenen Fortführung im Unterfall 1.1.2.1.2.

$\Rightarrow$ {Übergang zur Sicht auf b_1}
$\exists ectr_1 \in \big(Case(IS)|_{\{b_1\}}.\mathcal{P}(\{b_1\})\big)^*, \exists ectr_2 \in \big(\mathcal{P}(\{b_1\}).Case(IS)|_{\{b_1\}}\big)^*, \exists i \in I\!N :$
$ectr_1.\{p_{k_{i-1}}\}.\delta.\{p_{k_i}\}.ectr_2 \in \mathcal{ECT}[\![IS]\!]|_{\{b_1\}}$ mit $\delta = \emptyset \wedge (e_{k_{i-1}}, p_{k_{i-1}}) \notin E.$

$\Rightarrow$ {Implikationen nach $(*_4)$ und Folgerungen nach $(*_5)$ im Beweis zu Satz 9.20}
$\exists ectr_1 \in \big(Case(IS)|_{\{b_1\}}.\mathcal{P}(\{b_1\})\big)^*, \exists ectr_2 \in \big(\mathcal{P}(\{b_1\}).Case(IS)|_{\{b_1\}}\big)^*, \exists i \in I\!N :$
$ectr_1.\{p_{k_{i-1}}\}.\delta.\{p_{k_i}\}.ectr_2 \in \mathcal{ECT}[\![IS]\!]|_{\{b_1\}}$ mit $\delta = \emptyset \wedge (p_{k_{i-1}}, p_{k_i}) \in \rightarrow_1.$

$\Rightarrow$ {Umindizierung $p_{k_x} \to p'_x$}
$\exists ectr_1 \in \big(Case(IS)|_{\{b_1\}}.\mathcal{P}(\{b_1\})\big)^*, \exists ectr_2 \in \big(\mathcal{P}(\{b_1\}).Case(IS)|_{\{b_1\}}\big)^*, \exists i \in I\!N, \exists p'_{i-1}, p'_i \in b_1 :$
$ectr_1.\{p'_{i-1}\}.\delta.\{p'_i\}.ectr_2 \in \mathcal{ECT}[\![IS]\!]|_{\{b_1\}}$ mit $\delta = \emptyset \wedge (p'_{i-1}, p'_i) \in \rightarrow_1.$
$\Rightarrow$
Widerspruch zur Satzaussage a)

Somit zeigt sich, dass die Vorgängerknotenforderung für die im Satz 9.22 angegebenen Konstruktion von IS (ohne e_0) notwendig ist, damit die Satzaussage a) und, als Folgerung daraus, auch b) erfüllt sind. □

A.4 Beweise aus Kapitel 10

Beweis von Satz 10.1 (Elementare Struktureigenschaften)

Es gelten die Bezeichnungen und Voraussetzungen aus dem Satz. Sei eine beliebige Ausführung Π von $V_I System(IS)$ gegeben, die, gemäß Definition 4.1, $z_0 z_1 z_2 \dots$ erzeugt. (Unter Bezugnahme auf Π wird im Beweis die Notation 3.12 verwendet.) Seien $t_0, t_1, t_2, \dots$ die durch die Definition gegebenen Globalzeitpunkte. Sei $i \in \{1, 2, \dots\}$ beliebig aber fest, jeweils aus dem zugeordneten Wertebereich.

Zu **a)**. Ohne Einschränkung gelte $z_{i+1} \neq \mathsf{F}$, andernfalls erfolgt eine Indexverschiebung von i.

Zu zeigen: Aus $z^{t_i}\langle V_{b(p)}\rangle(p) = \mathsf{F}$ und $z^{t_{i+1}}\langle V_{b(p)}\rangle(p) \neq \mathsf{F}$ und $z^{t_j}\langle V_{b(p)}\rangle(p) = 1$ mit $i < j$ folgt mindestens einer der drei folgenden Fälle :

i) $E(p) \neq \emptyset$.

ii) $K(p) \setminus b(p) \neq \emptyset$.

iii) $\exists t', t_i < t' < t_j : z^{t'}\langle V_{b(p)}\rangle(p) = 0$.

Im Fall iii) gilt Teil a) mit $k := max\{j \in I\!N_0 \mid t_j \leq t'\}$.

Beweis mittels Fallunterscheidung:

Fall 1). $z^{t_{i+1}}\langle V_{b(p)}\rangle(p) \neq 1$.

Fall 1.1). $z^{t_{i+1}}\langle V_{b(p)}\rangle(p) = q$ mit $q \in b(p) \setminus \{p\}$.

Es gibt keine Aktion aus A1 bis A13, durch die bei $V_{b(p)}$ ein direkter Wechsel der Phasenqualität von p, von $\_z(p) = \mathsf{F}$ nach $\_z(p) = q$, hervorgerufen werden kann.

$\Rightarrow$
Fall 1.1) kann nicht eintreten.

Fall 1.2). $z^{t_{i+1}}\langle V_{b(p)}\rangle(p) = 0$.
$\Rightarrow$
Die Aussage iii) gilt mit $t' := t_{i+1}$.

Fall 2). $z^{t_{i+1}}\langle V_{b(p)}\rangle(p) = 1$.

Bei $V_{b(p)}$ findet zum Globalzeitpunkt t_{i+1}, ausgehend von $\_z(p) = \mathsf{F}$, eine Zuweisung $\_z(p) := 1$ statt. Als mögliche Aktion kommt nur A13.i in Frage. Die Aktionsbeschreibung fordert ($\_e_{out}(p)$ oder (*not* $\_e_{out}(p)$ und *not* $e_{in}(p)$ und *not* $\_s(v)$ für eine K-Nachbarphase v von p.)).

Fall 2.1). $e_{out}^{t_{i+1}}\langle V_{b(p)}\rangle(p) = true$.

Gemäß der Auswertung von $\_e_{out}(p)$ existieren ein Bereich $b' \in B$ und eine Phase $x \in P$ mit $(p,x) \in E\langle b(p), b'\rangle$ und $in^{t_{i+1}}\langle V_{b(p)}\rangle(x) = true$. Insbesondere gilt also $E(p) \neq \emptyset$.
$\Rightarrow$
Aussage i) gilt.

Fall 2.2). $e_{out}^{t_{i+1}}\langle V_{b(p)}\rangle(p) = false$ und $e_{in}^{t_{i+1}}\langle V_{b(p)}\rangle(p) = false$ und $\forall v \in K(p) : s^{t_{i+1}}\langle V_{b(p)}\rangle(v) = false$.

Fall 2.2.1). $E(p) \neq \emptyset$ oder $K(p) \setminus b(p) \neq \emptyset$.
$\Rightarrow$
Aussage i) oder ii) gilt direkt.

Fall 2.2.2). $E(p) = \emptyset$ und $K(p) \setminus b(p) = \emptyset$.

Aus der „Zu zeigen“-Voraussetzung $z^{t_i}\langle V_{b(p)}\rangle(p) \neq \mathsf{F}$ und der für jede Ausführung vorgegebenen Anfangsbelegung $z^{t_0}\langle V_{b(p)}\rangle(p) = 1$ folgt die Existenz von Globalzeitpunkten t^1 und t^2, $t^1 < t^2 < t_{i+1}$, so dass gilt: $\forall t \in [t^1, t^2[: z^t\langle V_{b(p)}\rangle(p) \neq \mathsf{F}$ und $\forall t \in [t^2, t_{i+1}[: z^t\langle V_{b(p)}\rangle(p) = \mathsf{F}$.
$\Rightarrow$
$V_{b(p)}$ führt zum Globalzeitpunkt t^2, ausgehend von $\_z(p) \neq \mathsf{F}$, eine Zuweisung $\_z(p) := \mathsf{F}$ aus. Diese Zuweisung findet sich nur bei Aktion A13.iii oder A13.iv. In beiden Fällen wird dabei (*not* $\_e_{out}(p)$ und ($\_e_{in}(p)$ oder $\_s(v)$ für eine K-Nachbarphase v von p)) vorausgesetzt.
$\Rightarrow$
Wegen der Fallvoraussetzung 2.2.2 muss $e_{in}^{t^2}\langle V_{b(p)}\rangle(p) = true$ gelten. Aus den Definitionen/Festlegungen von $e_{in}^{\cdot}\langle\cdot\rangle(\cdot)$ bzw. $\_e_{in}(\cdot)$ folgt die Existenz von $v \in P$ mit $(v,p) \in E$, und es gilt $in^{t^2}\langle V_{b(p)}\rangle(v) = true$. $(*_1)$
$\Rightarrow$
Bei $V_{b(p)}$ findet zu einem Globalzeitpunkt t^3 eine letzte Zuweisung $\_in(v) := true$ vor t^2 statt. Als Aktionen kommen hierzu nur A1 oder A10 in Frage. Beide Aktionen führen zu einem abschließenden Update-Aufruf (A13). Zum Zeitpunkt des Updates gilt im Fall A1 $\_mark(v) = false$ als Initialisierungsvoraussetzung, und im Fall A10 gilt $\_mark(v) = false$ aufgrund der dem Update direkt vorangegangenen Zuweisung. Das Update hat als Konsequenz die zum Globalzeitpunkt t^2 stattfindende Zuweisung $\_z(p) := \mathsf{F}$. Zwischenzeitlich kann sich die Belegung nicht ändern. Es gilt demnach $mark^{t^2}\langle V_{b(p)}\rangle(v) = false$. $(*_2)$
$\Rightarrow$
Die Beobachtung 2 aus dem Beweis von Satz 3.14 liefert zusammen mit $(*_1)$ und $(*_2)$: $z^{t^2}\langle V_{b(v)}\rangle(v) \neq 0$. $(*_3)$

Annahme: $\exists t^4 \in]t^2, t_{i+1}] : e_{in}^{t^4}\langle V_{b(p)}\rangle(p) = false$.
$\Rightarrow$
Gemäß den Definitionen/Festlegungen von $e_{in}^{\cdot}\langle\cdot\rangle(\cdot)$ bzw. $\_e_{in}(\cdot)$ muss insbesondere $in^{t^4}\langle V_{b(p)}\rangle(v) = false$ gelten.
$\Rightarrow$
Es existiert ein Globalzeitpunkt $t^5 \in]t^2, t^4]$, zu dem bei $V_{b(p)}$ die Zuweisung $\_in(v) := false$ ausgeführt wird. Als Aktion kommt nur A10 in Frage. Nach der Zuweisung erfolgt bei A10

$\_mark(v) := false$. Somit existiert ein Globalzeitpunkt $t^6 \in]t^5, t_{i+1}]$ mit $in^{t^6}\langle V_{b(p)}\rangle(v) = false$ und $mark^{t^6}\langle V_{b(p)}\rangle(v) = false$.
$\Rightarrow$
Analog zur Beobachtung 2 aus Satz 3.14 lässt sich $z^{t^6}\langle V_{b(v)}\rangle(v) = 0$ schließen. $(*_4)$
Dabei ist noch zu berücksichtigen, dass als Initialbelegung $\_mark(x) = \_in(x) = false$ für jede Phase $x \in P$ bei jeder Komponente von $V_I System(IS)$ gilt.

Richtet man sich nach der Definition 4.1 des Verhaltens und berücksichtigt $(*_3)$ und $(*_4)$, dann existieren globale Aktivitätszustände $z'_0, z'_1, \ldots, z'_k, z'_{k+1}, \ldots, z'_{k+l}, \ldots \in GZustand(IS)$, $k, l \in I\!N$, so dass $z'_0 z'_1 \ldots z'_k z'_{k+1} \ldots z'_{k+l} \ldots \in \mathcal{V}[\![IS]\!]$ mit $z'_k(p) = z'_{k+1}(p) = \ldots = z'_{k+l}(p) = z^{t^2}\langle V_I System(IS)\rangle(p) = \mathsf{F}$ und $z'_k(v) = z^{t^2}\langle V_I System(IS)\rangle(v) \neq \mathsf{0}$ und $z'_{k+l}(v) = z^{t^6}\langle V_I System(IS)\rangle(v) = \mathsf{0}$.
$\Rightarrow$
$\exists k' \in I\!N,\ k \leq k' < k+l : z'_{k'}(p) \neq \mathsf{0} \wedge z'_{k'}(v) \neq \mathsf{0} \wedge z'_{k'+1}(v) = \mathsf{0}$.
$\Rightarrow$
Wegen $(v, p) \in E$ liegt nun ein Widerspruch zu Satz 4.6.a vor, demnach $z'_{k'+1}(v) \neq \mathsf{0}$ gelten muss. Folglich ist die Annahme falsch.

Insbesondere gilt somit $e_{in}^{t_{i+1}}\langle V_{b(p)}\rangle(p) = true$, was ein Widerspruch zur übergeordneten Fallvoraussetzung 2.2 ist.
$\Rightarrow$
Fall 1.1) kann nicht eintreten.

Damit ist die Fallunterscheidung zu a) abgeschlossen. Für alle auftretenden Fälle wurde i, ii) oder iii) gezeigt. Es gilt Teil a) des Satzes.

Zu **b)**. Zu zeigen: Aus $z^{t_i}\langle V_{b(p)}\rangle(p) = \mathsf{F}$ folgt mindestens einer der beiden folgenden Fälle:

i) $\exists v \in P \setminus b(p) : ((v, p) \in E \wedge z^{t_i}\langle V_{b(v)}\rangle(v) \neq \mathsf{0})$.

ii) $\exists v \in P \setminus b(p) : ((v, p) \in K \wedge (\exists w \in b(v), \exists t', t' \leq t_i : z^{t'}\langle V_{b(v)}\rangle(w) \in \{v, \mathsf{F}\}))$.

Im Fall ii) gilt Teil b) mit $j := max\{j' \in I\!N_0 \mid t_{j'} \leq t'\}$.

Beweis: Es gelte $z^{t_i}\langle V_{b(p)}\rangle(p) = \mathsf{F}$.

Wegen der initial vorgegebenen Belegung $z^{t_0}\langle V_{b(p)}\rangle(p) = 1$ existiert ein Globalzeitpunkt $t^1 \in]t_0, t_i]$, zu dem $V_{b(p)}$, ausgehend von $\_z(p) \neq \mathsf{F}$, eine letzte Zuweisung $\_z(p) := \mathsf{F}$ vor/bei t_i ausführt. Diese Zuweisung findet sich nur bei Aktion A13.iii oder A13.iv. In beiden Fällen wird dabei (*not* $\_e_{out}(p)$ und ($\_e_{in}(p)$ oder $\_s(v)$ für eine K-Nachbarphase v von p)) vorausgesetzt. Es gilt somit der folgende Fall 1) oder/und Fall 2).

Fall 1). $e_{in}^{t^1}\langle V_{b(p)}\rangle(p) = true$.

Aus den Definitionen/Festlegungen von $e_{in}\langle\cdot\rangle(\cdot)$ bzw. $\_e_{in}(\cdot)$ folgt die Existenz von $v \in P \setminus b(p)$ mit $(v, p) \in E$, und es gilt $in^{t^1}\langle V_{b(p)}\rangle(v) = true$. $(*_1)$
$\Rightarrow$
Bei $V_{b(p)}$ findet zu einem Globalzeitpunkt t^2 eine letzte Zuweisung $\_in(v) := true$ vor t^1 statt. Als Aktionen kommen hierzu nur A1 oder A10 in Frage. Beide Aktionen führen zu einem abschließenden Update-Aufruf (A13). Zum Zeitpunkt des Updates gilt im Fall A1 $\_mark(v) = false$ als Initialisierungsvoraussetzung und im Fall A10 gilt $\_mark(v) = false$ aufgrund der dem Update direkt vorangegangenen Zuweisung. Das Update hat als Konsequenz die zum Globalzeitpunkt t^1 stattfindende Zuweisung $\_z(p) := \mathsf{F}$. Zwischenzeitlich kann sich die Belegung nicht ändern. Es gilt demnach $mark^{t^1}\langle V_{b(p)}\rangle(v) = false$. $(*_2)$
$\Rightarrow$
Die Beobachtung 2 aus dem Beweis von Satz 3.14 liefert zusammen mit $(*_1)$ und $(*_2)$: $z^{t^1}\langle V_{b(v)}\rangle(v) \neq \mathsf{0}$. $(*_3)$

Annahme: $\exists t^3 \in]t^1, t_i] : z^{t^3}\langle V_{b(v)}\rangle(v) = \mathsf{0}$.

Richtet man sich nach der Definition 4.1 des Verhaltens und berücksichtigt $(*_3)$ und die Annahme,

dann existieren globale Aktivitätszustände $z'_0, z'_1, \ldots, z'_k, z'_{k+1}, \ldots, z'_{k+l}, \ldots \in GZustand(IS)$, $k, l \in I\!N$, so dass $z'_0 z'_1 \ldots z'_k z'_{k+1} \ldots z'_{k+l} \ldots \in \mathcal{V}[\![IS]\!]$ mit $z'_k(p) = z'_{k+1}(p) = \ldots = z'_{k+l}(p) = z^{t^1}\langle V_I System(IS)\rangle(p) = \mathsf{F}$ und $z'_k(v) = z^{t^1}\langle V_I System(IS)\rangle(v) \neq 0$ und $z'_{k+l}(v) = z^{t^3}\langle V_I System(IS)\rangle(v) = 0$.
$\Rightarrow$
$\exists k' \in I\!N,\ k \leq k' < k+l : z'_{k'}(p) \neq 0 \wedge z'_{k'}(v) \neq 0 \wedge z'_{k'+1}(v) = 0$.
$\Rightarrow$
Wegen $(v,p) \in E$ liegt nun ein Widerspruch zu Satz 4.6.a vor, demnach $z'_{k'+1}(v) \neq 0$ gelten muss. Folglich ist die Annahme falsch.

Insbesondere gilt somit $z^{t_i}\langle V_{b(v)}\rangle(v) \neq 0$.
$\Rightarrow$
Aussage i) gilt, mit $(v,p) \in E$ wegen $(*_1)$.

Fall 2). $s^{t^1}\langle V_{b(p)}\rangle(v) = true$ für eine K-Nachbarphase v von p. $(*'_1)$

Wegen der initial vorgegebenen Belegung $s^{t_0}\langle V_{b(p)}\rangle(v) = false$ existiert ein Globalzeitpunkt $t^2 \in]t_0, t^1]$, zu dem $V_{b(p)}$, ausgehend von $\_s(v) = false$, eine Zuweisung $\_s(v) := true$ ausführt. Diese Zuweisung findet sich nur bei den Aktionen A6 und A11.

A6 entfällt als Möglichkeit, da, durch die Vorbedingung vorgegeben, $M \subseteq b(p)$ gelten muss, mit der Konsequenz $v \in b(p)$. Dies ist aufgrund der Disjunktheit der Bereiche nicht der Fall.

Bei A11 erfolgt die Zuweisung als Reaktion auf den Empfang einer Nachricht $solicit(M)$, mit $v \in M$, gesendet von einer Komponente $V_{b'}$. Das Senden von $solicit(M)$ durch $V_{b'}$ erfolgt ausschließlich mittels Aktion A6. Aus dessen Vorbedingung folgt $b' = b(v)$.

Die Ausführung von A6 erfordert einen Aufruf aus einer anderen Aktion, ausgeführt von $V_{b(v)}$. In Frage kommen hierzu nur die Aktionen A4 oder A5.

Fall 2.1). $V_{b(v)}$ führt zu einem Globalzeitpunkt $t^3 \in]t_0, t^2]$ den Solicitation-Aufruf bzgl. $\{v\}$ aus A4 heraus aus.

Als Bedingung für den Aufruf liefert die Aktionsbeschreibung (unter Beachtung der gültigen Bezeichnungen) $\_z(w) = v$ für eine Phase $w \in b(v)$. Es gilt $z^{t^3}\langle V_{b(v)}\rangle(w) = v$.
$\Rightarrow$
Aussage ii) gilt mit $t' := t^3$, mit $(v,p) \in K$ und $v \notin b(p)$ wegen $(*'_1)$.

Fall 2.2). $V_{b(v)}$ führt zu einem Globalzeitpunkt $t^3 \in]t_0, t^2]$ den Solicitation-Aufruf bzgl. $b(v) \setminus \{w\}$, mit $w \in b(v)$, aus A5 heraus aus.

w ist die am Anfang von A5 bei $V_{b(v)}$ aktuelle Phase, d.h. es gilt dort $\_z(w) \neq 0$. Die Vorbedingung fordert $\_z(w) = \mathsf{F}$ als Ausführungsvoraussetzung der Aktion. Folglich existiert ein Globalzeitpunkt $t^4 \in]t_0, t^3[$ mit $z^{t^4}\langle V_{b(v)}\rangle(w) = \mathsf{F}$.
$\Rightarrow$
Aussage ii) gilt mit $t' := t^4$, mit $(v,p) \in K$ und $v \notin b(p)$ wegen $(*'_1)$.

Damit ist der Beweis zu b) abgeschlossen. Für alle auftretenden Fälle wurde i oder ii) gezeigt. Es gilt Teil b) des Satzes.

Zu **c)**.

$z_i(p) = \mathsf{F}$
$\Rightarrow$ {Satz 10.1.b}
$\exists v \in P \setminus b(p) : (((v,p) \in E \wedge z_i(v) \neq 0) \vee ((v,p) \in K \wedge (\exists w \in b(v), \exists j \in I\!N, j \leq i : z_j(w) \in \{v, \mathsf{F}\})))$
$\Rightarrow$ {Abschwächung}
$\exists v \in P \setminus b(p) : ((v,p) \in E \vee (v,p) \in K)$
$\Rightarrow$ {Definition 2.3}
$\exists v \in P \setminus b(p) : (v \in E^{-1}(p) \vee v \in K^{-1}(p))$
$\Rightarrow$ {Umformung}
$E^{-1}(p) \setminus b(p) \neq \emptyset \vee K^{-1}(p) \setminus b(p) \neq \emptyset$

$\Rightarrow$ {Es gilt immer: $E^{-1}(p) \cap b(p) = \emptyset$, K ist symmetrisch}
$E^{-1}(p) \neq \emptyset \vee K(p) \setminus b(p) \neq \emptyset$.

Die Implikationskette liefert die Gültigkeit von Teil c) des Satzes.

Zu **d)**. Es gelte neben den Satzvoraussetzungen von d): $(p, v) \in K \wedge K(q) \setminus b(q) = \emptyset = K(r) \setminus b(r)$.

Zu zeigen: Aus $z^{t_i}\langle V_{b(v)}\rangle(v) = \mathsf{F}$ folgt mindestens einer der beiden folgenden Fälle:

i) $K(v) \setminus (b(v) \cup b(p)) \neq \emptyset$.

ii) $E^{-1}(v) \neq \emptyset$.

Beweis: Es gelte $z^{t_i}\langle V_{b(v)}\rangle(v) = \mathsf{F}$.

Wegen der initial vorgegebenen Belegung $z^{t_0}\langle V_{b(v)}\rangle(v) = 1$ existiert ein Globalzeitpunkt $t^1 \in]t_0, t_i]$, zu dem $V_{b(v)}$, ausgehend von $\_z(v) \neq \mathsf{F}$, eine letzte Zuweisung $\_z(v) := \mathsf{F}$ vor/bei t_i ausführt. Diese Zuweisung findet sich nur bei Aktion A13.iii oder A13.iv. In beiden Fällen wird dabei (*not* $\_e_{out}(v)$ und ($\_e_{in}(v)$ oder $\_s(x)$ für eine K-Nachbarphase x von v)) vorausgesetzt. Es gilt somit der folgende Fall 1) oder/und Fall 2).

Fall 1). $e_{in}^{t^1}\langle V_{b(v)}\rangle(v) = true$.

Aus den Definitionen/Festlegungen von $e_{in}^{\cdot}\langle\cdot\rangle(\cdot)$ bzw. $\_e_{in}(\cdot)$ folgt die Existenz von $x \in P \setminus b(v)$ mit $(x, v) \in E$, und es gilt $in^{t^1}\langle V_{b(v)}\rangle(x) = true$.
$\Rightarrow$
Insbesondere gilt $E^{-1}(v) \neq \emptyset$.
$\Rightarrow$
Aussage ii) gilt.

Fall 2). $s^{t^1}\langle V_{b(v)}\rangle(x) = true$ für eine K-Nachbarphase x von v. $(*_1)$

Wegen der initial vorgegebenen Belegung $s^{t_0}\langle V_{b(v)}\rangle(x) = false$ existiert ein Globalzeitpunkt $t^2 \in]t_0, t^1]$, zu dem $V_{b(v)}$, ausgehend von $\_s(x) = false$, eine Zuweisung $\_s(x) := true$ ausführt. Diese Zuweisung findet sich nur bei den Aktionen A6 und A11.

A6 entfällt als Möglichkeit, da, durch die Vorbedingung vorgegeben, $M \subseteq b(v)$ gelten muss, mit der Konsequenz $x \in b(v)$. Dies ist aufgrund der Disjunktheit der Bereiche nicht der Fall.

Bei A11 erfolgt die Zuweisung als Reaktion auf den Empfang einer Nachricht $solicit(M)$, mit $x \in M$, gesendet von einer Komponente $V_{b'}$. Das Senden von $solicit(M)$ durch $V_{b'}$ erfolgt ausschließlich mittels Aktion A6. Aus dessen Vorbedingung folgt $b' = b(x)$.

Die Ausführung von A6 erfordert einen Aufruf aus einer anderen Aktion, ausgeführt von $V_{b(x)}$. In Frage kommen hierzu nur die Aktionen A4 oder A5.

Fall 2.1). $V_{b(x)}$ führt zu einem Globalzeitpunkt $t^3 \in]t_0, t^2]$ den Solicitation-Aufruf bzgl. $\{x\}$ aus A4 heraus aus.

Als Bedingung für den Aufruf liefert die Aktionsbeschreibung (unter Beachtung der gültigen Bezeichnungen) $\_z(y) = x$ für eine aktuell bei $V_{b(x)}$ eingenommene Phase $y \in b(x)$.

Die Zuweisung $\_z(y) := x$ kann $V_{b(x)}$ nur bei Aktion A3 ausführen, d.h. es gibt einen Globalzeitpunkt $t^4 \in]t_0, t^3]$, zum dem $V_{b(x)}$ die Aktion von A3 beginnt. Voraussetzung ist die Erfüllung der Vorbedingung. Diese fordert $b(x)$ autonom, d.h. $b(x) \notin \underline{B}$.
$\Rightarrow$
$b(x) \neq b(p)$, da $b(p) \in \underline{B}$ nach Voraussetzung.
$\Rightarrow$ {mit $(*_1)$}
$x \notin b(p) \wedge x \notin b(v) \wedge (x, v) \in K$.
$\Rightarrow$
$K(v) \setminus (b(p) \cup b(v)) \neq \emptyset$.
$\Rightarrow$
Aussage i) gilt.

Fall 2.2). $V_{b(x)}$ führt zu einem Globalzeitpunkt $t^3 \in]t_0, t^2]$ den Solicitation-Aufruf bzgl. $b(x) \setminus \{y\}$, mit $y \in b(x)$, aus A5 heraus aus.

y ist die am Anfang von A5 bei $V_{b(x)}$ aktuelle Phase, d.h. es gilt dort $\_z(y) \neq \mathsf{0}$. Die Vorbedingung fordert $\_z(y) = \mathsf{F}$ als Ausführungsvoraussetzung der Aktion. Folglich existiert ein Globalzeitpunkt $t^4 \in]t_0, t^3[$, zu dem die Aktion beginnt, mit $z^{t^4}\langle V_{b(x)}\rangle(y) = \mathsf{F}$.

Annahme: $b(x) = b(p)$.

Als strukturelle Voraussetzung ist für den Fall d) $K(q) \setminus b(q) = \emptyset$ und $K(r) \setminus b(r) = \emptyset$ vorgegeben. Folglich gilt zu jedem Zeitpunkt t einer Ausführung von $V_I System(IS)$: (not $\_k(q)$ und not $\_mark(q')$ für jede K-Nachbarphase q' von q, falls $y \neq q$), sowie (not $\_k(r)$ und not $\_mark(r')$ für jede K-Nachbarphase r' von r, falls $y = q$). $(*_2)$

Die Aktionsbeschreibung für A5 definiert die Menge M_2 für $V_{b(x)}$ bei aktueller Phase y (d.h. $\_z(y) \neq \mathsf{0}$) als $M_2 := \{q' \in b(x) \setminus \{y\} \mid not\ \_k(q')$ und $not\ \_mark(v)$ für jede K-Nachbarphase v von q' $\}$. Als weitere Voraussetzung gilt $b(x) = b(q) = b(r)$. Mit $(*_2)$ folgt somit ($q \in M_2$, falls $y \neq q$) und ($r \in M_2$, falls $y = q$).
$\Rightarrow$
Fall c) der Aktionsbeschreibung von A5 ist immer erfüllt.
$\Rightarrow$
Der Solicitation-Aufruf bzgl. $b(x) \setminus \{y\}$ in A5, der not a) und not b) und not c) voraussetzt, wird bei $V_{b(x)}$ zu keinem Zeitpunkt t stattfinden.
$\Rightarrow$
Es liegt ein Widerspruch zur Bedingung für Fall 2.2 vor. Die Annahme ist somit falsch.

Aus der Ungültigkeit der Annahme folgt $b(x) \neq b(p)$.
$\Rightarrow \{$mit $(*_1)\}$
$x \notin b(p) \wedge x \notin b(v) \wedge (x, v) \in K$.
$\Rightarrow$
$K(v) \setminus (b(p) \cup b(v)) \neq \emptyset$.
$\Rightarrow$
Aussage i) gilt.

Damit ist der Beweis zu d) abgeschlossen. Für alle auftretenden Fälle wurde i) oder ii) gezeigt. Es gilt Teil d) des Satzes. $\square$

Symbolverzeichnis

(P, B, C, K)	Struktur eines Lose Gekoppelten Systems; Seite 94
$(P, B, \underline{B}, K, E)$	Struktur eines I–Systems; Seite 10
$(c_0.c_1.c_2 \ldots)\lfloor_T$	Folge von Cases mit Sicht auf Bereichsmenge T; Seite 102
$(c_0.\delta_1.c_1.\delta_2.c_2 \ldots)\lfloor_T$	alternierende Folge von Cases und Bereichsmengen mit Sicht auf Bereichsmenge T; Seite 103
$(z_0.z_1.z_2 \ldots)\lfloor_T$	Folge von globalen Aktivitätszuständen mit Sicht auf Bereichsmenge T; Seite 100
$[p_1\!<\cdot>, \ldots, p_n\!<\cdot>]$	Notation für globalen Aktivitätszustand; Seite 15
$\lfloor ztr \rfloor$	zc-Transformation einer Folge ztr von globalen Aktivitätszuständen; Seite 45
$\lfloor ztr \rfloor^{e}$	Erweiterte zc-Transformation einer Folge ztr von globalen Aktivitätszuständen; Seite 50
$\exists!$	Es existiert genau ein…; Seite 14
Π, Π', Π_1	Bezeichnungen für Ausführungen eines $\mathrm{V_I}$Systems; Seite 24
$\alpha^{\Pi,t}\langle V_I System(IS)\rangle$	$\alpha \in \{mark, in, s, e_{in}, e_{out}, k, z\}$, α-Globalbelegung bzgl. I–System IS und Globalzeitpunkt t von Ausführung Π; Seite 24
$\alpha^{\Pi,t}\langle V_b\rangle(p)$	$\alpha \in \{mark, in, s, e_{in}, e_{out}, k, z\}$, Wert der Variablen $\_\alpha(p)$ bei Komponente V_b zum Globalzeitpunkt t bezogen auf Ausführung Π; Seite 24
$a_1 a_2 \ldots \infty$	Trace, deren Länge unendlich ist; Seite 34
$a_1 a_2 \ldots$	Trace, deren Länge endlich oder unendlich ist; Seite 34
A^+	positive Hülle einer endlichen Menge A; Seite 34
A^*	Kleene'sche Hülle einer endlichen Menge A; Seite 34
$AB(IS)$	Menge der autonomen Bereiche des I–Systems IS; Seite 10
$b(p)$	Bereich der Phase p; Seite 10
$c\lfloor_T$	Case c mit Sicht auf Bereichsmenge T; Seite 102
$cz(c)$	globaler Aktivitätszustand zu Case c; Seite 16
$Case(IS)$	Menge aller Cases von I–System IS; Seite 14
$Case(IS)\lfloor_T$	Menge aller Cases von I–System IS mit Sicht auf Bereichsmenge T; Seite 102
$CG(IS)$	Casegraph von I–System IS; Seite 78
$CG(LCS)$	Casegraph von Lose Gekoppeltem System LCS; Seite 95

$CG(IS)\lfloor_T$	Casegraph von I–System IS mit Sicht auf Bereichsmenge T; Seite 105
$\mathcal{CT}[\![IS]\!]$	Casetrace-Semantik von I–System IS; Seite 41
$\mathcal{CT}[\![IS]\!](c_0)$	Casetrace-Semantik von I–System IS bzgl. Start-Case c_0; Seite 41
$\mathcal{CT}[\![IS]\!]\lfloor_T$	Casetrace-Semantik von I–System IS mit Sicht auf Bereichsmenge T; Seite 102
$\mathcal{CT}[\![IS]\!](c_0)\lfloor_T$	Casetrace-Semantik von I–System IS mit Sicht auf Bereichsmenge T bzgl. Start-Case c_0; Seite 102
$\mathcal{CT}^{\mathrm{i}}[\![IS]\!]$	Interleaving Casetrace-Semantik von I–System IS; Seite 68
$\mathcal{CT}^{\mathrm{i}}[\![IS]\!](c_0)$	Interleaving Casetrace-Semantik von I–System IS bzgl. Start-Case c_0; Seite 68
$\mathcal{CT}^{\mathrm{i}}[\![IS]\!]\lfloor_T$	Interleaving Casetrace-Semantik von I–System IS mit Sicht auf Bereichsmenge T; Seite 104
$\mathcal{CT}^{\mathrm{i}}[\![IS]\!](c_0)\lfloor_T$	Interleaving Casetrace-Semantik von I–System IS mit Sicht auf Bereichsmenge T bzgl. Start-Case c_0; Seite 104
$ECG(IS)$	Erweiterter Casegraph von I–System IS; Seite 82
$ECG(IS)\lfloor_T$	Erweiterter Casegraph von I–System IS mit Sicht auf Bereichsmenge T; Seite 105
$\mathcal{ECT}[\![IS]\!]$	Erweiterte Casetrace-Semantik von I–System IS; Seite 47
$\mathcal{ECT}[\![IS]\!](c_0)$	Erweiterte Casetrace-Semantik von I–System IS bzgl. Start-Case c_0; Seite 48
$\mathcal{ECT}[\![IS]\!]\lfloor_T$	Erweiterte Casetrace-Semantik von I–System IS mit Sicht auf Bereichsmenge T; Seite 103
$\mathcal{ECT}[\![IS]\!](c_0)\lfloor_T$	Erweiterte Casetrace-Semantik von I–System IS mit Sicht auf Bereichsmenge T bzgl. Start-Case c_0; Seite 103
$\mathcal{ECT}^{\mathrm{i}}[\![IS]\!]$	Erweiterte Interleaving Casetrace-Semantik von I–System IS; Seite 64
$\mathcal{ECT}^{\mathrm{i}}[\![IS]\!](c_0)$	Erweiterte Interleaving Casetrace-Semantik von I–System IS bzgl. Start-Case c_0; Seite 64
$\mathcal{ECT}^{\mathrm{i}}[\![IS]\!]\lfloor_T$	Erweiterte Interleaving Casetrace-Semantik von I–System IS mit Sicht auf Bereichsmenge T; Seite 104
$\mathcal{ECT}^{\mathrm{i}}[\![IS]\!](c_0)\lfloor_T$	Erweiterte Interleaving Casetrace-Semantik von I–System IS mit Sicht auf Bereichsmenge T bzgl. Start-Case c_0; Seite 104
$first(tr)$	erstes Element einer Trace tr; Seite 34
$GZustand(IS)$	Menge aller globalen Aktivitätszustände von I–System IS; Seite 15
$GZustand(IS)\lfloor_T$	Menge aller globalen Aktivitätszustände von I–System IS mit Sicht auf Bereichsmenge T; Seite 100
$IS\lfloor_M$	Teilsystem des I–Systems IS; Seite 12
$IS_1 \sim_{S,T} IS_2$	Trace-Äquivalenz von I–Systemen IS_1 und IS_2 bzgl. Semantiktyp S und Bereichsmenge T; Seite 145
$IS_1 \sim_{G,T} IS_2$	Graph-Äquivalenz von I–Systemen IS_1 und IS_2 bzgl. Graphentyp G und Bereichsmenge T; Seite 145
$ISystem$	Menge aller I–Systeme; Seite 10

$Kontroll(IS)$	Menge aller Kontrollbereiche von I–System IS; Seite 99
$last(tr)$	letztes Element einer Trace tr; Seite 34
$LCSystem$	Menge aller Lose Gekoppelten Systeme; Seite 94
$LZustand(b)$	Menge aller lokalen Aktivitätszustände von Bereich b; Seite 15
$Pre(p)$	Menge der Vorgängerknoten von Knoten p in einem gerichteten Graphen; Seite 177
$R(p)$, $R^{-1}(p)$	Nach- und Vorbereich der Relation R bzgl. Phase p; Seite 11
$R\langle b_1, b_2\rangle$	Schnittmenge von Relation R und Kreuzprodukt der Bereiche b_1 und b_2; Seite 11
$Relevanz(IS)$	Menge aller Relevanzbereiche von I–System IS; Seite 99
$RelGZustand(IS)$	Menge aller relevanten globalen Aktivitätszustände von I–System IS; Seite 29
$RelGZustand(IS)\lfloor_T$	Menge aller relevanten globalen Aktivitätszustände von I–System IS mit Sicht auf Bereichsmenge T; Seite 100
$StabGZustand(IS)$	Menge aller stabilen globalen Aktivitätszustände von I–System IS; Seite 15
$StabLZustand(b)$	Menge aller stabilen lokalen Aktivitätszustände von Bereich b; Seite 15
$Suc(p)$	Menge der Nachfolgeknoten von Knoten p in einem gerichteten Graphen; Seite 177
V_b	Komponente eines $\mathrm{V_I}$Systems; Seite 18
$VG(IS)$	Verhaltensgraph von I–System IS; Seite 74
$VG(IS)\lfloor_T$	Verhaltensgraph von I–System IS mit Sicht auf Bereichsmenge T; Seite 105
$\mathcal{V}[\![IS]\!]$	Verhalten von I–System IS; Seite 35
$\mathcal{V}[\![IS]\!](z_0)$	Verhalten von I–System IS bzgl. Start-Aktivitätszustand z_0; Seite 35
$\mathcal{V}[\![IS]\!]\lfloor_T$	Verhalten von I–System IS mit Sicht auf Bereichsmenge T; Seite 100
$\mathcal{V}[\![IS]\!](z_0)\lfloor_T$	Verhalten von I–System IS mit Sicht auf Bereichsmenge T bzgl. Start-Aktivitätszustand z_0; Seite 100
$\mathcal{V}^{\mathrm{i}}[\![IS]\!]$	Interleaving Verhalten von I–System IS; Seite 59
$\mathcal{V}^{\mathrm{i}}[\![IS]\!](z_0)$	Interleaving Verhalten von I–System IS bzgl. Start-Aktivitätszustand z_0; Seite 59
$\mathcal{V}^{\mathrm{i}}[\![IS]\!]\lfloor_T$	Interleaving Verhalten von I–System IS mit Sicht auf Bereichsmenge T; Seite 104
$\mathcal{V}^{\mathrm{i}}[\![IS]\!](z_0)\lfloor_T$	Interleaving Verhalten von I–System IS mit Sicht auf Bereichsmenge T bzgl. Start-Aktivitätszustand z_0; Seite 104
$V_ISystem(IS)$	$\mathrm{V_I}$System, das dem I–System IS zugeordnet ist; Seite 18
$z\|_b$	Abbildung z, Definitionsbereich beschränkt auf b; Seite 15
$z\lfloor_T$	globaler Aktivitätszustand z mit Sicht auf Bereichsmenge T; Seite 100
$z\langle b\rangle$	lokaler Aktivitätszustand von Bereich b; Seite 14
$zc(z)$	Case zu globalem Aktivitätszustand z; Seite 16

Literaturverzeichnis

[1] R. Alur, L. de Alfaro, T. A. Henzinger und F. Y. C. Mang. Automating modular verification. CONCUR'99, LNCS 1664, 82–97. Springer, 1999.

[2] R. Alur und T. A. Henzinger. Reactive modules. Formal Methods in System Design, Vol. 15, 7–48. Kluwer Academic Publishers, 1999.

[3] D. J. Andrews, J. F. Groote und C. A. Middelburg (Hrsg.). Semantics of Specification Languages. Springer, 1994.

[4] K. R. Apt und E.-R. Olderog. Programmverifikation. Springer, 1994.

[5] E. Astesiano, A. Giovini und G. Reggio. Processes and data types: observational semantics and logic. Semantics of Systems of Concurrent Processes, LNCS 469, 1–20. Springer, 1990.

[6] J. W. Bakker, W.-P. de Roever und G. Rozenberg (Hrsg.). Semantics: Foundations and Applications, LNCS 666. Springer, 1993.

[7] K. S. Barber. Dynamic adaptive autonomy in agent-based systems. Fourth International Symposium on Autonomous Decentralized Systems (ISADS 99). IEEE Computer Society, 1999.

[8] G. Bengel. Verteilte Systeme. Vieweg, 2000.

[9] B. Berard, M. Bidoit, A. Finkel, F. Laroussinie, A. Petit, L. Petrucci und P. Schnoebelen. Systems and Software Verification, Model-Checking Techniques and Tools. Springer, 2001.

[10] E. Best. Semantik. Vieweg, 1995.

[11] B. Bloom und A. Meyer. Experimenting with process equivalence. Semantics for Concurrency, Workshops in Computing, 81–95. Springer, 1990.

[12] S. Bornot, G. Gössler und J. Sifakis. On the construction of live timed systems. TACAS/ETAPS 2000, LNCS 1785, 109–126. Springer, 2000.

[13] B. Boschmann. Zur Verhaltensanalyse in Verteilten Systemen. Diplomarbeit, Universität Bonn, Dezember 1984.

[14] H.-D. Bremer. Anwendung des Modells der Interaktionssysteme bei der Planung in Großunternehmen. Diplomarbeit, Universität Bonn, Januar 1984.

[15] M. Broy. Compositional refinement of interactive systems modelled by relations. COMPOS'97, LNCS 1536, 130–149. Springer, 1998.

[16] R. Busch. Entwurf und Darstellung von Lösungsspielräumen bei Zuordnungsproblemen mit Hilfe der losen Kopplung. Dissertation, Rheinische Friedrich-Wilhelms-Universität zu Bonn, 1977.

[17] P. Caspi, A. Girault und D. Pilaud. Automatic distribution of reactive systems for asynchronous networks of processors. IEEE Transactions on Software Engineering, 25(3):416–427, 1999.

[18] G. Castelli, F. de Cindio, G. de Michelis und C. Simone. The GCP language and its implementation. Proc. of the IFIP workshop Language for Automation, New Orleans, Oktober 1984.

[19] K. M. Chandy und J. Misra. Parallel Program Design. Addison-Wesley, 1989.

[20] M. Charpentier und K. M. Chany. Towards a compositional approach to the design and verification of distributed Systems. FM'99, Vol. I, LNCS 1708, 570–589. Springer, 1999.

[21] F. Christian and C. Fetzer. Probabilistic internal clock synchronization. Thirteenth Symposium on Reliable Distributed Systems, Dana Point, Ca., 1994.

[22] E. M. Clarke, O. Grumberg und D. A. Peled. Model Checking. MIT Press, 1999.

[23] G. Coulouris, J. Dollimore und T. Kindberg. Distributed Systems: Concepts and Design. Addison-Wesley, 2001.

[24] R. J. Cypser. Communications for Cooperating Systems - OSI, SNA, and TCP/IP. Addison-Wesley, 1991.

[25] P. Darondeau. Concurrency and computability. Semantics of Systems of Concurrent Processes, LNCS 469, 223–238. Springer, 1990.

[26] W.-P. de Roever, F. de Boer, U. Hannemann und J. Hooman. Concurrency Verification: Introduction to Compositional and Noncompositional Methods. Cambridge University Press, 2001.

[27] V. Diekert und G. Rozenberg. The Book of Traces. World Scientific, 1995.

[28] J. Engelfriet. Determinacy $\rightarrow$ (observation equivalence $=$ trace equivalence). Theoretical Computer Science, 36:21–25, 1985.

[29] R. Fehling. Hierarchische Petrinetze: Beiträge zur Theorie und formale Basis für zugehörige Werkzeuge. Dissertation, Universität Dortmund, 1991.

[30] E. Fromentin und M. Raynal. Local states in distributed computations: a few relations and formulas. Operating Systems Review, 28(2):65–71, 1994.

[31] V. K. Garg. Principles of Distributed Systems. Kluwer Academic Publishers, 1996.

[32] R. J. van Glabbeek. The linear time - branching time spectrum. CONCUR '90. Theories of Concurrency: Unification and Extension, LNCS 458, 278–297. Springer, 1990.

[33] R. J. van Glabbeek und U. Goltz. Equivalences and refinement. Semantics of Systems of Concurrent Processes, LNCS 469, 309–333. Springer, 1990.

[34] P. Godefroid und P. Wolper. A partial approach to model checking. Information and Computation, 110(2):305–326, 1994.

[35] K. Goto, H. Matsubara und S. Myojo. A mobile guide system for visually disabled persons. Fourth International Symposium on Autonomous Decentralized Systems (ISADS 99). IEEE Computer Society, 1999.

[36] M. G. Gouda. Elements of Network Protocol Design. John Wiley and Sons, 1998.

[37] V. Gruhn und A. Thiel. Komponentenmodelle. Addison-Wesley, 2000.

[38] I. Guessarian (Hrsg.). Semantics of Systems of Concurrent Processes, LNCS 469. Springer, 1990.

[39] D. Harel. Statecharts: a visual formalism for complex systems. Science of Computer Programming, 8:231–274, 1987.

[40] G. T. Heineman und W. T. Councill. Component-Based Software Engineering. Addison-Wesley, 2001.

[41] C. Heitmeyer und D. Mandrioli (Hrsg.). Formal Methods for Real-Time Computing. Wiley and Sons, 1996.

[42] T. A. Henzinger und S. K. Rajamani. Fair bisimulation. Tools and Algorithms for the Construction and Analysis of Systems (ETAPS/TACAS'00), LNCS 1785, 299–314. Springer, 2000.

[43] R. G. Herrtwich und G. Hommel. Kooperation und Konkurrenz. Springer, 1989.

[44] C. A. R. Hoare. Communicating Sequential Processes. Prentice Hall, 1985.

[45] J.-E. Hong und D.-H. Bae. Incremental scenario modeling using Hierarchical Object-Oriented Petri Nets. International Journal of Software Engineering and Knowledge Engineering, 11(3):357–386, 2001.

[46] J. E. Hopcroft und J. D. Ullman. Einführung in die Automatentheorie, formale Sprachen und Komplexitätstheorie. Addison-Wesley, 1989.

[47] International Symposium on Autonomous Decentralized Systems (ISADS). IEEE Computer Society.

[48] V. Issarny, L. Bellissard, M. Riveill und A. Zarras. Component-based programming of distributed applications. Distributed Systems, LNCS 1752, 327–353. Springer, 2000.

[49] P. Jalote. Fault Tolerance in Distributed Systems. Prentice Hall, 1994.

[50] K. Jensen. Coloured Petri Nets – Analysis Methods. Springer, 1997.

[51] M. Joseph. Real-time Systems - Specification, Verification and Analysis. Prentice Hall, 1996.

[52] B. Kahlbrandt. Software-Engineering. Springer, 1998.

[53] S. Katz und D. Peled. Verification of distributed programs using representative interleaving sequences. Distributed Computing, 6:107–120, 1992.

[54] R. Khanna (Hrsg.). Distributed Computing: Implementation and Management Strategies. Prentice Hall, 1994.

[55] U. Kiencke. Ereignisdiskrete Systeme. Oldenbourg, 1997.

[56] S. Kostadinow. Eine Entwurfsmethode für verteilte digitale Systeme auf der Grundlage von Restriktionen. Dissertation, Technische Hochschule Karl-Marx-Stadt, Dezember 1985.

[57] N. Kraft. Zum Aufbau einer Theorie verteilter Systeme auf der Interaktion der Komponenten. Diplomarbeit, Universität Bonn, August 1980.

[58] M. Z. Kwiatkowska, M. W. Shields und R. M. Thomas (Hrsg.). Semantics for Concurrency. Workshops in Computing. Springer, 1990.

[59] C. Lakos und J. Lamp. The incremental modeling of the Z39.50 protocol with Object Petri Nets. Application of Petri Nets to Communication Networks, LNCS 1605, 37–68. Springer, 1999.

[60] L. Lamport. Time, clocks, and the ordering of events in a distributed system. Communications of the ACM, 21(7):558–565.

[61] L. Lamport. The Temporal Logic of Actions. ACM Trans Program Lang Syst, 16(3):872–923, 1994.

[62] L. Lamport. Fairness and hyperfairness. Distributed Computing, 13:239–245, 2000.

[63] M. Lin, J. Malec und S. Nadjm-Tehrani. On semantics and correctness of reactive rule-based programs. PSI'99, LNCS 1755, 235–246. Springer, 2000.

[64] N. A. Lynch. Distributed Algorithms. Morgan Kaufmann, 1996.

[65] A. Maggiolo-Schettini, H. F. Wedde und J. Winkowski. Modeling a solution for a control problem in distributed systems by restrictions. Theoretical Computer Science, 13:61–83, 1981.

[66] Z. Manna und A. Pnueli. The Temporal Logic of Reactive and Concurrent Systems. Springer, 1992.

[67] R. Milner. A Calculus of Communicating Systems, LNCS 92. Springer, 1980.

[68] R. H. B. Netzer und J. Xu. Necessary and sufficient conditions for consistent global snapshots. IEEE Transactions on Parallel and Distributed Systems, 6(2):165–169, 1995.

[69] F. Nielson, H. R. Nielson und C. Hankin. Principles of Program Analysis. Springer, 1999.

[70] E.-R. Olderog. Nets, Terms and Formulas. Cambridge University Press, 1991.

[71] D. Park. Concurrency and automata on infinite sequences. GI Conference on Theoretical Comuter Science, LNCS 104. Springer, 1981.

[72] C. A. Petri. Kommunikation mit Automaten. Dissertation, Institut für Instrumentelle Mathematik, Universität Bonn, 1962.

[73] D. K. Pradhan. Fault-Tolerant Computer System Design. Prentice Hall, 1996.

[74] J. Raasch. Systementwicklung mit strukturierten Methoden. Hanser, 1992.

[75] W. Reisig. Elements of Distributed Algorithms. Springer, 1998.

[76] G. Rozenberg und A. Salomaa (Hrsg.). Handbook of Formal Languages, Vol 1. Springer, 1997.

[77] P. Sander, W. Stucky und R. Herschel. Automaten, Sprachen, Berechenbarkeit. Teubner Stuttgart, 1995.

[78] R. Schwarz und F. Mattern. Detecting causal relationships in distributed computations: in search of the holy grail. Distributed Computing, 7:149–174, 1994.

[79] J.-O. P. Siepmann. Entwurf und Implementierung der Kommunikation im Dragon Slayer II System. Diplomarbeit, Universität Dortmund, März 1998.

[80] P. H. Starke. Analyse von Petri-Netz Modellen. Teubner Stuttgart, 1990.

[81] C. Stirling. Modal and Temporal Properties of Processes. Springer, 2001.

[82] D. J. Walker. Bisimulation and divergence. Information and Computation, 85:202–241, 1990.

[83] H. F. Wedde. Lose Kopplung von Systemkomponenten. Berichte der GMD Nr. 96, Gesellschaft für Mathematik und Datenverarbeitung Bonn, 1975.

[84] H. F. Wedde. An iterative and starvation-free solution for a general class of distributed control problems based on interaction primitives. Theoretical Computer Science, 24:1–20, 1983.

[85] H. F. Wedde. Interaction systems: a theory of distributed systems and its practical relevance. Part II: application studies. Wayne State University, Detroit, 1993.

[86] H. F. Wedde und W. Freund. Harmonious internal clock synchronization. Euromicro Workshop on Real-Time Systems, 175–182, Stockholm, Schweden, 2000.

[87] H. F. Wedde und J.-O. Siepmann. A universal framework for managing metadata in the distributed Dragon Slayer system. Euromicro Workshop on Multimedia and Telecommunications. IEEE Computer Society Press, 2000.

[88] H. F. Wedde und A. Wedig. Explicit modeling of influences, and of their absence, in distributed systems. Technical Report 742, Universität Dortmund, März 2001. http://ls3-www.cs.uni-dortmund.de/IS/P/techrep742.ps.

[89] H. F. Wedde und A. Wedig. Explicit modeling of influences, and of their absence, in distributed systems. Tools and Algorithms for the Construction and Analysis of Systems (ETAPS/TACAS'02), LNCS 2280, 127–141. Springer, 2002.

[90] H. F. Wedde und E. Zannoni. Interaction systems: a theory of distributed systems and its practical relevance. Part I: theoretical foundations. Wayne State University, Detroit, 1993.

[91] A. Wedig. Modellierung verteilter Systeme mit I–Systemen - Ein Beispiel. Formale Beschreibungstechniken für verteilte Systeme, 10. GI/ITG Fachgespräch FBT2000, Lübeck. Shaker, Juni 2000.

[92] J. Winkowski. Protocols of accessing overlapping sets of resources. Information Processing Letters, 12(5), 1981.