

Lehrstuhl für Informatik VII
der Technischen Universität München

**Grundlagen strukturierten Spezifizierens
mit höheren Netzen**

Andreas Heise

Vollständiger Abdruck der von der Fakultät für Informatik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften (Dr. rer. nat.)

genehmigten Dissertation.

Vorsitzender: Univ.-Prof. Dr. Tobias Nipkow

Prüfer der Dissertation: 1. Univ.-Prof. Dr. Dr. hc. Wilfried Brauer

2. Univ.-Prof. Dr. Hartmut Ehrig,
Technische Universität Berlin

Die Dissertation wurde am 22.03.2001 bei der Technischen Universität München eingereicht und durch die Fakultät für Informatik am 03.07.2002 angenommen.

Abstrakt

Ziel dieser Arbeit ist es, die Grundlagen für ein modulares Spezifizieren nebenläufiger Systeme mit Petrinetzen zu legen und das modulare Analysieren von Systemeigenschaften zu ermöglichen. Dabei liegt der Schwerpunkt auf der Entwicklung der dazu notwendigen Theorie, deren Praxisbezug aber stets erkennbar bleibt. Mit den Algebraischen High-Level-Netzen erweitern wir eine der für praktische Anwendungen geeignetesten Petrinetzklassen und bringen sie, durch die Einführung von Kompositions- und Parameterisierungsoperationen, dem Einsatz im Bereich nebenläufiger Systeme einen wesentlichen Schritt näher. Den einheitlichen Rahmen für Spezifikationen und deren Semantik bilden Konstruktionen der Kategorientheorie.

Grundlage für das Verstehen und Analysieren von Systemen ist die Unterscheidung zwischen Spezifikationen als abstrakte Beschreibungen notwendiger Eigenschaften und deren Konkretisierung in Form von Modellen. Dabei sollten alle spezifizierten Eigenschaften in den Modellen nachweisbar und alle implementierungsbedingten Anpassungen möglich sein. Im Unterschied zu der in der Literatur üblichen Modellbildung, die zwar eine Variation des Datentypanteils in Form verschiedener Algebren zuläßt, aber die Netzstruktur unverändert in jedes Modell übernimmt, interpretieren wir beide Teile einer Systemspezifikation lose. Für das übliche asymmetrische Vorgehen gibt es in der Literatur weder inhaltliche noch anwendungsbezogene Begründungen.

Ob eine Spezifikation das gewünschte Verhalten beschreibt, ist nur mit Hilfe von Modellen nachweisbar, die neben den spezifizierten Eigenschaften über keine weiteren verfügen. Ein derartiges initiales Modell ist im Rahmen unseres Ansatzes für jede Spezifikation konstruierbar. Um eine detaillierte Beschreibung der in einem System möglichen Abläufe zu erhalten, ordnen wir jedem Spezifikationsmodell eine operationale Semantik in Form von Prozeßtermen zu. Diese kompositionale true concurrency Semantik stellt das für die Petrinetztheorie essentielle Verhältnis von Nichtdeterminismus und Nebenläufigkeit in den Vordergrund und läßt sich direkt in die übliche Prozeßdarstellung übertragen.

Systeme in der Praxis sind meist deutlich komplexer als die im Bereich der Petrinetztheorie verwendeten Beispiele. Sie erfordern daher, neben einer eindeutigen Spezifikation, Möglichkeiten zur deren strukturierter Erstellung. Die hier eingeführten Kompositions- und Parameterisierungsoperationen, mit ihren bis auf Isomorphie eindeutigen Ergebnissen, schaffen dafür die Grundlagen. Ihnen stehen entsprechende Eigenschaften auf Seiten der Modell- und Prozeßtermsemantik gegenüber, ohne die ihr Nutzen für die Praxis stark eingeschränkt wäre.

Danksagungen

Eine Reihe von Personen haben, jede auf ihre Weise, zum Gelingen dieser Arbeit beigetragen. Zu aller erst gilt mein ganz besonderer Dank meiner Lebensgefährtin, die mich mit Liebe, Geduld und großem Engagement über die gesamte Entstehungszeit, durch alle Höhen und Tiefen dieser Arbeit begleitet hat und damit zurecht kam, daß ich oft nicht erreichbar war. Ganz besonderen Dank auch an meine beiden Töchter Lara und Lea für Ihre bedingungslose Zuneigung und dafür, daß Sie mich immer wieder in Ihre Welt entführt haben.

Des weiteren gilt mein spezieller Dank Prof. Dr. Wilfried Brauer. Die freundschaftliche und konstruktive Betreuung sowie seine nie endende Geduld und der ansteckende Optimismus, waren eine große Hilfe bei der Überwindung auftauchender Probleme. Prof. Dr. Hartmut Ehrig danke ich für den Anstoß zu dieser Arbeit und für seine kritischen Kommentare, die mir vorhandene Mängel aufgezeigt und mich gezwungen haben, Motivationen und Ergebnisse klar herauszuarbeiten.

Für die freundschaftliche Aufnahme an der Universität Tübingen, viele fruchtbare Diskussionen und freundschaftliche Gespräche, bin ich Prof. Dr. Klaus Jörn Lange zu großem Dank verpflichtet. Die Art und Weise, in der mir Unterstützung von ihm und seiner Gruppe (Markus, Klaus, Henning u. Muriel) zu Teil wurde, hat mich außerordentlich beeindruckt.

Besonderen Dank an Markus Holzer. Seine Anteilnahme, die vielen ergiebigen Gespräche, das Finden des verwendeten Diagramm-Formats, die Einführung in das Grafiksystem „MetaPost“ nebst Erstellung grundlegender Makros und vor allem unsere Freundschaft, haben einen unschätzbaren Beitrag zur Entstehung des vorliegenden Werkes geleistet.

Darüber hinaus danke ich den vielen Menschen, deren unerschütterlicher Glaube an diese Arbeit und ihren, nicht immer vom Erfolg verwöhntem Autor, mich stets aufs neue motiviert hat.

Inhaltsverzeichnis

Abstrakt	iii
Einleitung	vii
1 P/T-Netze mit individuellen Marken	1
1.1 CP/T-Netze und -Morphismen	2
1.2 CP/T-Netz-Komposition und -Zerlegung	8
1.3 CP/T-Netz-Semantik	12
2 Systemspezifikationen	23
2.1 Algebraische Netzspezifikationen	24
2.2 Algebraische Netze	30
2.3 Operationale A-Netz-Semantik	38
3 Operationen auf Spezifikationen	45
3.1 Systemspezifikationsmorphismen	46
3.2 Generalisierte Modellmorphismen	54
3.3 Komposition von Systemspezifikationen	60
3.4 Parameterisierung von Systemspezifikationen	74
4 Spezifikation eines „FFS“	81
4.1 Bearbeitungszentren	83
4.2 Komposition des FFS	91
4.3 Parameterisiertes FFS	105

5	Kategorielle Konstruktionen	113
5.1	Grundlagen	114
5.2	Freie abelsche Gruppen	125
5.3	Spezifikationen und Modelle	142
6	Datentypspezifikationen	147
6.1	Signaturen	148
6.2	Spezifikationen	157
6.3	Spezifikationen mit Variablen	172
	Zusammenfassung und Ausblick	181
	Anhang	185
	Beweise zu Kapitel 1	185
	Beweise zu Kapitel 2	197
	Beweise zu Kapitel 3	200
	Beweise zu Kapitel 5	256
	Beweise zu Kapitel 6	284
	Literaturverzeichnis	307

Einleitung

Motivation und Zielsetzung

Als die Petrinetztheorie in den 60er Jahren von C.A. Petri [Pet62] begründet wurde, standen vor allem die grundlegenden Aspekte der Beschreibung von Verteiltheit und Nebenläufigkeit im Vordergrund; der Einsatz dieser Methode bei der Spezifikation realer und damit meist umfangreicher Systeme war nicht aktuell. Diese Situation hat sich seit dem grundlegend gewandelt. Zum einen hat die vorhandene Hardware ein Leistungsniveau erreicht, das eine rechnergestützte, detaillierte Beschreibung und Simulation derartiger Systeme ermöglicht, zum anderen gewinnen Netzwerke und verteilte Rechnerarchitekturen immer mehr an Bedeutung. Diese Entwicklungen haben dazu geführt, daß Verteiltheit und Nebenläufigkeit in der Praxis eine immer größere Rolle spielen und damit auch die Methoden, diese Aspekte zu modellieren. Dabei stößt die Beschreibung komplexer Systeme allein mit den Mitteln der klassischen Petrinetztheorie schnell an ihre Grenzen. Die entstehenden Netze werden unüberschaubar.

Inwieweit sich die Petrinetztheorie als Spezifikationsmethode in der Praxis neben den prozeßalgebraischen Spezifikationssprachen, wie CCS ([Mil80]), CSP ([BHR84]) und LOTOS ([Int87]) bewähren kann, ist nicht zuletzt von ihrer Handhabbarkeit bei der Spezifikation realer und damit in der Regel komplexer Systeme abhängig. Ein wesentlicher Aspekt dabei ist die Möglichkeit zu einer modularen Vorgehensweise, die es erlaubt, ein System aus überschaubaren, separat spezifizierten Subsystemen zusammenzusetzen und die Eigenschaften des komplexen Systems aus den Eigenschaften der Subsysteme abzuleiten.

Im Gegensatz zu CCS, CSP und LOTOS, die Systeme mit Hilfe von „atomaren“ Aktionen „bottom-up“ zusammensetzen und damit prinzipiell Grundzüge von Kompositionalität aufweisen, muß diese den petrinetzbasierten Ansätzen hinzugefügt werden. Entsprechende Mechanismen existieren zumeist nur für „höhere Netze“, d.h. für Netze mit individuellen Token. Doch auch für diese Netzklasse, die ja vor allem unter Anwendungsgesichtspunkten entwickelt wurde, werden Kompositionsoperationen oft nur am Rande betrachtet oder durch technische Einschränkungen in ihrem Strukturierungspotential gemindert. So können Netze häufig nur über gleichnamige Netzknoten verknüpft werden ([Jen92, Jen94], [BDCM88, BCML91], [BG91]), was

unter Umständen künstliche Systemabgrenzungen zur Folge hat.¹ In anderen Fällen muß entweder der Datentyp- oder der Netzanteil fix sein ([Lil91]), oder die zugrundeliegenden Netze sind von vornherein in ihrer Struktur beschränkt, z.B. auf „*state machines*“ ([BDCM88, BCML91]). Ein weiteres Manko vieler Formalismen, die eine Komposition von Netzen ermöglichen, ist das Fehlen einer Semantik der Kompositionsoperationen ([DHP91], [Lil91, Lil95]), ohne die aber weder Eigenschaften von den Komponenten auf das Ergebnis, noch in umgekehrter Richtung übertragen werden können.

Zwei Klassen von Eigenschaften lassen sich allen Systemen zuordnen, einmal die notwendigen Eigenschaften, ohne die sie ihre Aufgabe nicht erfüllen, und Eigenschaften, die durch ihre Konkretisierung entweder als mathematisches-, als reales- oder als implementiertes Objekt entstehen. Diese für das Verstehen und Analysieren von Systemen wichtige Unterscheidung macht eine Trennung zwischen Spezifikation als abstrakte Beschreibung der notwendigen Eigenschaften und deren Konkretisierung in Form von Modellen notwendig. Dabei sollten alle spezifizierten Eigenschaften in den Modellen nachweisbar und alle implementierungsbedingten Änderungen möglich sein. Vorbild in dieser Hinsicht ist die Theorie abstrakter Datentypen mit loser Semantik. Vergleichbare Ansätze im Bereich der Petrinetztheorie, bei denen die angesprochene Differenzierung für alle Systemaspekte möglich ist, d.h. sowohl für die Ereignisabfolge als auch für die daraus entstehenden Zustände, gibt es bisher nicht.

Viele Systeme wie Maschinen, Kommunikationseinrichtungen oder Prozeßsteuerungen sind von vornherein dafür vorgesehen, mit leichten Modifikationen in unterschiedlichen Umgebungen zu funktionieren. Um sie nicht in jedem Kontext neu beschreiben und als korrekt beweisen zu müssen, braucht man Mechanismen, mit deren Hilfe umgebungsabhängige Spezifikationsteile ausgezeichnet und ersetzt werden können. Dabei müssen sich die Effekte möglicher Anpassungen in die Analyse der Systemeigenschaften einbeziehen lassen. Diese Anforderungen werden im Rahmen der Spezifikation sequentieller Systeme, z.B. in der Theorie abstrakter Datentypen, durch „*Parameterisierung*“ und die „*Parameterübergabe*“ erfüllt. Für petrinetzbasierte Spezifikationsansätze fehlen derartige Mechanismen bisher.

Ziel dieser Arbeit ist es, die Grundlagen für ein modulares Spezifizieren nebenläufiger Systeme mit Petrinetzen zu legen und das modulare Analysieren von Systemeigenschaften zu ermöglichen. Dabei liegt der Schwerpunkt auf der Entwicklung der dazu notwendigen Theorie, deren Praxisbezug aber erkennbar bleibt, d.h. am Ende steht kein fertiges, am ingenieurmäßigen Einsatz orientiertes Spezifikationswerkzeug, es werden aber auch keine praxisfernen Einschränkungen verwendet. Die vorgestellten Objekte und Operationen — Systemspezifikationen mit loser Semantik, Komposition, Parameterisierung und Parameterübergabe mit entsprechender Semantik — lassen sich direkt anwenden oder in bestehende Petrinetzwerkzeuge integrieren.

¹z.B. in Fällen, in denen die Komponenten über Teilsysteme zusammenhängen und ihr Verhalten erst in Zusammenhang mit diesen Systemen verstanden werden kann

Als Orientierung dienen uns die Konzepte zum modularen Aufbau komplexer Systeme, die sich im Bereich der algebraischen Datentypspezifikationen ([EM85, EM90]) bewährt haben. Sie sind ausreichend abstrakt und mächtig, um auf verschiedene Objekte, z.B. Netze, Daten und Prozesse, übertragbar zu sein, und stellen eine breite Palette von Ergebnissen und Methoden (Parameterisierung, Module, Spezifikationslogik) zur Verfügung.

Einbettung und Abgrenzung

Der Anspruch, größere Systeme auf Petrinetzbasis strukturiert zu beschreiben, führt fast zwangsläufig zur Verwendung „höherer Petrinetze“. Bei diesem Netztyp werden die Objekte eines Systems, z.B. Daten, Werkzeuge oder Prozessoren, separat spezifiziert und nicht in die Graphstruktur des Netzes kodiert. Die daraus entstehenden Systembeschreibungen sind, gegenüber den klassischen Petrinetzen, kompakter und verständlicher, da Objekte und Objektfluß mit den dafür jeweils am besten geeigneten Mitteln beschrieben werden können.

Bei den höheren Petrinetzen lassen sich zwei Klassen unterscheiden; zum einen Netze, denen sich ohne weitere Interpretation ein operationales Verhalten zuordnen läßt; zum anderen abstrakte Petrinetze, bei denen erst die zugeordneten Modelle ein Verhalten haben. Der namhafteste Vertreter der ersten Klasse sind die „*Coloured Nets*“ von Kurt Jensen, bei denen die Datentypen über Mengen und Funktionen ([Jen81]) bzw. über programmiersprachliche Konstrukte ([Jen90]) beschrieben werden. Die dieser Klasse fehlende Differenzierung zwischen Spezifikationen als abstrakten Beschreibungen notwendiger Eigenschaften und deren Konkretisierungen macht eine Abgrenzung von Systemeigenschaften und Implementierungsdetails unmöglich.

Die Kombination aus abstrakter Datentypspezifikation zur Beschreibung der Objekteigenschaften und abstraktem Petrinetz zur Beschreibung von Konflikten, kausalen Zusammenhängen und Nebenläufigkeit, kennzeichnet die zweite Klasse von höheren Petrinetzen. Sie ist Gegenstand einer Reihe von Arbeiten ([Vau86], [Hei88], [DHP90], [Rei91], [Lil91, Lil95], [Pad96]) und bildet die Grundlage der hier vorgestellten Systemspezifikationen.

Im Unterschied zu der in der Literatur üblichen Modellbildung, die eine Variation des Datentypanteils in Form verschiedener Algebren zuläßt, aber die Netzstruktur unverändert in jedes Modell übernimmt, interpretieren wir beide Teile einer Systemspezifikation „*lose*“. Das heißt, auch die Netzstruktur unterliegt einer kontrollierten Modifikation beim Übergang von der Spezifikation zu den Modellen. Dabei bleiben die essentiellen Systemeigenschaften erhalten. Aspekte, die aus der Notwendigkeit entstanden sind, die Spezifikation in eine syntaktische Form zu bringen oder die in irgendeiner Form implementationsabhängig sind, können dagegen modifiziert oder ergänzt werden.

Die Gleichbehandlung beider Aspekte einer Systemspezifikation bei der Konstruktion ihrer Semantik entspricht ihrer gleichgewichtigen Bedeutung und ist Voraussetzung

für Vergleiche von Spezifikationen mit bereits vorhandenen Systemen, bei denen die Repräsentation von Daten und Nebenläufigkeit vom initialen Modell abweicht. Für das übliche asymmetrische Vorgehen gibt es dagegen in der Literatur weder inhaltliche noch anwendungsbezogene Begründungen.

Die im Rahmen unserer Modellbildung zugelassenen Modelle einer Systemspezifikation bilden die Objekte einer Modellkategorie, die mit dem initialen ein bis auf Isomorphie eindeutiges, „*minimales Modell*“ enthält, das genau das spezifizierte Systemverhalten aufweist. Darüber hinaus erlaubt eine Generalisierung der Modellmorphisamen mit Hilfe eigenschaftserhaltender Abbildungen zwischen den Spezifikationen eine Strukturierung der Menge aller Modelle innerhalb einer globalen Modellkategorie. In Darstellung und Funktionalität sind die Modelle mit den Coloured Nets von Kurt Jensen vergleichbar, bieten aber durch die formale Beschreibung der verwendeten Datentypen entscheidende Vorteile beim Beweisen von Systemeigenschaften.

Um eine detaillierte Beschreibung der in einem System möglichen Abläufe zu erhalten, ordnen wir jedem Spezifikationsmodell eine operationale Semantik in Form von Prozeßtermen zu. Dabei steht jeder Prozeßterm für die Äquivalenzklasse von Prozessen im herkömmlichen Sinn (\rightarrow [Rei85]), die sich aus der Abstraktion von der Reihenfolge ergibt, in der mit ununterscheidbaren Token verfahren wird. Eine derartige „*true concurrency*“ Semantik war, mit Ausnahme der „*strikt monoidalen Petrinetzkatogorien*“ in [Lil95], für höhere Petrinetze bisher nicht verfügbar. Im Unterschied zu den strikt monoidalen Petrinetzkatogorien, deren grundlegend andere Konstruktion auf rein abstrakt algebraischer Ebene erfolgt, stellt unsere Semantikonstruktion das für die Petrinetztheorie essentielle Verhältnis von Nichtdeterminismus und Nebenläufigkeit in den Vordergrund.

Die Verwendung höherer Petrinetze allein reicht zur Beschreibung komplexer Systeme nicht aus. Die vorliegende Arbeit sieht daher zwei Operationen vor, die ein strukturiertes Erstellen großer Spezifikationen unterstützen. Zum einen die Komposition, mit deren Hilfe sich Spezifikationen über Schnittstellen² zusammenfügen lassen und zum anderen die Parameterisierung zusammen mit der Parameteraktualisierung, die eine Anpassung an unterschiedliche Kontexte erlaubt. Beide Operationen haben ihre Entsprechungen auf der Semantikebene.

Kompositionen in der von uns verwendeten Form stellen, aufgrund der allgemeineren Morphismen, eine Erweiterung der „*Unions*“ aus [Pad96] und der entsprechenden Diagramme aus [Lil95] dar, deren Nutzen anhand von praxisnahen Beispielen (\rightarrow Kapitel 4) belegt wird. Der wesentlichere Unterschied liegt jedoch darin, daß hier der Komposition als Operation eine Semantik zugeordnet wird. Mit deren Hilfe läßt sich jedes Modell und jeder Modellmorphismus einer komponierten Spezifikation eindeutig auf Modelle bzw. Modellmorphisamen der Komponenten projizieren und als Komposition dieser Projektionen darstellen. Umgekehrt existiert zu jeder Auswahl aus miteinander verträglichen Komponentmodellen ein entsprechendes Modell

²im Unterschied zu [Jen92, Jen94], [BDCM88, BCML91], [BG91] handelt es sich dabei um komplette Spezifikationen

der komponierten Spezifikation. Dessen strukturelle Eigenschaften und operationales Verhalten sind aus den Eigenschaften bzw. dem Verhalten der Komponentmodelle berechenbar.

Parameterisierung und Parameteraktualisierung orientieren sich in an entsprechenden Konstruktionen aus dem Bereich algebraischer Datentypspezifikationen (\rightarrow [EM85]). Vergleichbare Ansätze für Petrinetze fehlten bisher, obwohl sie zumindest bei den Formalismen, die neben den Daten auch die Netzstruktur in eine algebraische Spezifikation abbilden, z.B. [BDCM88, BCML91] oder [Kra87, Sch88], prinzipiell naheliegend wären.

Auf der Modellebene wird die Parameterisierung durch eine Semantik ergänzt, die den Effekt aller möglichen Aktualisierungen des Parameters wiedergibt. Dazu wird jeder parameterisierten Spezifikation ein Funktor zwischen der Kategorie der Parametermodelle, die insbesondere die relevanten Teile der initialen Modelle zukünftiger Aktualisierungen beinhaltet, und der Modellkategorie der Gesamtspezifikation zugeordnet. Diese Abbildung ist bezüglich der Projektion der Spezifikations- auf die Parametermodelle „frei konstruiert“. Das schließt Nebeneffekte der Aktualisierung auf den Spezifikationsrumpf aus und erlaubt es, Eigenschaften zu bestimmen, die unter allen Aktualisierungen erhalten bleiben.

Gliederung

Die vorliegende Arbeit gliedert sich in zwei Teile. Ihren Kern bildet der erste Teil mit den Kapiteln 1–4, in denen Systemspezifikationen und Modelle sowie Operationen zu deren Komposition und Parameterisierung eingeführt werden. Der zweite Teil schafft die theoretischen Grundlagen in den Bereichen Kategorientheorie, freie abelsche Gruppen und algebraische Spezifikationen. Trotz der inhaltlichen Bezüge zwischen erstem und zweiten Teil ist es möglich und beabsichtigt sich allein mit Hilfe des ersten Teils einen Überblick über die wesentlichen Inhalte dieser Arbeit zu verschaffen. Alternativ zu dieser und der normalen sequentiellen Leseweise, ist es aber auch möglich mit dem zweiten Teil zu beginnen und die Arbeit damit „*bottom up*“ zu lesen.

Mit den „*Coloured-Place/Transition-Nets*“ (kurz CP/T-Netze), einer Erweiterung der Stellen/Transitions-Netze (\rightarrow [Rei85]) um individuelle Marken, legen wir im ersten Kapitel die Grundlagen für die Beschreibung der Nebenläufigkeitsaspekte in Systemspezifikationen und deren Modellen. Die Menge der CP/T-Netze wird durch Abbildungen (CP/T-Netzmorphismen) strukturiert, die sowohl die Eigenschaften der Marken als auch die Abhängigkeiten und Konflikte der möglichen Ereignisse bewahren. Sie sind ein wesentlicher Bestandteil von Diagrammen, über die CP/T-Netze komponiert werden können. Als Semantik ordnen wir CP/T-Netzen und Morphismen Prozeßterme bzw. strukturerhaltenden Abbildungen zwischen diesen Termen zu.

In Kapitel 2 führen wir „*AN-Spezifikationen*“, bestehend aus einer algebraischen Spezifikation der vorhandenen Objekte und einer Beschreibung des Objektflusses in Form

eines CP/T-Netzes, als Spezifikation nebenläufiger Systeme ein. Die Modelle dieser Systemspezifikationen entstehen durch Interpretation des Datentypanteils in Form einer Algebra und durch Interpretation des Netzanteils durch ein CP/T-Netz mit der Algebra als Markenbereich. Anstelle von Funktionen bei Coloured Nets stehen Terme mit Variablen über der Algebra an den Kanten der Modelle. Aus den Belegungen dieser Variablen mit Werten der Algebra ergeben sich die Schaltmodi und Schalteffekte des Modells. Mit Hilfe der Schaltmodi läßt sich das Modell in ein CP/T-Netz entfalten, dessen operationale Semantik (Prozeßterme) das Modellverhalten beschreibt.

Inhalt von Kapitel 3 sind Operationen zum strukturierten Erzeugen von Systemspezifikationen und deren Semantik. Dazu werden zunächst eigenschaftserhaltende Abbildungen (Spezifikationsmorphismen) zwischen Systemspezifikationen eingeführt. Als deren Semantik definieren wir „*Vergißfunktoren*“ zwischen den Modellkategorien von Bild und Urbild des jeweiligen Spezifikationsmorphismus. Mit den anschließend eingeführten Kompositionen lassen sich Systemspezifikationen über Schnittstellen zusammenfügen. Grundlage dafür sind „*Kompositionsdiagramme*“ in der Kategorie der Systemspezifikationen und deren Colimites. Über die Modellsemantik der Komposition lassen sich alle Modelle und Modellmorphismen komponierter Spezifikationen als Komposition von Modellen und Modellmorphismen der Komponentenspezifikationen darstellen.

Ein weiteres Mittel zur Strukturierung komplexer Spezifikationen wird mit der Parameterisierung und der Parameteraktualisierung eingeführt. Dabei werden parameterisierte Spezifikationen durch Inklusionsmorphismen zwischen dem Parameter und dem Spezifikationsrumpf repräsentiert. Im Rahmen der Parameterersetzung, bei der ein passender aktueller Parameter durch Angabe eines vom Parameter ausgehenden Spezifikationsmorphismus ausgewählt wird, ergibt sich die aktualisierte Spezifikation als Komposition von Spezifikationsrumpf und aktuellem Parameter. Auf der Semantikebene wird die Parameterisierung durch die Zuordnung einer „*freien Konstruktion*“ von den Parametermodellen in die Rumpfmodelle ergänzt.

Anhand der schrittweisen Spezifikation eines „*Flexiblen Fertigungssystems*“ veranschaulichen wir in Kapitel 4 die Verwendung von Systemspezifikationen und der darauf definierten Operationen sowie deren Semantik. Diese aus Maschinen zur spanenden Bearbeitung von Metallteilen, automatischen Förderverbindungen und Spannplätzen als Schnittstelle zu den menschlichen Operateuren bestehenden komplexen Systeme werden unter anderem im Automobilbau eingesetzt und zeichnen sich durch ein hohes Maß an nebenläufig ausführbaren Aktionen aus.

Kapitel 5 bietet zunächst eine kurze Einführung in einige Aspekte der Kategorientheorie. Anschließend werden freie abelsche Gruppen und lineare Abbildungen, die wesentlichen Anteil an Definition und Eigenschaften von CP/T-Netzen haben, betrachtet. Zusätzliche Eigenschaften einiger linearer Abbildungen, wie „*Monotonie*“ oder ihre „*freie Erzeugbarkeit*“ aus Funktionen, zeichnen wichtige Unterkategorien innerhalb der Kategorie der freien abelschen Gruppen aus. Darüber hinaus werden Einschränkungen für Diagramme untersucht, die die Existenz von Colimites sicher-

stellen. Der letzte Abschnitt von Kapitel 5 befaßt sich mit dem Zusammenhang von Spezifikationen und Modellen auf kategorieller Grundlage.

Die Beschreibung, der in den Systemspezifikationen vorkommenden Datentypen, erfolgt durch algebraische Spezifikationen im Stil von H. Ehrig/B. Mahr [EM85]. Die im Rahmen dieser Arbeit wichtigen Definitionen aus diesem Bereich finden sich in Kapitel 6. Sie werden ergänzt durch die Definition von Spezifikationen und Algebren mit Variablen. Diese Erweiterungen entstehen durch hinzufügen einer Variablenmenge als zusätzliche Konstanten zu den Spezifikationen und durch Abbildung von deren Algebren mittels der freien Konstruktion, die vom Zusammenhang der ursprünglichen mit der erweiterten Spezifikation induziert wird.

P/T-Netze mit individuellen Marken

Bei der Beschreibung von Systemen unterscheiden wir zwei Aspekte; zum einen die Objekte, die während eines Ablaufs im System erzeugt und manipuliert werden, zum anderen die Konflikte und kausalen Zusammenhänge der Ereignisse, durch die diese Veränderungen bewirkt werden. In diesem Kapitel gehen wir davon aus, daß die Struktur der Objekte und die Operationen zu deren Manipulation in Form einer Algebra gegeben sind. Die Eigenschaften dieser Algebren werden in den folgenden Kapiteln durch algebraische Spezifikationen (\rightarrow Kapitel 6) festgelegt. Die Beschreibung der in einem System möglichen Ereignisse und ihrer Abhängigkeiten untereinander in Form von „*Colored-Place/Transition-Netzen*“ (kurz „*CP/T-Netze*“) sind Inhalt dieses Kapitels.

Die im weiteren eingeführten CP/T-Netze bilden eine Erweiterung der aus der klassischen Petrinetztheorie bekannten *Stellen/Transitions-Netze* (\rightarrow [Rei85]). Dabei treten Elemente aus den Trägermengen einer Algebra an die Stelle von ununterscheidbaren Marken. Wie bei den Stellen/Transitions-Netzen, die sich als Spezialfall der CP/T-Netze darstellen lassen, können beim Eintreten von Ereignissen mehrere identische Objekte von den betroffenen Stellen abgezogen bzw. ihnen hinzugefügt werden. Im Unterschied zu der wohl bekanntesten Erweiterung der Stellen/Transitions-Netze, den „*Colored Nets*“ von K. Jensen (\rightarrow [HJS90]), bei denen der Effekt eines Ereignisses von dem Modus¹ abhängt, in dem es eintritt, ist dieser Zusammenhang bei den CP/T-Netzen konstant. Die Menge aller CP/T-Netze wird durch Abbildungen, die „*CP/T-Netzmorphismen*“, strukturiert. Diese bewahren sowohl die in der Algebra festgelegten Eigenschaften der Marken, als auch die in der Netzstruktur kodierten Abhängigkeiten und Konflikte der möglichen Ereignisse.

CP/T-Netze und CP/T-Netzmorphismen werden in Abschnitt 1.1 definiert. Die Eigenschaft von CP/T-Netzmorphismen, in ihren Zielen vollständige Unternetze auszuzeichnen, wird in Abschnitt 1.2 zur Definition von CP/T-Netz-Kompositionen und -Zerlegungen genutzt. Im Fall der Komposition beschreiben die Unternetze Schnittstellen in den Komponenten, über denen die Verbindung erfolgt. Die Zerlegung separiert das vom Morphismus ausgezeichnete Unternetz von seiner Umgebung.

In Abschnitt 1.3 wird den CP/T-Netzen eine operationale Semantik in Form von Prozeßtermen zugeordnet. Dabei handelt es sich um eine algebraische Form der Prozeßsemantik für Petrinetze (\rightarrow [Rei85]), basierend auf der von P. Degano, J. Meseguer und U. Montanari in [DMM89] vorgeschlagenen Axiomatisierung von Abläufen auf Stellen/Transitions-Netzen. Sie stellt mit der sequentiellen - und der nebenläufigen

¹d.h. der entsprechenden Variablenbelegung

Komposition zwei Operationen zur Verfügung über die jeder Prozeßterm aus „atomaren“ Termen zusammengesetzt werden kann. Den CP/T-Netzmorphismen entsprechen auf Seiten der operationalen Semantik Abbildungen zwischen Prozeßtermen. Diese Abbildungen bewahren die Struktur der atomaren Terme und sind mit den Kompositionsoperatoren kompatibel.

1.1 CP/T-Netze und -Morphismen

Die Standardnotation für Petrinetze (\rightarrow [BF86], [Rei85]) ist direkt aus deren graphischer Repräsentation als bipartite, kantenbeschriftete Graphen abgeleitet. Diese intuitiv erscheinende Beschreibung hat einige Nachteile. Zum einen ist sie aus mathematischer Sicht schwerfällig, da sie mit Mengen, Relationen, Projektionen und Abbildungen von Relationen auf natürlichen Zahlen (bei Stellen/Transitions-Netzen) eine ganze Reihe mathematischer Objekte mit unterschiedlichen Eigenschaften verwendet, zum anderen muß der wesentliche Aspekt der Netze, ihr Verhalten, relativ umständlich aus der statischen Graphstruktur errechnet werden.

Unsere Notation für CP/T-Netze, die deren Verhalten in den Vordergrund stellt, basiert auf den Arbeiten von G. Winskel ([Win87]) und J. Meseguer/U. Montanari ([MM90]) über Stellen/Transitions-Netze. Dabei lassen sich mit nur einem Konzept, den freien abelschen Gruppen² und ihren linearen monotonen Abbildungen, die Netze, deren Zustände, das Schaltverhalten und die Netzabbildungen beschreiben. Zudem handelt es sich bei den freien abelschen Gruppen um eine gut untersuchte Struktur mit einer Vielzahl von Eigenschaften. Zwischen dieser Notation und der gewohnten Repräsentation der Netze als Graphen besteht ein eindeutiger Zusammenhang.

Um im Rahmen der CP/T-Netze mit ihren individuellen Marken über den Einfluß von Transitionen auf Markierungen reden zu können, müssen die Stellen im Netz mit dem Wertebereich der Marken verknüpft werden. Da im weiteren anstatt einzelner Stellen allgemeiner Stellensummen und damit Elemente der freien abelschen Gruppe über den Stellen betrachtet werden und die Individuenbereiche der Marken in Form von Algebren gegeben sind, verknüpft die nachfolgend definierte Abbildung „ \otimes “ freie abelsche Gruppen mit Algebren. Die aus dieser Verknüpfung resultierenden freien abelschen Gruppen liefern die Markierungen der CP/T-Netze.

Notation: Wir notieren die freie abelsche Gruppe über einer „Basis“ genannten Menge A (\rightarrow 5.2.1) durch A^{\otimes} , deren Elemente als formale Summen der Form

$$\bigoplus_{a \in A} x_a * a \quad \text{mit } x_a \in \mathbb{Z} \quad \text{für alle } a \in A$$

²In [Win87]/[MM90] werden anstelle von freien abelschen Gruppen, freie abelsche Monoide verwendet. Der Vorteil von Gruppen liegt darin, daß durch das Vorhandensein von Inversen, auch Summen mit negativen Einträgen behandelt werden können, wie sie z.B. bei Invarianten auftreten. Freie abelsche Gruppen werden in Abschnitt 5.2 definiert.

und die Kategorie der freien abelschen Gruppen mit monotonen Morphismen³ durch \mathcal{AG}_m (\rightarrow Abschnitt 5.2). Um auszudrücken, wie oft ein Basiselement a in einer Summe v aus A^\otimes enthalten ist (x_a), benutzen wir die Funktionsschreibweise $v(a)$.⁴

Wenn im weiteren von Algebren die Rede ist, beziehen wir uns immer auf mehrsortige Algebren, denen eine als „*Signatur*“ bezeichnete Struktur zugrunde liegt. Deren „*Sorten*“ legen die Trägermengen fest, während „*Operationssymbole*“ Wertigkeit und Zielmenge der Algebraoperationen beschreiben. Die Gesamtheit aller Algebren bildet zusammen mit den zwischen ihnen verkehrenden strukturerhaltenden Abbildungen, den „*generalisierten Algebromorphismen*“⁵, die Kategorie $\mathcal{GSTG-ALG}$. Als Modelle von Signaturen bzw. von Spezifikationen, werden Algebren in Kapitel 6 ausführlich behandelt. Für weiterführende Betrachtungen verweisen wir auf [EM85] und [EBO91].

1.1.1 Definition (Verbindung von Stellen und Algebren):

Sei \mathcal{AG}_m die Kategorie der freien abelschen Gruppen mit monotonen Morphismen, $\mathcal{GSTG-ALG}$ die Kategorie der Algebren mit generalisierten Morphismen und $\|_-\|$ der Vergißfunktorkontraktor von Algebren in Mengen (\rightarrow 6.1.14). Wir definieren die Abbildung

$$\otimes: \mathcal{AG}_m \times \mathcal{GSTG-ALG} \rightarrow \mathcal{AG}_m$$

für alle \mathcal{AG}_m -Morphismen $f: G_1^\otimes \rightarrow G_2^\otimes$ und generalisierten Algebromorphismen $k: A_1 \rightarrow A_2$ durch:

$$\begin{aligned} \otimes(f, k) &\stackrel{\text{def}}{=} f \otimes k: (G_1 \times \|A_1\|)^\otimes \rightarrow (G_2 \times \|A_2\|)^\otimes \\ &\text{mit } f \otimes k(g, a) \stackrel{\text{def}}{=} \bigoplus_{g' \in G_2} f(g)(g') * (g', k(a)) \quad \text{6} \end{aligned}$$

für alle G_1 -Elemente g und $\|A_1\|$ -Elemente a .⁷

□

Ein wesentliches Merkmal der aus Gruppen- und Algebromorphismen zusammengesetzten Morphismen in 1.1.1 ist die allein vom Gruppenmorphismus bestimmte additive Struktur der formalen Summen ihrer Bilder. So entsteht beispielsweise aus einem Gruppenmorphismus f mit $f(p) = q \oplus 2r$ und einem Algebromorphismus k , ein Morphismus der für ein Algebraelement a die Zuordnung $(p, a) \mapsto (q, k(a)) \oplus 2 * (r, k(a))$ enthält. Dagegen kann die Summe $(q, b) \oplus 2 * (r, b')$ mit $b \neq b'$ nie in einem Bild von

³Lineare Abbildungen, die Basiselemente auf Summen aus Basiselementen (ohne deren Inverse) abbilden (\rightarrow 5.2.4).

⁴Die Elemente von freien abelschen Gruppen können auch als Abbildungen von den Basiselementen nach \mathbb{Z} betrachtet werden.

⁵bestehend aus einer Abbildung zwischen den Signaturen und einer Familie von, die Sorten und Stelligkeit der Operationen bewahrenden, Abbildungen zwischen den Trägermengen der Algebren

$$\text{6 } f(g) \in G_2^\otimes \implies f(g) = \bigoplus_{g' \in G_2} x_{g'} * g' \implies f(g)(g') = x_{g'} \in \mathbb{Z}$$

⁷der Effekt von \otimes auf Gruppen und Algebren ergibt sich aus dem Effekt auf die entsprechenden Identitäten, d.h. $G \otimes A = \text{dom}(id_G \otimes id_A) = (G \times \|A\|)^\otimes$.

(p, a) unter $f \otimes k$ liegen. Motivation für diese Einschränkung ist die Trennung zwischen der Netz- und der Markenstruktur. Sie entspricht den unterschiedlichen Aspekten — Kausalität, Nichtdeterminismus und Nebenläufigkeit beim Netz und Eigenschaften der erreichbaren Zustände bei den Marken — die damit beschrieben werden.

Die Abbildung \otimes bewahrt die Struktur der Kategorien \mathcal{AG}_m und $\mathcal{GSIG}\text{-}\mathcal{ALG}$. Dadurch können die Morphismenkomposition und die Identitäten im Bild von \otimes auf die entsprechenden Konstruktionen in \mathcal{AG}_f und $\mathcal{GSIG}\text{-}\mathcal{ALG}$ zurückgeführt werden.

1.1.2 Lemma (Funktoreigenschaft):

Die Abbildung $\otimes: \mathcal{AG}_m \times \mathcal{GSIG}\text{-}\mathcal{ALG} \rightarrow \mathcal{AG}_m$ ist ein Funktor. □

(Beweis \rightarrow Anhang S. 185)

Notation: Im Sinne einer intuitiveren Notation bezeichnen wir das Bild einer Gruppe G^\otimes und einer Algebra A unter \otimes ⁸ mit $G^\otimes \otimes A$.

CP/T-Netze bestehen aus fünf Komponenten:

- einer freien abelschen Gruppe über einer Menge T von Transitionen, genannt „Schaltvektoren“,⁹
- einer freien abelschen Gruppe über einer Menge P von Stellen, genannt „Stellenvektoren“,
- einer Algebra als Individuenbereich der „Marken“ und
- zwei monotonen linearen Abbildungen „Pre“, „Post“ von den Schaltvektoren in die Markierungen.

Die Markierungen eines CP/T-Netzes ergeben sich aus der Verknüpfung der Stellenvektoren mit den Marken über \otimes und definieren die möglichen Gesamtzustände des Netzes. *Pre* und *Post* beschreiben den Effekt von Schaltvektoren als die Markierung, die von einem Zustand abgezogen (*Pre*) bzw. ihm hinzugefügt (*Post*) wird.¹⁰ Aus der Linearität dieser Morphismen folgen für jeden Schaltschritt $\bigoplus_{t \in T} x_t * t$ die Gleichungen

$$Pre\left(\bigoplus_{t \in T} x_t * t\right) = \bigoplus_{t \in T} x_t * Pre(t) \quad \text{und} \quad Post\left(\bigoplus_{t \in T} x_t * t\right) = \bigoplus_{t \in T} x_t * Post(t).$$

Damit reicht es aus, *Pre* und *Post* auf der Menge der Transitionen anzugeben, um sie für ihren gesamten Definitionsbereich festzulegen.

⁸genauer unter der von \otimes generierten Objektabbildung (\rightarrow 117)

⁹in der Literatur oft als „Schaltschritte“ bezeichnet

¹⁰Zwei Morphismen sind notwendig weil wir „Schlingen“ im Netz zulassen.

1.1.3 Definition (CP/T-Netz):

Ein CP/T-Netz ist ein Quintupel $(T^{\otimes}, P^{\otimes}, A, Pre, Post)$ für das gilt:

- (I) T^{\otimes} ist eine freie abelsche Gruppe,
- (II) P^{\otimes} ist eine freie abelsche Gruppe,
- (III) A ist eine Algebra aus $\mathcal{GSIG-ALG}$,
- (IV) $Pre: T^{\otimes} \rightarrow P^{\otimes} \otimes A$ ist ein monotoner Morphismus in den freien abelschen Gruppen und
- (V) $Post: T^{\otimes} \rightarrow P^{\otimes} \otimes A$ ist ein monotoner Morphismus in den freien abelschen Gruppen.

□

Notation: Im folgenden werden wir CP/T-Netze wahlweise als Quintupel oder als Diagramme der Form:

$$T^{\otimes} \begin{array}{c} \xrightarrow{Pre} \\ \xrightarrow{Post} \end{array} P^{\otimes} \otimes A$$

notieren. Entsprechend der in der Literatur üblichen graphischen Notation, symbolisieren wir Transitionen durch Quadrate, Stellen durch Kreise, und die Algebraelemente als Kanten- bzw. als Stelleninschriften. Pre und $Post$ fassen wir unter der Bezeichnung „*Markenflußmorphismen*“ zusammen, aus ihnen ergeben sich in der graphischen Netzrepräsentation die Kanten. Danach verläuft eine Kante

- von einer Stelle p in eine Transition t , wenn es im Bild von t unter Pre ein Paar gibt, dessen erste Komponente p ist und
- von einer Transition t in einer Stelle p , wenn es das Bild von t unter $Post$ ein Paar enthält, dessen erste Komponente p ist.

Die jeweils zweiten Komponenten im Bild von t werden aufsummiert den Kanten als Anschrift hinzugefügt (\rightarrow 1.1.5).

In der Literatur (\rightarrow z.B. [Rei85], [Jen81], [Rei91]) wird zu Petrinetzen üblicherweise eine „*Schaltregel*“ angegeben, die festlegt, unter welchen Bedingungen ein Ereignis eintreten kann und welche Auswirkungen es auf den Systemzustand hat. Diese prinzipiell in den Bereich der (operationalen) Semantik gehörende Regel, wird üblicherweise separat beschrieben und bildet damit ein weder zur Struktur noch zur Semantik eines Petrinetzes gehörendes Element. Im Rahmen unseres Formalismus wurde die Schaltregel in die operationale CP/T-Netz-Semantik integriert. Sie entspricht der für Stellen/Transitions-Netze üblichen Regel und läßt sich, wie in Abschnitt 1.3 (\rightarrow S. 15) gezeigt, ohne weiteres aus der dort eingeführten operationalen Netzsemantik extrahieren .

CP/T-Netze mit vergleichbarem Verhalten (\rightarrow Abschnitt 1.3) stehen über „*CP/T-Netzmorphismen*“ miteinander in Beziehung. Diese nachfolgend definierten Morphismen

men bestehen aus je einer strukturerhaltenden Abbildung für die drei CP/T-Netz-Komponenten Transitions-/Stellengruppe und Algebra; wobei wir uns mit dem Begriff „Struktur“ hier zum einen auf die additive Struktur der freien abelschen Gruppen und zum anderen auf die Sorten- und Operationsstruktur der Algebren beziehen. Neben dem Erhalt der Struktur durch die Komponenten ist deren Verträglichkeit mit den Markenflußmorphisms der ausschlaggebende Faktor für die Bewahrung des Netzverhaltens durch die CP/T-Netzmorphisms.

1.1.4 Definition (CP/T-Netzmorphisms):

Seien $N_i = (T_i^{\otimes}, P_i^{\otimes}, A_i, Pre_i, Post_i)$ $i = 1, 2$ CP/T-Netze, dann bildet ein Tripel (f^T, f^P, f^A) einen CP/T-Netzmorphismus $f: N_1 \rightarrow N_2$ wenn gilt:

- (I) $f^T: T_1^{\otimes} \rightarrow T_2^{\otimes}$ ist ein monotoner Morphismus in den freien abelschen Gruppen,
- (II) $f^P: P_1^{\otimes} \rightarrow P_2^{\otimes}$ ist ein monotoner Morphismus in den freien abelschen Gruppen,
- (III) $f^A: A_1 \rightarrow A_2$ ist ein generalisierter Algebromorphismus und
- (IV) die beiden folgenden Diagramme kommutieren.

$$\begin{array}{ccc}
 T_1^{\otimes} & \xrightarrow{Pre_1} & P_1^{\otimes} \otimes A_1 \\
 f^T \downarrow & \curvearrowright & \downarrow f^P \otimes f^A \\
 T_2^{\otimes} & \xrightarrow{Pre_2} & P_2^{\otimes} \otimes A_2
 \end{array}
 \qquad
 \begin{array}{ccc}
 T_1^{\otimes} & \xrightarrow{Post_1} & P_1^{\otimes} \otimes A_1 \\
 f^T \downarrow & \curvearrowright & \downarrow f^P \otimes f^A \\
 T_2^{\otimes} & \xrightarrow{Post_2} & P_2^{\otimes} \otimes A_2
 \end{array}$$

□

1.1.5 Beispiel (CP/T-Netze und CP/T-Netzmorphismus):

Abbildung 1.1 zeigt die CP/T-Netze

$$\begin{aligned}
 N_1 &= (\{t_1\}^{\otimes}, \{p_1, p_2\}^{\otimes}, (\{\bullet\}, \emptyset), Pre_1: t_1 \mapsto 2 * (p_1, \bullet), Post_1: t_1 \mapsto 2 * (p_2, \bullet)) \\
 N_2 &= (\{t'_1, t''_1, t_2\}^{\otimes}, \{p_1, p'_2, p''_2, p_3, p_4\}^{\otimes}, (\{\bullet\}, \emptyset), \\
 &\quad Pre_2: t'_1 \mapsto (p_1, \bullet), Post_2: t'_1 \mapsto (p'_2, \bullet) \\
 &\quad t''_1 \mapsto (p_1, \bullet) \quad t''_1 \mapsto (p''_2, \bullet) \\
 &\quad t_2 \mapsto (p_3, \bullet) \quad t_2 \mapsto 2 * (p_1, \bullet) \oplus (p_4, \bullet))
 \end{aligned}$$

in der üblichen graphischen Notation, verbunden durch den CP/T-Netzmorphismus

$$f = \left(\begin{array}{ccc}
 f^T: t_1 \mapsto t'_1 \oplus t''_1, & f^P: p_1 \mapsto p_1, & id \\
 & p_2 \mapsto p'_2 \oplus p''_2 &
 \end{array} \right).$$

Die schraffierten Stellen/Transitions in N_2 kennzeichnen das Bild von N_1 unter f .

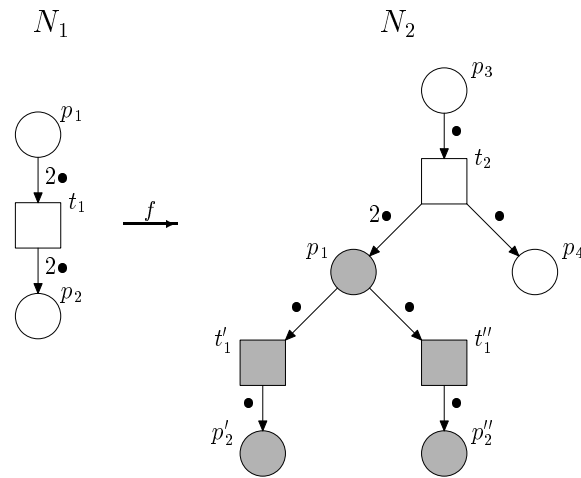


Abbildung 1.1: Ein CP/T-Netzmorphismus

Formale Summen, wie sie z.B. in den Bildern der Markenflußmorphismen vorkommen, können auch als Funktionen von der Basismenge, der entsprechenden freien abelschen Gruppe in die ganzen Zahlen aufgefaßt werden. Das ist dann von Nutzen, wenn hervorgehoben werden soll, ob und wie oft eine Marke von einer Stelle abgezogen bzw. ihr hinzugefügt wird. So beschreibt beispielweise der Ausdruck $Post_2(t_2)(p_1, \bullet) = 2$ das Hinzufügen von zwei Marken des Typs \bullet auf die Stelle p_1 beim Schalten von t_2 .

□

CP/T-Netze bilden zusammen mit den CP/T-Netzmorphismen eine Kategorie, die wir mit \mathcal{CPTN} bezeichnen. Deren Identitäten ergeben sich aus den Identitäten der Netzkomponenten, die jeweils selbst Objekte einer Kategorie ($\mathcal{AG}_m, \mathcal{GSIG-ALG}$) sind. Entsprechend wird die Komposition von Morphismen komponentweise in den jeweiligen Kategorien durchgeführt.

1.1.6 Definition (Kategorie der CP/T-Netze):

Wir definieren eine Kategorie \mathcal{CPTN} mit

- (I) den CP/T-Netzen als Objekten,
- (II) den CP/T-Netzmorphismen als Morphismen,
- (III) den Morphismen $(id_{T^\otimes}, id_{P^\otimes}, id_A): N \rightarrow N$ mit $N = (T^\otimes, P^\otimes, A, Pre, Post)$ als Identitäten und
- (IV) der komponentweisen Komposition

$$(f_2^T \circ f_1^T, f_2^P \circ f_1^P, f_2^A \circ f_1^A)$$

für alle komponierbaren CP/T-Netzmorphismen $f_i = (f_i^T, f_i^P, f_i^A) \ i = 1, 2$.

□

1.2 CP/T-Netz-Komposition und -Zerlegung

Im folgenden Abschnitt stellen wir Konstruktionen vor, mit deren Hilfe CP/T-Netze aus Teilnetzen komponiert, oder in Teilnetze zerlegt werden können. Die Auszeichnung der Teilnetze erfolgt jeweils durch CP/T-Netzmorphismen. Sowohl Komposition und Zerlegung als auch die nachfolgend definierte Projektion von CP/T-Netzen und CP/T-Netzmorphismen auf ihren Algebraanteil, spielen eine wichtige Rolle in den folgenden Kapiteln.

CP/T-Netze und CP/T-Netzmorphismen enthalten Algebren und Algebromorphismen als separate Teile, die von der Projektion \mathcal{P}^A als Ergebnis geliefert werden. Die Funktoreigenschaft der Projektion folgt direkt aus der komponentweisen Verknüpfung der CP/T-Netzmorphismen und der Definition der Identitäten in \mathcal{CPTN} (\rightarrow 1.1.6). Damit genügt es, wie bei Funktoren üblich, \mathcal{P}^A auf Morphismen zu definieren; die Bilder der Objekte ergeben sich aus der Abbildung der jeweiligen Identitäten.

1.2.1 Definition (Projektion):

Wir definieren die Projektion von den CP/T-Netzen und CP/T-Netzmorphismen auf deren Algebraanteil als Funktor $\mathcal{P}^A: \mathcal{CPTN} \rightarrow \mathcal{GSIG-ALG}$ durch:

$$\mathcal{P}^A((f^T, f^P, f^A)) = f^A$$

für alle CP/T-Netzmorphismen (f^T, f^P, f^A) . □

Die Komposition von CP/T-Netzen realisieren wir mit einem vielseitigen und mächtigen Instrument der Kategorientheorie, den „*Colimites*“ über Diagrammen. Dabei werden die für die Komposition maßgeblichen Zusammenhänge zwischen den Objekten durch Morphismen beschrieben. Formal verstehen wir unter einem Diagramm einen Funktor, von einer die Form des Diagramms beschreibenden, endlichen Kategorie \mathcal{S} ¹¹, in die Kategorie, über der das Diagramm interpretiert werden soll, z.B. \mathcal{AG}_m , $\mathcal{GSIG-ALG}$ oder \mathcal{CPTN} . Dabei wird \mathcal{S} zumeist durch den zugrundeliegenden Graphen charakterisiert, in dem die Identitäten und alle durch Komposition erzeugbaren Morphismen weggelassen sind.

Ein Beispiel für die Charakterisierung von Kompositionen mit Hilfe von Morphismen sind die im weiteren häufig verwendeten Diagramme der Form

$$Obj_1 \xleftarrow{m_1} Schnittstelle \xrightarrow{m_2} Obj_2$$

mit einer durch den Graphen $s_1 \leftarrow s_0 \rightarrow s_2$ generierten Kategorie \mathcal{S} . Interpretiert in \mathcal{CPTN} , beschreiben sie die Vereinigung zweier CP/T-Netze über die Teilnetze, die durch das Bild des jeweiligen CP/T-Netzmorphismus ausgezeichnet werden. Dabei sorgt der gemeinsame Ursprung der Morphismen dafür, daß die Teilnetze miteinander

¹¹ von „Shape“-Kategorie

kompatibel sind, d.h. alle Eigenschaften gemeinsam haben, die für das \mathcal{CPTN} im Ursprung der Morphismen gelten und von CP/T-Netzmorphismen bewahrt werden.

Der Colimes eines Diagramms \mathcal{D} mit Codomäne \mathcal{C} besteht aus einem \mathcal{C} -Objekt C und einer Familie von Morphismen $(c_s)_{s \in \text{Obj}(\mathcal{S})}$ mit jeweils einem Diagrammobjekt $\mathcal{D}(s)$ als Domäne und C als Codomäne. Diese Morphismen müssen mit den Diagrammmorphismen verträglich sein, d.h. für jeden \mathcal{S} -Morphismus $m: s \rightarrow s'$ muß

$$\begin{array}{ccc} \mathcal{D}(s) & \xrightarrow{\mathcal{D}(m)} & \mathcal{D}(s') \\ & \searrow c_s & \downarrow c_{s'} \\ & & C \end{array} \quad \circlearrowright$$

kommutieren. Unter diesen als „Senken“ bezeichneten Paaren $((c_s)_{s \in \text{Obj}(\mathcal{S})}, C)$ zeichnen sich Colimites dadurch aus, daß von ihnen in jedes andere Objekt einer Senke genau ein¹² mit den $(c_s)_{s \in \text{Obj}(\mathcal{S})}$ verträglicher Morphismus existiert. Eine kurze Einführung zu diesem Thema findet sich zusammen mit den Definitionen von Diagramm und Colimites in Abschnitt 5.1, ausführlich wird es in [HS73] oder [BW90] behandelt.

CP/T-Netze sind relativ komplexe Objekte, daher erscheint es zunächst schwierig, sie zu komponieren, d.h. Colimites zu Diagrammen in \mathcal{CPTN} zu bilden. Hier hilft uns die Gliederung der CP/T-Netzmorphismen in drei, voneinander unabhängige, das Bild eines CP/T-Netzes vollständig determinierende Komponenten¹³. Aufgrund dieser Zusammenhänge können wir im folgenden die Komposition von CP/T-Netzen auf die Komposition ihrer Komponenten, der freien abelschen Gruppen und Algebren, zurückführen und damit wesentlich vereinfachen. Unter welchen Bedingungen Colimites in den entsprechenden Kategorien existieren und wie sie gebildet werden, wird für die freien abelschen Gruppen in Abschnitt 5.2 und für die Algebren in Abschnitt 6.2 erläutert.

1.2.2 Lemma (Komponentweise Konstruktion von Colimites):

Sei \mathcal{S} eine endliche Kategorie

$$\begin{aligned} \mathcal{D}^T: \mathcal{S} &\rightarrow \mathcal{AG}_m \text{ ein Funktor mit Colimes } ((c_s^T)_{s \in \mathcal{S}}, T_c^\otimes), \\ \mathcal{D}^P: \mathcal{S} &\rightarrow \mathcal{AG}_m \text{ ein Funktor mit Colimes } ((c_s^P)_{s \in \mathcal{S}}, P_c^\otimes), \\ \mathcal{D}^A: \mathcal{S} &\rightarrow \mathcal{GSLG-ALG} \text{ ein Funktor mit Colimes } ((c_s^A)_{s \in \mathcal{S}}, A_c) \end{aligned}$$

und $\mathcal{D}: \mathcal{S} \rightarrow \mathcal{CPTN}$ ein Funktor mit

$$\mathcal{D}(m) = (\mathcal{D}^T(m), \mathcal{D}^P(m), \mathcal{D}^A(m))$$

¹²bis auf Isomorphie

¹³sowohl ihre Domänen als auch ihre Codomänen sind paarweise disjunkt und entsprechen den Domänen/Codomänen der Markenflußmorphismen, deren Bilder damit (\rightarrow 1.1.4 (IV)) festgelegt sind.

für jeden Morphismus m in \mathcal{S} , dann ergibt sich der Colimes von \mathcal{D} in \mathcal{CPTN} zu

$$\left((c_s^T, c_s^P, c_s^A)_{s \in \text{Obj}(\mathcal{S})}, T_c^\otimes \begin{array}{c} \xrightarrow{Pre_c} \\ \xrightarrow{Post_c} \end{array} P_c^\otimes \otimes A_c \right).$$

Die Markenflußmorphisme Pre_c und $Post_c$ ergeben sich eindeutig als kommutative Ergänzungen der nachfolgenden Diagramme, in denen für alle Objekte s in \mathcal{S} , Pre_s und $Post_s$ die Markenflußmorphisme in $\mathcal{D}(s)$ bezeichnen.

$$\begin{array}{ccc} T_s^\otimes & \xrightarrow{c_s^T} & T_c^\otimes \\ \downarrow Pre_s & \curvearrowright & \downarrow Pre_c \\ P_s^\otimes \otimes A_s & \xrightarrow{c_s^P \otimes c_s^A} & P_c^\otimes \otimes A_c \end{array} \quad \begin{array}{ccc} T_s^\otimes & \xrightarrow{c_s^T} & T_c^\otimes \\ \downarrow Post_s & \curvearrowright & \downarrow Post_c \\ P_s^\otimes \otimes A_s & \xrightarrow{c_s^P \otimes c_s^A} & P_c^\otimes \otimes A_c \end{array} \quad \square$$

(Beweis \rightarrow Anhang S. 186)

Mit Hilfe von Morphismen lassen sich CP/T-Netze nicht nur komponieren, sie können auch zu deren Zerlegung dienen. Definition 1.1.4 stellt sicher, daß die Bilder von CP/T-Netzmorphismen wieder, über einer Menge von Stellen mit ihrer Umgebung verbundene, CP/T-Netze sind¹⁴. Diese Kontaktstellen bilden, für sich und zusammen mit der Umgebung des Bildes, ein CP/T-Netz, so daß durch einen CP/T-Netzmorphismus drei Teilnetze ausgezeichnet werden: sein Bild, dessen Komplement zusammen mit den Kontaktstellen und die Kontaktstellen selbst. Bei der nachfolgend definierten, auf dieser Eigenschaft der CP/T-Netzmorphismen basierenden Zerlegung, wird die Algebra im Zielnetz des CP/T-Netzmorphismus vollständig in alle drei CP/T-Netze des Resultats übernommen. Die Gründe hierfür liegen in den Algebren, die sich aufgrund der Abschlußeigenschaft ihrer Trägermengen bezüglich der enthaltenen Operationen nicht den Teilnetzen entsprechend zerlegen lassen.

Wir geben die Zerlegung eines CP/T-Netzes als Diagramm an, bei dem das von den Kontaktstellen gebildete Teilnetz jeweils über einen Inklusionsmorphismus mit dem Bild und mit dessen Umgebung verbunden ist. Der Vorteil dieser Darstellung liegt in der Verfügbarkeit von sowohl den einzelnen Teilnetzen als Bilder des Diagrammfunktors als auch der gesamten Struktur. Letztere kann z.B. als Grundlage für Diagrammoperation wie der Colimesbildung genutzt werden (\rightarrow 1.2.5).

1.2.3 Definition (Zerlegung von CP/T-Netzen):

Sei \mathcal{S} die durch den Graphen „ $im \leftarrow is \rightarrow co$ “ generierte Kategorie und

$$f = (f^T, f^P, f^A): N_1 \rightarrow N_2 \quad \text{mit} \quad N_i = (T_i^\otimes, P_i^\otimes, A_i, Pre_i, Post_i), \quad i = 1, 2$$

ein CP/T-Netzmorphismus. Wir definieren die Zerlegung von N_2 über f als Funktor $\mathcal{DC}_f: \mathcal{S} \rightarrow \mathcal{CPTN}$ durch:

¹⁴d.h. die Bilder von CP/T-Netzmorphismen sind immer „stellenberandet“

$$(I) \mathcal{DC}_f(im) = T_{im}^{\otimes} \begin{array}{c} \xrightarrow{Pre_{im}} \\ \xrightarrow{Post_{im}} \end{array} P_{im}^{\otimes} \otimes A_2 \text{ mit}$$

$$\begin{aligned} T_{im} &= \{ t \in T_2 \mid \exists t' \in T_1: f^T(t')(t) > 0 \} \\ P_{im} &= \{ p \in P_2 \mid \exists p' \in P_1: f^P(p')(p) > 0 \} \\ Pre_{im} &= Pre_2 \Big|_{T_{im}^{\otimes}} \\ Post_{im} &= Post_2 \Big|_{T_{im}^{\otimes}}, \end{aligned}$$

$$(II) \mathcal{DC}_f(co) = T_{co}^{\otimes} \begin{array}{c} \xrightarrow{Pre_{co}} \\ \xrightarrow{Post_{co}} \end{array} P_{co}^{\otimes} \otimes A_2 \text{ mit}$$

$$\begin{aligned} T_{co} &= T_2 \setminus T_{im} \\ P_{co} &= (P_2 \setminus P_{im}) \cup \{ p \in P_2 \mid \exists t \in T_{co}, a \in A_2: \\ &\quad Pre_2(t)(p, a) > 0 \vee Post_2(t)(p, a) > 0 \} \\ Pre_{co} &= Pre_2 \Big|_{T_{co}^{\otimes}} \\ Post_{co} &= Post_2 \Big|_{T_{co}^{\otimes}}, \end{aligned}$$

$$(III) \mathcal{DC}_f(is) = \emptyset \Longrightarrow (P_{im} \cap P_{co})^{\otimes} \otimes A_2 \text{ und}$$

$$(IV) \text{ f\"ur alle } m: s \rightarrow s' \text{ in } \mathcal{S} \text{ durch: } \mathcal{DC}_h(m) = (\hookrightarrow_{T_s^{\otimes}, T_{s'}^{\otimes}}, \hookrightarrow_{P_s^{\otimes}, P_{s'}^{\otimes}}, id_{A_2}).$$

□

1.2.4 Beispiel (Dekomposition):

Der CP/T-Netzmorphismus in Abbildung 1.1 f\"uhrt unter Verwendung von Definition 1.2.3 zu der in Abbildung 1.2 gezeigten Zerlegung. Das Netz N_2 kann durch Colimesbildung (\rightarrow 1.2.2) aus seiner Zerlegung zur\"uckgewonnen werden.

□

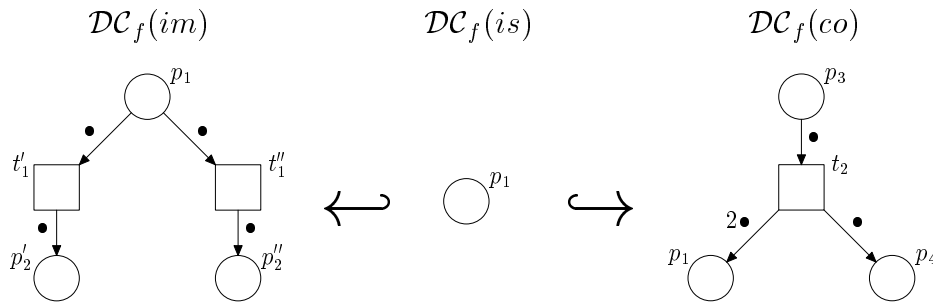
Die Zerlegung eines CP/T-Netzes \u00fcber einen CP/T-Netzmorphismus bildet ein Diagramm mit dem Ausgangsnetz Colimesobjekt, d.h. bei Zerlegung von CP/T-Netzen in der beschriebenen Weise werden weder Teile weggelassen noch hinzugef\u00fcgt.

1.2.5 Lemma (Zerlegungen bewahren die Netzstruktur):

Sei \mathcal{DC}_f die Zerlegung eines CP/T-Netzes N \u00fcber einen CP/T-Netzmorphismus f , dann bildet $((\hookrightarrow, \hookrightarrow, id)_{s \in Obj(\mathcal{S})}, N)$ einen Colimes zu \mathcal{DC}_f .

□

(Beweis \rightarrow Anhang S. 188)

Abbildung 1.2: Zerlegung von N_2 durch f .

1.3 CP/T-Netz-Semantik

Im letzten Abschnitt dieses Kapitels ordnen wir den CP/T-Netzen ein Verhalten in Form der möglichen Ereignisabfolgen zu und geben eine Transformation der CP/T-Netzmorphismen in Abbildungen zwischen diesen Verhaltensbeschreibungen an. Die dadurch definierte Semantik umfaßt alle wichtigen Petrinetzeigenschaften wie Kausalität, Nebenläufigkeit, Konflikte und Nichtdeterminismus. Diese lassen sich direkt aus der Verhaltensbeschreibung ablesen und werden von den transformierten CP/T-Netzmorphismen bewahrt.

Für die klassischen Petrinetze, speziell für B/E-Systeme, existiert mit der Zuordnung von „Prozessen“ zu Netzen (vergl. [Rei85]) eine gut untersuchte Semantik mit den eingangs beschriebenen Eigenschaften. Aufgrund der damit verbundenen expliziten Unterscheidung von Nichtdeterminismus und Nebenläufigkeit, fällt sie in die Klasse der sogenannten „True-Concurrency-Semantiken“. Diese Verhaltensbeschreibung wurde in den Arbeiten von P. Degano, J. Meseguer und U. Montanari [MM90], [DMM89] in eine algebraische Form gebracht und auf Stellen/Transitions-Netze erweitert. Dabei werden Prozesse als Terme dargestellt, die sich über zwei Operatoren sequentiell (Symbol „;“) und nebenläufig (Symbol „||“) ¹⁵ komponieren lassen. Auf diese Weise läßt sich jeder „Prozeßterm“ aus „atomaren Termen“ ¹⁶ zusammensetzen. Das Verhältnis der Operatoren zueinander ist ein wichtiges Ergebnis der zitierten Arbeiten und findet in der Gleichung

$$(v; v') \parallel (w; w') = (v \parallel w); (v' \parallel w')$$

seinen Ausdruck.¹⁷ In ihrem Kontext interpretiert besagt diese Gleichung, daß bei der nebenläufigen Komposition zweier sequentieller Prozesse die Ereignisse des einen

¹⁵In den Originalarbeiten durch wird die nebenläufige Verknüpfung durch „ \otimes “ symbolisiert.

¹⁶Terme, die nur ein Ereignis enthalten

¹⁷Sie ist nur unter der Voraussetzung, daß die sequentiellen Verknüpfungen der linken Seite definiert sind, von rechts nach links anwendbar. „;“ ist assoziativ und „||“ ist assoziativ und kommutativ.

- in keinem kausalen Zusammenhang zu den Ereignissen des anderen Prozesses stehen. Neben dem modularen Aufbau der Prozeßterme, liegt ein weiterer Vorzug dieser Prozeßbeschreibung in seiner Variabilität. Zu jedem Prozeß im klassischen Sinn existiert eine Äquivalenzklasse von Prozeßtermen, die es erlaubt, je nach Anwendung Markierungen¹⁸ ganz oder teilweise auszublenden.

Die im folgenden beschriebene operationale Semantik für CP/T-Netze, basiert auf der algebraischen Prozeßbeschreibung von P. Degano, J. Meseguer und U. Montanari. Als ersten Schritt in diese Richtung definieren wir eine eingeschränkte Form der Prozeßterme, die „*Schrittsequenzen*“. Damit lassen sich Prozesse beschreiben, die aus der sequentiellen Verknüpfung gleichzeitig eintretender Ereignisse¹⁹ bestehen. Die Menge der Schrittfolgen zu einem CP/T-Netz wird über eine Äquivalenzrelation in Klassen zerlegt. Dabei werden diejenigen Sequenzen zusammengefaßt, die ausgehend von der selben Anfangsmarkierung, die gleichen unabhängigen Ereignisse in unterschiedlicher Reihenfolge enthalten. Die Abfolge der Ereignisse in diesen Sequenzen ist nicht kausal bedingt, sie stellt vielmehr eine mögliche, d.h. die kausalen Abhängigkeiten respektierende Sequenzialisierung der Ereignisse dar. Demgemäß existiert zu jedem Prozeß im herkömmlichen Sinne eine Klasse von Schrittsequenzen. Die Kausalordnung der Ereignisse im Prozeß ist in der entsprechenden Ordnung auf der Schrittsequenz enthalten und beide Verhaltensdarstellungen verfügen über die gleichen Anfangs- und Endmarkierungen.

Für ein gegebenes CP/T-Netz, bilden die Äquivalenzklassen über dessen Schrittsequenzen, zusammen mit den möglichen Markierungen eine Kategorie. Dabei werden zwei Markierungen m und m' durch einen Morphismus $[v]$ verbunden, wenn im Netz ein Schritt v existiert, der m von den Stellen abzieht ($Pre(v) = m$) und m' hinzufügt ($Post(v) = m'$). Jede Markierung, als Objekt der Kategorie, definiert gleichzeitig ihren Identitätsmorphimus und die Komposition von Morphismen entspricht der sequentielle Verknüpfung von Schritten.

1.3.1 Definition (Schrittsequenzsemantik):

Sei $N = (T^{\otimes}, P^{\otimes}, A, Pre, Post)$ ein CP/T-Netz,

(I) Mor die kleinste Menge für die gilt:

$$(a) m \in P^{\otimes} \otimes A \implies m: m \rightarrow m \in Mor,$$

$$(b) v \in T^{\otimes} \implies v: Pre(v) \rightarrow Post(v) \in Mor,$$

$$(c) f, g \in (T \cup (P \times \|A\|))^{\otimes} \\ \implies f \oplus g: dom(f) \oplus dom(g) \rightarrow cod(f) \oplus cod(g) \in Mor \text{ und}$$

$$(d) f, g \in Mor \wedge cod(f) = dom(g) \implies f; g: dom(f) \rightarrow cod(g) \in Mor;$$

¹⁸Markierungen werden in diesem Zusammenhang als Ereignisse ohne Effekt dargestellt; sie bilden die neutralen Elemente bzgl. der sequentiellen Komposition „;“

¹⁹d.h. formale Summen über den Transitionen eines Netzes, siehe 1.1

(II) $\approx \subseteq Mor \times Mor$ ist der reflexive, symmetrische und transitive Abschluß der Relation $\sim \subseteq Mor \times Mor$ gegeben durch:

- (a) $v \oplus w \sim v \oplus dom(w); w \oplus cod(v)$ für alle $v \oplus w$ in Mor und
- (b) $v; w \sim v'; w'$, wenn $v; w$ in Mor , $v \sim v'$ und $w \sim w'$;

dann bezeichnen wir die Kategorie \mathcal{ST}_N mit

(III) den Elementen aus $P^\otimes \otimes A$ als Objekten,

(IV) den Äquivalenzklassen aus Mor/\approx als Morphismen mit

$$dom([f]_{\approx}) = dom(f) \quad \text{und} \quad cod([f]_{\approx}) = cod(f)$$

für alle Morphismen $[f]_{\approx}$,

(V) den Morphismen $[m]_{\approx}$ mit m in $P^\otimes \otimes A$ als Identitäten und

(VI) der Komposition „ \circ “ definiert durch:

$$[w]_{\approx} \circ [v]_{\approx} \stackrel{\text{def}}{=} [v; w]_{\approx},$$

für alle komponierbaren Morphismen $[v]_{\approx}, [w]_{\approx}$,

als „*Schrittsequenzsemantik zu N*“ (kurz „*N-Schrittsequenzsemantik*“). □

Schrittsequenzen stehen in einem engen Zusammenhang zur üblichen Schrittsemantik für Petrinetze (\rightarrow [BF86]). Jeder Morphismus in \mathcal{ST}_N und dementsprechend jede Klasse von Schrittsequenzen, läßt sich äquivalent als Komposition von Morphismen der Form:

$$[m_n] \circ [v_n] \circ [m_{n-1}] \circ \cdots \circ [m_1] \circ [v_1] \circ [m_0]$$

darstellen, in der die $[m_i]$ Identitäten und die v_j Schaltvektoren²⁰ bezeichnen. Sie entspricht der Schrittfolge:

$$m_0 [v_1 > m_1 \cdots m_{n-1} [v_n > m_n$$

in der üblichen Schreibweise.

Die Entsprechung von Schrittsequenzen und Schrittfolgen gilt nur in der angegebenen Richtung, da die Schritte in den Schrittsequenzen immer die gesamte Markierung „konsumieren“ unter der sie aktiviert sind. Als Konsequenz daraus müssen Teile einer

²⁰Summe von Transitionen

Markierung, die von einem Schaltschritt unberührt bleiben, diesem in der Schrittsequenz explizit in Form des entsprechenden Identitätsmorphismus hinzugefügt werden (\rightarrow 1.3.2). Das heißt, im allgemeinen entspricht die folgende Schrittsequenz

$$[m_n] \circ [v_n \oplus m_n''] \circ [m_{n-1}] \circ \cdots \circ [m_1] \circ [v_1 \oplus m_1''] \circ [m_0]$$

mit $m_i = m_i' \oplus m_i''$ und $m_{i-1} = m_i'' \oplus m_{i-1}'$ für $i = 1, \dots, n$, der zuvor gezeigten Schrittfolge.

Notation: Um zu einer, im Sinne der Verhaltensbeschreibung, intuitiveren Darstellung der Komposition von Schrittsequenzklassen zu gelangen, bringen wir deren Operanden in die ihrer sequentiellen Abfolge entsprechende Reihenfolge. Zu diesem Zweck notieren wir die Komposition $[g] \circ [f]$ zweier Morphismen in \mathcal{ST}_N durch $[f]; [g]$. Daraus ergibt sich für die Schrittsequenz aus dem vorangehenden Absatz die Darstellung:

$$[m_0]; [v_1]; [m_1]; \cdots; [m_{n-1}]; [v_n]; [m_n] \quad ,$$

die sich nur noch durch die Verknüpfungssymbole von der entsprechenden Schrittfolge in der herkömmlichen Notation unterscheidet.

Aus den Zusammenhängen zwischen Schritten und Markierungen in den Schrittsequenzsemantiken der CP/T-Netze, läßt sich eine „Schaltregel“ im üblichen Sinn (\rightarrow [Rei85], [BF86]) extrahieren. Die Aktivierungsbedingung für einen Schaltvektor v unter einer Markierung m (notiert „ $m[v >$ “), ergibt sich danach zu

$$m[v > \iff Pre(v) \leq m \quad ^{21}$$

und der Folgezustand m' bezüglich des Schaltens von v unter m (notiert „ $m[v > m'$ “) zu

$$m[v > m' \implies m' = m \ominus Pre(v) \oplus Post(v). \quad ^{22}$$

Diese Bedingungen entsprechen der für Petrinetze üblichen Schaltregel²³. Für CP/T-Netze sind sie obsolet, da deren Verhaltensbeschreibung ohne sie auskommt.

1.3.2 Beispiel (Equivalente Schrittsequenzen):

Abbildung 1.3 zeigt ein CP/T-Netz mit einer einelementigen Algebra $((\{\bullet\}, \emptyset))$ und der Anfangsmarkierung $3(p1, \bullet) \oplus 2(p2, \bullet)$.

Die Schrittsequenzsemantik zu diesem Netz enthält den Morphismus

$$[t \oplus 2(p1, \bullet) \oplus (p2, \bullet); 2t \oplus (p3, \bullet)]: 3(p1, \bullet) \oplus 2(p2, \bullet) \rightarrow 2(p2, \bullet) \oplus 3(p3, \bullet),$$

²¹ $\bigoplus_{a \in A} x_a * a \leq \bigoplus_{a \in A} y_a * a \iff \forall a \in A: x_a \leq y_a$

²² $\left(\bigoplus_{a \in A} x_a * a\right) \ominus \left(\bigoplus_{a \in A} y_a * a\right) = \bigoplus_{a \in A} x_a - y_a * a$

²³ z.B. in [Rei85], [Jen81], [BF86], [Rei91]

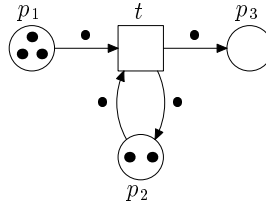


Abbildung 1.3: Netz mit drei möglichen Abläufen.

dessen Bezeichner $([\dots])$ die drei in diesem Netz möglichen Ereignisabfolgen — t gefolgt von t nebenläufig zu t , t gefolgt von t gefolgt von t und t nebenläufig zu t gefolgt von t — zusammenfaßt, d.h.

$$\begin{aligned} & t \oplus 2(p1, \bullet) \oplus (p2, \bullet) ; 2t \oplus (p3, \bullet) \\ \approx & t \oplus 2(p1, \bullet) \oplus (p2, \bullet) ; t \oplus (p1, \bullet) \oplus (p2, \bullet) \oplus (p3, \bullet) ; t \oplus (p2, \bullet) \oplus 2(p3, \bullet) \\ \approx & 2t \oplus (p1, \bullet) ; t \oplus (p2, \bullet) \oplus 2(p3, \bullet). \end{aligned}$$

Mit der Konvention, die Komposition mit vertauschten Operanden und dem Operationssymbol „;“ zu notieren, folgt daraus:

$$\begin{aligned} & [t \oplus 2(p1, \bullet) \oplus (p2, \bullet)] ; [2t \oplus (p3, \bullet)] \\ = & [t \oplus 2(p1, \bullet) \oplus (p2, \bullet)] ; [t \oplus (p1, \bullet) \oplus (p2, \bullet) \oplus (p3, \bullet)] ; [t \oplus (p2, \bullet) \oplus 2(p3, \bullet)] \\ = & [2t \oplus (p1, \bullet)] ; [t \oplus (p2, \bullet) \oplus 2(p3, \bullet)] \end{aligned}$$

□

Petritnetzprozesse im üblichen Sinn (\rightarrow [Rei85]) lassen sich als Ergebnis von sequentiellen und nebenläufigen Verknüpfungen kausaler Ereignissequenzen betrachten. Um entsprechende Terme, im weiteren als „Prozeßterme“ bezeichnet, erzeugen zu können, fehlt bis jetzt eine Möglichkeit zur nebenläufigen Komposition von Schrittsequenzen. Sie wird zu jedem CP/T-Netz N durch die Definition eines Funktors von $\mathcal{ST}_N \times \mathcal{ST}_N$ nach \mathcal{ST}_N ergänzt. Dieser Funktor kann als zweistelliger Operator auf den Schrittsequenzklassen aufgefaßt werden und gestattet die Abbildung von Prozeßtermen auf Schrittsequenzen (\rightarrow 1.3.4).

1.3.3 Definition (Paralleloperator auf Schrittsequenzsemantiken):

Sei \mathcal{ST}_N die Schrittsequenzsemantik eines CP/T-Netzes N . Wir definieren eine Abbildung $\|_N: \mathcal{ST}_N \times \mathcal{ST}_N \rightarrow \mathcal{ST}_N$ durch

$$\|_N ([f], [g]) = \begin{cases} [f \oplus g], & f, g \in (T \cup (P \times \|A\|))^{\otimes}; \\ \|_N ([f''], [g'']) \circ \|_N ([f'], [g']), & f = f' ; f'' \wedge g = g' ; g''; \end{cases}$$

für jedes Morphismenpaar $[f], [g]$ aus \mathcal{ST}_N .

□

Notation: Um die Lesbarkeit der Prozeßterme erhöhen, verwenden wir den Funktor \parallel_N in Infixnotation und lassen den Index weg, wenn dieser aus dem Kontext hervorgeht, d.h. wir schreiben $[v] \parallel [w]$ für $\parallel_N([v], [w])$. Darüber hinaus nehmen wir an, daß die parallele Verknüpfung „ \parallel “ stärker bindet als die sequentielle Verknüpfung „ $;$ “.

1.3.4 Beispiel (Von Prozeßtermen zu Prozessen):

Die folgenden Terme der Schrittsequenzsemantik zu N_2 aus Abbildung 1.1,

$$[(p_3, \bullet)]; [t_2]; [(p_4, \bullet)] \parallel [(p_1, \bullet); t'_1; (p'_2, \bullet)] \parallel [(p_1, \bullet); t''_1; (p''_2, \bullet)],$$

$$[t_2]; [t'_1] \parallel [t''_1] \parallel [(p_4, \bullet)],$$

lassen sich unter Verwendung von 1.3.1 (II) und 1.3.3 zur Schrittsequenz

$$[t_2; t'_1 \oplus t''_1 \oplus (p_4, \bullet)]$$

auswerten. Sie entspricht dem Prozeß in Abbildung 1.4.

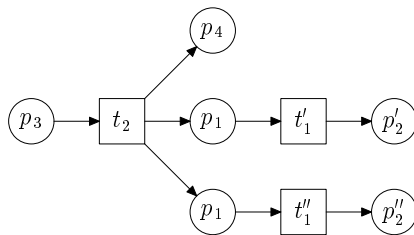


Abbildung 1.4: Der Prozeß zum Term $[t_2; t'_1 \oplus t''_1 \oplus (p_4, \bullet)]$.

□

1.3.5 Lemma (Eigenschaften des Paralleloperators):

Sei \mathcal{ST}_N die Schrittsequenzsemantik eines CP/T-Netzes N , dann

(I) ist $\parallel_N: \mathcal{ST}_N \times \mathcal{ST}_N \rightarrow \mathcal{ST}_N$ ein Funktor und

(II) für alle \mathcal{ST}_N -Morphismen $[f], [g]$ gilt: $\parallel_N([f], [g]) = \parallel_N([g], [f])$.

□

(Beweis \rightarrow Anhang S. 190)

Als Funktor ist der von einem CP/T-Netz N generierte Paralleloperator \parallel_N mit der Komposition der Morphismen innerhalb der N -Schrittsequenzsemantik verträglich. Betrachtet man CP/T-Netze als Beschreibungen nebenläufiger Systeme und die zugehörigen Schrittsequenzsemantiken als mögliche Ereignisabfolgen in diesen Systemen, dann entspricht diese Eigenschaft einem grundsätzlichen Zusammenhang zwischen Nebenläufigkeit und Kausalität von Ereignissen. Dieser Zusammenhang läßt

sich für Ereignisse v_1, v_2, w_1, w_2 in Form von Gleichungen folgendermaßen ausdrücken

$$\begin{aligned} (v_1; v_2) \parallel (w_1; w_2) &= (v_1 \parallel w_1); (v_2 \parallel w_2) \\ &= (v_1 \parallel id_{dom(w_1)}); (v_2 \parallel w_1); (w_2 \parallel id_{cod(v_2)}) \quad 24 \\ &= (w_1 \parallel id_{dom(v_1)}); (v_1 \parallel w_2); (v_2 \parallel id_{cod(w_2)}) \end{aligned}$$

und besagt zum einen, daß kausal abhängige Ereignisse (z.B. v_1, v_2) in jedem Fall nacheinander stattfinden und zum anderen, daß Ereignisse die in keinem kausalen Zusammenhang stehen (z.B. v_1, w_1) entweder nebenläufig oder in beliebiger sequentieller Reihenfolge auftreten können. Die Identitäten innerhalb der Gleichungen können dahingehen interpretiert werden, daß die den zugehörigen Markierungen entsprechenden Ressourcen im Zuge des jeweiligen Ereignisses unangetastet bleiben.

1.3.6 Beispiel (Umformung von Schrittsequenzen):

Die Gleichung

$$[t \oplus 2(p1, \bullet) \oplus (p2, \bullet)]; [2t \oplus (p3, \bullet)] = [2t \oplus (p1, \bullet)]; [t \oplus (p2, \bullet) \oplus 2(p3, \bullet)]$$

über Schrittsequenzen zum CP/T-Netz aus Abbildung 1.3, läßt sich mit Hilfe der im vorangehenden Absatz gezeigten Gleichungen folgendermaßen herleiten:

$$\begin{aligned} &[t \oplus 2(p1, \bullet) \oplus (p2, \bullet)]; [2t \oplus (p3, \bullet)] \\ &= [t \oplus (p1, \bullet)] \parallel [(p1, \bullet) \oplus (p2, \bullet)]; [t \oplus (p3, \bullet)] \parallel [t] \\ &= \left([t \oplus (p1, \bullet)]; [t \oplus (p3, \bullet)] \right) \parallel \left([(p1, \bullet) \oplus (p2, \bullet)]; [t] \right) \\ &= \left([t \oplus (p1, \bullet)]; [t \oplus (p3, \bullet)] \right) \parallel \left([t]; [(p2, \bullet) \oplus (p3, \bullet)] \right) \quad 25 \\ &= [t \oplus (p1, \bullet)] \parallel [t]; [t \oplus (p3, \bullet)] \parallel [(p2, \bullet) \oplus (p3, \bullet)] \\ &= [2t \oplus (p1, \bullet)]; [t \oplus (p2, \bullet) \oplus 2(p3, \bullet)]. \end{aligned}$$

□

Mit Hilfe der Schrittsequenzsemantiken und den darauf aufbauenden Prozeßtermen, in denen sich alle wichtigen Netzeigenschaften widerspiegeln, läßt sich unsere Behauptung, die CP/T-Netzmorphismen würden diese Netzeigenschaften bewahren, belegen. Zu diesem Zweck konstruieren wir zu jedem CP/T-Netzmorphismus f einen Funktor \mathcal{ST}_f zwischen den Schrittsequenzsemantiken der jeweiligen Netze mit den Eigenschaften

$$\mathcal{ST}_f([v]; [w]) = \mathcal{ST}_f([v]); \mathcal{ST}_f([w])$$

²⁴In jeder Kategorie gilt bzgl. jedes Morphismus f , $f = f \circ id_{dom(f)} = id_{cod(f)} \circ f$. Bezogen auf \mathcal{ST}_N folgt daraus $f = id_{dom(f)}; f = f; id_{cod(f)}$.

²⁵ $id_{dom([t]); [t]} = [t]; id_{cod([t])}$

und

$$\mathcal{ST}_f([v] \parallel [w]) = \mathcal{ST}_f([v]) \parallel \mathcal{ST}_f([w]).$$

Da alle elementaren Morphismen einer Schrittsequenzsemantik, d.h. alle Morphismen die sich nicht als Ergebnis von Kompositionen darstellen lassen, aus Markierungen und Schaltschritten des zugrundeliegenden CP/T-Netzes zusammengesetzt sind, können sie direkt mit den dafür zuständigen CP/T-Netzmorphismus-Komponenten f^T resp. $f^P \otimes f^A$ abgebildet werden. Die Bilder aller übrigen Morphismen lassen sich aus der Komposition der Bilder ihrer elementaren Bestandteile gewinnen. Damit ist gleichzeitig die Verträglichkeit von \mathcal{ST}_f mit der sequentiellen Verknüpfung „;“ sichergestellt.

1.3.7 Definition (Abbildung zwischen Schrittsequenzsemantiken):

Sei $f = (f^T, f^P, f^A): N_1 \rightarrow N_2$ ein CP/T-Netzmorphismus und

$$\hat{f}: (T_1 \cup (P_1 \times \parallel A_1 \parallel))^\otimes \rightarrow (T_2 \cup (P_2 \times \parallel A_2 \parallel))^\otimes$$

ein Morphismus in den freien abelschen Gruppen mit

$$\hat{f}(v) = \begin{cases} f^T(v), & v \in T_1; \\ f^P \otimes f^A(v), & v \in P_1 \times \parallel A_1 \parallel; \end{cases}$$

für alle v aus $(T_1 \cup (P_1 \times \parallel A_1 \parallel))$. Wir definieren die Abbildung \mathcal{ST}_f von den \mathcal{ST}_{N_1} - in die \mathcal{ST}_{N_2} -Morphismen durch:

$$\mathcal{ST}_f([k]) = \begin{cases} [\hat{f}(k)], & k \in (T_1 \cup (P_1 \times \parallel A_1 \parallel))^\otimes; \\ \mathcal{ST}_f([k_2]) \circ \mathcal{ST}_f([k_1]), & k = k_1; k_2; \end{cases}$$

für jeden \mathcal{ST}_{N_1} -Morphismus $[k]$.

□

Die parallele Verknüpfung von Schrittsequenzklassen läßt sich auf die Gruppenoperation „ \oplus “ und die Morphismenkomposition zurückführen (\rightarrow 1.3.3). Die Gruppenoperation wird von dem Gruppenmorphismus \hat{f} bewahrt, auf dem \mathcal{ST}_f in allen Fällen aufbaut, in denen die Definition des Paralleloperators „ \parallel “ die Operation „ \oplus “ verwendet. Zusammen mit der Bewahrung der Morphismenkomposition ergibt sich daraus die Verträglichkeit von \mathcal{ST}_f mit der parallelen Komposition.

1.3.8 Lemma (Eigenschaften der Schrittsemantikabbildung):

Für jeden CP/T-Netzmorphismus $f: N_1 \rightarrow N_2$ gilt:

(I) $\mathcal{ST}_f: \mathcal{ST}_{N_1} \rightarrow \mathcal{ST}_{N_2}$ ist ein Funktor und

(II) das folgende Diagramm kommutiert:

$$\begin{array}{ccc}
 \mathcal{ST}_{N_1} \times \mathcal{ST}_{N_1} & \xrightarrow{\parallel_{N_1}} & \mathcal{ST}_{N_1} \\
 \mathcal{ST}_f \times \mathcal{ST}_f \downarrow & \curvearrowright & \downarrow \mathcal{ST}_f \\
 \mathcal{ST}_{N_2} \times \mathcal{ST}_{N_2} & \xrightarrow{\parallel_{N_2}} & \mathcal{ST}_{N_2}
 \end{array}$$

□

(Beweis → Anhang S. 192)

1.3.9 Beispiel (Von Morphismen zu Abbildungen zwischen Prozeßtermen):

Der CP/T-Netzmorphismus f in Abbildung 1.1 generiert den Funktor \mathcal{ST}_f , mit dessen Hilfe sich die Prozeßterme von N_1 in Prozeßterme zu N_2 übertragen lassen, z.B.

$$\begin{aligned}
 \mathcal{ST}_f([2 * (p_1, \bullet); t_1; (p_2, \bullet)]) &= \mathcal{ST}_f([(p_2, \bullet)]) \circ \mathcal{ST}_f([t_1]) \circ \mathcal{ST}_f([2 * (p_1, \bullet)]) \\
 &= [(p'_2, \bullet) \oplus (p''_2, \bullet)] \circ [t'_1 \oplus t''_1] \circ [2 * (p_1, \bullet)] \\
 &= [2 * (p_1, \bullet); [t'_1 \oplus t''_1]; [(p'_2, \bullet) \oplus (p''_2, \bullet)]] \\
 &= [t'_1] \parallel [t''_1]; [(p'_2, \bullet)] \parallel [(p''_2, \bullet)] \\
 &= [t'_1; (p'_2, \bullet)] \parallel [t''_1; (p''_2, \bullet)] \\
 &= [(p_1, \bullet); t'_1; (p'_2, \bullet)] \parallel [(p_1, \bullet); t''_1; (p''_2, \bullet)].
 \end{aligned}$$

Das ergibt in der graphischen Prozeßnotation den in Abbildung 1.5 dargestellten Zusammenhang.

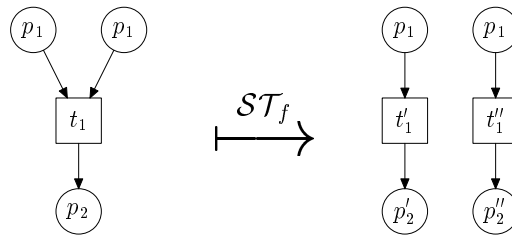


Abbildung 1.5: Die Abbildung eines Prozesses von N_1 aus Abb. 1.1.

□

Die Definitionen 1.3.1, 1.3.3 ordnen jedem CP/T-Netz seine Schrittsequenzsemantik und einen Operator zu deren nebenläufiger Komposition zu. Zusammen mit der Algebra des jeweiligen CP/T-Netzes ergibt sich daraus dessen „erweiterte Prozeßtermsemantik“. Diese Erweiterung um die Algebren ist notwendig, da sonst die Informationen über die Operationen der in den Netzen enthaltenen Algebren in deren

Semantik nicht mehr enthalten wären. Wir dehnen die erweiterte Prozeßtermsemantik auf die CP/T-Netzmorphismen aus, in dem wir jedem CP/T-Netzmorphismus ein Paar, bestehend aus seiner Algebrakomponente und dem von ihm generierten Schrittsequenzsemantikfunktorkomplex, zuordnen und erhalten damit eine, zu \mathcal{CPTN} korrespondierende, Semantikkategorie. Im weiteren werden wir, abgesehen von Fällen in denen wir die Algebrakomponente besonders hervorheben wollen, die erweiterte Prozeßtermsemantik der CP/T-Netze und CP/T-Netzmorphismen einfach als deren „Prozeßtermsemantik“ bezeichnen.

1.3.10 Definition (Erweiterte Prozeßtermsemantik):

Wir definieren \mathcal{PTS} als die kleinste Kategorie, die

- (I) für jedes CP/T-Netz N ein Tripel $(\mathcal{P}^A(N), \mathcal{ST}_N, \|_N)$ als Objekt,
- (II) für jeden CP/T-Netzmorphismus $f: N_1 \rightarrow N_2$ ein Tupel $(\mathcal{P}^A(f), \mathcal{ST}_f)$ als Morphismus mit

$$\begin{aligned} \text{dom}((\mathcal{P}^A(f), \mathcal{ST}_f)) &= (\mathcal{P}^A(N_1), \mathcal{ST}_{N_1}, \|_{N_1}) \\ \text{cod}((\mathcal{P}^A(f), \mathcal{ST}_f)) &= (\mathcal{P}^A(N_2), \mathcal{ST}_{N_2}, \|_{N_2}), \end{aligned}$$

- (III) die Paare $(\mathcal{P}^A(id_N), id_{\mathcal{ST}_N})$ für jedes CP/T-Netz N als Identitäten und

- (IV) die komponentweise Komposition als Kompositionsoperator enthält.

□

Die Schrittsequenzsemantik eines CP/T-Netzes N enthält, bis auf die Operationen der Algebra, alle Informationen von N und ist allein dadurch vollständig determiniert. Damit existiert zu jedem CP/T-Netz genau ein Tripel aus Algebra, Schrittsequenzsemantik und Kompositionsoperator. Das Entsprechende gilt für jeden CP/T-Netzmorphismus und dem ihm als Semantik zugeordneten Paar. Aus der Verknüpfung beider Eigenschaften ergibt sich die Isomorphie von \mathcal{CPTN} und \mathcal{PTS} .

1.3.11 Korollar (Semantikfunktorkomplex):

Die Abbildung

$$\mathcal{PC}: \mathcal{CPTN} \rightarrow \mathcal{PTS} \quad \text{mit} \quad \mathcal{PC}(f) \stackrel{\text{def}}{=} (\mathcal{P}^A(f), \mathcal{ST}_f)$$

für jeden CP/T-Netzmorphismus f , ist ein bijektiver Funktorkomplex.

□

Systemspezifikationen

Im ersten Abschnitt dieses Kapitels stellen wir ein formales Beschreibungsmittel für nebenläufige Systeme, die „*Algebraischen Netzspezifikationen*“ (kurz „*AN-Spezifikationen*“) vor. Abschnitt 2.2 führt eine denotationelle Semantik in Form von Modellen für AN-Spezifikationen ein und Abschnitt 2.3 ordnet jedem Modell eine operationale Semantik in Form von halbgeordneten Abläufen zu.

Unter einem nebenläufigen System soll hier eine endliche Menge aktiver Komponenten¹ verstanden werden, die voneinander unabhängig agieren und über passive Komponenten verknüpft sein können. Diese Verknüpfung ist statisch, d.h. sie ist in jedem Systemzustand gleich. Selbstmodifizierende Systeme, wie sie z.B. mit Hilfe von R. Milners „ *π -Kalkül*“ ([Mil95]) oder R. Valks „*Self-Modifying Nets*“ ([Val81]) beschrieben werden können, lassen sich mit AN-Spezifikationen nur indirekt erfassen, indem die sich ändernde Struktur in die von den aktiven Komponenten verbrauchten bzw. erzeugten Ressourcen kodiert wird. Sequentielle Systeme können, als Spezialfälle nebenläufiger Systeme, bei denen nie mehr als eine Komponente gleichzeitig agiert, mit AN-Spezifikationen beschrieben werden. Für diese Systemklasse existiert allerdings eine Reihe bewährter Beschreibungsmethoden, die mächtiger oder bei vergleichbaren Eigenschaften einfacher zu handhaben sind.

Die Aktionen in nebenläufigen Systemen erfolgen asynchron, d.h. ein globaler Takt wird nicht vorausgesetzt. Kommunikation und Synchronisation werden über das Bereitstellen und Abziehen von Ressourcen auf den passiven Komponenten realisiert. Typische Vertreter nebenläufiger Systeme sind: Telekommunikationsanlagen, Fertigungssysteme, verteilte Datenbanken, Prozeßsteuerungen und Rechnernetze. Diese Systeme sind in der Regel sehr komplexer Natur, so daß ein dafür angemessenes Beschreibungsmittel über ausreichend mächtige Strukturierungsmechanismen verfügen sollte. Eine häufig angewandte, unter dem Schlagwort „*Objektorientierung*“ bekannte Möglichkeit zur Strukturierung der Beschreibung komplexer Systeme, die Verwendung von Datentypen, kommt in diesem Kapitel bei der Beschreibung der Ressourcen zum Einsatz. Komposition und Parameterisierung als weitere Mechanismen sind Inhalt von Kapitel 3.

Um formale Systembeschreibungen von informellen abzugrenzen, verwenden wir dafür im weiteren den Begriff „*Systemspezifikation*“, wohl wissend, daß auch diese Bezeichnung in der Literatur zum Teil für nicht oder nur in Teilen formale Beschreibungen

¹das schließt nicht aus, daß unendlich viele verschiedene Aktionen ausgeführt werden können, da jede Komponente in verschiedenen Modi arbeiten kann, vergl. dazu Definition der operationalen Semantik in Abschnitt 2.3

verwendet wird.² Neben der in allen Punkten festgelegten Syntax, unterscheiden sich Systemspezifikationen vor allem durch das Vorhandensein einer eindeutigen Semantik von informellen Systembeschreibungen.

Semantiken, ob denotationell oder operational, treten hauptsächlich in zwei Formen auf: entweder als „*lose Semantik*“, die oft auch als „*klassische Semantik*“ bezeichnet wird oder als „*initiale Semantik*“³. Erstere bezeichnet eine Menge von Modellen, die die spezifizierten Eigenschaften erfüllen. Diese Modelle können darüber hinaus Eigenschaften besitzen, die sich nicht aus der Spezifikation ableiten lassen. Typisch sind lose Semantiken für Spezifikationen von Objekten der klassischen Mathematik, z.B. Mengen, Halbgruppen, Gruppen, Relationen und Ordnungen. So ist z.B. jede Gruppe auch eine Halbgruppe. Ein loser Zusammenhang zwischen Spezifikation und Modellen ist aber auch in anderen Bereichen üblich, so legt z.B. ein Pflichtenheft bei der Auftragserteilung nur die erwünschten Eigenschaften eines Produkts fest, ein erweiterter Funktionsumfang oder die Verwendung von Hilfsoperationen ist damit aber nicht ausgeschlossen. Lose Semantiken im allgemeinen haben den Nachteil, daß nicht von vornherein klar ist, ob es überhaupt Modelle zur Spezifikation gibt.

Initiale Semantiken zeichnen sich dadurch aus, daß zu jeder Spezifikation genau ein Modell existiert. In den meisten Fällen kann es direkt aus der Spezifikation generiert werden. Beispiele dafür sind die „*Herbrand-Modelle*“ für Prädikatenlogiken erster Stufe oder die „*Quotiententermalgebra*“ (\rightarrow [EM85]) als initiales Modell einer algebraischen Spezifikation. Mit Hilfe von initialen Modellen kann durch direkten Vergleich getestet werden, ob eine Spezifikation wirklich das Objekt beschreibt, z.B. die natürlichen Zahlen oder ein Telefonsystem, das dem Spezifikateur oder dem späteren Nutzer vorschwebte. Initiale Semantiken sind dagegen wenig geeignet, Objekte zu klassifizieren oder Implementierungsschritte, mit den dazu notwendigen Umgebungsanpassungen zu validieren.

Bei den in Abschnitt 2.2 und Abschnitt 2.3 eingeführten Semantiken für Systemspezifikationen handelt es sich um lose Semantiken mit initialen Modellen. Sie vereinen die Vorteile beider Semantikformen. Dabei spielen die initialen Modelle die Rolle von Referenzobjekten.

2.1 Algebraische Netzspezifikationen

Wie die meisten „*High-Level-Spezifikationen*“ nebenläufiger Systeme (z.B. [Vau86], [Rei91], [Jen81], [Jen90], [ISO83]) bestehen die im folgenden vorgestellten AN-Spezifikationen im wesentlichen aus zwei Teilen: einer Datentypspezifikation zur Beschreibung der systemrelevanten Objekte und einer Spezifikation der im System möglichen Ereignisse und ihrer Abhängigkeiten, die wir im weiteren als „*Nebenläufigkeitsstruktur*“ bezeichnen. Der Datentypanteil wird mit Hilfe algebraischer Spezifikationen

²z.B. für Flußdiagramme

³es existiert mit der „*terminalen Semantik*“ noch eine dritte Form, die aber selten benutzt wird

im Stil von [EM85] und [EBO91] beschrieben. Dabei werden die Operationen zum Erzeugen und Manipulieren der Objekte mit Hilfe von Gleichungen spezifiziert. Eine kurze Einführung dazu, mit Beispielen, sowie die im weiteren verwendeten Definitionen und Sätze sind in Kapitel 6 zusammengefaßt.

Die Vorteile der algebraischen Datentypspezifikation, z.B. gegenüber der in Jensen [Jen81] und [Jen90] gewählten Beschreibung, mit Hilfe von Mengen und Funktionen oder des in einigen kommerziellen Spezifikationswerkzeugen ([JCHH91]) implementierten Beschreibungsverfahrens mittels programmiersprachlicher Konstrukte, liegen einmal in der Verfügbarkeit eines Beweiskalküls für Datentypereigenschaften (vergl. [EM85]) und zum anderen in den Strukturierungsmöglichkeiten über Komposition und Parameterisierung. Darüber hinaus eröffnen vorhandene Termersetzungsverfahren ([EW91]) die Möglichkeit, Terme algorithmisch zu manipulieren und damit für rechnergestützte Werkzeuge handhabbar zu machen.

Die Nebenläufigkeitsstruktur der Ereignisse wird in Form von „höheren Petrinetzen“ (\rightarrow CP/T-Netze, Kapitel 1.1) angegeben. Neben der intuitiven graphischen Darstellung sprechen vor allem die für höhere Netze vorhandenen bzw. auf sie übertragbaren Beweismethoden wie z.B. der „Invariantenkalkül“ oder „Deadlock-/Trap-Eigenschaften“. Ein weiterer wesentlicher Grund ist die für klassische Petrinetze entwickelte Halbordnungssemantik, die in Abschnitt 2.3 auf höhere Netze übertragen wird. Im Gegensatz zu „Interleaving-Semantiken“ bei prozeßalgebrabasierten Ansätzen wie „CCS“ ([Mil80]) und „TCSP“ ([BHR84]) bei denen Ereignisse, zunächst nach ihrem Eintreten und bei Nebenläufigkeit zufällig, totalgeordnet werden, läßt die Halbordnungssemantik eine Unterscheidung zwischen Nichtdeterminismus und Nebenläufigkeit zu. Damit wird das Beweisen von Systemeigenschaften erleichtert, da sich aus den halbgeordneten Abläufen direkt Systemzustände ablesen lassen, die im Zuge einer „künstlichen“ Totalordnung der Ereignisse nicht mehr darstellbar sind (\rightarrow [Rei92],[DGK⁺92]) und damit im Rahmen einer Systemanalyse nicht mehr zur Verfügung stehen. Darüber hinaus lassen sich mit Hilfe von halbgeordneten Abläufen Aussagen über den Grad an Parallelismus in Systemen treffen, um daraus beispielsweise auf deren Effizienz zu schließen.

AN-Spezifikationen enthalten neben der Datentypspezifikation und dem die Nebenläufigkeitsstruktur beschreibenden CP/T-Netz, eine Menge von Gleichungen zu jeder Transition im Netz. Die darin auftretenden Variablen finden sich in den Termen wieder, mit denen die Kanten des Netzes beschriftet sind. Diese Terme entstammen der Termalgebra zu einer Erweiterung der Datentypspezifikation um eine Variablenfamilie⁴ als zusätzliche nullstellige Operationen. Sie erlauben es Objekte, die im Zuge eines Ereignisses entstehen oder verbraucht werden, zu Gruppen zusammen zu fassen. Diese Gruppen lassen sich durch die Gleichungsmengen einschränken. Da das Auftreten von Ereignissen vom Vorhandensein der Objekte abhängt, die vom Ereignis

⁴Die hinzu gefügten Variablen sind im wesentlichen für alle Spezifikationen gleich. Sie haben eine grundsätzlich andere Funktion als die Variablen der zur Datentypspezifikation gehörenden Gleichungen (\rightarrow 6.2.1).

verbraucht werden, schränken die Gleichungen gleichzeitig die in einem Systemzustand möglichen Ereignisse ein.

Um die technischen Konstruktionen zu vereinfachen, nehmen wir im folgenden an, daß die Variablenmenge X für alle Signaturen im wesentlichen gleich ist. Wir erhalten ein derartiges X , indem wir von einer festen Variablenmenge X_{fix} ausgehen, die mit den Sorten SO der jeweiligen Signatur indiziert wird, d.h. $X = X_{fix} \times SO$. Aufgrund dieser globalen Definition der Variablen können wir auf eine individuelle Festlegung als zusätzliche AN-Spezifikations-Komponente verzichten.

Mit Hilfe von X kann jede Spezifikation $Spec$ zu $Spec(X)$ erweitert werden, indem wir die Variablen als zusätzliche Konstanten den Operationen von $Spec$ hinzufügen (\rightarrow 6.3.4). Diese Erweiterung läßt sich kanonisch auf Spezifikationsmorphismen übertragen. Ein dermaßen von $h: Spec \rightarrow Spec'$ zu $h^x: Spec(X) \rightarrow Spec'(X)$ erweiterter Morphismus verhält sich auf den Sorten und Operationen aus $Spec$ wie h und auf X wie die Identität. Analog zu Spezifikationen können auch Algebren mit X erweitert werden. Für eine $Spec$ -Algebra A notieren wir die so erweiterte Algebra mit $A(X)$. Dabei handelt es sich um eine A -Quotiententalgebra (\rightarrow 6.2.15) zu $Spec(X)$, die bezüglich der Inklusion von $Spec$ in $Spec(X)$ frei über A erzeugt ist (\rightarrow 6.2.17).

Wir definieren nachfolgend für jedes CP/T-Netz die Menge der Variablen in der Umgebung jeder Transition. Darunter fallen alle Variablen, die in den Termen an den Kanten einer Transition vorkommen, d.h. alle Variablen im Bild der Transition unter $Pre/Post$.

2.1.1 Definition (Variablen in der Umgebung von Transitionen):

Gegeben ein CP/T-Netz $(T^\circ, P^\circ, A(X), Pre, Post)$, dann definieren wir

(I) die Abbildung $env: T \rightarrow 2^{P \times \|A(X)\|}$ durch:

$$env(t) \stackrel{\text{def}}{=} \left\{ (p, a) \in P \times \|A(X)\| \mid Pre(t)(p, a) > 0 \vee Post(t)(p, a) > 0 \right\}$$

(II) und die Abbildung $var: T \rightarrow 2^X$ durch:

$$var(t) \stackrel{\text{def}}{=} \bigcup_{(p,a) \in env(t)} var(a)$$

dabei bezeichnet $var: \|A(X)\| \rightarrow 2^X$ die Abbildung von $A(X)$ -Termen auf ihre Variablen (\rightarrow 6.3.1).

□

2.1.2 Definition (Variablen in Gleichungsmengen):

Gegeben eine algebraische Spezifikation $Spec$ und die Menge EQ_{Spec} aller Gleichungen zur $Spec$ -Signatur mit Variablen aus X , dann definieren wir die Abbildung $var: 2^{EQ_{Spec}} \rightarrow 2^X$ für alle Teilmengen E aus EQ_{Spec} durch:

$$var(E) \stackrel{\text{def}}{=} \bigcup_{(X', l, r) \in E} X'.$$

□

2.1.3 Definition (Algebraische Netzspezifikationen):

Eine „Algebraische Netzspezifikation“ (kurz „AN-Spezifikation“) ist ein Tripel

$$(Spec, Cond, N)$$

mit den folgenden Eigenschaften:

- (I) $Spec$ ist eine algebraische Spezifikation,
- (II) N ist ein CP/T-Netz mit endlicher Stellen- und Transitionsmenge, das die Gleichung:

$$\mathcal{P}^A(N) = T_{OP(X)} \quad ^5$$

erfüllt, wobei $T_{OP(X)}$ die Termalgebra zur $Spec$ -Erweiterung $Spec(X)$ bezeichnet,

- (III) $Cond: T \rightarrow 2^{EQ_{Spec}}$ ist eine Abbildung von der Menge der N -Transitionen in die Menge aller Gleichungen zur $Spec$ -Signatur mit Variablen aus X und
- (IV) für alle Transitionen t und Variablen $x_s, y_{s'}$ aus $var(t) \cup var(Cond(t))$ gilt:

$$s \neq s' \implies x \neq y$$

wobei s, s' die jeweiligen Sorten der Variablen bezeichnen.

□

Im Unterschied zu vergleichbaren Definitionen von Netzen mit algebraisch spezifizierten Tokentypen, z.B. bei [Vau86] oder in [Rei91], wird bei AN-Spezifikationen auf die Sortentreue der Stellen verzichtet, d.h. Transitionen können Terme zu verschiedenen Sorten von einer Stelle abziehen oder ihr hinzufügen. Sortentreue ist zwar eine häufig gestellte und vom Standpunkt des Software-Engineering auch sinnvolle Zusatzforderung, für die mathematische Behandlung von Systemspezifikationen und deren Semantik aber ohne Bedeutung⁶. Aus diesem Grunde wurde sie zu Gunsten einer mathematisch einfacheren Handhabbarkeit der Theorie weggelassen.

⁵ \mathcal{P}^A (\rightarrow 1.2.1) ist die Projektion der CP/T-Netze auf ihre Algebrakomponente.

⁶Sollte die Sortentreue einzelner Stellen für den Beweis von Systemeigenschaften notwendig sein, kann sie durch eine entsprechende Gestaltung der Terme in deren Umgebung sichergestellt werden.

2.1.4 Beispiel (Die Spezifikation eines „Alternating-Bit-Protokolls“):

In Abbildung 2.1 wird ein Kommunikationsprotokoll, das in der Literatur unter dem Namen „*Alternating-Bit-Protokoll*“⁷ bekannt ist, als AN-Spezifikation beschrieben. Das Protokoll soll das im folgenden kurz skizzierte Problem lösen: Ein Sender und ein Empfänger sind über einen möglicherweise fehlerhaften Kanal miteinander verbunden. Der Sender verschickt Mitteilungen an den Empfänger und erwartet von diesem eine Empfangsbestätigung. Solange die Empfangsbestätigung nicht eingetroffen ist, geht der Sender davon aus, daß die Mitteilung nicht empfangen wurde und wiederholt das Versenden in gewissen Zeitabständen. Da Sender und Empfänger nicht synchronisiert sind und sich auch nicht synchronisieren können (oder wollen), kann es vorkommen, daß sich das Versenden von Mitteilung und Bestätigung überschneiden. Außerdem wird davon ausgegangen, daß sowohl Mitteilungen als auch Bestätigungen im Kanal verloren gehen können.

Das Alternating-Bit-Protokoll stellt sicher, daß jede Mitteilung solange gesendet wird, bis sie empfangen wurde und eine Bestätigung dafür beim Sender vorliegt. Den Mitteilungen sind Typen zugeordnet, und es können mehrere Mitteilungen unterschiedlichen Typs unabhängig von einander versendet werden. Ein wesentliches Merkmal dieses Protokolls ist die konstante Länge der Kontrollinformation, die den Mitteilungen hinzugefügt wird. Diese Länge ist nur von der Anzahl der Mitteilungstypen abhängig und nicht von der Anzahl der notwendigen Sendeveruche.

Für die Protokollspezifikation gehen wir davon aus, daß die Variablenmenge X die Variablen „ m “ zur Sorte „Message“ und „ c “ zur Sorte „Control“ enthält. Die Abbildung von Transitionen in Gleichungsmengen haben wir in die graphische Darstellung des zur Spezifikation gehörenden CP/T-Netzes integriert.⁸ Darin gehören die Stellen und Transitionen auf der linken Seite des Netzgraphen zum Sender und die Stellen und Transitionen auf der rechten Seite zum Empfänger. Verbunden sind sie durch die, den Kanal symbolisierenden, Stellen „ $chan_m$ “ und „ $chan_a$ “.

Die Stelle „ to_send “ enthält alle zu versendenden Mitteilungen. Abgesendete Mitteilungen werden auf der Stelle „ buf “ kopiert, um im Bedarfsfall, d.h. wenn die Empfangsbestätigung ausbleibt, durch die Transition „ t_{out} “ erneut verschickt werden zu können. Bestätigungen, die nicht das korrekte Kontrollbit enthalten, werden von der Transition „ $clear$ “ aus dem Kanal entfernt. Korrekte Bestätigungen werden durch die Transition „ rec_a “ dem Kanal entnommen und führen zu einer Umstellung des aktuellen Kontrollbits für diesen Mitteilungstyp.

Beim Empfänger werden die Kontrollbits der eintreffenden Mitteilungen überprüft (Transition „ rec_m “). Mitteilungen mit korrektem Kontrollbit landen ohne Kontrollbit auf der Stelle „ $recvd$ “. Die Transition „ $send_m$ “ ist für das Versenden der Bestätigungen zuständig. Durch verloren gegangene Bestätigungen können Mitteilungen mit

⁷genaugenommen firmieren zahlreiche Protokollvarianten unter diesem Namen

⁸die entsprechende Zeile ist durch eine Linie mit der zugehörenden Transition $senden_m$ verbunden

Alternating-Bit-Protocol =

$$\{ \{ \text{Message, Control, MessContr} \},$$

$$\{ _ _ : \text{Message, Control} \rightarrow \text{MessContr},$$

$$\text{cont} : \text{Message} \rightarrow \text{Control},$$

$$| _ | : \text{Control} \rightarrow \text{Control},$$

$$- : \text{Control} \rightarrow \text{Control} \}$$

$$\{ | \text{cont}(n) | = \text{cont}(n)$$

$$| - \text{cont}(n) | = \text{cont}(n)$$

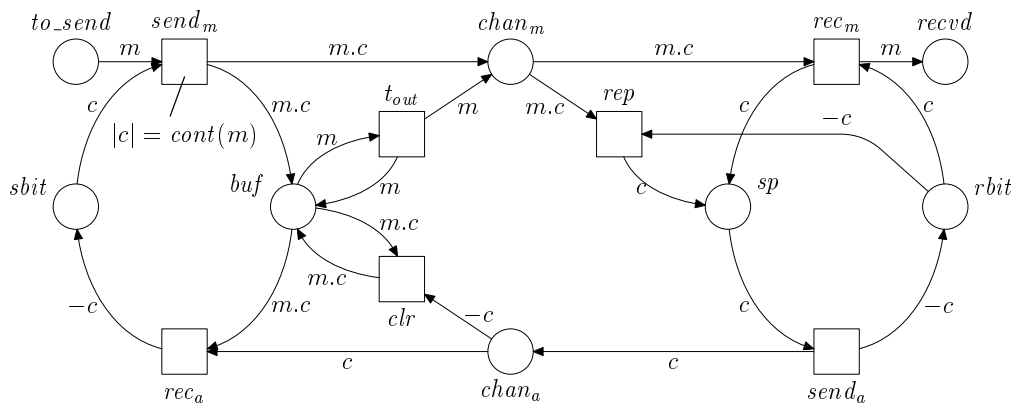
$$-(-x) = x \}$$


Abbildung 2.1: Ein Alternating-Bit-Protokoll.

falschen Kontrollbits entstehen. Sie werden von der Transition „*repl*“ dem Kanal entnommen. Das Ablegen des Kontrollbits dieser Mitteilungen auf der Pufferstelle des Empfängers („*sp*“) erlaubt es, die entsprechende Bestätigung erneut zu verschicken.

Auf den Stellen „*sbit*“ beim Sender und „*rbit*“ beim Empfänger liegt für jeden Mitteilungstyp jeweils das aktuelle Kontrollbit (Wert von „*cont*“), das vom Sender den Mitteilungen hinzugefügt und vom Empfänger überprüft wird. Jedes korrekte Empfangen von Mitteilungen beim Empfänger und Bestätigungen beim Sender führt zur Umstellung des Kontrollbits für den jeweiligen Mitteilungstyp. Vor dem Start des Protokolls müssen die Stellen „*sbit*“ und „*rbit*“ mit den Kontrollbits für alle Mitteilungstypen initialisiert werden und zwar so, daß für jeden Mitteilungstyp jeweils ein gleichwertiges Kontrollbit auf beiden Stellen vorhanden ist. Die n -fache Umstellung eines Kontrollbits ergibt für gerade n wieder das ursprüngliche Bit. Diese für die Funktion des Protokolls wesentliche Datentypeigenschaft ist in den Gleichungen der Datentypspezifikation festgelegt.

□

Die Spezifikation des Alternating-Bit-Protokolls (\rightarrow 2.1.4) enthält getrennte Kanäle für Mitteilungen und Bestätigungen. Für diese Designentscheidung gibt es keine Notwendigkeit im Protokoll, sie wurde vielmehr aus Gründen der Übersichtlichkeit ge-

treffen. Darüber hinaus werden die potentiellen Fehler im Kanal nicht mitspezifiziert. Zum einen gibt es verschiedene Fehlermöglichkeiten, die sich nicht alle gleichzeitig erfassen lassen, z.B Störungen bei getrennten Kanälen, die nur einen Kanal betreffen oder Signale, die dem Kanal hinzugefügt werden, zum anderen läßt sich das Protokoll auch auf stabile Kanäle anwenden. Sowohl die Beschaffenheit der Kanäle (ein oder zwei Kanäle) als auch die Art der möglichen Fehler finden sich in den verschiedenen Modellen (\rightarrow 2.2, 2.3) zur Protokollspezifikation wieder, die wir im folgenden Abschnitt einführen. Hier lassen sich, im Rahmen der von der Spezifikation gesetzten Grenzen, alle Anpassungen an konkrete Systeme vornehmen und deren Auswirkungen untersuchen.

2.2 Algebraische Netze

Bei der im folgenden vorgestellten Modellsemantik für AN-Spezifikationen, handelt es sich um eine lose Semantik mit initialen Modellen⁹. Im Unterschied zu den bekannten Semantiken für höhere,¹⁰ petrinetzbasierte Systemspezifikationen, z.B. [Vau86], [DHP90] und [Rei91], die bei der Modellbildung eine Variation des Datentypanteils in Form verschiedener Algebren zulassen, aber die Netzstruktur der Spezifikation unverändert in jedes Modell übernehmen, interpretieren wir beide Teile einer Systemspezifikation lose. Das heißt, auch die Netzstruktur unterliegt einer kontrollierten Modifikation beim Übergang von der Spezifikation zu den Modellen. Dabei bleiben die essentiellen Eigenschaften der Systemspezifikation, wie der Effekt der Operationen beim Datentypanteil und Kausalität, Nebenläufigkeit und Konflikte beim Netzanteil, erhalten. Dagegen können Aspekte, die aus der Notwendigkeit entstanden sind, die Spezifikation in eine syntaktische Form zu bringen, in irgend einer Form implementationsabhängig¹¹ sind oder außerhalb der Sicht der Systemspezifikation liegen, modifiziert werden. Dazu zählen wir generell die Namensgebung einzelner Komponenten sowie beim Datentypanteil die Menge der Objekte und bei der Netzstruktur die Menge der möglichen Ereignisse sowie die Synchronie des Markenflusses, d.h. das Zusammenfassen von Markierungen auf Stellen.

Die Modelle einer AN-Spezifikation, die wir als „*Algebraische Netze*“ oder kurz als „*A-Netze*“ bezeichnen, bestehen neben der Spezifikation zu der sie gehören, aus einem Datentypspezifikationsmorphismus mit der Datentypkomponente der AN-Spezifikation als Domäne, einer Algebra zur Codomäne des Datentypspezifikationsmorphismus, einem CP/T-Netz, das nicht mit dem in der Spezifikation enthaltenen Netz identisch sein muß und einem CP/T-Netzmorphismus. Letzterer beschreibt den notwendigen Zusammenhang beider Netze. Durch diese, gegenüber der üblicherweise erzwungenen Identität, lose Koppelung der Netze in Spezifikation und Modell, sind

⁹Es existieren zu jeder Spezifikation mehrere initiale Modelle, die sich aber nur durch die Benennung der Komponenten unterscheiden.

¹⁰d.h. mit individuellen Token arbeitende

¹¹der Begriff der Implementation wird hier im Sinne von Verfeinerung verstanden und bezieht sich nicht auf die Transformation auf eine konkrete Hardware

sowohl Änderungen als auch Ergänzungen der Graphstruktur (\rightarrow 1.1.4) bei der Modellbildung möglich.

In einem weiteren Aspekt, dessen Charakter allerdings weniger grundlegend ist als der vorangehend diskutierte, verallgemeinern Algebraische Netze die übliche Modellbildung höherer Petrinetze mit algebraisch spezifizierten Tokentypen. Bei A-Netzen ist nicht die enthaltene Algebra, sondern nur ihr Vergißbild bezüglich eines in der Definition enthaltenen Spezifikationsmorphismus (\rightarrow 6.1.11), eine Algebra zur Datentypkomponente der Systemspezifikation. Handelt es sich bei dem enthaltenen Spezifikationsmorphismus um die Identität, wie in allen Beispielen dieses Kapitels, sind Vergißbild und Algebra gleich und man erhält den üblichen Zusammenhang.

Durch die vergleichsweise lose Anbindung der Algebrakomponente eines A-Netzes an die Datentypkomponente der zugehörigen Systemspezifikation, kann die Algebra Operationen und Trägermengen enthalten die keine Entsprechung in der Datentypkomponente haben. Darüber hinaus verfügt sie aber über alle in der Datentypkomponente der Systemspezifikation beschriebenen Eigenschaften. Zum Tragen kommen die Erweiterungsmöglichkeiten der Algebra hauptsächlich in Kapitel 3, wo sie eine Voraussetzung für die Vergißfunktorkonstruktion (\rightarrow 3.1.4) und die Verträglichkeit der Komposition von Systemspezifikationen mit deren Semantiken (\rightarrow 3.3.7) bildet.

Die nachfolgend definierten Algebraischen Netze sind mit den aus der Literatur bekannten „*Colored Nets*“ vergleichbar. Dabei übernehmen Terme mit Variablen über der Algebra des A-Netzes die Rolle der Funktionen an den Kanten der Netzstruktur in [Jen81] bzw. die der „ML“-spezifizierten Operationen in [HJS90].

2.2.1 Definition (Algebraisches Netz):

Ein „*Algebraisches Netz*“ (kurz „*A-Netz*“) ist ein Quintupel

$$(ANS, g, A, N^o, eval)$$

mit den folgenden Eigenschaften:

- (I) $ANS = (Spec, Cond, N)$ ist eine AN-Spezifikation,
- (II) $g: Spec \rightarrow Spec'$ ist ein Spezifikationsmorphismus,
- (III) A ist eine $Spec'$ -Algebra,
- (IV) N^o ist ein CP/T-Netz mit einer endlichen Stellen- und Transitionsmenge, das die Gleichung:

$$\mathcal{P}^A(N^o) = A(X)$$

erfüllt, in der $A(X)$ die bezüglich der Inklusion $Spec' \subset Spec'(X)$ frei konstruierte A -Quotiententermalgebra bezeichnet (\rightarrow 6.2.17),

(V) $eval: N \rightarrow N^o$ ist ein CP/T-Netzmorphimus mit

$$\mathcal{P}^A(eval) = (g^x, eval_{\mathcal{V}(A(X))}): T_{OP(X)} \rightarrow A(X),$$

wobei $g^x: Spec(X) \rightarrow Spec'(X)$ (\rightarrow 6.3.4, $g^x = \mathcal{T}^S(g)$) die kanonische Erweiterung von g bezeichnet und $eval_{\mathcal{V}(A(X))}$ den Evaluierungsmorphismus der Signaturalgebren zu $Spec(X)$,

(VI) für alle Transitionen t', t'' aus N und t aus N^o gilt:

$$eval^T(t')(t) > 0 \wedge eval^T(t'')(t) > 0 \implies g^*(Cond(t')) = g^*(Cond(t'')),$$

wobei g^* die kanonische Fortsetzung der g -generierten Abbildung zwischen Gleichungen (\rightarrow 6.2.3) auf Gleichungsmengen und $eval^T$ die Transitionskomponente von $eval$ bezeichnet

(VII) und für alle Transitionen t aus N^o und Variablen $x_s, y_{s'}$ aus $var(t)$ gilt:

$$s \neq s' \implies x \neq y$$

wobei s, s' die jeweiligen Sorten der Variablen bezeichnen.

□

In Analogie zu den generierten Elementen einer Algebra im Rahmen von algebraischen Spezifikationen¹² bezeichnen wir die Stellen und Transitionen im Bild der Evaluierungsfunktion $eval$ als generiert.

2.2.2 Beispiel (Ein Modell der Alternating-Bit-Protokoll-Spezifikation):

Das A-Netz in Abbildung 2.2 ist ein Modell¹³ der AN-Spezifikation in Abbildung 2.1. Es gibt 10^n verschiedene Mitteilungen, die mit den Werten aus $-10^n, \dots, 10^n$ als Kontrollbits versehen werden. Nach dem Empfang einer korrekt markierten Mitteilung durch den Empfänger bzw. dem Empfang einer korrekt markierten Bestätigung durch den Sender, ändert sich das Vorzeichen des jeweiligen Kontrollbits.

Die Netzstruktur des Modells weist zwei wesentliche Unterschiede gegenüber der Struktur der Spezifikation auf. Erstens existiert nur ein Kanal, der sowohl für den Transport von Mitteilungen als auch von Bestätigungen dient. Der zweite Unterschied ist von größerer Bedeutung: Mitteilungen und Bestätigungen können verloren gehen.¹⁴ Die Existenz von Modellen, die in dieser Weise fehlerbehaftet arbeiten, ist die Motivation von großen Teilen des spezifizierten Protokolls.

□

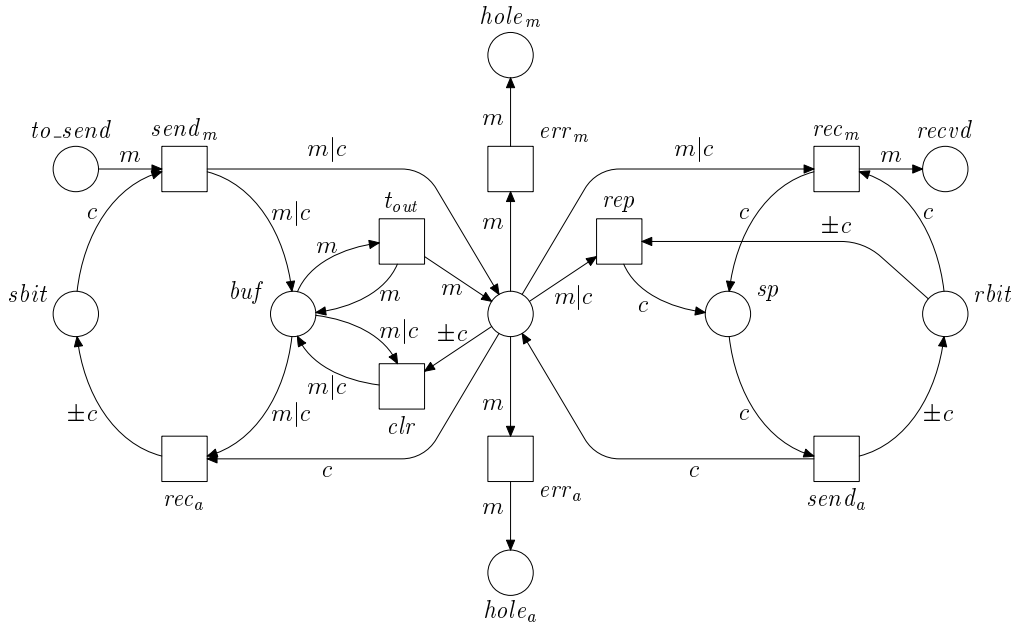
¹²die im Bild der jeweiligen Evaluierungsfunktion liegen

¹³Die Reihenfolge der Beispielkomponenten entspricht hier, wie in allen übrigen Beispielen, dem in Definition 2.2.1 angegebenen Quintupel.

¹⁴in diesem Fall durch das Schalten der Transitionen „ err_m “ und err_a

ABP-Modell 1: Alternating-Bit-Protokoll, *id*,

$$\begin{aligned} & \{ \{ \{ mess_0, \dots, mess_{10^n} \}, \{-10^n, \dots, 10^n\}, \\ & \quad \{ mess_0, \dots, mess_{10^n} \} \times \{-10^n, \dots, 10^n\} \} \\ & \{ |, \downarrow, \Delta, \pm \text{ mit} \\ & \quad m | x = (m, x), \downarrow m_i = i, \Delta x = x^2/x, \pm x = -1 * x \} \end{aligned}$$



$$\begin{aligned} id, \quad e^P : chan_{ms} &\mapsto chan & eval_{A(X)} : m.c &\mapsto ap(m, c) \\ &chan_{ac} &\mapsto chan &cod(m) &\mapsto mc(m) \\ &\vdots &\mapsto \vdots &|c| &\mapsto ab(c) \\ & & &-c &\mapsto cg(c) \end{aligned}$$

Abbildung 2.2: Modell des Alternating-Bit Protokolls mit fehlerhaftem Kanal.

Die Menge der Modelle einer Systemspezifikation wird durch Abbildungen, die wir als „*A*-Netzmorphismen“ bezeichnen, strukturiert. Dabei handelt es sich um Paare, bestehend aus einem Algebra- und einem CP/T-Netzmorphismus. Beide Bestandteile bewahren die Eigenschaften ihrer jeweiligen Domäne, d.h. der Algebra-Morphismus ist kompatibel mit den Algebraoperationen (\rightarrow 6.2.4), während der CP/T-Netzmorphismus den Effekt, die Unabhängigkeit und den kausalen Zusammenhang von Transitionen bewahrt (\rightarrow 1.1.4).

A-Netzmorphismen erfüllen im weiteren eine Reihe von Funktionen, mit ihrer Hilfe lassen sich die initialen Modelle einer Systemspezifikation (\rightarrow 2.2.6) bestimmen, sie können ein A-Netz als Modell einer Systemspezifikation auszeichnen und sie spielen als Teile generalisierter Morphismen in Kapitel 3 eine zentrale Rolle bei der Definition der Semantiken für die Komposition und Parameterisierung von Systemspezifikationen.

Wie schon bei den AN-Spezifikationen und den A-Netzen, wird der Zusammenhang von Datentyp- und Netzabbildung bei A-Netzmorphismen (\rightarrow 2.2.3) mit Hilfe der Projektion \mathcal{P}^A (\rightarrow 2.2.3 III) beschrieben. Um die Existenz initialer Modelle¹⁵ zu ermöglichen, müssen die A-Netzmorphismen mit den Evaluierungsmorphismen der A-Netze, d.h. den CP/T-Netzmorphismen, die den Zusammenhang der Netzstrukturen in Systemspezifikation und Modell beschreiben, verträglich sein (\rightarrow 2.2.3 IV). Für den Algebra-Teil eines A-Netzmorphismus ist diese Forderung immer erfüllt, da für jeden Algebramorphismus $f: A \rightarrow B$ gilt: $eval_A \circ f = eval_B$ (\rightarrow 6.1.1, 6.1.9)).

2.2.3 Definition (A-Netzmorphismus):

Gegeben zwei A-Netze $AN_i = (ANS_i, g_i, A_i, N_i^o, eval_i)$. Ein Paar (f^G, f^N) bildet einen A-Netzmorphismus

$$(f^G, f^N): AN_1 \rightarrow AN_2,$$

wenn folgende Bedingungen erfüllt sind:

- (I) $f^G: A_1 \rightarrow A_2$ ist ein generalisierter Algebramorphismus,
- (II) $f^N: N_1^o \rightarrow N_2^o$ ist ein CP/T-Netzmorphismus,
- (III) für die Projektion von f^N auf den Algebraanteil gilt:

$$\mathcal{P}^A(f^N) = \mathcal{T}^A(f^G): A_1(X) \rightarrow A_2(X),$$

wobei \mathcal{T}^A (\rightarrow 6.3.9) die Transformation von generalisierten Algebramorphismen in generalisierte Morphismen zwischen Algebren mit Variablen bezeichnet,

- (IV) $f^N \circ eval_1 = eval_2$ und
- (V) für die Einschränkung von f^N auf $\mathcal{DC}_{eval_1}(co)$ gilt:

$$f^N \Big|_{\mathcal{DC}_{eval_1}(co)} \in Mor(\mathcal{CPTN})[\mathcal{DC}_{eval_1}(co), \mathcal{DC}_{eval_2}(co)].$$

□

Bedingung 2.2.3.V erzwingt die Abbildung nichtgenerierter Elemente aus der Domäne eines A-Netzmorphismus auf nichtgenerierte Elemente seiner Codomäne. Notwendig wird diese Einschränkung für die Definition von Vergißfunktoren in Abschnitt 3.2. Diese Funktoren bilden zwischen Modellkategorien unterschiedlicher Systemspezifikationen ab und „vergessen“ dabei Teile der generierten Modellstruktur (\rightarrow 3.1.4).

¹⁵deren Datentyp und Netzstruktur vollständig generiert sind

2.2.4 Beispiel (Ein weiteres Modell des Alternating-Bit-Protokolls):

Das A-Netz in Abbildung 2.3 zeigt ein weiteres Modell des Alternating-Bit-Protokolls (\rightarrow 2.1). Im Unterschied zu dem strukturell ähnlichen Modell in Beispiel 2.2 sind die Stellen dieses Modells vollständig von seiner Evaluierungsfunktion überdeckt, d.h. sie liegen im Bild der Netzstruktur zur Spezifikation.

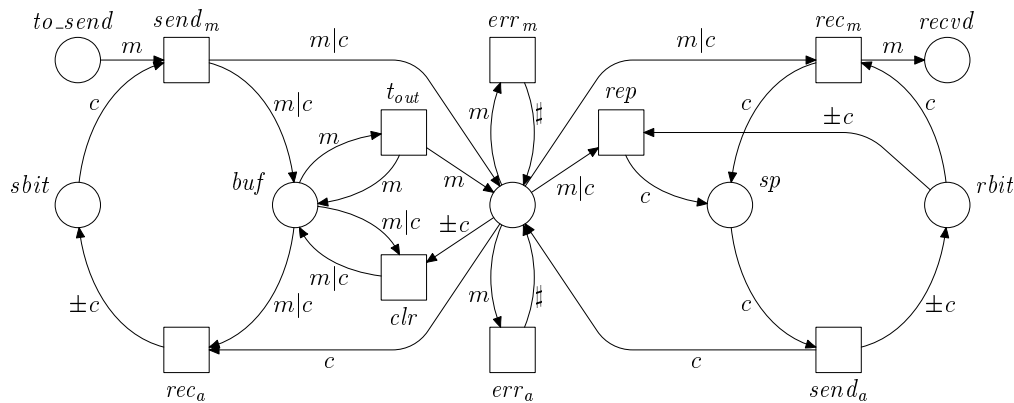
ABP Modell 2: Alternating-Bit Protokoll, ,

$$\{\{String, \mathbb{Z}, String \times \mathbb{Z}\}$$

$$\{ |, \downarrow, \Delta, \pm \text{ mit}$$

$$m | x = (m, x), \Delta x = x^2/x, \pm x = -1 * x$$

$$\downarrow s = \left\{ \begin{array}{ll} 0, & s = \emptyset; \\ \downarrow s' * 256 + \text{ascii}(a), & s = s'a; \end{array} \right\}$$



$$\begin{array}{ll} id, & e^P : chan_{ms} \mapsto chan \quad eval_{A(X)} : m.c \mapsto ap(m, c) \\ & chan_{ac} \mapsto chan \quad cod(m) \mapsto mc(m) \\ & \vdots \mapsto \vdots \quad |c| \mapsto ab(c) \\ & \quad \quad \quad -c \mapsto cg(c) \end{array}$$

Abbildung 2.3: Ein Modell, das falsche Mitteilungen produzieren kann.

Die strukturellen Unterschiede zwischen den Modellen 1 und 2 schlagen sich auch in deren Verhalten nieder. Während in Modell 1 lediglich Nachrichten verloren gehen können, was aufgrund des speziell dafür ausgelegten Protokolls (\rightarrow 2.1.4) solange nicht tragisch ist wie überhaupt noch Nachrichten durchkommen, können darüber hinaus in Modell 2 Nachrichten und Bestätigungen in den Kanal gelangen¹⁶, die vom Absender bzw. vom Empfänger nie versendet wurden. Dieser Fall wurde bei der Spezifikation nicht berücksichtigt und kann in ungünstigen Fällen, d.h. wenn z.B. zufällig bekannte Kontrollbits durch „err_a“ in den Kanal gelangen, die Kommunikation völlig aus dem Takt bringen.

¹⁶durch „err_m“ oder „err_a“ bei denen die Variable # beliebig belegt werden kann

Aufgrund von Bedingung 2.2.3.V kann es vom Modell 1 in das Modell 2 keinen A-Netzmorphismus geben, da die Stellen „ $hole_m$ “ und „ $hole_a$ “ in Modell 1 nicht generiert sind, nichtgenerierte Stellen wieder auf nichtgenerierte Stellen abgebildet werden müssen und in Modell 2 alle Stellen generiert sind.

Auch in umgekehrter Richtung, also von Modell 2 in das Modell 1 kann es keinen A-Netzmorphismus geben. Grund dafür sind die Transitionen „ err_m “ und „ t_{out} “, die in Modell 2 beide mit der gleichen, den Kanal simulierende Stelle, in Modell 1 aber mit unterschiedlichen Stellen verbunden sind. Jeder A-Netzmorphismus, der die Transition „ t_{out} “ auf die gleichnamige Transition im Modell 1 abbildet, muß, da seine Netzkomponente mit den Markenflußmorphismen verträglich ist, auch die den Kanal simulierende Stelle wieder auf die entsprechende Stelle in Modell 1 abbilden. Diese Abbildung ist mit dem Effekt der Transition „ err_m “ als Bild der gleichnamigen Transition aus Modell 2 unverträglich. Der Versuch die Transition „ t_{out} “ oder „ err_m “ auf andere Transitionen im Modell 1 abzubilden scheitert aus den selben Gründen.

□

Die Modelle und Modellmorphismen einer Systemspezifikation lassen sich zu einer Modellkategorie zusammenfassen. Identitäten und die Komposition von Morphismen werden dabei komponentenweise gebildet, d.h. separat auf der Algebra- und der Netzkomponente.

2.2.5 Korollar (Modellkategorien):

Gegeben eine AN-Spezifikation ANS , dann bilden die A-Netze mit ANS als erster Komponente zusammen mit den dazwischen verkehrenden A-Netzmorphismen eine Kategorie (genannt $\mathcal{MOD}(ANS)$). Die Identitäten und die Komposition von Morphismen werden komponentenweise in den Kategorien der jeweiligen Komponenten gebildet.

□

Jede Modellkategorie einer Systemspezifikation enthält ein initiales Modell.¹⁷ Es wird aus dem initialen Modell der Datentypspezifikation (\rightarrow [EM85] 2.4–2.9), der Quotiententermalgebra und der Netzstruktur der Systemspezifikation gebildet, indem in der Netzstruktur alle Terme an den Kanten durch die entsprechenden Äquivalenzklassen ersetzt werden. Die Evaluierungsfunktion von der Netzstruktur der Systemspezifikation in die Netzstruktur des initialen Modells besteht dementsprechend aus den Identitäten als Transitions- und Stellenmorphismen sowie aus der Evaluierungsfunktion von der Termalgebra als initialem Objekt des Datentyps in die Quotiententermalgebra.¹⁸

¹⁷das initiale Modell ist nur bis auf Isomorphie eindeutig, d.h. es existiert genau genommen eine Menge isomorpher initialer Modelle

¹⁸jede Algebra zu einer Spezifikation ist gleichzeitig eine Algebra zu deren Signatur

Entsprechend der Definition initialer Objekte in Kategorien, (\rightarrow 5.1.9) existiert vom initialen Modell einer Systemspezifikation in jedes andere Modell genau ein Modellmorphismus. Dessen Transitions- und Stellenkomponente ist durch die entsprechende Abbildung der Evaluierungsfunktion des Zielmodells (\rightarrow 2.2.1) gegeben, als Algebra-komponente dient der, aus dem Spezifikationsmorphimus des Zielmodells und der Evaluierungsfunktion von der Quotiententermalgebra in die Algebra des Zielmodells gebildete, generalisierte Morphismus. Aus der Eindeutigkeit der Algebromorphismen mit der Quotiententermalgebra als Domäne ergibt sich, zusammen mit Bedingung (III) der A-Netzmorphimus-Definition (\rightarrow 2.2.3), die Eindeutigkeit der Modellmorphimen vom initialen in jedes andere Modell einer AN-Spezifikation, d.h. „Es kann nur einen geben!“¹⁹

Notation: Im folgenden notieren wir an allen Stellen wo ein generalisierter Algebromorphismus erwartet wird, generalisierte Morphimen mit Identitäten als erster Komponente durch ihre zweite Komponente, d.h. wir notieren (id_{Spec}, f^A) durch f^A .

2.2.6 Satz (Initiale Modelle):

Gegeben eine AN-Spezifikation

$$ANS = \left(Spec, Cond, T^{\otimes} \begin{array}{c} \xrightarrow{Pre} \\ \xrightarrow{Post} \end{array} P^{\otimes} \otimes T_{OP(X)} \right),$$

dann ist das A-Netz

$$\left(ANS, id_{Spec}, T_{Spec}, T^{\otimes} \begin{array}{c} \xrightarrow{Pre_I} \\ \xrightarrow{Post_I} \end{array} P^{\otimes} \otimes T_{Spec(X)}, eval_I \right)$$

mit $eval_I = (id_{T^{\otimes}}, id_{P^{\otimes}}, eval_{T_{Spec(X)}})$ und

$$\begin{aligned} Pre_I &\stackrel{\text{def}}{=} id_{P^{\otimes}} \otimes eval_{T_{Spec(X)}} \circ Pre \\ Post_I &\stackrel{\text{def}}{=} id_{P^{\otimes}} \otimes eval_{T_{Spec(X)}} \circ Post \end{aligned}$$

das initiale Modell der Kategorie $\mathcal{MOD}(ANS)$.

□

(Beweis \rightarrow Anhang S. 197)

¹⁹Zitat aus dem Film „Der Highlander“

2.3 Operationale A-Netz-Semantik

Im folgenden Abschnitt ordnen wir jedem A-Netz ein Verhalten in Form seiner Abläufe zu. Unter Abläufen verstehen wir in diesem Zusammenhang Prozeßterme, die als Semantik der CP/T-Netze (\rightarrow 1.3) eingeführt wurden. Dabei handelt es sich um eine algebraische Form der ursprünglich für B/E-Systeme eingeführten und später auf Stellen/Transitions-Netze erweiterten Prozesse,²⁰ basierend auf Konstruktionen²¹ von P. Degano, J. Meseguer und U. Montanari ([DMM89]). Diese, aus Sicht der A-Netze initiale Semantik, definiert eine lose Semantik für die in 2.1 eingeführten Systemspezifikationen.

Der wesentliche Schritt beim Erzeugen der operationalen Semantik für A-Netze besteht in deren Entfaltung. Dabei wird jede Transition in eine Menge von Paaren, bestehend aus dem Transitionsnamen und einer Belegung der Variablen ihrer Umgebung übersetzt. Diese Belegungen müssen mit den Gleichungen derjenigen Transitionen der Systemspezifikation verträglich sein, die auf die Transition im Modell abgebildet werden.

Eine Besonderheit bei der Auswahl zulässiger Variablenbelegungen im Vergleich zum üblichen Vorgehen bei Entfaltungen höherer Netze (vergl. [Hum89], [Vau86], [Rei91]) besteht darin, daß als Belegungen nicht nur Elemente der Algebra des A-Netzes in Frage kommen, sondern auch Terme mit Variablen über den Operationen der Algebra. Dadurch erhält man abstrakte Belegungen und im weiteren abstrakte Abläufe, die für eine Menge von konkreten, ebenfalls in der operationalen Semantik vorhandenen, Abläufen stehen. Das erlaubt zum einen eine kompakte Darstellung von erwünschtem oder unerwünschtem Verhalten, zum anderen erhält man damit eine Grundlage für die Reduzierung des Zustandsraumes bei der Verifikation²² von Systemeigenschaften.

Den A-Netzmorphismen wird auf Seiten der Semantik eine Transformation der Abläufe des Ursprungsnetzes in die Abläufe des Zielnetzes zugeordnet. Diese Transformationen werden durch Funktoren zwischen den Semantikkategorien beschrieben und erhalten sowohl die sequentielle Abfolge der Schaltschritte sowie deren Halbordnung. In Abschnitt 3.2 wird die hier gegebene Definition auf generalisierte A-Netzmorphismen erweitert und damit die Möglichkeit geschaffen, Abläufe von A-Netzen, auch über Grenzen der Semantik einzelner Systemspezifikationen hinweg, in Beziehung zu setzen.

Wir definieren nachfolgend für jede Transition eines A-Netzes, die Menge der zulässigen Belegungen ihrer Variablen. Zu diesen Variablen zählen neben den Variablen in den Beschriftungen der mit der Transition verbundenen Kanten, bei generierten Transitionen²³ auch die Variablen aus den Gleichungsmengen *Cond* zu Transitionen der AN-Spezifikation, die durch die Evaluierungsfunktion auf die Transition abgebildet

²⁰Beispiel 1.3.4 veranschaulicht die Umwandlung von Prozeßtermen in Prozesse im üblichen Sinn

²¹ $\pi[N]$ u. C-Prozesse

²²etwa durch „Model-Checking“

²³Transitionen im Bild der Evaluierungsfunktion des A-Netzes

werden. Letztere sind in der Praxis zumeist in ersteren enthalten und schränken damit deren mögliche Belegungen ein, es gibt aber sinnvolle Ausnahmen, z.B. wenn Variablen dazu benutzt werden, die Gemeinsamkeiten zulässiger Belegungen anderer Variablen auszudrücken.²⁴

Um zu entscheiden, ob eine Belegung zulässig ist, wird zunächst festgestellt, ob die betreffende Transition generiert ist. Ist das nicht der Fall, kann die Transition uneingeschränkt schalten; es werden alle möglichen Belegungen zugelassen. Andernfalls sind nur die Belegungen der Variablen einer Transition möglich, die jede der Gleichungen zu den Transitionen der AN-Spezifikation erfüllen, die von der Evaluierungsfunktion auf die Transition abgebildet werden.

2.3.1 Definition (Variablenbelegungen):

Sei $((Spec, Cond, N), g, A, N^o, eval)$ ein A-Netz, T^o die Transitionsmenge von N^o , $Spec'$ die Codomäne von g und $EQ_{Spec'}$ die Menge aller Gleichungen zur $Spec'$ -Signatur mit Variablen aus X , dann definieren wir

(I) die Abbildung $Cond^o: T^o \rightarrow 2^{EQ_{Spec'}}$ für alle t aus T^o durch:

$$Cond^o(t) = \bigcup_{t' \in N \mid eval^T(t')(t) > 0} g^*(Cond(t'))$$

wobei g^* die Fortsetzung der g -generierten Abbildung zwischen Gleichungen (\rightarrow 6.2.3) auf Gleichungsmengen und $eval^T$ die Transitionskomponente von $eval$ bezeichnet und

(II) die Abbildung

$$Ass: T^o \rightarrow \bigcup_{t \in T^o} 2^{[var(t) \cup var(Cond^o(t)) \rightarrow A(X)]}$$

für alle t aus T^o durch:

$$Ass(t) \stackrel{\text{def}}{=} \{ ass: var(t) \cup var(Cond^o(t)) \rightarrow A(X) \mid \\ \forall (X', l, r) \in Cond^o(t): eval^{ass}(l) = eval^{ass}(r) \},$$

wobei $eval^{ass}$ (\rightarrow 6.1.9) die Evaluierungsfunktion mit Variablenbelegung von der $Spec'$ -Termalgebra mit Variablen nach $A(X)$ bezeichnet. □

Aus jedem Algebra-Morphismus läßt sich eine Transformation aller möglichen Belegungen der Variablen seiner Ursprungsalgebra in Belegungen seiner Zielalgebra gewinnen. Dazu werden Urbild (Variablenmenge) und Bild (Algebra) der Belegungen mit Hilfe des Algebromorphismus abgebildet.

²⁴Die Variable „ m “ in den Gleichungen zur Transition „ em “ der Transportspezifikation (\rightarrow S. 93) bildet ein Beispiel hierfür.

Als letzten Schritt auf dem Weg zu einer operationalen Semantik verknüpfen wir die Transitionen mit ihren Variablenbelegungen zu freien abelschen Gruppen und Transitionsmorphismen mit den, sich aus ihrer Algebrakomponente ergebenden, Transformationen zu Gruppenmorphismsen. In Analogie zu der in Abschnitt 1.1 eingeführten Verknüpfung „ \oplus “ notieren wir diese Verknüpfung in Operatorschreibweise, d.h. „Gruppe \odot Belegungen“ und „Morphismus \odot Transformation“. Der dadurch implizierte Zusammenhang zwischen den Operationen rechtfertigt sich zum einen durch deren analoge Verwendung, zum anderen durch die Ähnlichkeit ihrer Definitionen. Abgesehen von der Tatsache, daß einmal Algebren und zum anderen Funktionen mit freien abelschen Gruppen verknüpft werden, bestehen die Unterschiede darin, daß „ \odot “ nur partiell definiert ist, d.h. nur eine Teilmenge des kartesischen Produkts: „Transitionen \times Variablenbelegungen“ zur Generierung der freien abelschen Gruppe benutzt. Diese Unterschiede sind der Grund dafür, daß die Operation \odot im Gegensatz zu \oplus nicht als Funktor definiert wird, obwohl sie ebenfalls die Identitäten und die Komposition von Morphismen bewahrt.

2.3.2 Definition (Verknüpfung von Transitionen mit Belegungen):

Gegeben ein $\mathcal{MOD}(ANS)$ -Morphismus

$$\left((f^S, f^A), (f^T, f^P, f^{G_X}) \right): AN_1 \rightarrow AN_2,$$

und $AN_i = (ANS, g, A_i, (T_i^\otimes, P_i^\otimes, A_i(X), Pre_i, Post_i), eval_i)$, dann definieren wir:

(I) eine freie abelsche Gruppe $T_i^\otimes \odot Ass_i$ durch:

$$T_i^\otimes \odot Ass_i \stackrel{\text{def}}{=} \{ (t, ass) \mid t \in T_i \wedge ass \in Ass_i(t) \}^\otimes,$$

(II) eine Abbildung

$$f^{As}: \bigcup_{t \in T_1} Ass_1(t) \rightarrow \bigcup_{t \in T_2} Ass_2(t),$$

die für alle Variablenbelegungen ass aus $dom(f^{As})$ das nachfolgende Diagramm kommutativ ergänzt

$$\begin{array}{ccc} X' & \xrightarrow{\cong} & \{ x_{f^{SO}(s)} \mid x_s \in X' \} \\ \text{ass} \downarrow & \curvearrowright & \downarrow f^{As}(ass) \\ A_1(X) & \xrightarrow{f^{G_X}} & A_2(X) \end{array} \quad 26$$

wobei f^{SO} die Sortenabbildung von f^S bezeichnet und

(III) einen monotonen Morphismus in den freien abelschen Gruppen

$$f^T \odot f^{G_X}: T_1^\otimes \odot Ass_1 \rightarrow T_2^\otimes \odot Ass_2$$

$$\text{durch: } f^T \odot f^{G_X}(t, ass) \stackrel{\text{def}}{=} \bigoplus_{t' \in T_2} f^T(t)(t') * (t', f^{As}(ass))$$

für all Paare (t, ass) in $T_1^{\otimes} \odot Ass_1$.

□

Mit Hilfe der vorangegangenen Definitionen läßt sich den A-Netzen und A-Netzmorphismen in zwei Schritten eine Semantik zuordnen. Zunächst wird jedes A-Netz in ein CP/T-Netz entfaltet und jeder A-Netzmorphismus in ein CP/T-Netzmorphismus zwischen den Entfaltungen seiner A-Netze transformiert. Im zweiten Schritt werden die so entstandenen CP/T-Netze auf die Kategorie ihrer Prozesse abgebildet (\rightarrow 1.3.1, 1.3.3) und die CP/T-Netzmorphismen auf Funktoren zwischen Prozeßkategorien (\rightarrow 1.3.7).

Bei der Entfaltung eines A-Netzes werden die Transitionen seiner Netzstruktur mit den zulässigen Variablenbelegungen verknüpft. Die Anschriften an den Kanten der daraus entstandenen Transitionen (Paare der Form (Transitionsname, Belegung)), ergeben sich aus der Belegung der Variablen in den Termen der entsprechenden Kanten im A-Netz. Stellen und die um X erweiterte Algebra des A-Netzes werden in die Entfaltung übernommen.

2.3.3 Definition (A-Netz-Entfaltung):

Sei

$$AN = (ANS, g, A, T^{\otimes} \xrightleftharpoons[Post]{Pre} P^{\otimes} \otimes A(X), eval)$$

ein A-Netz, dann definieren wir dessen „CP/T-Netz-Entfaltung“

$$UF_{AN} \stackrel{\text{def}}{=} T^{\otimes} \odot Ass \xrightleftharpoons[Post_u]{Pre_u} P^{\otimes} \otimes A(X)$$

$$\text{durch:} \quad \begin{aligned} Pre_u(t, ass) &\stackrel{\text{def}}{=} id \otimes \widetilde{ass}(Pre(t)) \\ Post_u(t, ass) &\stackrel{\text{def}}{=} id \otimes \widetilde{ass}(Post(t)) \end{aligned}$$

für alle Paare (t, ass) in $T^{\otimes} \odot Ass$, wobei \widetilde{ass} die kanonische Erweiterung von ass auf Terme mit Variablen bezeichnet (\rightarrow 6.3.2).

□

2.3.4 Beispiel (Ein Prozeßterm zum Modell des AB-Protokolls in Abbildung 2.2):

Der „abstrakte“ Prozeßterm in Abbildung 2.4 ist Teil der Semantik des A-Netzes in Abbildung 2.2. Wir bezeichnen ihn als abstrakt, weil er die Variable „m“ enthält, die mit einer Mitteilung substituiert einen konkreten Ablauf des Modells ergibt. Auf diese Weise lassen sich alle Kommunikationszyklen, die fehlerfrei und ohne Mehrfachversendung von Informationen auskommen, aus dem angegebenen Prozeßterm erzeugen. Diese konkreten Prozeßterme sind ebenfalls alle in der Semantik des A-Netzes enthalten. Wie sich diese Prozeßterme in die für Petrinetze übliche graphische Prozeßdarstellung übertragen lassen, wird exemplarisch im Beispiel 1.3.4 gezeigt.

²⁶der Isomorphismus existiert aufgrund von 2.2.1(VII)

Stellen, die in Spalten mit dem Operator „ \oplus “ zusammengefaßt sind, bilden erreichbare Markierungen des Prozesses. Durch die sequentielle Verknüpfung „;“ (\rightarrow S. 15) sind sie mit Paaren bestehend aus einer Transition und einer Variablenbelegung verbunden. Diese Bezeichner für Transitionen der A-Netz-Entfaltung überführen jeweils die links von ihnen stehende Markierung in die rechts stehende. Aus Platzgründen mußte der Prozeßterm in Abbildung 2.4 über mehrere Zeilen gefaltet werden.

$$\begin{aligned}
& \left[\begin{array}{c} (to_send, m) \\ \oplus \\ (sbit, mc(m)) \\ \oplus \\ (rbit, mc(m)) \end{array} \right] ; [(send_m, ass: c \mapsto mc(m))] ; \left[\begin{array}{c} (chan, ap(m, mc(m))) \\ \oplus \\ (rbit, mc(m)) \\ \oplus \\ (buf, ap(m, mc(m))) \end{array} \right] ; \\
& \left[(rec_m, ass: c \mapsto mc(m)) \right] ; \left[\begin{array}{c} (recvd, m) \\ \oplus \\ (sp, mc(m)) \\ \oplus \\ (buf, ap(m, mc(m))) \end{array} \right] ; \\
& \left[(send_a, ass: c \mapsto mc(m)) \right] ; \left[\begin{array}{c} (chan, mc(m)) \\ \oplus \\ (rbit, cg(mc(m))) \\ \oplus \\ (buf, ap(m, mc(m))) \end{array} \right] ; \\
& [(rec_a, ass: c \mapsto mc(m))] ; \left[\begin{array}{c} (sbit, cg(mc(m))) \\ \oplus \\ (rbit, cg(mc(m))) \end{array} \right]
\end{aligned}$$

Abbildung 2.4: Prozeßterm zum A-Netz in Abb. 2.2. □

Die Abbildung von A-Netzmorphismen auf CP/T-Netzmorphismen ergibt sich unmittelbar aus der Entfaltung der A-Netze. Von den drei für die CP/T-Netzmorphismen relevanten Netzkomponenten ändert sich nur die Transitionskomponente und die dabei neu hinzukommenden Variablenbelegungen lassen sich mit Hilfe der Algebra-komponente der A-Netzmorphismen abbilden.

2.3.5 Definition (A-Netzmorphismen \rightarrow CP/T-Netzmorphismen):

Gegeben eine AN-Spezifikation ANS , dann definieren wir eine Abbildung

$$\mathcal{UF}_{ANS}: Mor(\mathcal{MOD}(ANS)) \rightarrow Mor(\mathcal{CPTN})$$

$$\text{durch: } \mathcal{UF}_{ANS}(f) \stackrel{\text{def}}{=} (f^T \odot f^{G_X}, f^P, f^{G_X}): \mathcal{UF}_{AN_1} \rightarrow \mathcal{UF}_{AN_2}$$

für alle $\mathcal{MOD}(ANS)$ -Morphismen $f = (f^G, (f^T, f^P, f^{G_X})): AN_1 \rightarrow AN_2$. □

Die Semantik von Objekten spiegelt deren Eigenschaften wieder. Liegen die Objekte als Kategorie vor, d.h. werden sie von eigenschaftserhaltenden Abbildungen strukturiert, muß eine sinnvoll gewählte Semantik diese Struktur bewahren. Lemma 2.3.6 zeigt dies für den Übergang von A-Netzen in ihre Entfaltungen. Die Funktoreigenschaften des zweiten Schritts bei der Semantikkonstruktion von den CP/T-Netzen in die Prozeßtermkategorien wurden schon in Lemma 1.3.11 gezeigt, sodaß die Kombination beider Schritte in 2.3.7 ebenfalls einen Funktor ergibt.

2.3.6 Lemma (Eigenschaft der Entfaltungen):

Für jede AN-Spezifikation ANS bildet \mathcal{UF}_{ANS} einen Funktor von $MOD(ANS)$ nach \mathcal{CPTN} . □

Beweis: Die Behauptung ergibt sich als Spezialfall (Einschränkung auf generalisierte Morphismen zwischen den Objekten einer Modellkategorie) aus Definition 3.2.4 und Lemma 3.2.6.

2.3.7 Definition (Operationale Semantik für AN-Spezifikationen):

Wir definieren die operationale Semantik einer AN-Spezifikation ANS als Funktor $\mathcal{OS}_{ANS}: MOD(ANS) \rightarrow \mathcal{PTS}$ durch:

$$\mathcal{OS}_{ANS} \stackrel{\text{def}}{=} \mathcal{PC} \circ \mathcal{UF}_{ANS}.$$

Dabei bezeichnet $\mathcal{PC}: \mathcal{CPTN} \rightarrow \mathcal{PTS}$ den Semantikfunktor von den CP/T-Netzen in die erweiterte Prozeßtermsemantik (\rightarrow 1.3.11). □

Indem wir den A-Netz-Modellen einer Systemspezifikation die Kategorie ihrer Prozesse und den A-Netzmorphismen Funktoren zwischen Prozeßkategorien als (initiale) Semantik zugeordnet haben, ergibt sich aus der Kombination von Modellbildung und Prozeßsemantik der Modelle die (lose) operationale Semantik der Systemspezifikationen. Mit den Entfaltungen der A-Netze ließe sich noch eine Semantikebene zwischen die Modelle und deren Prozesse legen. Da die Entfaltungen aber weder den Vorteil einer kompakten Repräsentation bieten, wie die A-Netze, noch deren nebenläufiges Verhalten in der Weise veranschaulichen können wie die Prozesse, sind sie für sich genommen von geringem Interesse.

Operationen auf Spezifikationen

In den vorangehenden Kapiteln haben wir Systemspezifikationen, deren Modelle und operationale Semantik eingeführt. Damit sind wir in der Lage, kleinere nebenläufige Systeme zu beschreiben und deren Eigenschaften zu ermitteln. Spezifikationen von realen und damit in aller Regel komplexen Systemen allein mit diesen Mitteln, werden dagegen sehr schnell unüberschaubar.

Strukturierungsmöglichkeiten, die Abhilfe schaffen könnten, existieren zwar für jede der beiden Komponenten einer Systemspezifikation: Operationen zur Komposition und Parameterisierung bei der Datentypbeschreibung (z.B. [EM85]) und Komposition über Diagramme für die Netzstruktur, es fehlen aber sowohl Regeln zu deren Kombination, als auch korrespondierende Operationen auf der Semantikebene. Dabei kommt gerade dem zweiten Punkt große Bedeutung zu. Der Weg über eine rein syntaktische Strukturierung, wie er mit der Zerlegung des Netzgraphen in einigen rechnergestützten Spezifikationswerkzeugen gegangen wird (z.B. in [HJS90, JCHH91]), führt zwar zu einer notationellen Erleichterung und — eine sinnvolle Separierung vorausgesetzt — auch zu mehr konzeptioneller Klarheit, Eigenschaften und Systemverhalten müssen jedoch auf der „flachen“ Spezifikation ermittelt werden. Eine Reduktion der Beschreibung durch Faltung des Netzgraphen kommt ebenfalls nicht ernsthaft in Betracht. Einmal ist der dadurch erzielbare Effekt durch die „Uniformität“ der Netzstruktur beschränkt.¹ Auf der anderen Seite sollte die Beschreibung eines Systems allein von dessen Komponenten und deren Eigenschaften abhängen. Andernfalls leidet nicht nur ihre Verständlichkeit, sondern auch die Flexibilität in Bezug auf Systemänderungen.

Im aktuellen Kapitel wollen wir mit der Komposition von AN-Spezifikationen und deren Parameterisierung, Möglichkeiten zur strukturierten Spezifikation großer Systeme schaffen. Dabei erlaubt die Komposition, über Schnittstellen (ausgezeichnete Subspezifikationen), das Zusammensetzen von Systemspezifikationen aus überschaubaren Einheiten, während die Parameterisierung eine Anpassung vorhandener Spezifikationen an unterschiedliche Kontexte ermöglicht. Beide Konstruktionen haben ihre Entsprechungen, sowohl in der Modell-, als auch in der operationalen Semantik und ermöglichen dadurch die Transformation von lokalen in globale Systemeigenschaften.

Komposition und Parameterisierung von Systemspezifikationen in der im weiteren vorgestellten Form, verstehen wir eher als eine exemplarische Anwendung der im

¹Die Transitionen in den Modellen einer Spezifikation, bewegen in jedem ihrer Schaltmodi die gleiche Anzahl von Token. Damit lassen sich nur die Transitionen aufeinanderfalten, die mit den selben Stellen verbunden sind und bei denen entsprechende Kanten die gleiche Anzahl von Termen in ihrer Beschriftung aufweisen.

Rahmen dieser Arbeit gelegten Basis — ohne damit deren Verwendbarkeit in Frage stellen zu wollen — denn als deren endgültige praktische Ausprägung. Eine zukünftige Weiterentwicklung in Hinblick auf die Erfordernisse des Softwareengineering liegt im Rahmen dieses Ansatzes. Ein Beispiel in dieser Richtung ist die Datentypbeschreibung mit Hilfe von „Spezifikationsmodulen“, wie sie in [EM90] eingeführt wurde.

Das folgende Kapitel gliedert sich in vier Abschnitte: Abschnitt 3.1 schafft mit der Definition von semantikverträglichen Abbildungen zwischen AN-Spezifikationen, der Konstruktion von Colimites zu den darüber gebildeten Diagrammen und der Zuordnung von Vergißfunktoren zwischen Modellkategorien zu diesen Abbildungen, die Grundlagen der Komposition und Parameterisierung auf der Spezifikationsebene. Mit der Erweiterung der Modellmorphismen (\rightarrow 2.2.3) zu die Grenzen der Modellkategorien überschreitenden Abbildungen, der Konstruktion von Colimites zu entsprechenden Diagrammen und der Anpassung der operationalen Semantik an diese Erweiterung, schaffen wir im Abschnitt 3.2 die semantischen Grundlagen der folgenden Abschnitte. Syntax und Semantik der Komposition von Systemspezifikationen sind Inhalt von Abschnitt 3.3. Die entsprechenden Definitionen für die Parameterisierung und Parameterübergabe schließen sich in Abschnitt 3.4 an.

3.1 Systemspezifikationsmorphismen

Eine Voraussetzung für die Komposition und Parameterisierung von AN-Spezifikationen, ist die Möglichkeit sie in Beziehung zu setzen, z.B. um Schnittstellen oder Parameter auszuzeichnen. Zu diesem Zweck führen wir in diesem Abschnitt Abbildungen zwischen AN-Spezifikationen, die „AN-Spezifikationsmorphismen“, ein.

Entsprechend der beiden grundlegenden Komponenten, algebraische Datentypspezifikation und Ablaufbeschreibung in Form eines CP/T-Netzes, aus denen sich die AN-Spezifikationen zusammensetzen, besteht jeder AN-Spezifikationsmorphismus aus einem Datentypspezifikations- und einem CP/T-Netzmorphismus. Sie müssen als Paar so gewählt werden, daß sowohl der über die Algebrakomponente der Netzabbildung gegebene Zusammenhang (\rightarrow 2.1.1) zwischen beiden Komponenten, als auch die in Form von Gleichungsmengen zu Transitionen spezifizierten Einschränkungen im Systemverhalten gewahrt bleiben.

Notation: Sei $h^S: Spec_1 \rightarrow Spec_2$ ein Datentypspezifikationsmorphismus, dann notieren wir im folgenden dessen Transformation $\mathcal{T}^S(h^S): Spec_1(X) \rightarrow Spec_2(X)$ (\rightarrow 6.3.4) auf die mit der Variablenfamilie X als zusätzliche Konstanten erweiterten Spezifikationen mit h^{S_x} .

3.1.1 Definition (AN-Spezifikationsmorphisimen):

Seien $ANS_i = (Spec_i, Cond_i, N_i)$, $i = 1, 2$, AN-Spezifikationen, dann bildet ein Paar (h^S, h^N) einen „Algebraischen Netzspezifikationsmorphisimus“ (kurz „AN-Spezifikationsmorphisimus“) mit Domäne ANS_1 und Codomäne ANS_2 , notiert

$$(h^S, h^N): ANS_1 \rightarrow ANS_2,$$

wenn die folgenden Bedingungen erfüllt sind:

(I) $h^S: Spec_1 \rightarrow Spec_2$ ist ein Spezifikationsmorphisimus,

(II) $h^N: N_1 \rightarrow N_2$ ist ein CP/T-Netzmorphisimus mit

$$\mathcal{P}^A(h^N) = (h^{S_x}, h^t): T_{OP_1(X)} \rightarrow T_{OP_2(X)},$$

wobei $T_{OP_i(X)}$ die Termalgebra zu $Spec_i(X)$ bezeichnet;

(III) für alle Transitionen t_1 in N_1 und t_2 in N_2

$$h^T(t_1)(t_2) > 0 \implies h^{S^*}(Cond_1(t_1)) = Cond_2(t_2),$$

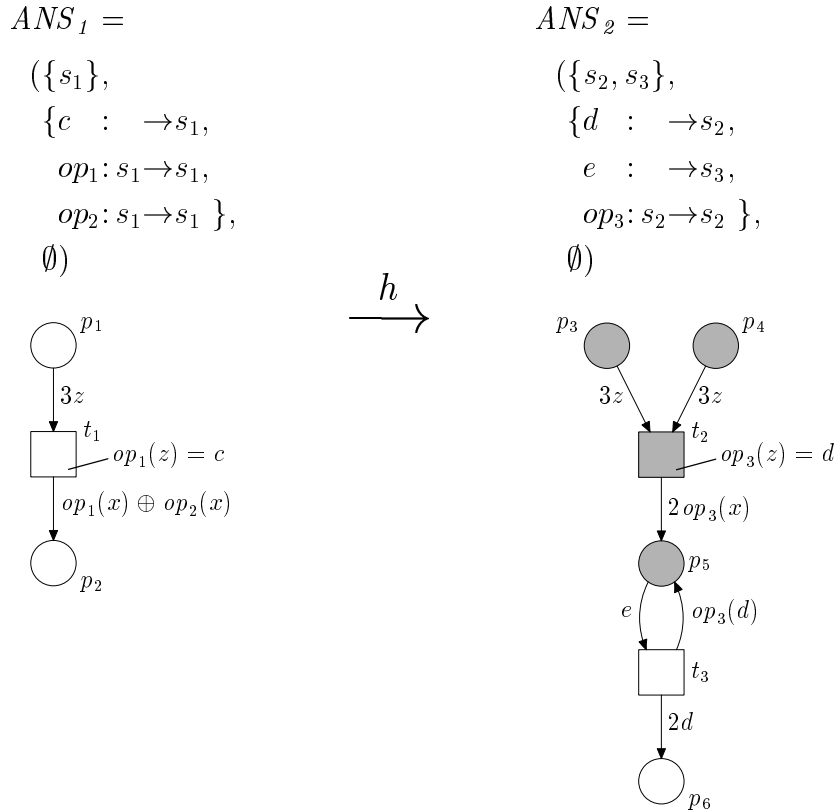
wobei h^T die Transitionskomponente von h^N bezeichnet und h^{S^*} die kanonische Fortsetzung der h^S -generierten Abbildung zwischen Gleichungen (\rightarrow 6.2.3) auf Gleichungsmengen.

□

Jede der beiden Komponenten eines AN-Spezifikationsmorphisimus bewahrt die wesentlichen Eigenschaften der Systemkomponente, die sie abbildet (\rightarrow 6.2, 1.1). Diese Eigenschaften legen ihrerseits, zusammen mit den ebenfalls bewahrten Beschränkungen durch die Gleichungsmengen, Systemeigenschaften wie den funktionalen Zusammenhang der Objekte, die möglichen Ereignisse und deren Nebenläufigkeit bzw. deren Nichtdeterminismus fest, sodaß diese Eigenschaften von den Spezifikationsmorphisimen bewahrt werden. Dagegen können kausale Abhängigkeiten von Ereignissen in der Domäne eines Spezifikationsmorphisimus, in dessen Codomäne, verloren gehen. Diese Möglichkeit besteht, wenn Systemstrukturen mit Hilfe von AN-Spezifikationsmorphisimen in größere Zusammenhänge eingebettet werden. Dabei können die Voraussetzungen für das Eintreten von Ereignissen, die in der Domäne des Spezifikationsmorphisimus von einem Ereignis abhängen, in dessen Codomäne durch unterschiedliche Ereignisse geschaffen werden.

3.1.2 Beispiel (Ein AN-Spezifikationsmorphisimus):

Abbildung 3.1 zeigt zwei AN-Spezifikationen verbunden durch den AN-Spezifikationsmorphisimus h , dessen Bild durch die markierten Stellen und Transitionen in ANS_2 hervorgehoben ist. Die Gleichungsmengen zu den Transitionen sind in die graphische Repräsentation der Netzkomponenten integriert.

Abbildung 3.1: Ein AN-Spezifikationsmorphismus h

h ist durch seine Datentypkomponente $h^S = (h^{SO}, h^{OP})$, die Transitionskomponente h^T und die Stellenkomponente h^P seiner Netzabbildung eindeutig beschrieben. Der generalisierte Algebromorphismus (h^{S_x}, h^t) von $T_{OP_1(X)}$ nach $T_{OP_2(X)}$ als Algebrakomponente der Netzabbildung, ergibt sich durch: $h^{S_x} = \mathcal{T}^S(h^S) = (h^{SO}, h^{OP_x})$ (\rightarrow 6.3.4) mit

$$h^{OP_x}(op) = \begin{cases} op & ; op \in X \\ h^{OP}(op) & ; \text{sonst} \end{cases}$$

für alle Operationen op aus $Spec_1(X)$ und

$$h^t(op(t_1, \dots, t_n)) = h^{OP_x}(op)(h^t(t_1), \dots, h^t(t_n))$$

(\rightarrow 3.1.1) für alle Terme $op(t_1, \dots, t_n)$ in $T_{OP_1(X)}$.

$$\begin{array}{llll} h^{SO} : s_1 \mapsto s_2, & h^{OP} : c \mapsto d, & h^T : t_1 \mapsto t_2, & h^P : p_1 \mapsto p_3 \oplus p_4 \\ & op_1 \mapsto op_3 & & p_2 \mapsto p_5 \\ & op_2 \mapsto op_3 & & \end{array}$$

□

3.1.3 Korollar (Die Kategorie der AN-Spezifikationen):

Die AN-Spezifikationen bilden zusammen mit ihren Morphismen, den komponentweise erzeugten Identitäten und der komponentweisen Komposition von Morphismen eine Kategorie (bezeichnet mit $\mathcal{AHLN}\text{-SPEC}$).

□

Kapitel 2 beschreibt die Semantik von AN-Spezifikationen in Form von Modellkategorien. Die im folgenden eingeführte Semantik der AN-Spezifikationsmorphisimen setzt diese Kategorien in Beziehung, indem sie jedem Modell und jedem Morphismus zur Zielspezifikation eines AN-Spezifikationsmorphisimus ein Modell bzw. einen Morphismus seiner Ursprungsspezifikation zuordnet. Der wesentliche Schritt dieser Zuordnung besteht in einer Reduktion auf den für die Ursprungsspezifikation relevanten Anteil. In diesem Sinne verhält sie sich analog zu den in [EM85] betrachteten Vergebisfunktionen (\rightarrow 6.1.11), die von Spezifikationsmorphisimen generiert, zwischen den Modellkategorien der Datentypspezifikationen abbilden.

In Bezug auf einen AN-Spezifikationsmorphisimus $h: ANS_1 \rightarrow ANS_2$, läßt sich jedes ANS_2 -Modell in drei Teile zerlegen. Zum einen in den vom ANS_1 -Bild unter h generierten Teil. Dieser erfüllt alle in ANS_1 spezifizierten Systemeigenschaften, da sie sowohl von h , als auch von der Modellbildung (bzgl. ANS_2) bewahrt werden. Ein weiterer Teil ist von der Umgebung des ANS_1 -Bildes generiert, er ist ANS_2 -spezifisch und steht in keiner Beziehung zu ANS_1 . Den dritten Teil bilden die nicht generierten Modellelemente. Sie sind spezifikationsunabhängig und beschreiben modellspezifische Aspekte. Diese Unterteilung, die sowohl für den Datentypanteil als auch für die Netzstrukturen gilt, bildet die Grundlage der im folgenden definierten Reduktion von ANS_2 - auf ANS_1 -Modelle. Dabei beschreibt der vom ANS_1 -Bild generierte - und der nicht generierte Teil zusammengenommen, alle Aspekte eines ANS_1 -Modells, der verbleibende Teil wird beim Übergang vom ANS_2 - zum entsprechenden ANS_1 -Modell entfernt.

3.1.4 Definition (Modellreduktion):

Gegeben ist ein AN-Spezifikationsmorphisimus

$$h = (h^S, h^N): ANS_1 \rightarrow ANS_2 \quad \text{mit} \quad h^N = (h^T, h^P, h^G)$$

und zwei Projektionen \mathcal{P}^T und \mathcal{P}^P die CP/T-Netze auf ihre Transitions- bzw. auf ihre Stellenmenge abbilden. Wir definieren zu jedem ANS_2 -Modell

$$AN = (ANS_2, g, A, N^o, eval)$$

dessen h -Redukt durch:

$$AN_{\mathcal{V}_h} \stackrel{\text{def}}{=} (ANS_1, g \circ h^S, A, N_{\mathcal{V}}^o, eval_{\mathcal{V}}).$$

Das darin enthaltene CP/T-Netz

$$N_{\mathcal{V}}^o \stackrel{\text{def}}{=} T_{\mathcal{V}}^{\otimes} \begin{array}{c} \xrightarrow{Pre_{\mathcal{V}}} \\ \xrightarrow{Post_{\mathcal{V}}} \end{array} P_{\mathcal{V}}^{\otimes} \otimes A(X),$$

ergibt sich mit Hilfe der Zerlegungen von N^o über die CP/T-Netzmorphismen $eval$, $eval \circ h^N$ (\rightarrow 1.2.3)

$$(I) T_{\mathcal{V}} \stackrel{\text{def}}{=} \mathcal{P}^T(\mathcal{DC}_{eval \circ h^N}(im)) \cup \mathcal{P}^T(\mathcal{DC}_{eval}(co)),$$

$$(II) P_{\mathcal{V}} \stackrel{\text{def}}{=} \mathcal{P}^P(\mathcal{DC}_{eval \circ h^N}(im)) \cup \mathcal{P}^P(\mathcal{DC}_{eval}(co)),$$

$$(III) Pre_{\mathcal{V}} \stackrel{\text{def}}{=} Pre \Big|_{T_{\mathcal{V}}^{\otimes}} \text{ und}$$

$$(IV) Post_{\mathcal{V}} \stackrel{\text{def}}{=} Post \Big|_{T_{\mathcal{V}}^{\otimes}}$$

wobei $Pre/Post$ die Markenflußmorphismen von N^o bezeichnen. Der Evaluierungsmorphismus $eval_{\mathcal{V}}$ ist gegeben durch:

$$eval_{\mathcal{V}} \stackrel{\text{def}}{=} \left(e^T \Big|_{T_{\mathcal{V}}^{\otimes}} \circ h^T, e^P \Big|_{P_{\mathcal{V}}^{\otimes}} \circ h^P, (\mathcal{T}^S(g \circ h^S), eval_{\mathcal{V}(A(X))}) \right)^2$$

wobei e^T , e^P die Transitions- bzw. Stellenkomponente von $eval$ bezeichnen. □

Der generierte Teil des bezüglich $h: ANS_1 \rightarrow ANS_2$ reduzierten ANS_2 -Modells AN gibt dessen ANS_1 -Eigenschaften wieder. Dem entsprechend enthält der Algebraanteil seiner Markenflußmorphismen nur Elemente, die in direktem Bezug zur ANS_1 -Datentypkomponente stehen. Dieser Zusammenhang läßt sich durch den von der h -Datentypkomponente h^{S_x} generiertem Vergißfunktors $\mathcal{V}_{h^{S_x}}$ (\rightarrow 6.1.11) zum Ausdruck bringen, d.h. die besagten Algebraelemente liegen im Bild der AN -Algebra unter $\mathcal{V}_{h^{S_x}}$ und stehen über die Evaluierungsfunktion

$$eval_{\mathcal{V}(A(X))}: T_{OP_1(X)} \rightarrow \mathcal{V}_{\mathcal{T}^S(g \circ h^S)}(A(X))$$

mit der Termalgebra zur erweiterten ANS_1 -Datentypspezifikation in Verbindung.

Dem gegenüber kann der modellspezifische Teil des h -Redukts von AN beliebige Elemente der AN -AlgebraKomponente enthalten. Aus diesem Grund muß die Algebra komplett in das reduzierte Modell übernommen werden. Da darin auch die Elemente ihres Vergißbildes unter \mathcal{V}_{h^S} enthalten sind, beschreibt sie die AlgebraKomponente des gesamten h -Redukts.

Die Reduktionen der Modellmorphismen zur Zielspezifikation eines AN-Spezifikationsmorphisms, ergeben sich direkt aus den entsprechenden Reduktionen ihrer

²Der Funktor \mathcal{T}^S (\rightarrow 6.3.4) bildet Spezifikationsmorphisms auf Spezifikationsmorphisms zwischen den um die Variablenfamilie X als zusätzliche Konstanten erweiterten Spezifikationen ab.

Domänen bzw. Codomänen. Dem entsprechend wird der enthaltene Algebromorphismus jeweils unverändert übernommen und die Stellen- bzw. die Transitionskomponente auf die Untergruppen eingeschränkt, die sich aus der Reduktion ihrer Domäne bzw. ihrer Codomäne ergeben.

3.1.5 Definition (Morphismenreduktion):

Sei $h: ANS_1 \rightarrow ANS_2$ ein AN-Spezifikationsmorphismus, dann definieren wir für jeden ANS_2 -Modellmorphismus

$$f = (f^G, (f^T, f^P, f^{Gx})): AN \rightarrow AN'$$

dessen h -Redukt $f_{\mathcal{V}_h}$ durch

$$f_{\mathcal{V}_h}: AN_{\mathcal{V}_h} \rightarrow AN'_{\mathcal{V}_h} \stackrel{\text{def}}{=} \left(f^G, \left(f^T \Big|_{T_{\mathcal{V}}^{\otimes}}^{T'_{\mathcal{V}}^{\otimes}}, f^P \Big|_{P_{\mathcal{V}}^{\otimes}}^{P'_{\mathcal{V}}^{\otimes}}, f^{Gx} \right) \right)$$

wobei $T_{\mathcal{V}}^{\otimes}/T'_{\mathcal{V}}^{\otimes}$ und $P_{\mathcal{V}}^{\otimes}/P'_{\mathcal{V}}^{\otimes}$ die Transitionen bzw. Stellen in der Netzstruktur von $\mathcal{V}_h(AN)/\mathcal{V}_h(AN')$ bezeichnen. □

Im Zuge der Modellreduktion bezüglich nicht surjektiver AN-Spezifikationsmorphismsen werden generierte Stellen und Transitionen der Modelle weggelassen. Aus diesem Grund erzwingt die Definition der Modellmorphismsen (\rightarrow 2.2.3), daß die Bilder von nichtgenerierten Bestandteilen eines A-Netzes keine generierten Stellen bzw. Transitionen enthalten dürfen. Alle nichtgenerierten Netzteile werden in das reduzierte Modell übernommen, die generierten Stellen und Transitionen dagegen nur zum Teil, sodaß ein Fehlen dieser Einschränkung zu undefinierten Reduktionen der Modellmorphismsen führen würde.

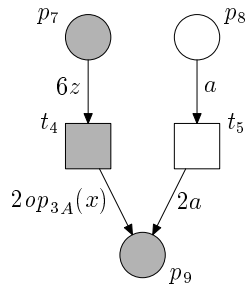
3.1.6 Beispiel (Vergißfunktork \mathcal{V}_h zwischen Modellkategorien):

Die rechte Seite von Abbildung 3.2 zeigt AN , ein Modell der Spezifikation ANS_2 aus Beispiel 3.1.2. Das von $h: ANS_1 \rightarrow ANS_2$ generierte AN -Redukt $AN_{\mathcal{V}_h}$ ist im linken Teil von Abbildung 3.2 wiedergegeben.

Die markierten Stellen und Transitionen in beiden A-Netzen zeichnen deren generierten Teil aus, d.h sie liegen im Bild der jeweiligen Evaluierungsmorphismsen, die durch ihre Komponenten e^T, e^P, e^A bzw. $e_{\mathcal{V}}^T, e_{\mathcal{V}}^P, e_{\mathcal{V}}^A$ gegeben sind. Die generierte Stelle p_{10} und die generierten Transitionen t_6, t_7 werden bei der Reduktion von AN auf $AN_{\mathcal{V}_h}$ „vergessen“. □

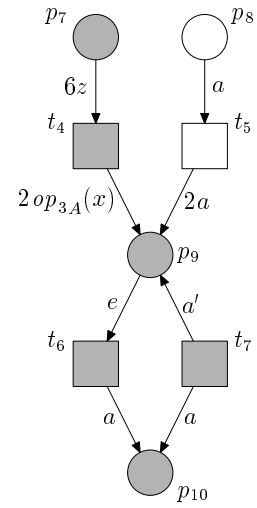
Die von einem AN-Spezifikationsmorphismus generierten Reduktionen der Modellmorphismsen (\rightarrow 3.1.5) erhalten die Identitäten und sind mit der Morphismsenkomposition verträglich. Zusammen mit der in 3.1.4 definierten Modellreduktion lassen sie sich zu einem Funktork vereinigen, der es ermöglicht Modelle und Modellmorphismsen

$$\begin{aligned}
AN_{\mathcal{V}_h}: & ANS_1, h^S, \\
& (\{a, a', \dots\}_{s_2}, \\
& \{e\}_{s_3}, \\
& \{d_A = a, \\
& \quad op_{3A}: a \mapsto a' \\
& \quad \quad a' \mapsto a\})
\end{aligned}$$



$$\begin{aligned}
e_{\mathcal{V}}^T: & t_1 \mapsto t_4 \\
e_{\mathcal{V}}^P: & p_1 \mapsto 2p_7 \\
& p_2 \mapsto p_9 \\
e_{\mathcal{V}}^A: & x \mapsto [x] \\
& c \mapsto [a] \\
& \vdots \mapsto \vdots
\end{aligned}$$

$$\begin{aligned}
AN: & ANS_2, id, \\
& (\{a, a', \dots\}_{s_2}, \\
& \{e\}_{s_3}, \\
& \{d_A = a, \\
& \quad op_{3A}: a \mapsto a' \\
& \quad \quad a' \mapsto a\})
\end{aligned}$$



$$\begin{aligned}
e^T: & t_2 \mapsto t_4 \\
& t_3 \mapsto t_6 \oplus t_7 \\
e^P: & p_3 \mapsto p_7 \\
& p_4 \mapsto p_7 \\
& p_5 \mapsto p_9 \\
& p_6 \mapsto p_{10} \\
e^A: & x \mapsto [x] \\
& d \mapsto [a] \\
& e \mapsto [e] \\
& \vdots \mapsto \vdots
\end{aligned}$$

Abbildung 3.2: h -generierte Reduktion (\mapsto Abb. 3.1) des ANS_2 -Modell AN .

zu unterschiedlichen AN-Spezifikationen in Beziehung zu setzen. Diese „Vergifunktoren“ bilden die Grundlage fr den berwiegenden Teil der folgenden Konstruktionen.

3.1.7 Lemma (Vergifunktork):

Sei $h: ANS_1 \rightarrow ANS_2$ ein AN-Spezifikationsmorphismus, dann ist die Abbildung $\mathcal{V}_h: MOD(ANS_2) \rightarrow MOD(ANS_1)$, gegeben durch:

$$\mathcal{V}_h(f) \stackrel{\text{def}}{=} f_{\mathcal{V}_h}$$

fr jeden ANS_2 -Modellmorphismus f ein Funktor.

□

(Beweis \rightarrow Anhang S. 202)

Die Zuordnung von Vergifunktoren zu AN-Spezifikationsmorphisimen ergnzt als deren Semantik die Modellsemantik der AN-Spezifikationen zu einer Semantik der gesamten Kategorie $\mathcal{AHLN-SPEC}$. Die Bewahrung von Identitten und Kompositionen im Rahmen dieser Zuordnung, bildet eine notwendige Voraussetzung fr die Definition der generalisierten Morphisimen im nchsten Abschnitt. Sie erlaubt es die $\mathcal{AHLN-SPEC}$ -Semantik als kontravarianten, d.h. die Richtung der Morphisimen umkehrenden, Funktor in die Kategorie der (kleinen) Kategorien³ aufzufassen. Kombiniert man diesen Funktor mit seiner Domne, der Kategorie $\mathcal{AHLN-SPEC}$, ergibt sich daraus ein „Spezifikationsrahmen“ (\rightarrow 5.3.1), der Syntax und Semantik der Systemspezifikationen zusammenfat.

3.1.8 Lemma (Spezifikationsrahmen):

Das Paar $(\mathcal{AHLN-SPEC}, MOD)$ mit $MOD: \mathcal{AHLN-SPEC}^{op} \rightarrow CATCAT$ gegeben durch:

$$MOD(h) \stackrel{\text{def}}{=} \mathcal{V}_h$$

fr alle AN-Spezifikationsmorphisimen h , bildet einen „Spezifikationsrahmen“.

□

(Beweis \rightarrow Anhang S. 208)

³Die Morphisimen zwischen jeweils zwei Objekten bilden Mengen, im Gegensatz zu Klassen bei Kategorien, die nicht klein sind.

3.2 Generalisierte Modellmorphismen

Die im letzten Abschnitt eingeführten AN-Spezifikationsmorphismen, bewahren wichtige Systemeigenschaften, wie die nebenläufige Ausführbarkeit von Ereignissen, Nichtdeterminismus bei deren Auswahl und die Funktionalität der Operationen, mit denen die Objekte im System manipuliert werden. Diese Eigenschaften bleiben, erzwungen durch die entsprechenden Definitionen (\rightarrow 2.2.1, 2.2.3), auch bei der Modellbildung erhalten. Es liegt daher nahe, Modelle und -morphismen von Systemspezifikationen, die durch einen AN-Spezifikationsmorphismus verbunden sind, in Beziehung zu setzen.

Wir schaffen die Zusammenhänge von Modellen unterschiedlicher Spezifikationen, durch die Definition „generalisierter Morphismen“, die im Unterschied zu den „normalen“ Modellmorphismen die Grenzen der Modellkategorien überschreiten können. Mit Hilfe dieser Morphismen lassen sich die Modelle aller Systemspezifikationen in einer Struktur, der Kategorie $\mathcal{GAHL-NETS}$, zusammenfassen, die darüber hinaus alle in 2.2.3 definierten Modellmorphismen als Spezialfälle enthält. Die darin ausdrückbaren Modellbeziehungen und die Möglichkeit in Beziehung stehende Modelle über Diagramme miteinander zu verknüpfen, bildet die Grundlage der Modellsemantik für die Komposition (\rightarrow 3.3) und Parameterisierung (\rightarrow 3.4) von Systemspezifikationen.

Generalisierte Modellmorphismen sind Paare, bestehend aus einem Spezifikations- und einem Modellmorphismus. Ersterer setzt die Spezifikationen der Modelle in Beziehung, während der Modellmorphismus das Modell im Ursprung des generalisierten Morphismus mit dem vom Spezifikationsmorphismus generierten Vergißbild des Zielmodells verknüpft, d.h. er gehört zur Modellkategorie der Domäne des Spezifikationsmorphismus. Diese Konstruktion erlaubt zum einen die Rückführung generalisierter Morphismen auf die schon bekannten Spezifikations- und Modellmorphismen, zum anderen enthält das Vergißbild des Zielmodells, alle für das Ursprungsmodell relevanten Teile, sodaß es für die Festlegung des Zusammenhangs zwischen Ursprungs- und Zielmodell des generalisierten Morphismus ausreicht.

Das Diagramm in Abbildung 3.3 veranschaulicht die Zusammenhänge von Spezifikations- und Modellkomponente generalisierter Morphismen $((h_1, f_1), (h_2, f_2))$ sowie deren Komposition.

Dabei werden die enthaltenen Spezifikationsmorphismen h_1, h_2 auf die übliche Weise komponiert. Der Modellmorphismus f_2 muß dagegen zunächst durch den Vergißfunktorktor \mathcal{V}_{h_1} in einen ANS_I -Modellmorphismus abgebildet werden, bevor eine Komposition mit f_1 in der Kategorie der ANS_I -Modelle möglich ist.

Eine derartige Komposition setzt voraus, daß bei der Konstruktion der Vergißfunktoren die Komposition der Spezifikationsmorphismen bewahrt wird. Bezogen auf unser Beispiel bedeutet das, die Gleichung

$$\mathcal{V}_{h_2 \circ h_1}(AN_g) = \mathcal{V}_{h_1}(\mathcal{V}_{h_2}(AN_g))$$

gilt. Die Erfüllung dieser Forderung folgt ebenso wie die Existenz der Identitäten in $\mathcal{GAHL-NETS}$, aus dem im Spezifikationsrahmen $(\mathcal{AHLN-SPEC}, MOD)$ (\rightarrow 3.1.8)

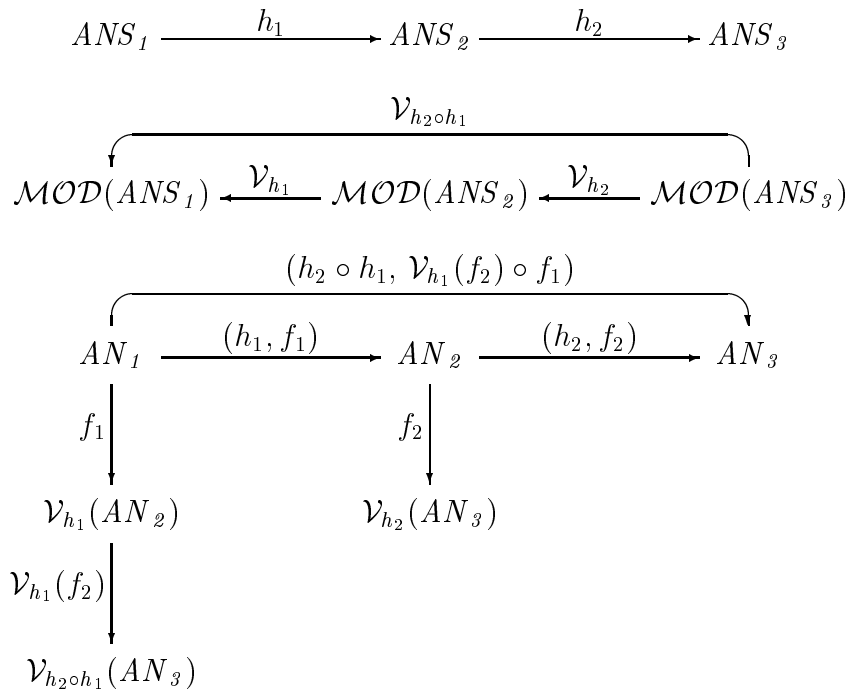
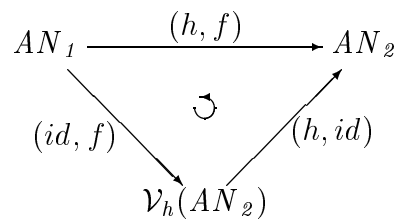


Abbildung 3.3: Komposition generalisierter Morphisamen.

festgelegten funktoriellen Zusammenhang von Systemspezifikationen und Modellsemantik. Einige weiterführende Betrachtungen und Ergebnisse zu diesem Thema, sowie die formalen Definitionen von Spezifikationsrahmen und der sich daraus ergebenden generalisierten Modellkategorien enthält Abschnitt 5.3.

Neben generalisierten Morphisamen mit initialen Modellen als Ursprung, spielen im weiteren zwei Arten von generalisierten Morphisamen eine gesonderte Rolle. Das sind zum einen die Entsprechungen der Morphisamen aus den Modellkategorien, die wir als „*einfach*“ bezeichnen, erkennbar an den Identitäten als erster Komponente und zum anderen Morphisamen der Form (h, id_{AN}) vom Vergißbild $\mathcal{V}_h(AN)$ eines Modells AN nach AN . Letztere bezeichnen wir als „*modellidentisch*“. Mit den Funktoreigenschaften der Vergißfunktorkonstruktion folgt aus der Kompositionsregel für generalisierte Morphisamen, daß sich, wie im folgenden Diagramm veranschaulicht, jeder generalisierte Modellmorphisamen in einen einfachen und einen modellidentischen Morphisamen zerlegen läßt.



Die mittels generalisierter Modellmorphisamen beschriebenen Modellbeziehungen, lassen sich in Morphisamen zwischen deren Netzstrukturen umsetzen. Das Ergebnis dieser

als „*Projektion*“ bezeichneten Abbildung sind CP/T-Netzmorphismen, die die Netzkomponenten von Domäne und Codomäne des generalisierten Netzmorphismus direkt verbinden. Dabei komponiert man die Projektion eines generalisierten Modellmorphisms aus den Projektionen seines einfachen und seines modellidentischen Anteils. Ersterer ist mit dem enthaltenen Modellmorphismus identisch, während die Projektion des modellidentischen Anteils einen Inklusionsmorphismus ergibt.

3.2.1 Definition (Projektionen generalisierter Morphismen):

Wir definieren die Abbildung

$$\mathcal{P}^N: \mathcal{GAHL}\text{-}\mathcal{NETS} \rightarrow \mathcal{CPTN}$$

für alle $\mathcal{GAHL}\text{-}\mathcal{NETS}$ -Morphismen $(h, (f^G, f^N)): AN_1 \rightarrow AN_2$ durch:

$$\mathcal{P}^N\left((h, (f^G, f^N))\right): N_1^o \rightarrow N_2^o = (\hookrightarrow_{T_{2,1}^{\otimes}, T_2^{\otimes}}, \hookrightarrow_{P_{2,1}^{\otimes}, P_2^{\otimes}}, id) \circ f^N$$

wobei N_i^o die Netzkomponente von AN_i bezeichnet und $T_{2,1}/P_{2,1}$ die Transitions-/Stellenmenge der Netzkomponente in $\mathcal{V}_h(AN_2)$. □

3.2.2 Lemma (Eigenschaften der Projektion):

- (I) Die Projektion \mathcal{P}^N ist ein Funktor von $\mathcal{GAHL}\text{-}\mathcal{NETS}$ nach \mathcal{CPTN} .
- (II) Sei $((h^S, h^N), f): AN_1 \rightarrow AN_2$ ein generalisierter Modellmorphismus, dann kommutiert das nachfolgende Diagramm in \mathcal{CPTN} :

$$\begin{array}{ccc} N_1 & \xrightarrow{h^N} & N_2 \\ \text{eval}_1 \downarrow & \circlearrowleft & \downarrow \text{eval}_2 \\ \mathcal{P}^N(AN_1) & \xrightarrow{\mathcal{P}^N((h, f))} & \mathcal{P}^N(AN_2) \end{array}$$

wobei $eval_i$ die Evaluierungsfunktion aus AN_i bezeichnet. □

(Beweis \rightarrow Anhang S. 211)

Wie Colimites zu Diagrammen in \mathcal{CPTN} (\rightarrow 1.2), lassen sich auch die Colimites in $\mathcal{GAHL}\text{-}\mathcal{NETS}$ komponentenweise bilden, d.h. wenn die Colimites der A-Netzkomponenten: Systemspezifikation, Algebra und CP/T-Netz, gegeben sind, läßt sich daraus der Colimes des Diagramms in $\mathcal{GAHL}\text{-}\mathcal{NETS}$ konstruieren. Für die übrigen Komponenten (Datentypspezifikations- und Evaluierungsmorphismus) braucht die Existenz entsprechender Colimites nicht vorausgesetzt zu werden, sie ergeben sich aus den Colimites ihrer Domänen und Codomänen.

Daß wir bisher noch keine Konstruktion für Colimites zu Diagrammen in der Systemspezifikationskategorie angegeben haben⁴, bildet für die Colimeskonstruktion in $\mathcal{GAHL}\text{-}\mathcal{NETS}$ kein Hindernis. Neben der Existenz eines entsprechenden Colimes setzen wir nur noch dessen komponentweise Konstruktion voraus, alles übrige ergibt sich aus den couniversellen Eigenschaften von Colimites (\rightarrow 5.1.15).

3.2.3 Lemma (Komponentweise Colimites):

Sei \mathcal{S} eine endliche Kategorie, $\mathcal{D}: \mathcal{S} \rightarrow \mathcal{GAHL}\text{-}\mathcal{NETS}$ ein Funktor,

$\mathcal{D}^S: \mathcal{S} \rightarrow \mathcal{SPEC}$ ein Funktor mit Colimes $((c_s^S)_{s \in \mathcal{S}}, \text{Spec}_c)$,

$\mathcal{D}^N: \mathcal{S} \rightarrow \mathcal{CPTN}$ ein Funktor mit Colimes $((c_s^N)_{s \in \mathcal{S}}, N_c)$,

$\mathcal{D}^{AS}: \mathcal{S} \rightarrow \mathcal{AHLN}\text{-}\mathcal{SPEC}$ ein Funktor mit Colimes $((c_s^S, c_s^N)_{s \in \mathcal{S}}, \text{ANS}_c)$,

$\mathcal{D}^A: \mathcal{S} \rightarrow \mathcal{GSPEC}\text{-}\mathcal{ALG}$ ein Funktor mit Colimes $((d_s^S, d_s^A)_{s \in \mathcal{S}}, A_c)$,

$\mathcal{P}^N \circ \mathcal{D}: \mathcal{S} \rightarrow \mathcal{CPTN}$ ein Funktor mit Colimes $((\dot{c}_s^N)_{s \in \mathcal{S}}, N_c^o)$

und es gelte:

$$\mathcal{D}^{AS}(m) = (\mathcal{D}^S(m), \mathcal{D}^N(m)),$$

$$\mathcal{D}(m) = (\mathcal{D}^{AS}(m), (\mathcal{D}^A(m), f_m^N)),$$

$$\mathcal{P}^A(\dot{c}_s^N) = \mathcal{T}^A((d_s^S, d_s^A)) \quad ^5$$

für alle Objekte s und Morphismen m in \mathcal{S} . Dann ergibt sich der Colimes von \mathcal{D} zu

$$\left(\left((c_s^S, c_s^N), ((d_s^S, d_s^A), d_s^N) \right)_{s \in \mathcal{S}}, \text{AN}_c \right)$$

mit

(I) $\text{AN}_c \stackrel{\text{def}}{=} (\text{ANS}_c, g_c, A_c, N_c^o, \text{eval}_c)$,

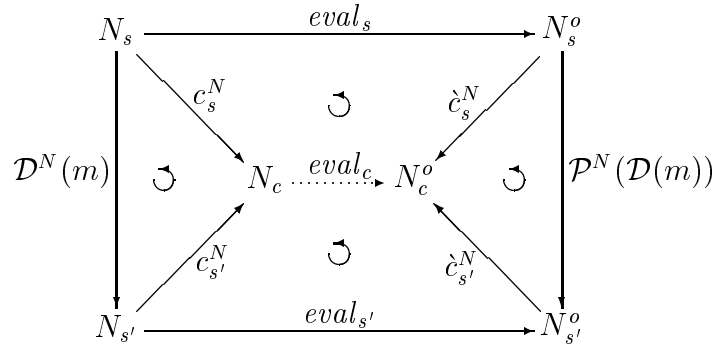
(II) g_c als eindeutige kommutative Ergänzung der nachfolgenden Diagramme für alle $m: s \rightarrow s'$ in \mathcal{S}

$$\begin{array}{ccccc}
 \text{Spec}_s & \xrightarrow{g_s} & & & \text{Spec}'_s \\
 \downarrow & \searrow c_s^S & \circlearrowleft & \swarrow d_s^S & \downarrow \\
 \mathcal{D}^S(m) & \circlearrowleft & \text{Spec}_c & \xrightarrow{g_c} & \text{Spec}'_c & \circlearrowleft & f_m^S \\
 \downarrow & \swarrow c_{s'}^S & \circlearrowleft & \swarrow d_{s'}^S & \downarrow \\
 \text{Spec}_{s'} & \xrightarrow{g_{s'}} & & & \text{Spec}'_{s'}
 \end{array}$$

worin f_m^S die Spezifikationskomponente von $\mathcal{D}^A(m)$ und $g_s/g_{s'}$ den Spezifikationsmorphismus in $\mathcal{D}(s)/\mathcal{D}(s')$ bezeichnet,

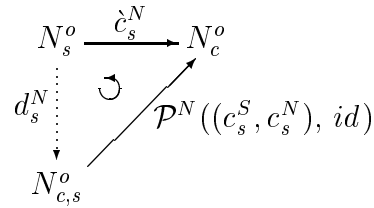
⁴wird in Abschnitt 3.3 nachgeholt

- (III) $eval_c$ als eindeutige kommutative Ergänzung der nachfolgenden Diagramme für alle $m: s \rightarrow s'$ in \mathcal{S}



worin $eval_s/eval_{s'}$ den Evaluierungsmorphismus in $\mathcal{D}(s)/\mathcal{D}(s')$ bezeichnet

- (IV) und für alle Objekte s in \mathcal{S} , ergibt sich sich d_s^N eindeutig als kommutative Ergänzung des folgenden Diagramms



worin $N_{c,s}^o$ die Netzstruktur in $\mathcal{V}_{(c_s^S, c_s^N)}(AN_c)$ bezeichnet.

□

(Beweis \rightarrow Anhang S. 213)

Im vorangegangenen Abschnitt haben wir Modellmorphisamen in Funktoren zwischen Prozeßkategorien transformiert (\rightarrow 2.3.5) und damit den strukturellen Zusammenhang der Modelle in einen operationalen Zusammenhang der Modellabläufe übersetzt. Generalisierte Modellmorphisamen verallgemeinern die strukturellen Zusammenhänge Modellkategorie übergreifend. Diese Verallgemeinerung wollen im weiteren auf die Ebene der Prozeßkategorien und Funktoren fortsetzen.

Die Umsetzung von Modellen und Modellmorphisamen in Prozeßkategorien und Funktoren erfolgte in zwei Schritten: zunächst wurde durch Entfaltung ein CP/T-Netz bzw. CP/T-Netzmorphismus erzeugt, dem dann mit Hilfe des Funktors \mathcal{PC} (\rightarrow 1.3.11) eine Prozeßkategorie bzw. ein Funktor zwischen Prozeßkategorien zugeordnet wurde. Aufgrund dieser Aufteilung muß nur die Entfaltung verallgemeinert werden, mit den daraus resultierenden CP/T-Netzen und Morphismen kann wie zuvor verfahren werden.

⁵ \mathcal{P}^A (\rightarrow 1.2.1) bildet CP/T-Netzmorphismen auf ihre Algebrakomponente ab und \mathcal{T}^A (\rightarrow 6.3.9) erweitert generalisierte Algebramorphismen auf generalisierte Algebramorphismen zwischen Algebren mit Variablen.

Da sich jeder generalisierte Morphismus als Komposition eines einfachen und eines modellidentischen Morphismus darstellen läßt (\rightarrow S. 55), bei der nur letzterer die Grenzen der Modellkategorie überschreitet, können wir uns im wesentlichen darauf beschränken, die Entfaltung für Morphismen dieser Form anzugeben. Alle übrigen Entfaltungen ergeben sich entweder direkt aus der ursprünglichen Definition oder können durch Komposition erzeugt werden.

Modellidentische Morphismen verbinden das Vergebild eines Modells mit dem Modell. Dabei sind Stellen und Transitionen des Vergebildes in den entsprechenden Mengen des Modells enthalten, die Algebren sind in beiden Fällen identisch. Aus diesen Zusammenhängen läßt sich der entsprechende Morphismus zwischen den Entfaltungen konstruieren.

3.2.4 Definition (Erweiterung der A-Netz-Entfaltungen):

Wir fassen die separat für jede AN-Spezifikation ANS definierten Funktoren $\mathcal{UF}_{ANS}: \mathcal{MOD}(ANS) \rightarrow \mathcal{CPTN}$ ⁶ zu einer Abbildung

$$\mathcal{UF}: \mathcal{GAHL-NETS} \rightarrow \mathcal{CPTN}$$

zusammen und erweitern sie durch

$$\mathcal{UF}((h, id_{\mathcal{V}_h(AN)})) \stackrel{\text{def}}{=} (\hookrightarrow \odot id, \hookrightarrow, id): \mathcal{UF}_{\text{dom}(h)}(\mathcal{V}_h(AN)) \rightarrow \mathcal{UF}_{\text{cod}(h)}(AN)$$

für alle AN-Spezifikationsmorphisimen h und $\text{cod}(h)$ -Modelle AN durch

$$\mathcal{UF}((h, f)) \stackrel{\text{def}}{=} \mathcal{UF}((h, id_{\mathcal{V}_h(AN_2)})) \circ \mathcal{UF}_{\text{dom}(h)}(f)$$

für alle Morphismen $(h, f): AN_1 \rightarrow AN_2$ in $\mathcal{GAHL-NETS}$, auf generalisierte Modellmorphisimen.

□

3.2.5 Definition (Erweiterung der operationalen Semantik):

Wir fassen die separat für jede AN-Spezifikation ANS definierten Funktoren $\mathcal{OS}_{ANS}: \mathcal{MOD}(ANS) \rightarrow \mathcal{PTS}$ ⁷ zu einer Abbildung

$$\mathcal{OS}: \mathcal{GAHL-NETS} \rightarrow \mathcal{PTS}$$

zusammen und erweitern sie durch

$$\mathcal{OS} \stackrel{\text{def}}{=} \mathcal{PC} \circ \mathcal{UF}$$

auf generalisierte Modellmorphisimen.

□

⁶siehe 2.3.3, 2.3.5

⁷siehe 2.3.7

3.2.6 Lemma (Eigenschaften der erweiterten operationalen Semantik):

(I) Die Abbildung \mathcal{UF} ist ein Funktor.

(II) Die Abbildung \mathcal{OS} ist ein Funktor.

□

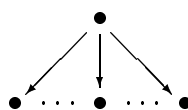
(Beweis \rightarrow Anhang S. 229)

3.3 Komposition von Systemspezifikationen

Kompositionsoperationen ermöglichen das Zusammensetzen umfangreicher Spezifikationen aus überschaubaren Teilen. Damit wird eine Standardvorgehensweise bei der Beschreibung komplexer Systeme, das Separieren von logisch zusammenhängenden Einheiten und deren Verknüpfung über Schnittstellen, im Rahmen unseres Spezifikationsansatzes unterstützt. Neben dem damit verbundenen Strukturierungsaspekt, ist die Erleichterung der Analyse von Systemeigenschaften aufgrund der geringeren Komplexität der Teilspezifikationen, ein weiterer Vorteil komponierter Spezifikationen. Grundvoraussetzung in beiden Fällen ist die Semantikverträglichkeit der Kompositionsoperationen, d.h. die Existenz korrespondierender Operationen auf Semantikebene.

Kompositionen sind partielle Operationen auf Systemspezifikationen. Wir bezeichnen deren Operanden im weiteren als „Komponenten“ und das Ergebnis als „komponierte Spezifikation“. Damit zwei oder mehr Spezifikationen komponiert werden können, müssen sie jeweils miteinander verträgliche, als „Schnittstellen“ bezeichnete Unterspezifikationen enthalten. Da wir uns mit der Verträglichkeit nur auf Eigenschaften nicht aber auf deren Repräsentation beziehen wollen, werden sie in Form einer „Musterschnittstelle“ als eigenständige Systemspezifikation beschrieben. Davon ausgehend, zeichnen Spezifikationsmorphismen die entsprechenden Strukturen in den Komponenten aus. Auf diese Weise wird den an einer Komposition beteiligten Spezifikationen eine spezifische Schnittstellensicht ermöglicht, was deren Integration in unterschiedliche Kontexte erleichtert.

Stelligkeit und Operanden der Kompositionsoperationen werden durch „Kompositionsdiagramme“ festgelegt. Wie alle im Rahmen dieser Arbeit verwendeten Diagramme, sind Kompositionsdiagramme Funktoren. Deren Domäne bestimmt den Kompositionstyp, d.h. Anzahl und Verknüpfung der Komponenten, während die zu komponierenden AN-Spezifikationen und die Auszeichnung ihrer Schnittstellen durch das Bild des Funktors erfolgt. Dabei schränken wir uns auf Kategorien der Form



als Domäne ein, die wir als „Kompositionsmuster“ bezeichnen.

Der Wert einer Komposition ergibt sich aus der Verschmelzung der Komponenten über deren Schnittstellen. Diese Spezifikationsanteile sind im Ergebnis nur noch als eine Struktur vorhanden, die übrigen Teile der Komponenten werden unverändert übernommen. Ein entsprechender Effekt wird durch die Bildung des Colimes zum Kompositionsdiagramm erreicht, der darüber hinaus mit Hilfe der Colimesmorphis- men jede Komponente mit der resultierenden AN-Spezifikation in Beziehung setzt und sicherstellt, daß diese Spezifikation die kleinste ist, die alle (durch die Morphismen bewahrten) Eigenschaften der Komponenten umfaßt.

Wie in den vorangegangenen Colimeskonstruktionen (\rightarrow 1.2.2, 3.2.3), lassen sich auch die Colimites zu Diagrammen in $\mathcal{AHLN}\text{-SPEC}$ komponentweise, d.h. aus den Co- limites der enthaltenen Datentypspezifikationen und Netzstrukturen, erzeugen. Die Objekte dieser Colimites bilden die entsprechenden Bestandteile der AN-Spezifikation im Colimes eines $\mathcal{AHLN}\text{-SPEC}$ -Diagramms. Die Zuordnung von Gleichungsmengen zu Transitionen als drittem Spezifikationsbestandteil, ergibt sich aus der Transfor- mation der Zuordnungen in den Diagrammobjekten mit Hilfe der Morphismen aus dem Datentypspezifikations- und dem Netzcolimes.

3.3.1 Lemma (Komponentweise Colimites in $\mathcal{AHLN}\text{-SPEC}$):

Sei \mathcal{S} eine endliche Kategorie, $\mathcal{D}: \mathcal{S} \rightarrow \mathcal{AHLN}\text{-SPEC}$ Funktor,

$$\begin{aligned} \mathcal{D}^S: \mathcal{S} &\rightarrow \text{SPEC ein Funktor mit Colimes } ((c_s^S)_{s \in \text{Obj}(\mathcal{S})}, \text{Spec}_c), \\ \mathcal{D}^N: \mathcal{S} &\rightarrow \text{CPTN ein Funktor mit Colimes } ((c_s^N)_{s \in \text{Obj}(\mathcal{S})}, N_c) \end{aligned}$$

und seien für alle Objekte s und Morphismen m in \mathcal{S} die folgenden Bedingungen erfüllt:

$$\begin{aligned} \mathcal{D}(m) &= (\mathcal{D}^S(m), \mathcal{D}^N(m)), \\ \mathcal{P}^A(c_s^N) &= (\mathcal{T}^S(c_s^S), c_s^t): T_{OP_s(X)} \rightarrow T_{OP_c(X)}, \end{aligned}$$

dann ist der Colimes von \mathcal{D} in $\mathcal{AHLN}\text{-SPEC}$ gegeben durch:

$$\left(((c_s^S, c_s^N))_{s \in \mathcal{S}}, (\text{Spec}_c, \text{Cond}_c, N_c) \right)$$

mit

$$\text{Cond}_c(t) \stackrel{\text{def}}{=} \bigcup_{s \in \text{Obj}(\mathcal{S})} \bigcup_{t' \in T_s: c_s^T(t')(t) > 0} c_s^{S*}(\text{Cond}_s(t')) \quad 8$$

für alle Transitionen t aus N_c , wobei T_s die Transitionsmenge in $\mathcal{D}(s)$ bezeichnet, c_s^T die Transitionskomponente von c_s^N und c_s^{S*} die c_s^S -generierte Transformation von Gleichungsmengen (\rightarrow 6.2.3).

□

(Beweis \rightarrow Anhang S. 237)

⁸Die Gleichungsmengen sind wohldefiniert, da jede Transition im Colimes eine Entsprechung in einer der Diagrammspezifikationen hat.

Um jeder Komposition eine bis auf Isomorphie eindeutige AN-Spezifikation als Wert zuordnen zu können, müssen die zugrunde liegenden Diagramme die zu verschmelzenden Elemente eindeutig festlegen. Diese Bedingung wird verletzt, wenn eine Stelle oder eine Transition der Schnittstellenspezifikation, von mehr als einem Diagrammorphismus auf eine Summe von Stellen bzw. von Transitionen abgebildet wird. In diesen Fällen existiert, wie das nachfolgende Beispiel (\rightarrow 3.3.2) zeigt, mehr als eine Möglichkeit die Elemente dem Diagramm entsprechend zu verschmelzen, mit jeweils unterschiedlichen Ergebnissen. Dieser Mangel an Eindeutigkeit aus Anwendersicht hat im Fehlen der entsprechenden Colimites für die Stellen- bzw. für die Transitionskomponente (\rightarrow 5.1.17, 5.2.7, 5.2.20) seine Entsprechung auf Seiten der mathematischen Konstruktion. Da sich mehrdeutige Diagramme in Hinblick auf jedes ihrer möglichen Ergebnisse dergestalt verändern lassen, daß sie diesen Wert eindeutig beschreiben (\rightarrow 3.3.2), können wir sie ohne an Ausdrucksmächtigkeit zu verlieren als Kompositionen ausschließen.

3.3.2 Beispiel (Mehrdeutige Komposition):

Das nachfolgende Diagramm (\rightarrow Abb. 3.4) über die vom Graphen „ $\bullet \leftarrow \bullet \rightarrow \bullet$ “ generierte Musterkategorie ist mehrdeutig, da die Stelle „ p “ der Schnittstellenspezifikation von beiden Diagrammorphismen auf Summen („ $p_1 \oplus p_2$ “ in der links stehenden bzw. „ $p_3 \oplus p_4$ “ in der rechts stehenden Diagrammkomponente) abgebildet wird.

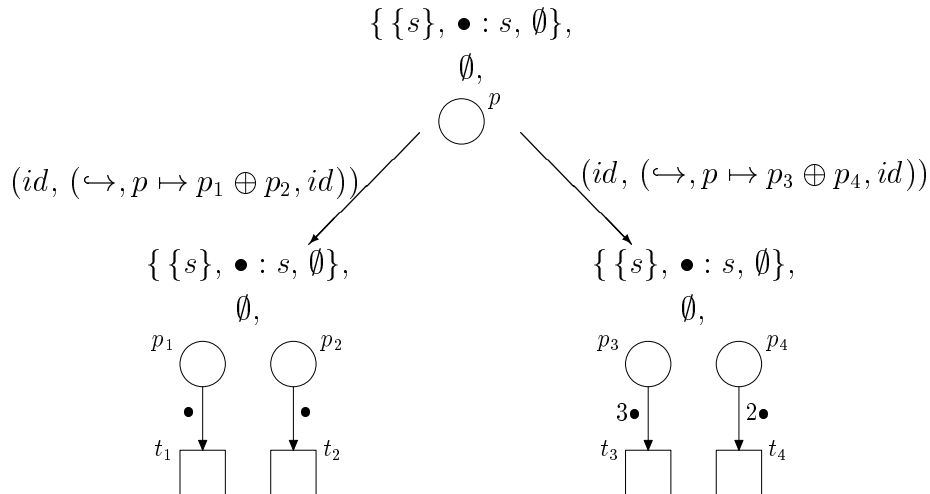


Abbildung 3.4: Mehrdeutige Komposition

Für die Verschmelzung der Schnittstellen, die im Fall der linken Spezifikation aus den Stellen p_1 und p_2 und im Fall der rechten Spezifikation aus den Stellen p_3 und p_4 bestehen, gibt es zwei gleichwertige Alternativen: zum einen die Verbindung von p_1 mit p_3 und von p_2 mit p_4 , zum anderen die Verbindung von p_1 mit p_4 und von p_2 mit p_3 . Die Netzkomponenten der daraus resultierenden Spezifikationen in Abb. 3.5 sind deutlich verschieden und damit auch die durch sie beschriebenen Systemeigenschaften.



Abbildung 3.5: Mögliche Resultate der Komposition in Abb. 3.4

Um eine der beiden möglichen Resultate des obigen Diagramms eindeutig zu beschreiben, muß die Netzkomponente der Schnittstellenspezifikation zwei Stellen enthalten, wie in Abb. 3.6 gezeigt. In diesem Fall bildet die linke der beiden AN-Spezifikationen in Abb. 3.5 das Ergebnis der Komposition.

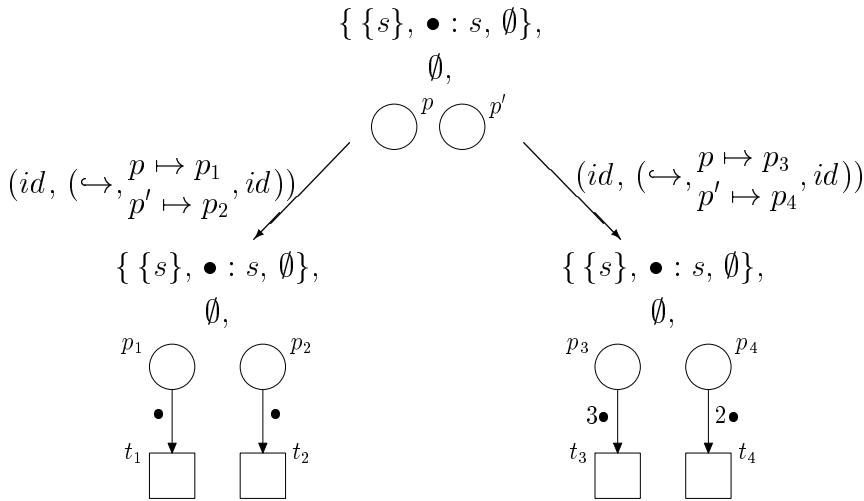


Abbildung 3.6: Eindeutige Komposition

□

Morphismen in den freien abelschen Gruppen, die frei über einer Funktion zwischen ihren Basismengen erzeugt sind, bezeichnen wir als funktionsgeneriert (\rightarrow 5.2.5). Unter funktionsgenerierten Morphismen werden Basiselemente immer auf Basiselemente abgebildet. Da in der Kategorie der Mengen und Funktionen (\mathcal{SET}) zu jedem endlichen Diagramm ein Colimes existiert und die „Freie Konstruktion“ von den Mengen in die freien abelschen Gruppen Colimites bewahrt, existiert auch in \mathcal{AG}_f zu jedem Diagramm, dessen Morphismen funktionsgeneriert sind ein Colimes.

3.3.3 Definition (Komposition):

(I) Eine endliche Kategorie \mathcal{S} ist ein „Kompositionsmuster“, wenn gilt:

$$\exists s \in \text{Obj}(\mathcal{S}), \forall s' \in \text{Obj}(\mathcal{S}), \forall m \in \text{Mor}(\mathcal{S}):$$

$$(\text{dom}(m) = s \vee m = \text{id}) \wedge \text{Mor}(\mathcal{S})[s, s'] \neq \emptyset$$

- (II) Ein „Kompositionsdiagramm“ (kurz eine „Komposition“) ist ein Funktor \mathcal{CP} von einem Kompositionsmuster \mathcal{S} nach $\mathcal{AHLN-SPEC}$, für den unter der Annahme $\mathcal{CP}(m) = (cp_m^S, cp_m^N)$ für alle Morphismen m in \mathcal{S} gilt:
- (a) für alle Morphismen m in \mathcal{S} ist die Transitionskomponente von cp_m^N funktionsgeneriert oder es existiert höchstens ein Morphismus m in \mathcal{S} , für den die Transitionskomponente von cp_m^N nicht funktionsgeneriert ist und für alle anderen Morphismen m' ist die Transitionskomponente von $cp_{m'}^N$ von einer injektiven Funktion generiert;
 - (b) für alle Morphismen m in \mathcal{S} ist die Stellenkomponente von cp_m^N funktionsgeneriert oder es existiert höchstens ein Morphismus m in \mathcal{S} für den die Stellenkomponente von cp_m^N nicht funktionsgeneriert ist und für alle anderen Morphismen m' ist die Stellenkomponente von $cp_{m'}^N$ von einer injektiven Funktion generiert.
- (III) Der „Wert einer Komposition“ ist ihr Colimes in $\mathcal{AHLN-SPEC}$. □

3.3.4 Satz (Kompositionsdiagramme):

Jedes Kompositionsdiagramm hat einen Wert. □

(Beweis → Anhang S. 241)

Notation: Alle im weiteren verwendeten Kompositionen beziehen sich auf eine mit \mathcal{S}_{cp} bezeichnete Kategorie als Kompositionsmuster. Damit stehen Domäne und Codomäne fest und wir notieren Kompositionen $\mathcal{CP}: \mathcal{S}_{cp} \rightarrow \mathcal{AHLN-SPEC}$ einfach durch \mathcal{CP} .

3.3.5 Beispiel (Eine strukturierte Spezifikation des Alternating-Bit Protokolls):

Vor die Aufgabe gestellt, ein Protokoll für die Kommunikation über eine unsichere Verbindung zu spezifizieren (→ 2.1.4), bietet es sich an, das Problem in einen sender- und einen empfängerspezifischen Teil zu gliedern. Dabei fungiert die Systemkomponente, auf die beide Teile Zugriff haben, die Kommunikationsverbindung als Schnittstelle. Mit Hilfe dieser Gliederung wollen wir unsere Protokollspezifikation aus Abbildung 2.1 neu entwerfen und im Anschluß durch einen Initialisierungsschritt erweitern.

Die Gliederung unserer Spezifikation erstreckt sich auf jede der drei AN-Spezifikationskomponenten (→ 2.1.3). So fehlen, gegenüber der ursprünglichen Spezifikation in der Datentypspezifikation des Empfängers (→ Abb. 3.8) die Operationen „cont“ und „|-“⁹, die nur vom Sender (→ Abb. 3.7) benötigt werden. Netzstruktur und Gleichungsmengen sind ebenfalls auf die für den Empfänger notwendige Funktionalität reduziert.

⁹ „cont“ ordnet jeder Nachricht eine Kontrollinformation zu, „|-“ ordnet jeder durch Anwendung der Operation „-“ manipultierten Kontrollinformation ihren ursprünglichen Wert zu

ABP-Sender =

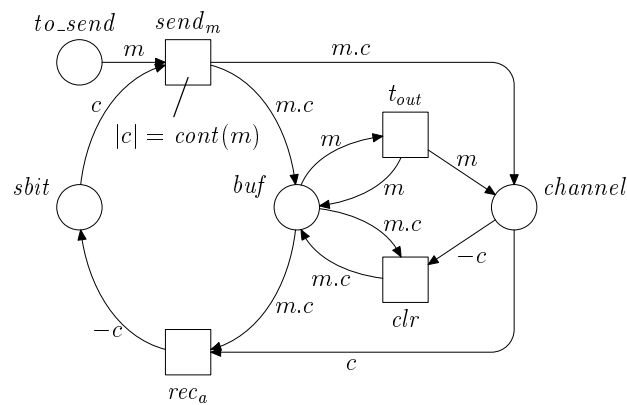
$$\begin{aligned} & \{ \{ Message, Control, MessContr \}, \\ & \{ _ . _ : Message, Control \rightarrow MessContr, \\ & \quad cont : Message \quad \rightarrow Control, \\ & \quad | _ | : Control \quad \rightarrow Control, \\ & \quad - : Control \quad \rightarrow Control \} \\ & \{ | cont(n) | = cont(n), \\ & \quad | - cont(n) | = cont(n), \\ & \quad -(-x) = x \} \end{aligned}$$


Abbildung 3.7: Alternating-Bit Protokoll / Sender.

ABP-Empfänger =

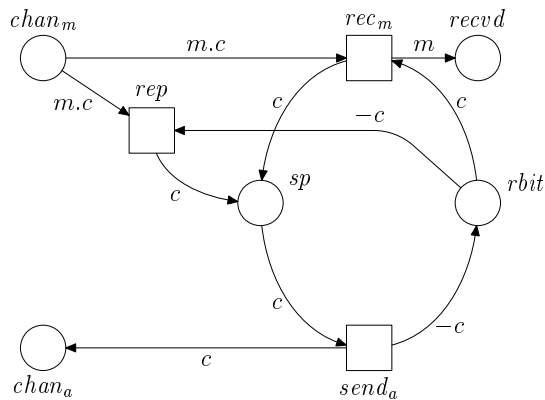
$$\begin{aligned} & \{ \{ Message, Control, MessContr \}, \\ & \{ _ . _ : Message, Control \rightarrow MessContr, \\ & \quad - : Control \quad \rightarrow Control \} \\ & \{ -(-x) = x \} \end{aligned}$$


Abbildung 3.8: Alternating-Bit Protokoll / Empfänger.

ABP-Kommunikation =

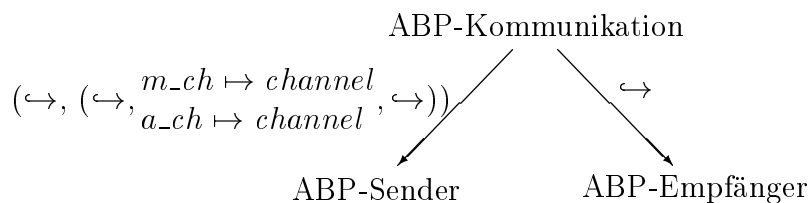
$$\begin{aligned} & \{ \{ Message, Control, MessContr \}, \\ & \{ _ _ : Message, Control \rightarrow MessContr, \\ & _ : Control \quad \rightarrow Control \}, \\ & \emptyset \} \\ \\ & \begin{array}{cc} m_ch & a_ch \\ \bigcirc & \bigcirc \end{array} \end{aligned}$$

Abbildung 3.9: Alternating-Bit Protokoll / Kommunikationsverbindung.

Darüber hinaus haben Sender und Empfänger eine unterschiedliche Sicht der Schnittstelle, während der Sender von einem gemeinsamen Kanal für Nachrichten und Bestätigungen ausgeht, arbeitet das Empfängerprotokoll mit getrennten Kanälen. Unterschiede dieser Art können z.B. auftreten, wenn die Komponenten von unterschiedlichen Personen innerhalb eines Teams oder zu verschiedenen Zeitpunkten (z.B. bei der Integration von Standardkomponenten) spezifiziert wurden. Solange die Eigenschaften der Schnittstellen verträglich sind, wie im vorliegenden Fall, können diese Unterschiede mit Hilfe der Schnittstellenspezifikation ausgeglichen werden.

Sender- und Empfängerteil des Protokolls sind über die Kommunikationsverbindung miteinander verknüpft. Deren AN-Spezifikation „ABP-Kommunikation“ beschreibt den „kleinsten gemeinsamen Nenner“, d.h. die detaillierteste Beschreibung, der verschiedenen Sichten dieser Schnittstelle in den beiden Komponenten.

Die Komposition von Sender- und Empfängerspezifikation über deren Kommunikationsverbindung, wird durch das Kompositionsdiagramm



beschrieben. Als Ergebnis der Komposition erhalten wir eine Protokollspezifikation, die bis auf den zu einer Stelle verschmolzenen Kommunikationskanal der Spezifikation in Abbildung 2.1 entspricht.

Um unsere Protokollspezifikation zu vervollständigen, führen wir einen Initialisierungsschritt ein (\rightarrow Abb. 3.10), der für jede Nachrichtenklasse¹⁰ ein Kontrollbit auf der Stelle „*cbit*“ erzeugt. Die Spezifikation verfügt über zwei Stellen, „*sbit*“ für den Sender und „*rbit*“ für den Empfänger, auf denen die Kontrollbits abgelegt sind. Hinter

¹⁰Die Nachrichten sind in Klassen eingeteilt. Nur innerhalb einer Nachrichtenklasse wird die Reihenfolge der Nachrichten bewahrt, Nachrichten verschiedener Klassen können parallel verschickt werden.

ABP-Initialisierung =

$$\begin{aligned} & \{ \{ Message, Control, MessContr \}, \\ & \{ _ . _ : Message, Control \rightarrow MessContr, \\ & \quad cont: Message \quad \rightarrow Control \}, \\ & \emptyset \} \end{aligned}$$

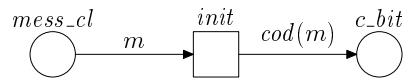


Abbildung 3.10: Alternating-Bit Protokoll / Protokollinitialisierung.

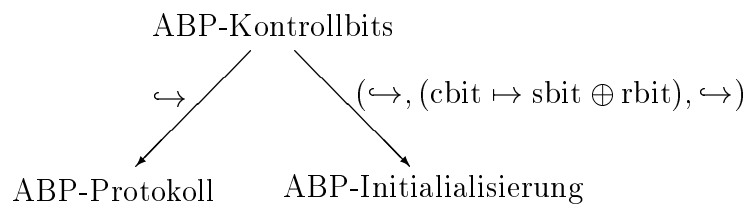
ABP Kontrollbits =

$$(\{ Message, Control, MessContr \}, \emptyset, \emptyset)$$



Abbildung 3.11: Alternating-Bit Protokoll / Kontrollstellen.

dieser Verteilung steht die Annahme, daß Sender und Empfänger an verschiedenen, möglicherweise weit voneinander entfernten Orten angesiedelt sind und daher der Zugriff auf eine globale Kontrollinformation nicht möglich ist. Als Konsequenz aus dieser Designentscheidung muß dafür gesorgt werden, daß vor der ersten Kommunikation beide Stellen mit den gleiche Werten belegt sind. Wir stellen diese Protokolleigenschaft sicher, indem wir die Initialisierung für einen Platz („*cbit*“ in Abb. 3.10) spezifizieren und diesen Platz mit den damit verbundenen Kanten bei der Komposition duplizieren. Dieser Effekt wird durch die Abbildung von „*cbit*“ auf die Summe „*sbit* \oplus *rbit*“ im Kompositionsdiagramm erreicht.



Im Ergebnis der Komposition (\rightarrow Abb. 3.12) sind dann anstelle von „*cbit*“ die zwei lokalen Stellen „*sbit*“, „*rbit*“ enthalten und die in „*cbit*“ mündende Kante ist aufgesplittet.

Die Namensgebung von Sorten, Operationen, Stellen und Transitionen im Ergebnis unserer Kompositionen, dient der Veranschaulichung von Zusammenhängen. Sie ist insofern willkürlich als die Ergebnisspezifikation nur bis auf Isomorphie eindeutig ist, d.h. die Systemeigenschaften liegen fest, nicht aber deren Repräsentation.

□

Alternating-Bit Protokoll + Init =

$\{ \{ Message, Control, MessContr \},$
 $\{ _ _ : Message, Control \rightarrow MessContr,$
 $cont: Message \rightarrow Control,$
 $| _ | : Control \rightarrow Control,$
 $- : Control \rightarrow Control \}$
 $\{ |cont(n)| = cont(n),$
 $| - cont(n)| = cont(n),$
 $-(-x) = x \}$

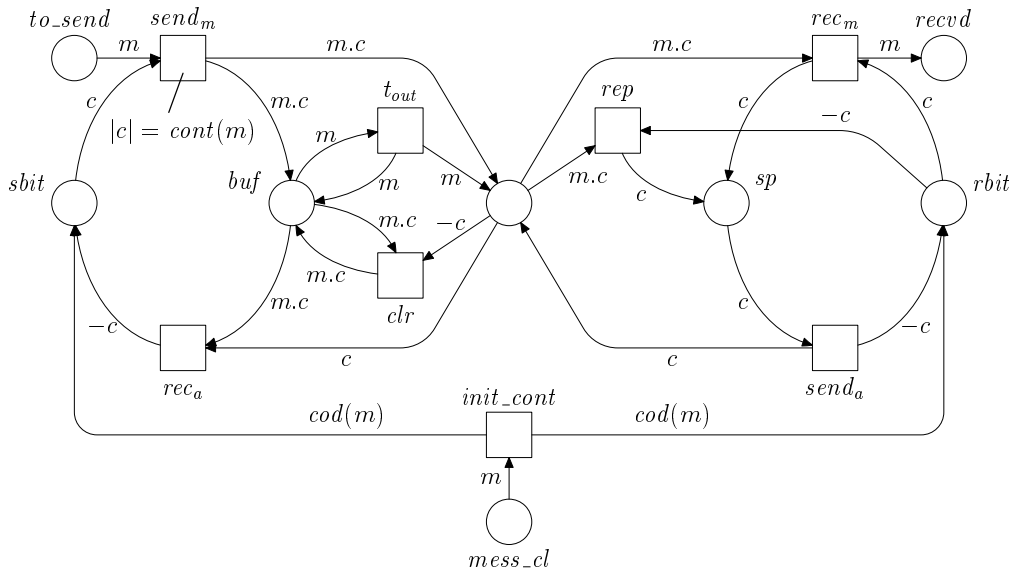


Abbildung 3.12: Alternating-Bit Protokoll mit Initialisierung.

Über Kompositionsmuster lassen sich neben AN-Spezifikationen auch deren Modelle verknüpfen, sofern sie über Morphismen in $\mathcal{GAHL-NETS}$ miteinander in Beziehung stehen. Voraussetzung dafür ist zum einen, daß der Spezifikationskomponente eines derartigen Diagramms ein Wert zugeordnet werden kann, d.h. daß es sich dabei um ein Kompositionsdiagramm handelt, zum anderen muß, aus den selben Gründen wie bei den Spezifikationen, die Verknüpfung der Netzkomponenten der Modelle eindeutig sein. Diese Form der Modellkomposition ist grundlegend für die im folgenden eingeführte Modellsemantik der Komposition und der Parameterisierung von Systemspezifikationen.

3.3.6 Satz (Kompositionsmuster und Colimites):

Sei \mathcal{CP} ein Kompositionsdiagramm und $\mathcal{D}: \mathcal{S}_{cp} \rightarrow \mathcal{GAHL-NETS}$ ein Funktor mit

$$\mathcal{D}(m) = (\mathcal{CP}(m), (f_m^G, f_m^N))$$

für alle Morphismen m in \mathcal{S}_{cp} und es gilt:

- (I) für alle Morphismen m in \mathcal{S}_{cp} ist die Transitionskomponente von f_m^N funktionsgeneriert oder es existiert höchstens ein Morphismus m in \mathcal{S}_{cp} für den die Transitionskomponente von f_m^N nicht funktionsgeneriert ist und für alle anderen Morphismen m' ist die Transitionskomponente von $f_{m'}^N$ von einer injektiven Funktion generiert;
- (II) für alle Morphismen m in \mathcal{S}_{cp} ist die Stellenkomponente von f_m^N funktionsgeneriert oder es existiert höchstens ein Morphismus m in \mathcal{S}_{cp} für den die Stellenkomponente von f_m^N nicht funktionsgeneriert ist und für alle anderen Morphismen m' ist die Stellenkomponente von $f_{m'}^N$ von einer injektiven Funktion generiert,

dann hat \mathcal{D} einen Colimes.

□

(Beweis → Anhang S. 242)

Das Zusammensetzen großer Systemspezifikationen aus Komponenten erleichtert die Spezifikation und Analyse komplexer Systeme nur dann wirkungsvoll, wenn die Gesamtheit der für die Komponentenspezifikationen geltende Systemeigenschaften genau die Eigenschaften der komponierten Spezifikation ausmacht. Gehen Eigenschaften verloren oder kommen sie kompositionsbedingt hinzu, muß für die Analyse und den Vergleich von Spezifikationen mit möglichen Modellen auf die Gesamtspezifikation zurückgegriffen werden, von der man wegen ihrer Größe und der damit verbundenen Unübersichtlichkeit wegkommen wollte.

Im Rahmen unserer Modellsemantik lassen sich daraus zwei Forderungen ableiten: zum einen muß zu jeder Auswahl aus miteinander verträglichen Modellen der Komponentenspezifikationen ein Modell der komponierten Spezifikation existieren, daß sich genau aus diesen Modellen zusammensetzt, zum anderen muß sich jedes Modell der

komponierten Spezifikation eindeutig in Modelle der Komponentenspezifikationen dekomponieren lassen. Die Einschränkung auf miteinander verträgliche Modelle, darunter verstehen wir Modelle, deren Vergißbilder in der Kategorie der Schnittstellenmodelle gleich sind, ist notwendig um zu gewährleisten, daß sich diese Modelle unverändert im korrespondierenden Modell der komponierten Spezifikation wiederfinden lassen.¹¹ Im nachfolgenden Satz verwenden wir äquivalent zur Gleichheit der Vergißbilder die Existenz von modellidentischen Morphismen (\rightarrow S. 55) vom Schnittstellenmodell in die Komponentmodelle, um deren Verträglichkeit zu gewährleisten.

3.3.7 Satz (Verträglichkeit von Komposition und Modellsemantik):

Sei \mathcal{CP} eine Komposition mit Wert $((c_s)_{s \in \mathcal{S}_{cp}}, ANS)$ und

$$\mathcal{D}: \mathcal{S}_{cp} \rightarrow \mathcal{GAHL-NETS}$$

ein Funktor mit $\mathcal{D}(m) = (\mathcal{CP}(m), id)$, dann

(I) hat \mathcal{D} einen Colimes in $\mathcal{GAHL-NETS}$, für dessen Objekt AN gilt:

$$\forall s \in \text{Obj}(\mathcal{S}_{cp}): \mathcal{V}_{c_s}(AN) = \mathcal{D}(s)$$

(II) und für alle ANS -Modelle \overline{AN} gilt:

$$\left(\forall s \in \text{Obj}(\mathcal{S}_{cp}): \mathcal{V}_{c_s}(\overline{AN}) = \mathcal{D}(s) \right) \implies \overline{AN} = AN.$$

□

(Beweis \rightarrow Anhang S. 244)

Die Eigenschaften: Existenz und Eindeutigkeit von komponierten Objekten und deren eindeutige Dekomponierbarkeit, lassen sich von den Modellen auf die Modellmorphis-men fortsetzen.¹² Damit läßt sich die gesamte Modellkategorie einer komponierten Systemspezifikation in Modellkategorien zu den Komponentenspezifikationen dekomponieren. Setzt man dagegen die Modellkategorien der Komponenten voraus, so ergibt sich die Modellkategorie der komponierten Spezifikation als Limes des Diagramms das entsteht, wenn man im Kompositionsdiagramm die Spezifikationen durch ihre Modellkategorien und Spezifikationsmorphis-men durch die generierten Vergißfunktoren ersetzt.

Die praktische Bedeutung von Satz 3.3.7 liegt vorallem in der eindeutigen Dekomponierbarkeit von Modellen komponierter Spezifikationen. Auf dieser Grundlage kann z.B. der Aufwand von Analysewerkzeugen wie „Modellprüfern“ erheblich gesenkt werden, indem man Eigenschaften auf den gegenüber dem Gesamtmodell viel kleineren

¹¹In vielen Fällen existiert auch zu unverträglichen Modellen ein korrespondierendes Modell der komponierten Spezifikation, aber in diesem Modell sind die von der Schnittstelle generierten Modelleile vereinheitlicht und das bedeutet, im Fall von unverträglichen Modellen, verändert.

¹²Satz 3.3.7 impliziert eine Amalgamierung im Sinne von [EBO91].

Modellkomponenten berechnet. Selbst wenn dabei nacheinander alle Komponenten eines Modells betrachtet werden müßten, würde man aufgrund der hyperlinearen Komplexität dieser Aufgabe immernoch gewinnen. Im Rahmen von Simulationswerkzeugen ergeben sich aus der Modelldekomposition auf Systemteile bezogene lokale Darstellungen von Modellabläufen, die z.B. die Fehlererkennung erheblich vereinfachen.

3.3.8 Beispiel (Dekomposition eines Modells zum Alternating-Bit Protokoll):

Im vorangehenden Beispiel (\rightarrow 3.3.5) haben wir eine Spezifikation des Alternating-Bit-Protokolls mit Initialisierung (\rightarrow Abb. 3.12) aus Teilspezifikationen (\rightarrow Abb. 3.7–3.10) komponiert.

M: Alternating-Bit Protokoll + Init, *id*,

$$\{ \{ \{ mess_0, \dots, mess_{10^n} \}, \{ -10^n, \dots, 10^n \}, \{ mess_0, \dots, mess_{10^n} \} \times \{ -10^n, \dots, 10^n \} \}$$

$$\{ |, \downarrow, \Delta, \pm \text{ mit}$$

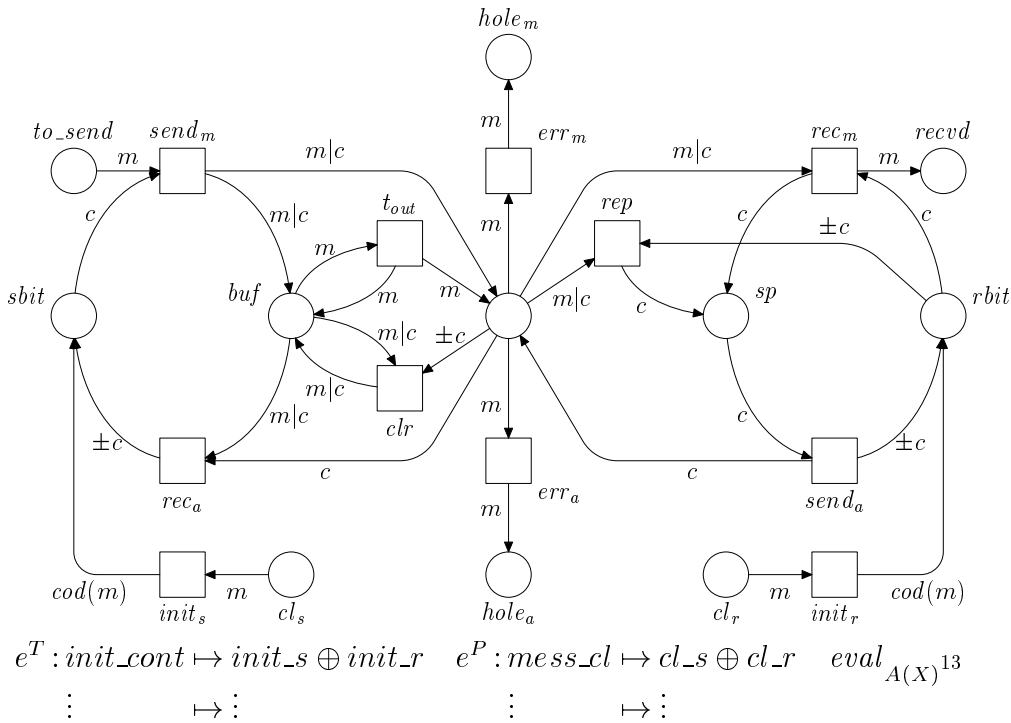
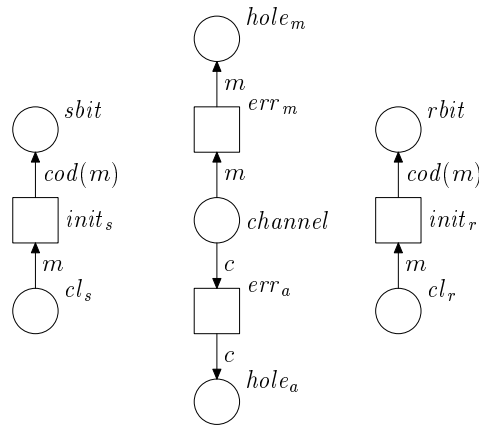
$$m | x = (m, x), \downarrow m_i = i, \Delta x = x^2/x, \pm x = -1 * x \}$$


Abbildung 3.13: Ein Modell des Alternating-Bit Protokolls aus Abb. 3.12.

¹³entspricht $eval_{A(X)}$ in Abb.2.2

Das A-Netz in Abbildung 3.13 ist ein Modell dieser Spezifikation und kann nach Satz 3.3.7 eindeutig in Modelle zu den Komponentenspezifikationen zerlegt werden. In den Abb. 3.14–3.16 zeigen wir einige dieser Modellreduktionen. Die Modelle zu den

$$\begin{aligned} \mathcal{V}_{c_2}(M): & \text{ ABP-Initialisierung, } c_2^S, \\ & \{\{\{mess_0, \dots, mess_{10^n}\}, \{-10^n, \dots, 10^n\}\}, \\ & \quad \{mess_0, \dots, mess_{10^n}\} \times \{-10^n, \dots, 10^n\}\} \\ & \{ |, \downarrow, \Delta, \pm \text{ mit} \\ & \quad m \mid x = (m, x), \downarrow m_i = i, \Delta x = x^2/x, \pm x = -1 * x \} \end{aligned}$$



$$\begin{aligned} e^T : \text{init_cont} &\mapsto \text{init_s} \oplus \text{init_r} & e^P : \text{mess_cl} &\mapsto \text{cl_s} \oplus \text{cl_r} & \text{eval}_{A(X)} \\ \vdots & \mapsto \vdots & \vdots & \mapsto \vdots \end{aligned}$$

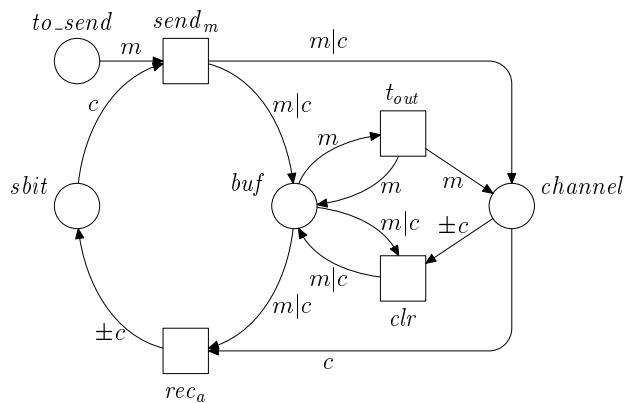
Abbildung 3.14: Projektion des Modells aus Abb. 3.13 auf den Initialisierungsteil.

jeweiligen Schnittstellenspezifikationen und das Modell zum Alternating-Bit Protokoll ohne Initialisierung¹⁴ wurden weggelassen. Wir nehmen in diesem Zusammenhang an, daß „ c_i “ ($i = 1, 2$) den Morphismus von der jeweiligen Spezifikationskomponente in das Kompositionsergebnis und „ c_i^S “ dessen Datentypkomponente bezeichnet.

Im Modell M sind die Transitionen „ err_m “, „ err_a “ und die Stellen „ $hole_m$ “, „ $hole_a$ “ nicht generiert, d.h. es existieren keine Entsprechungen in der Netzstruktur der Spezifikation. Das hat zur Folge, daß diese Stellen und Transitionen sowie die Stelle „ $channel$ “ in allen Modellreduktionen präsent sind (\rightarrow 3.1.4). Die generierte Stelle „ $channel$ “ wird hinzugenommen, weil sie mit den nichtgenerierten Transitionen verbunden ist und Dekompositionen von CP/T-Netzen (\rightarrow 1.2.3) als Teil der Modellreduktion immer zu vollständigen Teilnetzen führen.

¹⁴entspricht dem Modell in Abb. 2.2

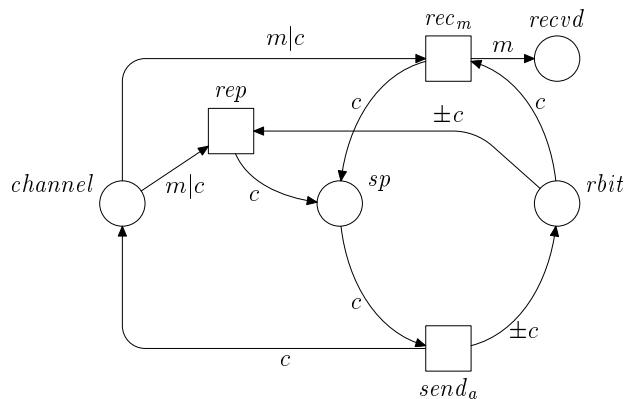
$\mathcal{V}_{c_1}(M)$: ABP-Sender, c_1^S ,
 $\{\{mess_0, \dots, mess_{10^n}\}, \{-10^n, \dots, 10^n\},$
 $\{mess_0, \dots, mess_{10^n}\} \times \{-10^n, \dots, 10^n\}\}$
 $\{ |, \downarrow, \Delta, \pm \text{ mit}$
 $m \mid x = (m, x), \downarrow m_i = i, \Delta x = x^2/x, \pm x = -1 * x\}$



$(\hookrightarrow, \leftrightarrow, eval_A(X))$

Abbildung 3.15: Projektion des Modell aus Abb. 3.13 auf den Senderteil.

$\mathcal{V}_{c_2}(M)$: ABP-Empfänger, c_2^S ,
 $\{\{mess_0, \dots, mess_{10^n}\}, \{-10^n, \dots, 10^n\},$
 $\{mess_0, \dots, mess_{10^n}\} \times \{-10^n, \dots, 10^n\}\}$
 $\{ |, \downarrow, \Delta, \pm \text{ mit}$
 $m \mid x = (m, x), \downarrow m_i = i, \Delta x = x^2/x, \pm x = -1 * x\}$



$(\hookrightarrow, \leftrightarrow, eval_A(X))$

Abbildung 3.16: Projektion des Modell aus Abb. 3.13 auf den Empfängerteil.

Mit den Modellen zur Initialisierung, zur Kontrollstellenspezifikation und zum Alternating-Bit Protokoll ohne Initialisierung, von denen hier nur das erste Modell gezeigt ist, endet der erste der beiden Zerlegungsschritte. Die weitere Zerlegung des Modells zum Alternating-Bit Protokoll ohne Initialisierung (entspricht dem A-Netz in Abb. 2.2) führt zu den A-Netzen in Abb. 3.15 und 3.16.

□

Die operationale Semantik von Systemspezifikationen leitet sich aus deren Modellsemantik ab (\rightarrow 2.3.7). Darüber hinaus läßt sich auch generalisierten Modellmorphismen eine Beziehung zwischen den operationellen Semantiken der entsprechenden Modelle zuordnen. Mit Hilfe des Umwegs über die Modelle sind wir damit in der Lage, die operationalen Semantiken der Komponenten und des Wertes einer Komposition ebenso in Beziehung zu setzen, wie deren Modellsemantiken.

Es ist aber nicht in jedem Fall nötig, daß zu den Modellen der Komponenten korrespondierende Modell der komponierten Spezifikation zu erzeugen. Dessen operationale Semantik enthält für jeden Zustand und jeden Ablauf aus einer der Komponentmodellsemantiken eine Entsprechung, sodaß sich bestimmte Fragestellungen, wie die Erreichbarkeit von Systemzuständen oder die Existenz von Abläufen, schon auf den operationalen Semantiken der Komponentmodelle klären lassen.

3.3.9 Korollar (Operationale Semantik und Komposition):

Sei \mathcal{CP} eine Komposition mit Wert $((c_s)_{s \in \mathcal{S}_{cp}}, ANS)$, $\mathcal{D}: \mathcal{S}_{cp} \rightarrow \mathcal{GAHL-NETS}$ ein Funktor mit

$$\mathcal{D}(m) = (\mathcal{CP}(m), id)$$

für alle Morphismen m in \mathcal{S}_{cp} und AN das ANS -Modell im Colimes zu \mathcal{D} , dann gilt für alle Objekte s in \mathcal{S}_{cp} :

$$\mathcal{OS}((c_s, id)): \mathcal{OS}(\mathcal{D}(s)) \rightarrow \mathcal{OS}(AN)$$

ist ein Morphismus in \mathcal{PTS} .

□

3.4 Parameterisierung von Systemspezifikationen

Viele Systeme, wie Maschinen, Kommunikationseinrichtungen oder Prozeßsteuerungen, sind von vornherein dafür vorgesehen, mit leichten Modifikationen in unterschiedlichen Umgebungen zu arbeiten. Um sie nicht jedesmal neu beschreiben und als korrekt beweisen zu müssen, braucht man Mechanismen mit deren Hilfe umgebungsabhängige Spezifikationsteile ausgezeichnet und ersetzt werden können. Der Effekt dieser Anpassungen muß auf die Semantik übertragbar sein. Diese Anforderungen werden im Rahmen der AN-Spezifikationen, durch die „Parameterisierung“ und die „Parameterübergabe“ erfüllt.

Parameterisierte Spezifikationen werden durch Inklusionsmorphismen repräsentiert. Deren Ursprung bezeichnet den „*Parameter*“, der durch die Inklusion in den „*Spezifikationsrumpf*“ eingebettet ist. Der Parameter wird bei der Anpassung an eine Umgebung durch eine passende Systemspezifikation ersetzt. Diesen Vorgang bezeichnen wir als „*Parameterübergabe*“ und die eingesetzte Spezifikation als „*Aktualisierung*“. Damit Parameterübergaben wirklich zu Anpassungen führen und nicht zu Spezifikationen mit völlig anderen Eigenschaften, müssen die im Parameter festgelegten Systemeigenschaften auch in der Aktualisierung gelten. Dieser Zusammenhang wird, als Teil der Parameterübergabe, durch einen Spezifikationsmorphismus zwischen Parameter und Aktualisierung sichergestellt.

3.4.1 Definition (Parameterisierte AN-Spezifikation):

Seien ANS und PAR AN-Spezifikationen, dann bezeichnet $ANS(PAR)$ eine parameterisierte Spezifikation, wenn die Inklusion

$$\hookrightarrow: PAR \rightarrow ANS$$

ein Morphismus in $\mathcal{AHLN}\text{-SPEC}$ ist. □

Parameterisierte Spezifikationen werden mit Hilfe von Parameterübergaben an die gewünschten Umgebungen angepaßt. Das Ergebnis einer Parameterübergabe, die aktualisierte Spezifikation, ergibt sich aus der Komposition der Aktualisierung mit der parameterisierten Spezifikation über den Parameter als Schnittstelle. Dabei entsteht in Folge der Inklusionsbeziehung von Parameter und parameterisierter Spezifikation, eine Inklusion zwischen Aktualisierung und aktualisierter Spezifikation.

3.4.2 Definition (Parameterübergabe):

Sei $ANS(PAR)$ eine parameterisierte AN-Spezifikation und ACT eine AN-Spezifikation, dann bezeichnen wir:

(I) einen AN-Spezifikationsmorphismus

$$h: PAR \rightarrow ACT$$

als „*Parameterübergabemorphismus*“ und

(II) die AN-Spezifikation $ANS_h(ACT)$ im Pushout¹⁵

$$\begin{array}{ccc} PAR & \xrightarrow{\hookrightarrow} & ANS \\ h \downarrow & \text{(PO)} & \downarrow c_1 \\ ACT & \xrightarrow{c_2} & ANS_h(ACT) \end{array}$$

als Ergebnis der von h induzierten Parameterübergabe. □

¹⁵Pushouts sind eine Kombination aus Diagramm und Colimes (\rightarrow 5.1.18)

Die Semantik parameterisierter Spezifikationen, muß den Effekt aller im Rahmen von Parameterübergaben möglichen Umgebungsanpassungen einer Systembeschreibung erfassen. Auf der Modellebene läßt sich das erreichen, indem man die Modellkategorie des Parameters mit der Modellkategorie des Rumpfes einer parameterisierten Spezifikation in Beziehung setzt. Der Parameter steht über die Parameterübergabemorphismen mit allen Aktualisierungen in Verbindung, während der Spezifikationsrumpf seinerseits über Morphismen mit allen aktualisierten Spezifikationen verbunden ist. Dadurch enthält die Modellkategorie des Parameters die Vergißbilder aller Aktualisierungsmodelle und die Modellkategorie der Rumpfspezifikation die Vergißbilder aller Modelle aktualisierter Spezifikationen. Die Vergißbilder haben bezüglich der von der Aktualisierung betroffenen Modellteile die gleichen Eigenschaften wie die ursprünglichen Modelle, sodaß ihre Betrachtung ausreicht um die Übertragung von Eigenschaften der Aktualisierungen in die aktualisierten Spezifikationen zu beschreiben.

Im Zuge einer Parameterübergabe, verändert sich in Abhängigkeit von der gewählten Aktualisierung jeweils nur der vom Parameter ausgezeichnete Teil einer parameterisierten Spezifikation. Um diesen Effekt auf die Modellsemantik zu übertragen, ordnen wir jedem Parametermodell das Modell der Rumpfspezifikation zu das entsteht, wenn man den vom Parameter generierter Teil des initialen Rumpfmodells durch das Parametermodell ersetzt. Realisiert wird diese Ersetzung durch eine Modellkomposition (\rightarrow 3.2.3) des initialen Rumpfmodells mit dem Parametermodell. Als Schnittstellenbeschreibung fungiert das initiale Parametermodell, das über generalisierte Modellmorphismen mit den beiden Komponenten in Beziehung steht.

3.4.3 Definition (Modelltransformation):

Sei $ANS(PAR)$ eine parameterisierte AN-Spezifikation, I_{ANS} das initiale ANS -Modell und I_{PAR} das initiale PAR -Modell, dann definieren wir für jedes PAR -Modell AN ein ANS -Modell $T_{\hookrightarrow}(AN)$ durch den Pushout

$$\begin{array}{ccc}
 I_{PAR} & \xrightarrow{(\hookrightarrow, eval)} & I_{ANS} \\
 (id, eval) \downarrow & (PO) & \downarrow (id, eval) \\
 AN & \xrightarrow{(\hookrightarrow, u_{AN})} & T_{\hookrightarrow}(AN)
 \end{array}$$

in $\mathcal{GAHL}\text{-}\mathcal{NETS}$.

□

Notation: Wenn sich die parameterisierte Spezifikation und der damit verbundene Inklusionsmorphismus zwischen Parameter und Rumpf auf die sich eine Transformation bzw. ein Vergißfunktorkontext her klar sind, lassen wir den Inklusionsmorphismus als Index weg und schreiben einfach $T(AN)$ für $T_{\hookrightarrow}(AN)$ bzw. \mathcal{V} für $\mathcal{V}_{\hookrightarrow}$.

Eine wesentliche Anforderung, die man allgemein an Parameterisierungen stellt, ist die Vermeidung von Nebeneffekten bei der Parameterübergabe. Übertragen auf unsere Modellsemantik bedeutet das, die Modelltransformationen dürfen außer den notwendigen Anpassungen an der Schnittstelle zwischen Rumpf- und Parametermodell keine weiteren Veränderungen bewirken, insbesondere keine Veränderungen der Modelleigenschaften. Letztere manifestierten sich in den von den Modellen ausgehenden Morphismen. So existieren z.B. vom initialen Modell (\rightarrow 2.2.6), das genau die spezifizierten Eigenschaften repräsentiert, Morphismen in alle anderen Modelle der Modellkategorie, während das Hinzufügen von Struktur und damit von Eigenschaften die Anzahl der Morphismen reduziert. Daher kann das Auftreten von Nebeneffekten in Bezug auf die Modelleigenschaften anhand der vom betreffenden Modell ausgehenden Morphismen nachgewiesen werden. Unsere Modelltransformation ist danach frei von Nebeneffekten, wenn sie die von den Modellen ausgehenden Morphismen bewahrt und keine weiteren hinzufügt.

Beim Nachweis dieser Eigenschaft helfen die von den AN-Spezifikationen generierten Vergißfunktoren. Sie erlauben es, die Bilder von Modelltransformationen in die Ursprungskategorie abzubilden. Dabei werden die Veränderungen deutlich, die im Zuge der Transformation am Modell vorgenommen wurden. Zwischen einem Modell AN und dem Bild seiner Transformation $\mathcal{V}(T(AN))$ existiert definitionsgemäß (\rightarrow 3.4.3) der Morphismus u_{AN} , d.h. die Modelleigenschaften werden von unserer Transformation zumindest bewahrt. Wenn darüber hinaus keine Eigenschaften hinzukommen und die Eigenschaften der Umgebung, in die die Transformation die Modelle eingebettet, erhalten werden, spricht man von „frei konstruierten Modellen“ [HS73, EM85, EBO91].

Das Bild $T(AN)$ eines Modells AN unter der von der Inklusion des Parameters in den Rumpf generierten Modelltransformation ist bezüglich des vom gleichen Morphismus generierten Vergißfunktors \mathcal{V} „frei konstruiert“, wenn zu jedem von AN ausgehenden Morphismus f ein Morphismus f' in der Modellkategorie des Rumpfes existiert, dessen Vergißbild das folgende Diagramm eindeutig kommutativ ergänzt.

$$\begin{array}{ccccc}
 AN & \xrightarrow{u_{AN}} & \mathcal{V}(T(AN)) & & T(AN) \\
 & \searrow f & \curvearrowright & \downarrow \mathcal{V}(f') & \downarrow f' \\
 & & & \mathcal{V}(AN') & AN'
 \end{array}$$

3.4.4 Satz (Freie Konstruktion):

- (I) Für jede parameterisierte AN-Spezifikation $ANS(PAR)$ und jedes PAR -Modell AN ist

$$\mathcal{F}_{\hookrightarrow}(AN) \stackrel{\text{def}}{=} T_{\hookrightarrow}(AN)$$

bezüglich des Vergißfunktors $\mathcal{V}_{\hookrightarrow}$ frei konstruiert, mit universeller Abbildung $u_{AN}: AN \rightarrow \mathcal{V}_{\hookrightarrow} \circ \mathcal{F}_{\hookrightarrow}(AN)$ (\rightarrow 3.4.3).

(II) Ist darüber hinaus die Datentypspezifikation $Spec_p$ aus PAR eine volle Unterspezifikation von $Spec$, der Datentypspezifikation aus ANS , d.h.

$$\forall s \in SO_p, op \in OP_s, eq \in EQ_s: op \in OP_{p_s} \wedge eq \in EQ_{p_s},$$

dann gilt:

$$u_{AN} = \left((\hookrightarrow, id), \left(id, id, (\hookrightarrow, id) \right) \right).$$

□

(Beweis \rightarrow Anhang S. 250)

Freie Konstruktionen können mit Hilfe der universellen Abbildungen zu Funktoren erweitert werden. Dabei ergibt sich das Bild eines Morphismus f , aufgrund der für freie Konstruktionen geltenden Eigenschaften, eindeutig aus der kommutativen Ergänzung des folgenden Diagramms

$$\begin{array}{ccc} AN_1 & \xrightarrow{f} & AN_2 \\ u_{AN_1} \downarrow & \curvearrowright & \downarrow u_{AN_2} \\ \mathcal{V}(\mathcal{F}(AN_1)) & \xrightarrow{\mathcal{V}(\mathcal{F}(f))} & \mathcal{V}(\mathcal{F}(AN_2)) \end{array}$$

Im Unterschied zur Parameterisierung, die alle möglichen Aktualisierungen des Parameters berücksichtigen muß, steht die Aktualisierung bei der Parameterübergabe fest (\rightarrow 3.4.2). Deren Semantik ergibt sich, mit Hilfe der Ergebnisse des vorangehenden Abschnitts (\rightarrow 3.3.7, 3.3.9), aus der Semantik der Komposition über die die aktualisierte Spezifikation erzeugt wird. Interpretiert man diese Ergebnisse im Rahmen von Parameterübergaben folgt daraus, daß sich jedes Modell einer aktualisierten Spezifikation eindeutig aus einem Modell der Aktualisierung und einem Modell der parameterisierten Spezifikation zusammensetzt. Damit ist die Semantik der Parameterübergabe mit der Semantik der Parameterisierung verträglich.

Parameterisierte Spezifikationen sind nur dann sinnvoll einsetzbar, wenn die in Parameter und Spezifikationsrumpf festgelegten Systemeigenschaften auf alle aktualisierten Spezifikationen übertragbar sind. Andernfalls würden parameterisierte Spezifikationen als Komponenten eines Gesamtsystems, eine Berechnung von dessen Eigenschaften in Hinblick auf spätere Umgebungsanpassungen durch Parameteraktualisierung unmöglich machen. Für die operationale Semantik ergibt sich daraus die Forderung nach der Übertragbarkeit von Abläufen der Parametermodelle in Abläufe der frei darüber konstruierten Rumpfm Modelle. Da Abläufe und erreichbare Markierungen eine Kategorie (die Prozeßkategorie) bilden, bedeutet Übertragbarkeit in diesem Fall die Existenz eines Funktors zwischen den Prozeßkategorien. Dieser läßt sich mit Hilfe des Semantikfunktors \mathcal{OS} (\rightarrow 3.2.5), aus den generalisierten Morphismen der Parametermodelle in ihre Bilder unter der freien Konstruktion gewinnen.

3.4.5 Satz (Operationale Semantik der Parameterisierung):

Sei $ANS(PAR)$ eine parameterisierte AN-Spezifikation, I_{PAR} , I_{ANS} die initialen Modelle vom Parameter bzw. vom Spezifikationsrumpf und AN ein Modell des Parameters, dann:

(I) kommutiert das folgende Diagramm in \mathcal{PTS} :

$$\begin{array}{ccc}
 \mathcal{OS}(I_{PAR}) & \xrightarrow{\mathcal{OS}(\hookrightarrow, eval)} & \mathcal{OS}(I_{ANS}) \\
 \mathcal{OS}(eval) \downarrow & \circlearrowleft & \downarrow \mathcal{OS}(eval) \\
 \mathcal{OS}(AN) & \xrightarrow{\mathcal{OS}(\hookrightarrow, u_{AN})} & \mathcal{OS}(\mathcal{F}(AN))
 \end{array}$$

wobei \hookrightarrow die Einbettung von PAR in ANS bezeichnet.

(II) Ist darüber hinaus die Datentypspezifikation $Spec_p$ aus PAR eine volle Unterspezifikation von $Spec$, der Datentypspezifikation aus ANS , d.h.

$$\forall s \in SO_p, op \in OP_s, eq \in EQ_s: op \in OP_{p_s} \wedge eq \in EQ_{p_s},$$

dann ist $\mathcal{OS}(AN)$ eine volle Unterkategorie von $\mathcal{OS}(\mathcal{V} \circ \mathcal{F}(AN))$, und für alle Morphismen $v: m \rightarrow m'$ in $\mathcal{OS}(\mathcal{V} \circ \mathcal{F}(AN))$ gilt:

$$v \in \mathcal{OS}(AN) \quad \text{oder} \quad v = id_m.$$

□

(Beweis \rightarrow Anhang S. 255)

Spezifikation eines „FFS“

„Flexible Fertigungssysteme“ (kurz „FFS“) für die Bohr- und Fräsbearbeitung prismatischer Werkstücke sind unabhängige logistische Einheiten, in denen mehrere, sich voll ersetzende „Bearbeitungszentren“ und eine „Werkstück-Ein/Ausgabestation“ mit Hilfe einer automatischen Werkstückversorgung zu einem Gesamtsystem verbunden sind. Dessen übergeordnete Steuerungs- und Dispositionsaufgaben werden durch einen ins System integrierten Leitreechner übernommen.

Die einzelnen Komponenten eines FFS sind konstruktiv und funktional aufeinander abgestimmt. Neben den schon erwähnten Bearbeitungszentren (kurz „BAZ“), der Ein/Ausgabestation (kurz „E/A-Station“), dem Werkstücktransportsystem und der Systemsteuerung, verfügen flexible Fertigungssysteme zumeist über einen „Werkstückspeicher“. Gegebenenfalls kommen als Zusatzeinrichtungen eine automatische Werkzeugversorgung, Waschmaschinen und automatische Meßeinrichtungen dazu.

Das Bedienpersonal arbeitet an der Schnittstelle des FFS zur Umgebung und ist weitestgehend vom Arbeitstakt der Maschinen entkoppelt. Grundlage dafür ist die Speicherung von Werkstückträgern mit in Vorrichtungen gespannten Roh- und Fertigteilen auf Abstellplätzen, die entlang des Fahrweges des Werkstücktransportsystems angeordnet sind. Diese Plätze werden in ihrer Gesamtheit als Werkstückspeicher bezeichnet. Die Hauptaufgabe des Bedienpersonals besteht im Auf- und Abspannen von Roh- bzw. Fertigteilen auf genormte Werkstückträger. Zu diesem Zweck verfügt die E/A-Station über einen oder mehrere Spannplätze.

Der Werkstücktransport erfolgt mit Hilfe von schienengebundenen oder induktiv gesteuerten Transportfahrzeugen.¹ Sie befördern die auf Werkstückträgern gespannten Teile zwischen den Bearbeitungszentren, dem Werkstückspeicher und den Spann- und Abstellplätzen.

Die Systemsteuerung koordiniert, steuert und überwacht alle Funktionen des automatischen Produktions- und Versorgungsablaufs in einem Fertigungssystem. Ihre Hauptaufgabe ist die zeitgerechte Versorgung der angeschlossenen Bearbeitungszentren und Sondermaschinen mit Werkstücken, Werkzeugen und NC-Programmen².

Die Anschaffung eines flexiblen Fertigungssystems stellt eine erhebliche Investition³ dar, die sich nur rechnet, wenn das System im tatsächlichen Betrieb reibungslos funktioniert und gut ausgelastet wird. Um dies zu gewährleisten müssen schon in der

¹in kleineren Systemen werden zum Teil auch Portalfördersysteme eingesetzt

²NC = Numeric Control

³in der Regel zwischen einer und mehreren Millionen DM

Planungsphase eine Vielzahl von Faktoren berücksichtigt werden, die je nach Fertigungssituation, d.h. nach Anzahl der zu fertigenden Teile, der benötigten Werkzeuge und der Bearbeitungszeiten in unterschiedlichem Maße den Durchsatz und die Kosten des Systems beeinflussen. Die wichtigsten Einflußgrößen sind

- die Anzahl der Bearbeitungszentren und die Größe ihrer Werkzeugmagazine,
- Typ und Anzahl der Transporteinrichtungen,
- die Anzahl der Werkzeuge im System,
- die Anzahl der Werkstückträger,
- die Anzahl der Werkstückspeicherplätze,
- die Anzahl der Spann- und Ablageplätze und
- die Anzahl des Bedienpersonals.

Aufgrund der hohen Komplexität flexibler Fertigungssysteme ist eine detaillierte Bewertung der in Frage kommenden Systemkonfigurationen durch analytische Berechnungsmethoden⁴ in der Regel nicht möglich, so daß in diesem Bereich verstärkt rechnergestützte Simulationsmethoden (→ [Vie85], [Chm85]) zum Einsatz kommen. Voraussetzung dafür ist eine formal eindeutige Systembeschreibung. Da diese Beschreibung im Idealfall eine wesentliche Rolle bei der Kommunikation zwischen Anwender und Produzent spielt, sollte sie anschaulich und überschaubar sein.

Im Rahmen dieser Arbeit dient die im weiteren schrittweise konstruierte Spezifikation eines flexiblen Fertigungssystems zur Veranschaulichung der in den vorangegangenen Kapiteln eingeführten Methoden anhand eines realitätsnahen Beispiels. Mit diesem Hintergrund wurden die folgenden vereinfachenden Annahmen getroffen:

- die vorhandenen Bearbeitungszentren können alle anfallenden Bearbeitungsaufträge ausführen, d.h. es existieren keine Sondermaschinen,
- die gesamte Bearbeitung eines Werkstücks kann ohne Unterbrechung, d.h. ohne zwischenzeitliches Waschen oder Vermessen, an einem Bearbeitungszentrum ausgeführt werden,
- von der Werkstückwechseleinrichtung der Bearbeitungszentren wird abstrahiert, d.h. die Fahrzeuge können die Werkstückträger direkt an das Maschinenbett übergeben,
- die in der Regel vorhandene automatische Werkzeugversorgung der Bearbeitungszentren bleibt unberücksichtigt und
- es wird von allen zeitlichen Aspekten wie Bearbeitungsdauer, Werkzeugstandzeit und Transportdauer abstrahiert.

⁴Warteschlangenmodelle o.ä.

Die Gliederung dieses Kapitels folgt seiner Funktion als Anwendungsbeispiel. Im ersten Abschnitt spezifizieren wir die Bearbeitungszentren und beschreiben anhand der Schrittsemantik des initialen und ausgewählter anderer Modelle die wichtigsten Abläufe.

Anschließend wird im zweiten Abschnitt die Spezifikation der Bearbeitungszentren mit den Spezifikationen der übrigen Systemkomponenten zu einer ersten FFS-Spezifikation komponiert. Der entgegengesetzte Vorgang, die Dekomposition eines FFS-Modells, veranschaulicht die Semantik der Komposition.

Im dritten und letzten Abschnitt betrachten wir die FFS-Spezifikation als parametrisierte Spezifikation mit der BAZ-Spezifikation als formalem Parameter und zeigen, wie sich mit Hilfe der freien Konstruktion als Semantik der Parameterisierung, die Modelle des ersten Abschnitts in FFS-Modelle einbetten lassen. Eine, in ihrer Funktionalität erweiterte und mit einem Werkzeugmagazin komponierte BAZ-Spezifikation ersetzt bei der Aktualisierung der FFS-Spezifikation den formalen Parameter. Im Unterschied zur Komposition im vorangehenden Abschnitt, enthält die Schnittstelle bei dieser Komposition ein komplettes Netz.

4.1 Bearbeitungszentren

Die Bearbeitung der Werkstücke erfolgt an numerisch gesteuerten Werkzeugmaschinen, den Bearbeitungszentren, deren „*Arbeitsspindeln*“ mit unterschiedlichen Werkzeugen bestückt werden können. Diese, mit einem maschinenseitig normierten Anschluß versehenen Werkzeuge, sind zwischen den Bearbeitungszentren austauschbar. Eine „*Werkzeugwechselvorrichtung*“ erlaubt den Austausch, des in der Spindel befindlichen, mit dem für die nächste Operation benötigten Werkzeug. Zu diesem Zweck muß die Spindel in die „*Werkzeugübergabeposition*“ verfahren werden.

Die auszuführenden Bearbeitungsschritte werden der Maschine in Form von Programmen übermittelt. Sie enthalten für jede Operation die Nummer des benötigten Werkzeugs und die auf die Achsen der Arbeitsspindel bezogenen Vortriebsdaten; wobei letztere bei der von uns gewählten Abstraktionsebene keine Rolle spielen. Wesentlich ist dagegen die Werkzeugnummer. Befindet sich das entsprechende Werkzeug nicht in der Spindel, wird die Bearbeitung unterbrochen und die Spindel in die Werkzeugübergabeposition verfahren.

Als ersten Schritt zur Spezifikation der Bearbeitungszentren definieren wir die verwendeten Datentypen. Dabei setzen wir die algebraischen Spezifikationen

- *Nat* (natürlichen Zahlen) (\rightarrow 6.2.5),
- *Bool* (boolsche Algebra) (\rightarrow 6.2.5),
- *Data* (formaler Parameter der durch jede Spezifikation aktualisiert werden kann) und

- $Liste(Data)$ (Listen mit beliebigen Einträgen deren Typ durch Aktualisierung des formalen Parameters festgelegt wird)

als gegeben voraus. Diese können ebenso wie alle weiteren Datentypspezifikationen mit Hilfe zweier Operationen, der Vereinigung bzw. der disjunkten Vereinigung, zu neuen Spezifikationen verknüpft werden. Das Ergebnis ergibt sich in beiden Fällen aus der entsprechenden Vereinigung der Spezifikationskomponenten. Abschließend ergänzen wir die Liste der Standarddatentypen durch die parameterisierte Spezifikation beliebiger Paare

$$\begin{aligned}
 Paar(Data \uplus Data) &= Data \uplus Data \cup \\
 \text{sorts } &PR; \\
 \text{opns } &(\ , \): DATA1, DATA2 \rightarrow PR.
 \end{aligned}$$

Im folgenden notieren wir das Ergebnis der Aktualisierung einer parameterisierten Spezifikation $Spec(Par)$ mit Hilfe einer Spezifikation Akt und eines Spezifikationsmorphisms $h: Par \rightarrow Akt$ durch $Spec_h(Akt)$. Ist der Spezifikationsmorphismus durch die Wahl der Aktualisierung bis auf Umbenennung festgelegt, im Fall von $Paar(Data \uplus Data)$ etwa dadurch, daß Akt nur eine Sorte enthält, kann er weggelassen werden. Ausführlichere Betrachtungen zu parameterisierten Datentypspezifikationen finden sich im Abschnitt 6.2 bzw. in [EM85].

Drei Objekttypen sind für die Beschreibung von Bearbeitungszentren auf dem von uns gewählten Abstraktionsniveau von Bedeutung: Werkstücke, Bearbeitungsoperationen (kurz Operationen) und Werkzeuge. Der bei deren Spezifikation gewählte Detaillierungsgrad ist speziell auf die Erfordernisse der Bearbeitungszentren ausgerichtet. Es besteht an dieser Stelle keine Notwendigkeit, Anforderungen zu berücksichtigen, die andere Teile eines FFS, beispielsweise die E/A-Stationen (\rightarrow S. 96), in Bezug auf Operationen und Eigenschaften an die Datentypen stellen.

Die (Bearbeitungs-)Operationen werden in der gleichnamigen Spezifikation beschrieben.

$$\begin{aligned}
 Operation(Data) &= Data \cup Nat \cup \\
 \text{sorts } &OPN; \\
 \text{opns } &\bowtie: NAT, DATA \rightarrow OPN.
 \end{aligned}$$

Deren einzige Operation „ \bowtie “ dient dem Erzeugen einer Bearbeitungsoperation aus einer natürlichen Zahl, der Werkzeugnummer, und einem DATA-Element stellvertretend für die übrigen Daten der Operation. Da bei der hier gewählten Betrachtung nur die Nummer des verwendeten Werkzeugs von Interesse ist, wurde der formale Parameter $Data$, der durch jede Spezifikation ersetzt werden kann, als Platzhalter für alle weiteren Operationsparameter⁵ eingesetzt.

⁵Vortriebsdaten für die Spindelachsen, Spindeldrehzahl etc.

Aus der Sicht der Bearbeitungszentren ist die als nächstes auszuführende Bearbeitungsoperation der wesentliche Aspekt der Werkstücke, repräsentiert durch die Spezifikation $Werkstück_B$.

$$\begin{aligned} Werkstück_B &= Operation \cup \\ \underline{\text{sorts}} \quad &WKST; \\ \underline{\text{opns}} \quad &+ : WKST, OPN \rightarrow WKST. \end{aligned}$$

Mit Hilfe der Operation „+“ läßt sich jedes Werkstück w , dessen Bearbeitung noch nicht abgeschlossen ist, durch den Term $w' + o$ darstellen. Der darin enthaltene Operand o legt die nächste Bearbeitungsoperation fest.

Darüber hinaus sind in $Werkstück_B$ keine weiteren Operationen, insbesondere keine erzeugenden, enthalten; so daß die initiale $Werkstück_B$ -Algebra keine Elemente zur Sorte WKST enthält. Damit wird der Tatsache Rechnung getragen, daß die Werkstücke in der E/A-Station eines FFS gebildet werden und die Bearbeitungszentren ohne die dort getroffenen Vorbereitungen nicht arbeiten können. An anderer Stelle dieses Abschnitts werden wir zeigen, wie sich im Rahmen der Modellbildung durch die Verwendung von Algebren zu der um die Variablenmenge X (\rightarrow S. 26 ff) erweiterten Spezifikation (\rightarrow 2.2), trotzdem Abläufe erzeugen lassen.

Beim Datentyp Werkzeug interessiert uns hier nur die Zuordnung von Nummern zu Werkzeugen. Sie manifestiert sich in der Operation „ ∇ “ der nachfolgenden Spezifikation.

$$\begin{aligned} Werkzeug &= Nat \cup \\ \underline{\text{sorts}} \quad &WKZ; \\ \underline{\text{opns}} \quad &\nabla : NAT \rightarrow WKZ. \end{aligned}$$

Da alle Bearbeitungszentren die gleiche Funktionalität haben, kann eine endliche Menge von ihnen in einer Spezifikation zusammengefaßt werden. Zu diesem Zweck werden anstelle von Werkstücken, Werkzeugen und Werkzeugnummern Paare verwendet, deren erste Komponente die Nummer der betreffenden Maschine angibt. Spezifiziert werden diese Paare, in dem Nat zusammen mit der jeweiligen Objektspezifikation als Aktualisierung der Spezifikation $Paar$ eingesetzt wird. Da die Zuordnung der beiden Sorten des formalen Parameters zu den Sorten der Aktualisierung im Fall der aus Werkstücken und Werkzeugen gebildeten Paare nicht eindeutig ist, muß sie durch einen Spezifikationsmorphismus festgelegt werden.

$$\begin{aligned} f: Data \uplus Data &\rightarrow Nat \cup Werkstück_M = \left(\begin{array}{l} f^{SO} : DATA1 \mapsto NAT \\ \phantom{f^{SO}} : DATA2 \mapsto WKST \end{array}, \emptyset \right) \\ g: Data \uplus Data &\rightarrow Nat \cup Werkzeug = \left(\begin{array}{l} g^{SO} : DATA1 \mapsto NAT \\ \phantom{g^{SO}} : DATA2 \mapsto WKZ \end{array}, \emptyset \right) \end{aligned} \quad 6$$

BAZ =

$\text{Paar}_f(\text{Nat} \cup \text{Werkstück}_B) \cup$

$\text{Paar}_g(\text{Nat} \cup \text{Werkzeug}) \cup$

$\text{Paar}(\text{Nat}),$

$\text{Cond}_B: ba \mapsto \emptyset$

$st \mapsto (x = n) = \text{False}$

$vf \mapsto \emptyset,$

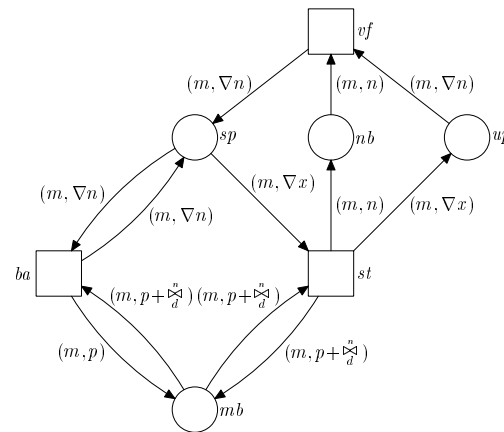


Abbildung 4.1: Ein Bearbeitungszentrum ohne lokalen Werkzeugvorrat.

Die Netzkomponente der AN-Spezifikation BAZ in Abbildung 4.1 enthält drei Transitionen. Davon ist „ ba “ für die Bearbeitung eines auf dem Maschinenbett („ mb “) befindlichen Werkstücks ($w + (n \bowtie d)$) zuständig, sofern noch eine Operation ($n \bowtie d$) vorgesehen ist, sich das dafür notwendige Werkzeug (∇n) in der Spindel („ sp “) und diese sich in Arbeitsposition befindet.

Stimmt die Nummer (x) des in der Spindel befindlichen Werkzeugs nicht mit der Nummer (n) des benötigten Werkzeugs überein, d.h. die „ st “ zugeordnete Gleichung $(x = n) = \text{False}$ ⁷ liefert auf beiden Seiten den booleschen Wert „ False “,⁸ dann schaltet st . Dadurch wird die Spindel in die Übergabeposition („ up “) gebracht und die Nummer des benötigten Werkzeugs (n) auf der Stelle „ nb “ angezeigt.

Nach dem Schalten von st bleibt das Bearbeitungszentrum passiv bis das Werkzeug in der Spindel gegen das benötigt Werkzeug ausgetauscht wird. Dieser Austausch muß

⁶da $\text{Data} \bowtie \text{Data}$ keine Operationen enthält bleibt die zweite Komponente der Spezifikationsmorphismen f, g leer

⁷das erste Gleichheitszeichen bezeichnet eine Operation auf natürlichen Zahlen mit einem booleschen Wert als Ergebnis und gehört nicht zur Gleichung!

⁸die Ungleichheit zweier Variablen läßt sich nur über Gleichungen spezifizieren

von außerhalb erfolgen und ist daher nicht mit spezifiziert. Anschließend schaltet „*vf*“ und bringt die Spindel wieder in seine Arbeitsposition.

Abbildung 4.2 zeigt eine Schrittsequenz (\rightarrow 1.3.1) des initialen *BAZ*-Modells.⁹ Um die Darstellung kompakt zu halten, wurde die Operation „ \oplus “ der freien abelschen Gruppe weggelassen und die Operanden stattdessen übereinander notiert

$$[(p, t) \oplus (p', t')] \iff \begin{bmatrix} (p, t) \\ (p', t') \end{bmatrix}.$$

Die erste Operation $n_1 \bowtie g_1$ des Werkstücks $(w + n_2 \bowtie g_2) + n_1 \bowtie g_1$ auf *BAZ* m kann ausgeführt werden, da sich das dafür benötigte Werkzeug ∇n_1 in der Spindel befindet. Für die anschließende Operation wird dagegen das Werkzeug ∇n_2 benötigt, so daß die Gleichung $(x = n) = \text{False}$ unter der Variablenbelegung $x = n_2$, $n = n_1$ und $p = w$ erfüllt (\rightarrow 2.3.1) ist. Damit sind wiederum die Vorbedingungen für die Kombination von st mit dieser Belegung in der Entfaltung (\rightarrow 2.3.3) des initialen Modells erfüllt und es existiert ein entsprechender Schaltschritt der operationalen Semantik. Ausgehend von der daraus resultierenden Markierung ist kein weiterer Schaltschritt mehr möglich.

$$\begin{bmatrix} \left(mb, \left(m, \left(w + \begin{smallmatrix} n_2 \\ \bowtie \\ g_2 \end{smallmatrix} \right) + \begin{smallmatrix} n_1 \\ \bowtie \\ g_1 \end{smallmatrix} \right) \right) \\ (sp, (m, \nabla n_1)) \end{bmatrix}; \begin{bmatrix} \left(ba, \text{ass} : n \mapsto n_1 \right. \\ \left. p \mapsto w + \begin{smallmatrix} n_2 \\ \bowtie \\ g_2 \end{smallmatrix} \right) \end{bmatrix}; \\ \begin{bmatrix} \left(mb, \left(m, w + \begin{smallmatrix} n_2 \\ \bowtie \\ g_2 \end{smallmatrix} \right) \right) \\ (sp, (m, \nabla n_1)) \end{bmatrix}; \begin{bmatrix} \left(st, \text{ass} : n \mapsto n_1 \right. \\ \left. x \mapsto n_2 \right. \\ \left. p \mapsto w \right) \end{bmatrix}; \begin{bmatrix} \left(mb, \left(m, w + \begin{smallmatrix} n_2 \\ \bowtie \\ g_2 \end{smallmatrix} \right) \right) \\ (up, (m, \nabla n_1)) \\ (nb, (m, n_2)) \end{bmatrix}$$

Abbildung 4.2: Schrittsequenz zum initialen *BAZ*-Modell.

Um die Spezifikation der Bearbeitungszentren in Form zweier Modelle aus dem Bereich der Automobilindustrie zu konkretisieren, gehen wir im folgenden davon aus, daß

- $Tl = \{\text{Motorblock}, \text{Zylinderkopf}, \text{Nockenwelle}\}$ die zu bearbeitenden Motorenteile bezeichnet,
- dafür die Werkzeuge $Wz = \{\text{Bohrer}, \text{Gewindebohrer}, \text{Fräse}\}$ benötigt werden,
- die von den Bearbeitungszentren ausführbaren Operationsparameter in der Menge $OpPar$ festgelegt sind,

⁹genau genommen existieren unendlich viele initiale Modelle, die sich aber alle bis auf Umbenennung gleichen

- $Op = \mathbb{N} \times OpPar$ gilt und
- OpT die kleinste Menge ist, die alle Op -Elemente, das Element „ \perp “ und für jedes Paar aus einem Op -Element „ o “ und einem OpT -Element „ t “ das Element „ $o.t$ “ enthält.

Außerdem nehmen wir die folgenden Funktionen als gegeben an

$$\diamond: \mathbb{N} \rightarrow Wz \quad \text{mit} \quad \diamond n = \begin{cases} \text{Bohrer,} & n \bmod 3 = 1; \\ \text{Gewindebohrer,} & n \bmod 3 = 2; \\ \text{Fräse,} & n \bmod 3 = 0; \end{cases}$$

$$\longleftrightarrow: \mathbb{N} \times OpPar \rightarrow Op \quad \text{mit} \quad \longleftrightarrow = \begin{matrix} n \\ b \end{matrix} = (n, b)$$

$$-.-: Tl \times OpT \times Op \rightarrow Tl \times OpT \quad \text{mit} \quad (t, l).o = \begin{cases} (t, o), & l = \perp; \\ (t, o.l), & \text{sonst;} \end{cases}$$

und bilden zusammen mit den vorher definierten Mengen die Algebren¹⁰

$$OpMotor = \mathbb{N} \cup (OpPar, \emptyset) \cup (Op, \longleftrightarrow) \in \mathcal{ALG}(Operation) \quad ^{11}$$

$$MotorWkst = OpMotor \cup (Tl \times OpT, -.-) \in \mathcal{ALG}(Werkstück_B)$$

$$MotorWkz = \mathbb{N} \cup (Wz, \diamond) \in \mathcal{ALG}(Werkzeug)$$

$$MNr_MotorWkst = \mathbb{N} \uplus MotorWkst \uplus (\mathbb{N} \times Tl \times OpT, (,))$$

$$MNr_MotorWkz = \mathbb{N} \uplus MotorWkz \uplus (\mathbb{N} \times Wz, (,))$$

$$MNr_WNr = \mathbb{N} \uplus (\mathbb{N} \times \mathbb{N}, (,)).$$

In der AN-Spezifikation *BAZ* sind die Minimalanforderungen festgelegt, die wir an die Bearbeitungszentren stellen. Demgegenüber verfügen diese Anlagen in der Praxis zumeist über einige Erweiterungen, wie automatische Werkzeugwechsellvorrichtungen oder einen lokalen Werkzeugvorrat in Form von Ketten- oder Palettenmagazinen. Abbildung 4.3 zeigt ein derartiges, gegenüber der Spezifikation in Abb. 4.1 um eine Werkzeugwechsellvorrichtung („*ww*“) mit Verriegelung („*mv*“) und einen Werkzeugübergabeplatz („*mu*“) erweitertes Modell. Bei Bearbeitungszentren dieses Typs kann ein Werkzeug per Hand oder durch eine automatische Transportvorrichtung in einen dafür vorgesehenen Platz abgelegt werden. Nach Freigabe dieser Position, erfaßt die Werkzeugwechsellvorrichtung in Form eines mechanischen Doppelgreifers das übergebene Werkzeug und das Werkzeug in der Spindel¹² und tauscht sie gegeneinander aus.

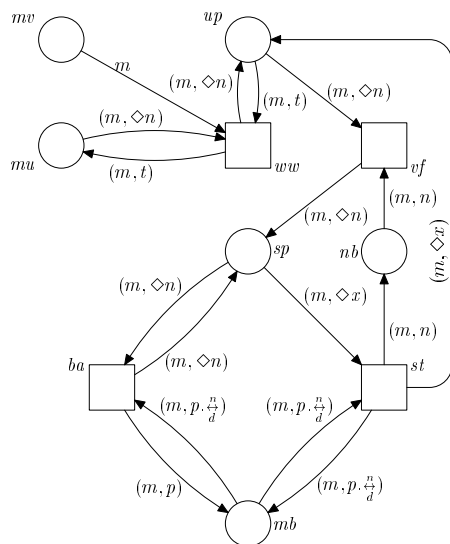
¹⁰ \mathbb{N} bezeichnet hier sowohl die Menge der natürlichen Zahlen als auch die entsprechende Algebra mit den Standardoperation „+“, „-“, „*“, „=“ etc.

¹¹die Vereinigung von Algebren erfolgt wie die Vereinigung von Spezifikationen komponentweise

¹²die sich dafür in der Übergabe position befinden muß

BAZ_WW: *BAZ*, *id*,

MNr_MotorWkst \cup *MNr_MotorWkz* \cup *MNr_WNr*



$(\leftrightarrow, \leftrightarrow, (id, eval))$

Abbildung 4.3: Ein Bearbeitungszentrum mit Werkzeugwechsler.

$$\left[\begin{array}{l} \mathcal{OS}_{BAZ}(eval)(s) \\ (mv, m) \\ (mu, (m, \diamond n_2)) \end{array} \right]; \left[\begin{array}{l} (ww, ass : m \mapsto m \\ n \mapsto n_2 \\ t \mapsto \diamond n_1) \end{array} \right]; \left[\begin{array}{l} (mb, (m, w \cdot \frac{n_2}{g_2})) \\ (mu, (m, \diamond n_1)) \\ (nb, (m, n_2)) \\ (up, (m, \diamond n_2)) \end{array} \right];$$

$$\left[\begin{array}{l} (vf, ass : m \mapsto m \\ n \mapsto n_2) \end{array} \right]; \left[\begin{array}{l} (mb, (m, w \cdot \frac{n_2}{g_2})) \\ (sp, (m, \diamond n_2)) \\ (mu, (m, \diamond n_1)) \end{array} \right]; s'$$

Abbildung 4.4: Schrittsequenz des *BAZ*-Modells *BAZ_WW*.

Das initiale *BAZ*-Modell und das Modell in Abbildung 4.3 sind durch den Modellmorphismus *eval* verknüpft.¹³ Damit läßt sich die Schrittsequenz zum initialen Modell aus Abbildung 4.2, im folgenden mit „*s*“ bezeichnet, mit Hilfe des von der Spezifikation generierten Semantikfunktors $\mathcal{OS}_{BAZ}(eval)$ (\rightarrow 2.3.7) in eine Schrittsequenz zum Modell in Abbildung 4.3 transformieren. Dabei werden in diesem Fall lediglich die Operationen umbenannt.

Die Schrittsequenz (\rightarrow Abb. 4.4) zum *BAZ*-Modell in Abbildung 4.3 enthält neben der transformierten Schrittsequenz aus Abbildung 4.2, das für die zweite Bearbeitungsoperation notwendige Werkzeug \diamond_{n_2} auf dem Werkzeugübergabepplatz (*mu*) und die Freigabe dieser Position (*mv*). Der nach Ablauf von $\mathcal{OS}_{BAZ}(eval)(s)$ aktivierte Werkzeugwechsler (*ww*) kann daraufhin das in der Spindel befindliche Werkzeug \diamond_{n_1} gegen \diamond_{n_2} tauschen. Nachdem anschließend die Spindel in eine Arbeitsposition verfahren wurde (*vf*), kann die Bearbeitung des Werkstücks in der hier nicht weiter ausgeführten Sequenz „*s'*“ fortgesetzt werden.

Neben einer automatischen Werkzeugwechsellvorrichtung verfügt das *BAZ*-Modell in Abbildung 4.5 über ein Werkzeugmagazin. Der darin gespeicherte Werkzeugvorrat ermöglicht eine in der Regel¹⁴ bedienerfreie und, abgesehen von Positionier- und Übergabezeiten unterbrechungsfreie Bearbeitung von Werkstücken. Das gilt insbesondere auch für Werkstücke zu deren Bearbeitung mehrere unterschiedliche Werkzeuge benötigt werden.

Im Unterschied zum Modell in Abbildung 4.3, das das initiale Modell lediglich ergänzt, unterscheidet sich die Netzstruktur in Abbildung 4.5 auch im von der Spezifikation generierten Teil. Die Stelle *nb*, die die Nummer des nächsten benötigten Werkzeugs enthält, wurde im Modell „*BAZ_Mag*“ dupliziert. Zum einen wird diese Information wie in den anderen Modellen dazu verwendet, um festzustellen, ob die Spindel in eine Arbeitsposition verfahren und die Bearbeitung fortgesetzt werden kann (*nb*), zum anderen wird sie vom Magazin zur Bereitstellung des Werkzeugs gebraucht („*ma*“).

Aus Sicht der Spezifikation kommt der Zusammenhang von *nb* mit der Stellensumme $nb \oplus ma$ in *BAZ_Mag* in der Stellenkomponente der Evaluierungsfunktion (\rightarrow Abb. 4.5) des Modells zum Ausdruck. Darüber hinaus ist er Teil des Modellmorphismus

$$h = (\leftarrow, h^P, id): BAZ_WW \rightarrow BAZ_Mag \quad \text{mit} \quad h^P: nb \mapsto nb \oplus ma,$$

$$\vdots$$

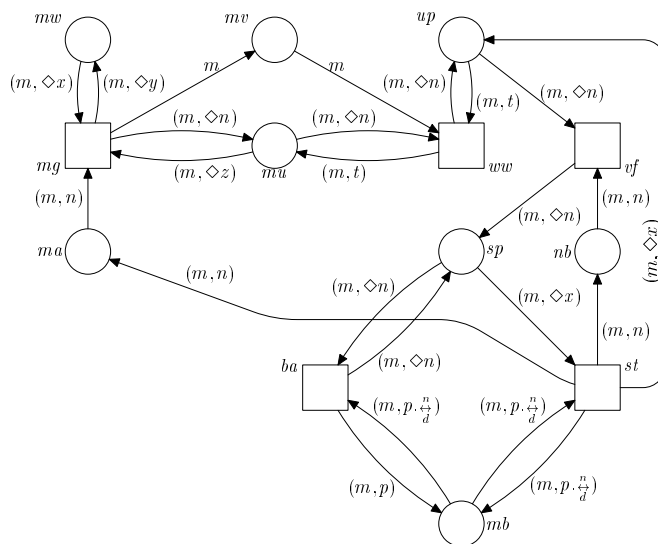
der die beiden Modelle des Bearbeitungszentrums miteinander in Beziehung setzt.

¹³vom initialen Modell existiert ein derartiger Morphismus in jedes andere Modell der Spezifikation (\rightarrow 2.2)

¹⁴abgesehen von Werkzeugbruch und Verschleiß

BAZ_Mag: *BAZ, id,*

MNr_MotorWkst \cup *MNr_MotorWkz* \cup *MNr_WNr*



$$\left(\begin{array}{l} e^P: nb \mapsto nb \oplus ma, \leftrightarrow, (id, eval) \\ \vdots \end{array} \right)$$

Abbildung 4.5: Ein Bearbeitungszentrum mit Werkzeugmagazin.

4.2 Komposition des FFS

Die im letzten Abschnitt eingeführten Bearbeitungszentren bilden nur eine Komponente flexibler Fertigungssysteme. Die übrigen im Rahmen dieses Beispiels relevanten Komponenten Werkstücktransportsystem, Werkstückablageplätze und die Ein/Ausgabestation werden, mit Ausnahme der rein passiven Ablageplätze, die wir als Teil des Transports behandeln, in diesem Abschnitt separat spezifiziert und anschließend zu dem Gesamtsystem FFS verknüpft.

Mit Hilfe des Werkstücktransportsystems entsteht aus den Einzelkomponenten eines FFS eine logistische Einheit. Die Steuerung dieses Systems macht einen wesentlichen Teil der Systemsteuerung aus und ist eng mit der Disposition des Werkstückflusses verbunden. Zu diesem Zweck muß sie den Belegungszustand der Systemkomponenten erfassen und laufend aktualisieren. Die algebraische Spezifikation *Systemzustand* liefert die Datengrundlage dafür.

Abgesehen von der Konstanten „ \perp “, die einen Dummy-Wert erzeugt, setzen sich die Operationsnamen der Spezifikation *Systemzustand* aus einem die Komponente bezeichnenden Präfix und einen den Zustand bezeichnenden Suffix zusammen. Dabei steht „M“ (Machine) für ein Bearbeitungszentrum, „B“ (Buffer) für einen Ablageplatz, „S“ (Station) für einen Platz der E/A-Station, „fr“ (free) für frei und „oc“ (occupied) für belegt. Der Operationsparameter bezeichnet die Nummer der je-

weiligen Komponente.

$$\begin{aligned}
 & \textit{Systemzustand} = \\
 & \quad \underline{\textit{sorts}} \quad \textit{SYSZ}; \\
 & \quad \underline{\textit{opns}} \quad \perp : \quad \rightarrow \textit{SYSZ} \\
 & \quad \quad \textit{Mfr} : \textit{NAT} \rightarrow \textit{SYSZ} \\
 & \quad \quad \textit{Moc} : \textit{NAT} \rightarrow \textit{SYSZ} \\
 & \quad \quad \textit{Bfr} : \textit{NAT} \rightarrow \textit{SYSZ} \\
 & \quad \quad \textit{Boc} : \textit{NAT} \rightarrow \textit{SYSZ} \\
 & \quad \quad \textit{Sfr} : \textit{NAT} \rightarrow \textit{SYSZ} \\
 & \quad \quad \textit{Soc} : \textit{NAT} \rightarrow \textit{SYSZ}.
 \end{aligned}$$

Um die werkzeugbedingten Wartezeiten gering zu halten, besteht bei einigen Bearbeitungszentren die Möglichkeit, das Bearbeitungsprogramm dem Werkstück voraus zu schicken. Damit haben Bearbeitungszentren, die über einen lokalen Werkzeugvorrat verfügen, die Möglichkeit, das für die erste Bearbeitungsoperation notwendige Werkzeug noch während des Werkstücktransports einzuwechseln. Die Spezifikation *Werkstück_T*, eine Erweiterung unserer bisherigen Werkstückspezifikation (\rightarrow S. 85), enthält zu diesem Zweck die Operation „Prg“, die zu jedem Werkstück das Bearbeitungsprogramm in Form einer Operationsliste liefert.

$$\begin{aligned}
 & \textit{Werkstück}_T = \textit{Werkstück}_B \cup \textit{Liste}(\textit{Operation}) \cup \\
 & \quad \underline{\textit{opns}} \quad \textit{Prg} : \textit{WKST} \rightarrow \textit{LST}; \\
 & \quad \underline{\textit{eqns}} \quad w : \textit{WKST}; o : \textit{OPN}; \\
 & \quad \quad \textit{Prg}(w + o) = \textit{Prg}(w) + o. \quad ^{15}
 \end{aligned}$$

Fahrtauftrag und Zustand (beladen oder unbeladen) sind die, im Zusammenhang mit den Transporteinheiten, relevanten Informationen. Die Form des Transports, d.h. schienengebunden, fahrerlos oder als Portal, spielt dagegen auf der hier gewählten Beschreibungsebene keine Rolle. Dementsprechend werden die Transporteinheiten in der Spezifikation „*Transportsystem*“ durch eine Konstante („*Tp*“) generiert. Darüber hinaus existiert auf den Transporteinheiten nur die Operation „Load“, die den Übergang von unbeladenen zu beladenen Einheiten ermöglicht. Dabei wird mit Hilfe der

¹⁵das „+“ auf der rechten Seite der Gleichung bezeichnet die Listenoperation die Elemente an eine bestehende Liste abhängt

Operationen „ $\rightarrow M$ “ (Ziel ist ein Bearbeitungszentrum), „ $\rightarrow B$ “ (Ziel ist ein Ablageplatz) und „ $\rightarrow S$ “ (Ziel ist eine Ein/Ausgabestation) das Ziel der Lastfahrt festgelegt.

$$\begin{aligned} \text{Transportsystem} &= \text{Werkstück}_T \cup \\ \text{sorts} & \text{ DST, TPS;} \\ \text{opns} & \text{ Tp} : \quad \quad \quad \rightarrow \text{TPS} \\ & \quad \rightarrow M : \text{NAT} \quad \quad \rightarrow \text{DST} \\ & \quad \rightarrow B : \text{NAT} \quad \quad \rightarrow \text{DST} \\ & \quad \rightarrow S : \text{NAT} \quad \quad \rightarrow \text{DST} \\ & \text{Load: TPS, WKST, DST} \rightarrow \text{TPS.} \end{aligned}$$

Damit die Transportsteuerung ihre Aufgabe wahrnehmen kann, muß für jede Komponente die Anzahl der im FFS vorhandenen Einheiten bekannt sein. Die Spezifikationen mengenmäßig unterschiedlich ausgelegter Systeme unterscheiden sich darüber hinaus jedoch nicht, so daß wir diese Werte als numerische Konstanten festlegen können. In diesem Zusammenhang steht

- „ q “ für die Anzahl der Bearbeitungszentren,
- „ r “ für die Anzahl der Plätze der Ein/Ausgabestation und
- „ s “ für die Anzahl der Werkstückablageplätze.

Jeder Transport läßt sich in die Abschnitte Leerfahrt und Lastfahrt untergliedern. Dabei fallen die Dispositionsaufgaben fast ausschließlich zu Beginn der Leerfahrt an. Bereits zu diesem Zeitpunkt werden sowohl Quelle als auch Ziel der anschließenden Lastfahrt festgelegt, wobei das Ziel zusätzlich reserviert wird. Mit der Übernahme des Werkstücks von der Quelle endet die Leerfahrt und die Lastfahrt beginnt.

Die Gliederung in Leer- und Lastfahrt findet sich in der Transport-Spezifikation (\rightarrow Abb. 4.6) wieder. Darin steuern die Transitionen („ em “, „ ep “, „ es “) der linken Hälfte die Leerfahrten, untergliedert nach deren Ziel (Bearbeitungszentrum, Ablageplatz oder E/A-Station), während die rechts stehenden Transitionen („ am “, „ ap “, „ as “) für die Lastfahrten, ebenfalls nach ihrem Ziel untergliedert, zuständig sind.

Die Festlegung von Quelle und Ziel eines Transports erfolgt in den Gleichungen der für die Leerfahrten zuständigen Transitionen. Dabei liegen die folgenden Regeln zugrunde:

- ein Fahrtauftrag „ $\rightarrow M_m$ “ von einer E/A-Station zum Bearbeitungszentrum m wird generiert ($\rightarrow es$), wenn das Werkstück p auf der E/A-Station noch bearbeitet werden muß ($Prg(p) \neq \emptyset$) und das Bearbeitungszentrum in der Steuerung „ ss “ als frei ($s_0 = Mfr(m)$) eingetragen ist,
- ein Fahrtauftrag „ $\rightarrow B_m$ “ von einer E/A-Station zum Ablageplatz m wird generiert ($\rightarrow es$), wenn das Werkstück auf der E/A-Station noch bearbeitet werden muß und alle Bearbeitungszentren als belegt eingetragen sind ($\bigwedge_{i=1}^q s_i = Moc(i)$),

- ein Fahrtauftrag „ $\rightarrow M_m$ “ von einem Ablageplatz zum Bearbeitungszentrum m wird generiert ($\rightarrow ep$), wenn das Werkstück auf dem Ablageplatz noch bearbeitet werden muß und das Bearbeitungszentrum als frei eingetragen ist,
- ein Fahrtauftrag „ $\rightarrow S_m$ “ von einem Ablageplatz zur E/A-Station m wird generiert ($\rightarrow ep$), wenn die Bearbeitung des Werkstücks abgeschlossen ist ($Prg(p) = \emptyset$) und die E/A-Station als frei eingetragen ist ($s_0 = Sfr(m)$),
- ein Fahrtauftrag „ $\rightarrow S_m$ “ von einem Bearbeitungszentrum zur E/A-Station m wird generiert ($\rightarrow em$), wenn die Bearbeitung des Werkstücks abgeschlossen ist und die E/A-Station als frei eingetragen ist und
- ein Fahrtauftrag „ $\rightarrow B_m$ “ von einem Bearbeitungszentrum zum Ablageplatz m wird generiert ($\rightarrow em$), wenn die Bearbeitung des Werkstücks abgeschlossen ist und alle E/A-Stationen als belegt eingetragen sind ($\bigwedge_{i=1}^r s_i = Soc(i)$).

Zusammen mit der Generierung des Fahrtauftrages wird die Leerfahrt ausgeführt. Sie überführt eine bis dahin passive und unbeladene („ pf “) in eine aktive beladene Transporteinheit („ af “). Handelt es sich bei dem Ziel der Leerfahrt um ein Bearbeitungszentrum, wird parallel zum Transport das Bearbeitungsprogramm an die entsprechende Maschine übertragen. Im Zuge der sich an eine Leerfahrt anschließenden Lastfahrt, wird die aktive Transporteinheit an ihrem Ziel entladen und in eine passive Einheit umgewandelt.

Ein/Ausgabestationen bilden die Schnittstelle eines FFS zu seinem Fertigungsumfeld und sind gleichzeitig die einzige Komponente, die im normalen Betrieb ein menschliches Eingreifen erfordert. Hier werden die zu bearbeitenden Teile auf Werkstückträger gespannt und mit den notwendigen Bearbeitungsprogrammen zu einer systemweit handhabbaren logistischen Einheit verbunden. Außerdem werden an dieser Stelle die fertig bearbeiteten Teile von den Werkstückträgern getrennt und verlassen das System.

In Bezug auf die E/A-Station ist als Systemzustand nur die Belegung der E/A-Plätze von Bedeutung, daher genügt hier die folgende Spezifikation

$$\begin{aligned}
 \text{Systemzustand } S &= \\
 \text{sorts } & \text{SYSZ}; \\
 \text{opns } & Sfr : \text{NAT} \rightarrow \text{SYSZ} \\
 & Soc : \text{NAT} \rightarrow \text{SYSZ}.
 \end{aligned}$$

Werkstückträger erlauben es, Teile unterschiedlicher Geometrie zu transportieren und zu bearbeiten, ohne Transportsysteme und Bearbeitungszentren auf das jeweilige Teil

Transport =

Systemzustand \cup *Transportsystem* \cup

$\text{Paar}_f(\text{Nat} \cup \text{Werkstück}_T) \cup \text{Paar}_k(\text{Nat} \cup \text{Liste}(\text{Operation}))$

$\text{Cond}_T : em \mapsto \{ \text{Prg}(p) = \emptyset,$

$$(s_0 = \text{Sfr}(m) \wedge s_{r+1} = \text{Soc}(m) \wedge d \Rightarrow S_m \wedge \bigwedge_{i=1}^r s_i = \perp) \vee$$

$$(s_0 = \text{Bfr}(m) \wedge s_{r+1} = \text{Boc}(m) \wedge d \Rightarrow B_m \wedge$$

$$\bigwedge_{i=1}^r s_i = \text{Soc}(i) = \text{True} \}$$

$ep \mapsto \{ (\text{Prg}(p) = \emptyset \wedge s_1 = \text{Sfr}(x) \wedge s_2 = \text{Soc}(x) \wedge d \Rightarrow S_x \wedge m = 0) \vee$

$$(\text{Prg}(p) \neq \emptyset \wedge s_1 = \text{Mfr}(m) \wedge s_2 = \text{Moc}(m) \wedge d \Rightarrow M_m)$$

$$= \text{True} \}$$

$es \mapsto \{ \text{Prg}(p) \neq \emptyset,$

$$(s_0 = \text{Mfr}(m) \wedge s_{q+1} = \text{Moc}(m) \wedge d \Rightarrow M_m \wedge \bigwedge_{i=1}^q s_i = \perp) \vee$$

$$(s_0 = \text{Bfr}(x) \wedge s_{q+1} = \text{Boc}(x) \wedge d \Rightarrow B_x \wedge m = 0 \wedge$$

$$\bigwedge_{i=1}^q s_i = \text{Moc}(i) = \text{True} \}$$

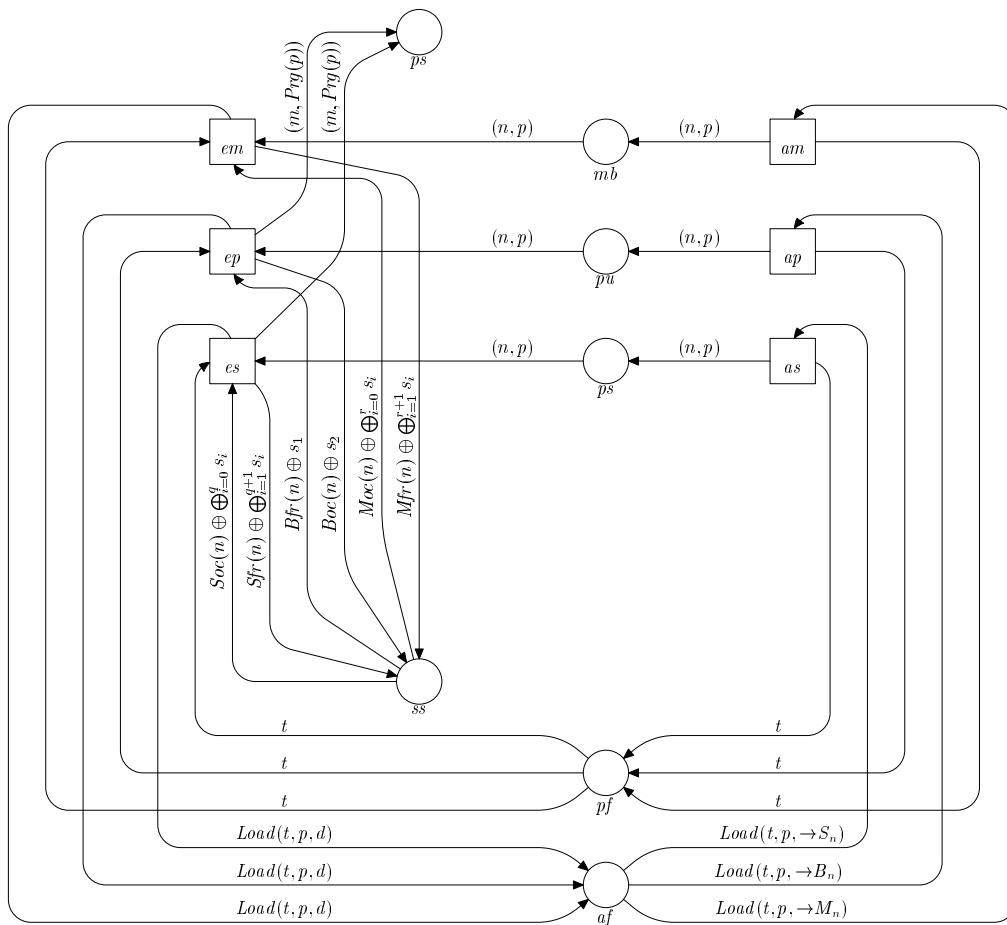


Abbildung 4.6: Transportspezifikation

umrüsten zu müssen.

$$\begin{aligned} & \textit{Werkstückträger} = \\ & \quad \underline{\text{sorts}} \quad \text{WKTR}; \\ & \quad \underline{\text{opns}} \quad \blacksquare : \rightarrow \text{WKTR}. \end{aligned}$$

An den zu bearbeitenden Teilen interessieren uns im Rahmen dieses Beispiels nur die auszuführenden Werkzeugoperationen. Sie werden geordnet nach ihrer Ausführungsreihenfolge in Listen zusammengestellt. Die Zuordnung von Bearbeitungsprogrammen zu diesen Teilen erfolgt in der nachfolgenden Datentypspezifikation durch die Operation „*Prg*“, die aufgrund ihrer Parameter- und Ergebnissorte eindeutig von der gleichnamigen *Werkstück_T*-Operation unterscheidbar ist.

$$\begin{aligned} \textit{Teil} &= \textit{Liste}(\textit{Operation}) \cup \\ & \quad \underline{\text{sorts}} \quad \text{TL}; \\ & \quad \underline{\text{opns}} \quad \textit{Prg}: \text{TL} \rightarrow \text{LST}. \end{aligned}$$

Für die E/A-Station wird der Datentyp *Werkstück* noch einmal erweitert. Hinzu kommen die Spezifikationen der Werkstückträger und Teile sowie die Operation „*—*“, die dem zusammenfügen dieser Komponenten mit den Bearbeitungsprogrammen dient.

$$\begin{aligned} \textit{Werkstück}_S &= \textit{Werkstück}_T \cup \textit{Teil} \cup \textit{Werkstückträger} \cup \\ & \quad \underline{\text{opns}} \quad \text{—}: \text{TL}, \text{WKTR}, \text{LST} \rightarrow \text{WKST}; \\ & \quad \underline{\text{eqns}} \quad t : \text{TL}; l : \text{LST}; \\ & \quad \textit{Prg}(\blacksquare t) = l. \end{aligned}$$

Die AN-Spezifikation der Ein/Ausgabestation in Abbildung 4.7 kommt mit je einer Transition für das Aufspannen („*us*“) und Abspannen („*bs*“) der Teile aus. Dabei wird die Information über den geänderten Belegungszustand der Ein/Ausgabeplätze an die Systemsteuerung (*ss*) weitergegeben.

Um die bisher separat beschriebenen FFS-Komponenten, Bearbeitungszentren, Werkstücktransport und Ein/Ausgabestation, zu einem System zu verknüpfen, müssen zunächst deren Schnittstellen festgelegt werden. Derartige Berührungspunkte existieren einmal in Form der Maschinenbetten zwischen Bearbeitungszentren und Transport, zum anderen in Form der E/A-Plätze und des Systemzustands zwischen dem Transport und den E/A-Stationen. Da diese Schnittstellen voneinander verschieden sind, muß die Komposition in zwei Schritten erfolgen.

Die Spezifikationen der Bearbeitungszentren und des Transports verfügen beide über eine die Maschinenbetten repräsentierende Stelle *mb* und über Paare aus Maschinennummer und Werkstück als Objekttypen, über die auf diese Stelle zugegriffen wird.

Ein/Ausgabestation =

$$Paar_f(Nat \cup Werkstück_S) \cup Systemzustand_S$$

$$Cond_S : us \mapsto \emptyset$$

$$bs \mapsto \emptyset$$

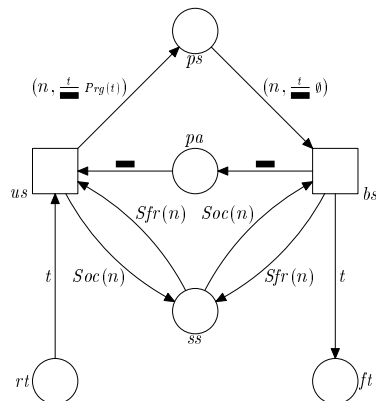


Abbildung 4.7: Ein/Ausgabestation

Diese Gemeinsamkeiten sind in der Schnittstellenspezifikation

$$Mb = (Paar_f(Nat \cup Werkstück_B), \emptyset, (\emptyset, \{mb\}, T_{OP}, \emptyset, \emptyset))$$

formalisiert. Dabei wurde die Werkstückspezifikation der Bearbeitungszentren verwendet, weil sie in der Werkstückspezifikation des Transports enthalten ist und damit den kleinsten gemeinsamen Nenner beider Systeme bildet.

Mit Hilfe des Kompositionsdiagramms

$$BAZ \leftrightarrow Mb \leftrightarrow Transport$$

können die Spezifikation der Bearbeitungszentren und die des Werkstücktransports miteinander verschmolzen werden. Die resultierende Spezifikation *ANS* bildet ein Zwischenergebnis auf dem Weg zur FFS-Spezifikation in Abbildung 4.8, die wir durch Komposition von *ANS* mit der Spezifikation der Ein/Ausgabestation (\rightarrow Abb. 4.7) erzeugen.

Wie bei der vorangegangenen Komposition besteht die Schnittstelle

$$EAp = (Paar_f(Nat \cup Werkstück_B) \cup Systemzustand_S, \emptyset, (\emptyset, \{ps, ss\}, T_{OP}, \emptyset, \emptyset))$$

bei der Verknüpfung der Spezifikationen *ANS* und *Ein/Ausgabestation* nur aus Stellen. Das liegt vor allem daran, daß bei dem hier gewählten Abstraktionsniveau die Werkstückplätze als rein passive Komponenten betrachtet werden.¹⁶ Der Datentypanteil entspricht dem Schnitt der Datentypkomponenten von *ANS* und *Ein/Ausgabestation*.

¹⁶Der nächste Abschnitt enthält unter anderem eine Komposition deren Schnittstelle von einer aktiven Komponente gebildet wird und dementsprechend ein „normales“ CP/T-Netz beinhaltet.

FFS =

Systemzustand \cup *Transportsystem* \cup
Paar_f(*Nat* \cup *Werkstück*) \cup *Paar_g*(*Nat* \cup *Werkzeug*) \cup
Paar_k(*Nat* \cup *Liste*(*Operation*)) \cup *Paar*(*Nat*)
Cond_B \cup *Cond_T* \cup *Cond_S*

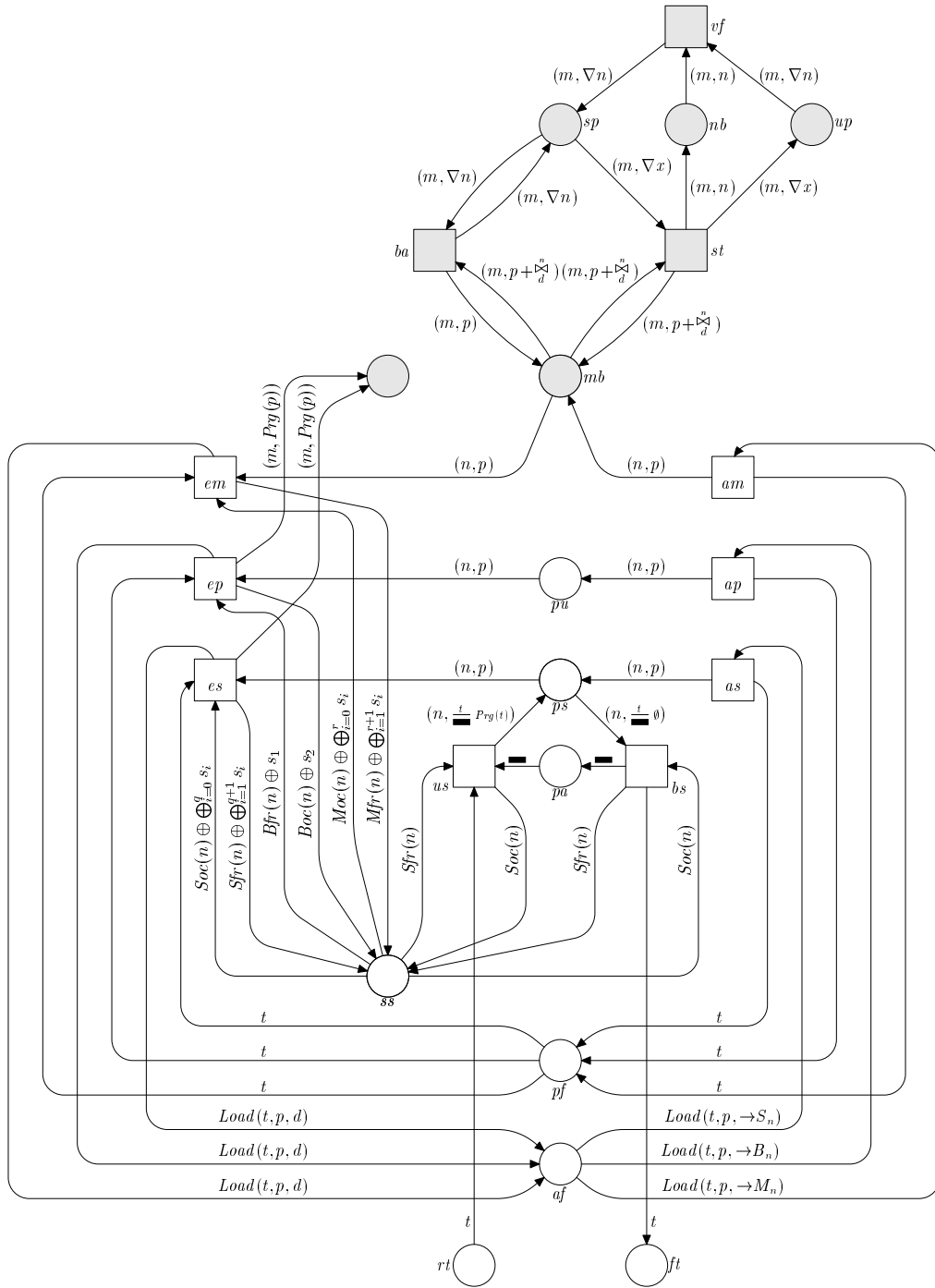


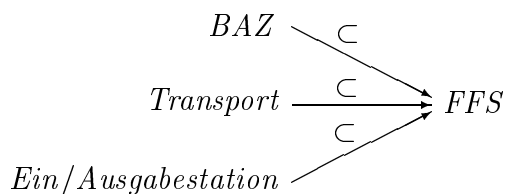
Abbildung 4.8: AN-Spezifikation eines FFS

Die Komposition

$$ANS \leftarrow EAp \hookrightarrow Ein/Ausgabestation$$

führt in ihrem Ergebnis¹⁷ zur Spezifikation in Abbildung 4.8 und beschreibt damit die grundlegenden Funktionen eines kompletten FFS.

Die durch die Komposition bis auf Isomorphie, d.h. in diesem Fall bis auf Umbenennung, eindeutig festgelegte Spezifikation *FFS*, vereinigt in sich die Netzstrukturen der drei von uns unterschiedenen FFS-Komponenten und deren Datentypspezifikationen. Da die Schnittstellen der zugrundeliegenden Kompositionen in beiden Fällen durch Inklusionen mit den Komponenten verbunden sind, ändern sich weder deren Netzstrukturen noch ihre Datentypen im Ergebnis der Komposition; sie werden lediglich ergänzt. So entspricht beispielsweise die *Werkstück*-Spezifikation in *FFS* der gleichnamigen Spezifikation in *Ein/Ausgabestation*, während die Datentypspezifikation *Systemzustand* mit der gleichnamigen Spezifikation des Transports übereinstimmt. Dieser Zusammenhang der Komponentenspezifikationen *BAZ*, *Transport* und *Ein/Ausgabestation* mit der komponierten Spezifikation *FFS* findet in der Existenz entsprechender Inklusionsmorphisimen seinen Niederschlag.



Im folgenden wollen wir eine, die *FFS*-Spezifikation als Gesamtes betreffende Eigenschaft, die mögliche Verklemmung des Systems anhand eines dazu führenden Ablaufs des initialen Modells untersuchen. Die Verklemmung eines *FFS* in der von uns spezifizierten Form ist immer dann möglich, wenn die Anzahl der darin vorgesehenen Werkstückträger gleich der Summe der Bearbeitungszentren, Ablage- und E/A-Plätze ist oder diese übersteigt.

Um den Ablauf überschaubar zu halten, gehen wir von einem kleinen System mit zwei Bearbeitungszentren, einem Ablageplatz, drei E/A-Plätzen und zwei Transporteinheiten aus und führen für eine Reihe von Systemschritten bzw. Schrittsequenzen

¹⁷genaugenommen handelt es sich bei dem Ergebnis um ein Diagramm (\rightarrow 3.3.3) auf dessen Objekt wir uns hier beziehen

abkürzende Notationen ein:

$$Tr_b(s, z, t') \stackrel{\text{def}}{=} \left[\left(\begin{array}{l} \text{ass} : n \mapsto s \\ m \mapsto z \\ x \mapsto 0 \\ p \mapsto \frac{t'}{\blacksquare} \text{Prg}(t') \\ t \mapsto Tp \\ \text{es}, \quad s_0 \mapsto Mfr(z) \\ s_1 \mapsto \perp \\ \vdots \\ s_q \mapsto \perp \\ s_{q+1} \mapsto Moc(z) \\ d \mapsto \rightarrow M_z \end{array} \right) \right] ; \left[\begin{array}{l} [(pg, (z, \text{Prg}(t')))] \\ \left[\begin{array}{l} (ss, Sfr(s)) \\ q * (ss, \perp) \\ (ss, Moc(z)) \end{array} \right] \\ \left[\left(\begin{array}{l} am, \text{ass} : n \mapsto z \\ p \mapsto \frac{t'}{\blacksquare} \text{Prg}(t') \\ t \mapsto Tp \end{array} \right) \right] \end{array} \right]$$

$$Tr_p(s, z, t') \stackrel{\text{def}}{=} \left[\left(\begin{array}{l} \text{ass} : n \mapsto s \\ m \mapsto 0 \\ x \mapsto z \\ p \mapsto \frac{t'}{\blacksquare} \text{Prg}(t') \\ t \mapsto Tp \\ \text{es}, \quad s_0 \mapsto Bfr(z) \\ s_1 \mapsto Moc(1) \\ \vdots \\ s_q \mapsto Moc(q) \\ s_{q+1} \mapsto Boc(z) \\ d \mapsto \rightarrow B_z \end{array} \right) \right] ; \left[\begin{array}{l} [(pg, (0, \text{Prg}(t')))] \\ \left[\begin{array}{l} (ss, Sfr(s)) \\ \bigoplus_{i=1}^q (ss, Moc(i)) \\ (ss, Boc(z)) \end{array} \right] \\ \left[\left(\begin{array}{l} ap, \text{ass} : n \mapsto z \\ p \mapsto \frac{t'}{\blacksquare} \text{Prg}(t') \\ t \mapsto Tp \end{array} \right) \right] \end{array} \right]$$

$$Sp(s, t') \stackrel{\text{def}}{=} \left[\left(\begin{array}{l} us, \text{ass} : n \mapsto s \\ t \mapsto t' \end{array} \right) \right]$$

Die gezeigte Verklemmung ist aber nicht auslegungsspezifisch, sie läßt sich unter vergleichbaren Umständen in Systemen beliebiger Größe erzeugen.

Für unseren Beispielablauf nehmen wir an, daß sich das FFS im Zustand m mit den folgenden Eigenschaften befindet,

$$m(pa, \blacksquare) \geq 6, \quad m(pf, Tp) = 2, \quad \sum_{(rt,t) \in P \times T_{Spec, Tl}} m(rt, t) \geq 6, \quad ^{18}$$

d.h. es existieren soviel Werkstückträger wie Bearbeitungszentren, Ablage- und E/A-Plätze zusammen und es sind noch mindestens sechs Teile zu bearbeiten.

¹⁸ $m(p, t)$ steht für die Anzahl der Token t auf der Stelle p im Zustand m (\rightarrow 1.1)

¹⁹die m_i enthalten alle von dem jeweiligen Schritt nicht berührten Token

$$m; \begin{array}{c} [Sp(1, t_1)] \\ [Sp(2, t_2)] \\ [Sp(3, t_3)] \end{array}; \begin{array}{c} [Tr_b(1, 1, t_1)] \\ [Tr_b(2, 2, t_2)] \end{array}; \begin{array}{c} [Tr_p(3, 1, t_3)] \\ m_3 \end{array}; \begin{array}{c} [Sp(1, t_4)] \\ [Sp(2, t_5)] \\ [Sp(3, t_6)] \end{array}; m' \quad 19$$

$m_1 \qquad m_2 \qquad m_3 \qquad m_4$

Abbildung 4.9: Ein Ablauf der zu einer Verklemmung des Systems führt.

Bei dem in Abbildung 4.9 gezeigten Ablauf werden zunächst nebenläufig drei Teile auf Werkstückträger gespannt und dann in zwei Etappen transportiert (es stehen nur zwei Transporteinheiten zur Verfügung). Die ersten beiden Werkstücke gelangen direkt zu den Bearbeitungszentren, ein Werkstück muß auf dem Ablageplatz zwischengelagert werden. Nach dem neuerlichen Aufspannen von drei Teilen sind alle im System vorhandenen Plätze belegt.

$$\sum_{i=1}^3 m'(ss, Sfr(i)) = 0, \quad m'(ss, Bfr(1)) = 0 \quad \text{und} \quad \sum_{i=1}^2 m'(ss, Mfr(i)) = 0$$

Unter diesen Umständen können die auf den Maschinen befindlichen Werkstücke noch bearbeitet werden, anschließend ist keine Aktion mehr möglich. Transporte können nicht mehr stattfinden, da alle Ziele belegt sind, die Maschinen können nicht mehr arbeiten, da neue Teile nicht mehr nachkommen und ein Auf- bzw. Abspannen von Teilen ist nicht mehr möglich, da der dafür benötigte Platz fehlt.²⁰

Sieht man davon ab, daß in unserem System nur zwei Transporte gleichzeitig durchgeführt werden können, sind die Transportschritte voneinander unabhängig. Dasselbe gilt für Transporte und Spannvorgänge, wenn sie nicht den selben Spannplatz betreffen. Da unabhängige Aktionen nebenläufig und nebenläufige Aktionen auch nacheinander ausgeführt werden können, läßt sich der in Abbildung 4.9 gezeigte Ablauf, mit Hilfe der in der Schrittsequenzsemantik (\rightarrow 1.3.1) festgelegten Regeln, in den äquivalenten Ablauf in Abbildung 4.10 umformen.

Jedes Modell einer Spezifikation, die sich als Ergebnis einer Komposition darstellen läßt, kann eindeutig in Modelle der Komponenten zerlegt werden. Dabei spielt es keine Rolle, ob die Spezifikation wirklich aus dieser oder überhaupt einer Komposition hervorgegangen ist. Um diesen Zusammenhang zu verdeutlichen, betrachten wir die FFS-Spezifikation in Abbildung 4.8 als Ergebnis der Komposition

$$BAZ^+ \leftarrow \left(Spec_{BAZ^+}, \emptyset, (\{pg, mb\}, \emptyset, T_{OP(X)}, \emptyset, \emptyset) \right) \hookrightarrow ANS_{Tp+EA}$$

²⁰im realen Betrieb könnten die un bearbeiteten Teile wieder abgespannt werden, das ist in unserer Spezifikation jedoch nicht vorgesehen

$$\begin{array}{ccccccc}
[Sp(2, t_2)] & [Sp(1, t_1)] & [Tr_b(1, 1, t_1)] & [Sp(1, t_4)] \\
m; [Sp(3, t_3)]; [Tr_b(2, 2, t_2)]; & [Tr_p(3, 1, t_3)]; & [Sp(2, t_5)]; & [Sp(3, t_6)]; m' \\
m'_1 & m'_2 & m'_3 & m'_4
\end{array}$$

Abbildung 4.10: Ein zum Ablauf in 4.9 äquivalenter Ablauf.

in der BAZ^+ der bisherigen BAZ -Spezifikation (\rightarrow Abb. 4.1), erweitert um die Stelle pg , entspricht,²¹ während ANS_{Tp+EA} das Ergebnis der Komposition

$$Transport \leftrightarrow EAp \hookrightarrow Ein/Ausgabestation$$

darstellt. Bezüglich dieser Komposition, deren Komponenten im Rahmen der Parametrisierung der FFS -Spezifikation im nächsten Abschnitt eine besondere Bedeutung erlangen, soll das FFS -Modell in Abbildung 4.11, auf dessen spezielle Bedeutung wir ebenfalls in Abschnitt 4.3 zurückkommen, dekomponiert werden.

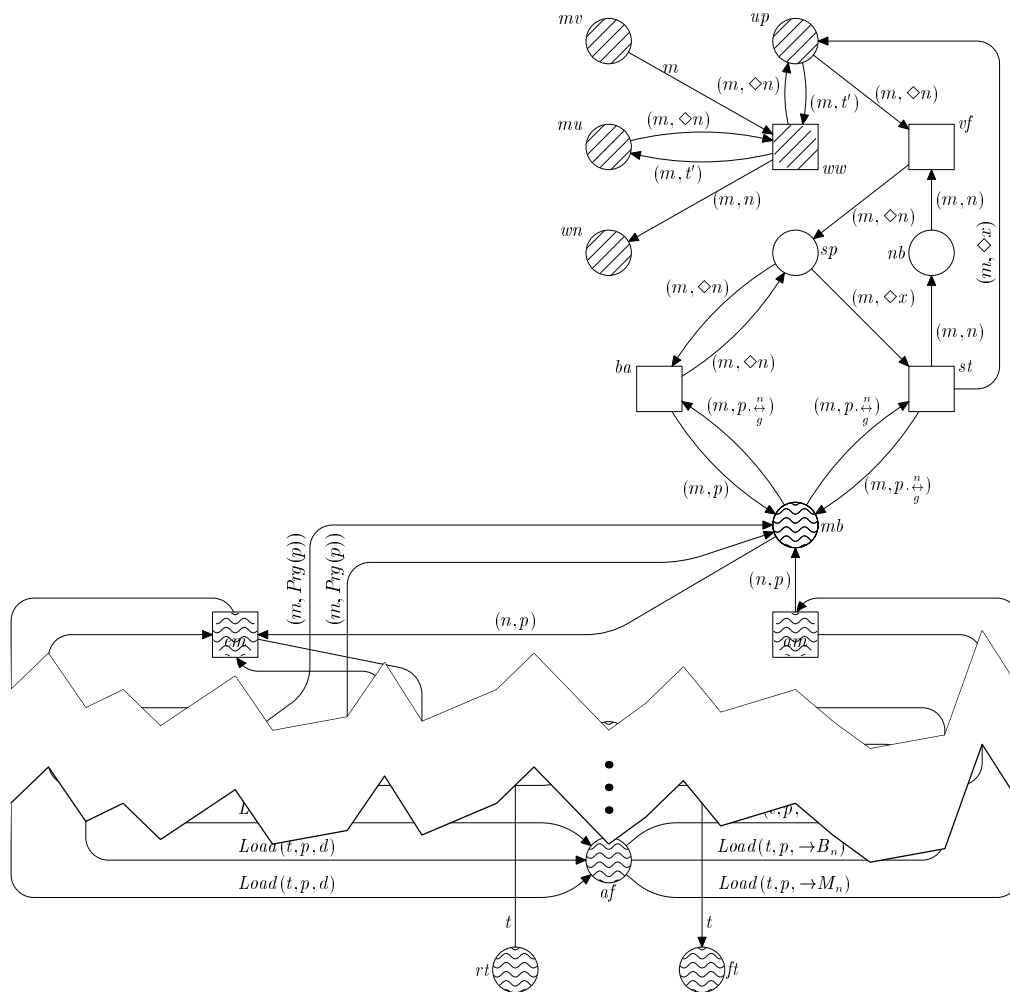
Die Dekomposition des Modells einer komponierten Spezifikation (\rightarrow 3.3.7) erfolgt mit Hilfe von Vergißfunktoren. Diese Vergißfunktoren werden von den Spezifikationsmorphismen generiert (\rightarrow 3.1.7), die die Komponenten mit dem Kompositionsergebnis verbinden. Im Fall der FFS -Komponente ANS_{Tp+EA} aus der vorangegangenen Komposition, ergibt sich das Ergebnis der Dekomposition des FFS -Modells FFS_{BAZ-WW} (\rightarrow Abb. 4.11) als Bild dieses Modells unter dem von der Inklusion generierten Vergißfunktoren $\mathcal{V}_{\hookrightarrow}$.

Um die Netzkomponente von $\mathcal{V}_{\hookrightarrow}(FFS_{BAZ-WW})$ zu konstruieren, müssen innerhalb des Modells drei Subnetze, in Abbildung 4.11 durch unterschiedliche Schraffuren hervorgehoben, unterschieden werden. Zum einen der nicht generierte Modellteil (diagonal schraffiert), bestehend aus den Stellen und Transitionen, die kein Urbild in der FFS -Spezifikation haben, ergänzt um die generierten Stellen (hier nur up), die durch Kanten mit nicht generierten Transitionen verbunden sind. Ein weiterer Teil, mit Wellenlinien markiert, besteht aus den von ANS_{Tp+EA} generierten Stellen, Transitionen und verbindenden Kanten. Den unmarkierten Rest bilden die von BAZ^+ generierten Stellen und Kanten. Da wir uns für den ANS_{Tp+EA} -spezifischen Modellteil interessieren, wird dieser letzte Netzanteil vom Vergißfunktoren eliminiert. Das Ergebnis ist als Netzanteil des ANS_{Tp+EA} -Modells in Abbildung 4.12 zu sehen.

Die Algebra wird ohne Reduktion vom ursprünglichen Modell in das ANS_{Tp+EA} -Modell übernommen. An Stelle dessen ändert sich der zum Modell gehörende Datentypspezifikationsmorphismus (\rightarrow 2.2.1) von der Identität im FFS -Modell zur Inklusion im ANS_{Tp+EA} -Modell. In dessen generiertem Teil kommen als Kantenanschriften nur

²¹in Abbildung 4.8 durch die grau hinterlegten Stellen und Transitionen kenntlich gemacht

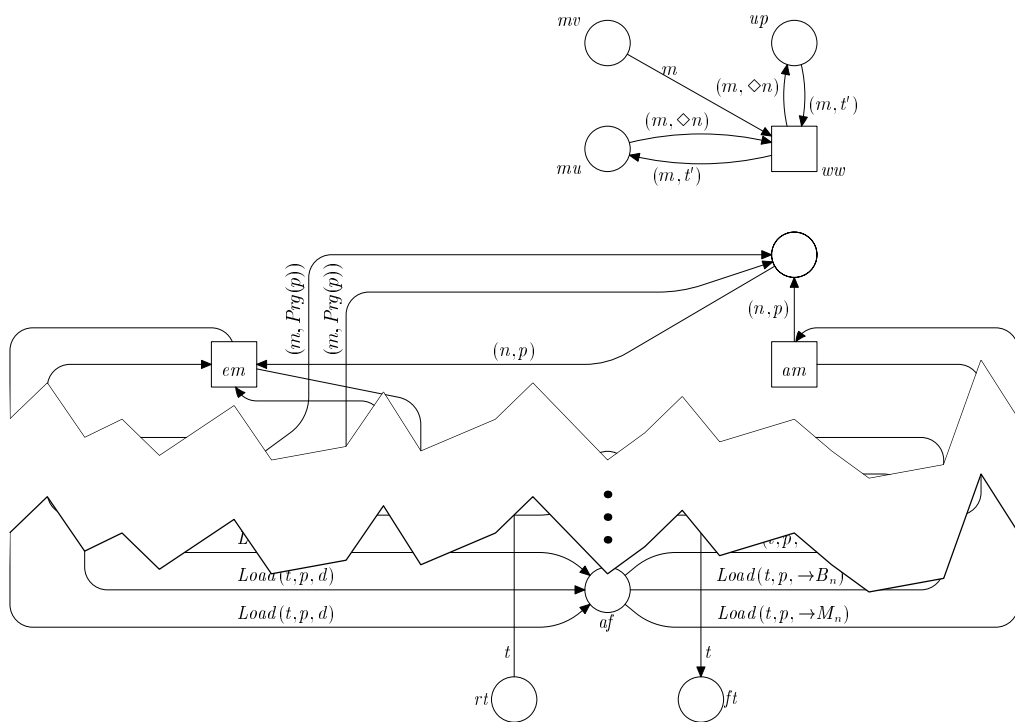
$FFS_{BAZ_WW}: FFS, id,$
 $\mathcal{F}(A_{BAZ_WW})$



$$\left(e^P: \begin{array}{l} \vdots \\ p_{mb} \mapsto p_{mb} \\ p_{pg} \mapsto p_{mb} \end{array}, \Leftrightarrow, (id, eval) \right)$$

Abbildung 4.11: Ein FFS-Modell

$\mathcal{V}_{\hookrightarrow}(FFS_{BAZ_WW}): ANS_{Tp+EA}, \hookrightarrow,$
 $\mathcal{F}(A_{BAZ_WW})$



$$\left(\begin{array}{l} e^P: \quad \vdots, \hookrightarrow, (id, eval) \\ p_{mb} \mapsto p_{mb} \\ p_{pg} \mapsto p_{mb} \end{array} \right)$$

Abbildung 4.12: Das ANS_{Tp+EA} -Modell $\mathcal{V}_{\hookrightarrow}(FFS_{BAZ_WW})$.

Algebraelemente vor, die im Bild des vom Datentypspezifikationsmorphismus generierten Vergißfunktors $\mathcal{V}_{\hookrightarrow}$ ²² liegen. Insofern entsprechen sich bei diesem Modellteil Netz- und Algebrakomponente.

Als nichtgenerierter Modellteil ist der Werkzeugwechsler (in Abb. 4.11 diagonal markiert) modellspezifisch, d.h. die von diesem Teil beschriebenen Ereignisse und Abläufe können nicht einer bestimmten *FFS*-Komponente zugeordnet werden. Das gilt auch dann, wenn der betreffende Modellteil wie im diesem Fall nur mit einer Komponente verbunden ist. Aus diesem Grund ist er in jedem aus der Dekomposition hervorgehenden Modell enthalten.

Das Fehlen einer Verbindung zwischen dem Werkzeugwechsler und dem restlichen Modell in Abbildung 4.12 spiegelt dessen Unabhängigkeit vom Werkstücktransport und der Ein/Ausgabestation wieder. Ein Ereigniszusammenhang besteht im Modell wie in Realität nur mit dem die Bearbeitungszentren repräsentierenden Modellteil, d.h. mit dem auf BAZ^+ bezogenen Dekompositionsergebnis. Dessen Netzteil stimmt mit dem CP/T-Netz des BAZ -Modells in Abbildung 4.3 überein. Der enthaltene Datentypspezifikationsmorphismus und die Algebra sind dagegen auf die gleiche Weise verändert wie im ANS_{Tp+EA} -Modell.

4.3 Parameterisiertes FFS

In unserer bisherigen *FFS*-Spezifikation haben wir uns auf die Beschreibung der grundlegenden Abläufe beschränkt. Das Resultat (\rightarrow Abb. 4.8) bleibt dadurch zum einen überschaubar, zum anderen gelten die dort beschriebenen Eigenschaften für eine breite Palette verschiedener flexibler Fertigungssysteme.

In dem wir die *FFS*-Spezifikation aus Abbildung 4.8 in die parameterisierte Spezifikation $FFS(BAZ^+)$ umwandeln, schaffen wir die Möglichkeit sie im Bereich der Bearbeitungszentren zu spezialisieren und die Ergebnisse dieser Spezialisierungen zu analysieren. Die dafür als Voraussetzung notwendige Inklusionsbeziehung (\rightarrow 3.4.1), zwischen der um die Stelle „*pg*“ erweiterten Spezifikation der Bearbeitungszentren BAZ^+ und *FFS*, gehört wie *FFS* zum Ergebnis der im letzten Abschnitt vorgestellten Komposition (\rightarrow S. 101) von BAZ^+ und ANS_{Tp+EA} .

Die zusätzliche Stelle von BAZ^+ gegenüber der ursprünglichen Spezifikation der Bearbeitungszentren BAZ (\rightarrow Abb. 4.1) dient der Aufnahme von Bearbeitungsprogrammen für die aktuellen bzw. demnächst zu bearbeitenden Werkstücke. Die Bearbeitungszentren haben damit die Möglichkeit zeitlich parallel zum Transport oder zur laufenden Bearbeitung, zukünftig benötigte Werkzeuge bereitzustellen bzw. einzuwechseln. Unsere Parameterspezifikation BAZ^+ macht davon keinen Gebrauch, erzwingt aber eine entsprechende Stelle bei allen als Aktualisierung in Frage kommenden Spezifikationen.

²²nicht zu verwechseln mit dem gleichnamigen Vergißfunktors zwischen AN-Spezifikation-Modellen

Obwohl sich BAZ und BAZ^+ in Bezug auf ihre Netzstruktur unterscheiden, existiert zu jedem der beiden im ersten Abschnitt vorgestellten BAZ -Modelle (\rightarrow Abb. 4.3, 4.5) ein BAZ^+ -Modell mit identischer Netzstruktur und Algebra (\rightarrow Abb. 4.13). Die BAZ^+ -Modelle unterscheiden sich von den BAZ -Modellen lediglich in der notwendigerweise unterschiedlichen Spezifikationskomponente (BAZ^+ anstelle von BAZ) und der Evaluierungsfunktion, die die zusätzliche Stelle (pg) auf das Maschinenbett abbildet. Da Modellmorphisamen auf Abbildungen zwischen den Netzstrukturen und Algebren beruhen, hat auch der die beiden BAZ -Modelle verbindende Modellmorphismus h (\rightarrow S. 90) sein Äquivalent in der Kategorie der BAZ^+ -Modelle.

$$\begin{array}{ccc}
\mathbf{BAZ_WW}: BAZ^+, id, & & \mathbf{BAZ_Mag}: BAZ^+, id, \\
A_{BAZ_WW} & & A_{BAZ_Mag} \\
N_{BAZ_WW} & & N_{BAZ_Mag} \\
\left(\begin{array}{l} e^P : p_{mb} \mapsto p_{mb}, \hookrightarrow, (id, eval) \\ p_{pg} \mapsto p_{mb} \\ \vdots \end{array} \right) & \xrightarrow{h} & \left(\begin{array}{l} e^P : p_{mb} \mapsto p_{mb}, \hookrightarrow, (id, eval) \\ p_{pg} \mapsto p_{mb} \\ \vdots \end{array} \right)
\end{array}$$

Abbildung 4.13: Die Modelle aus 4.3 und 4.5 als BAZ^+ -Modelle.

Die freie Konstruktion als Semantik parameterisierter AN-Spezifikationen (\rightarrow 3.4.4) ermöglicht die Abbildung der Parametermodelle in Modelle des Spezifikationsrumpfes. Betrachtet man in diesem Zusammenhang die Parametermodelle als die initialen Modelle möglicher Aktualisierungen, dann stehen ihre Bilder unter der freien Konstruktion für die (initialen) Modelle der entsprechend aktualisierten Spezifikationen. Auf diese Weise gibt die Semantik die durch Parameterersetzung möglichen Variationen der parameterisierten Spezifikation wieder.

In unserem Beispiel lassen sich damit die BAZ^+ -Modelle aus Abbildung 4.13 in das initiale FFS -Modell einbetten. Die Einbettungen von BAZ_WW haben wir als FFS -Modell FFS_{BAZ_WW} (\rightarrow Abb. 4.12) bereits im letzten Abschnitt eingeführt und bezüglich der Komposition, die den Parameter BAZ^+ mit der übrigen FFS -Spezifikation verbindet, dekomponiert. Das Ergebnis dieser Dekomposition in Bezug auf BAZ^+ macht zum einen den Effekt der freien Konstruktion auf den vom Spezifikationsrumpf erzeugten Modellteil deutlich, die Stellen pg und mb werden im Bild zu einer Stelle verschmolzen, zum anderen werden dadurch die Veränderungen des ursprünglichen Modells (\rightarrow Abb. 4.3) im Zuge der Einbettung erkennbar. Dabei zeigt sich dessen Netzteil unverändert, während der Algebraanteil lediglich ergänzt wurde.²³

²³wobei diese Ergänzungen im Netzteil nicht verwendet werden

Die Inklusionsbeziehung zwischen Parameter und Rumpf einer parameterisierten AN-Spezifikation, impliziert eine entsprechende Beziehung zwischen deren Datentypspezifikationen, d.h. bezogen auf unser Beispiel

$$BAZ^+ \subset FFS \implies Spec_{BAZ^+} \subset Spec_{FFS}$$

und damit die parameterisierte Datentypspezifikation $Spec_{FFS}(Spec_{BAZ^+})$. Da die Semantik parameterisierter Spezifikationen sowohl bei AN- als auch bei Datentypspezifikationen über freie Konstruktionen definiert ist und sich deren Bilder analog konstruieren lassen (\rightarrow 5.3.4), ist die Algebra des frei konstruierten FFS -Modells $\mathcal{F}(BAZ_WW)$ (\rightarrow 4.11) ebenfalls frei konstruiert.²⁴

Im nächsten Schritt wollen wir exemplarisch für alle möglichen Aktualisierungen des FFS -Parameters ein gegenüber BAZ^+ in seiner Funktionalität erweiterten und damit spezialisierteren Typ von Bearbeitungszentren spezifizieren. Bei diesen Bearbeitungszentren kommen die folgenden Funktionen hinzu:

- eine „*vorausschauende Werkzeuganforderung*“, mit deren Hilfe das nächste benötigte Werkzeug schon während des Transports zum Bearbeitungszentrum oder während der laufenden Bearbeitungsoperation angefordert werden kann,
- ein lokaler Werkzeugvorrat in Form eines „*Werkzeugmagazins*“, daß eine bedienerfreie Bearbeitung der Werkstücke auch dann ermöglicht, wenn dazu mehr als ein Werkzeug notwendig ist und
- eine „*Werkzeugwechselvorrichtung*“, die den Austausch eines vom Werkzeugmagazin bereitgestellten Werkzeugs mit dem in der Arbeitsspindel befindlichen Werkzeug bewerkstelligt.

Die Spezifikation der erweiterten Bearbeitungszentren gliedert sich in zwei Teilbereiche; im ersten beschreiben wir die, um eine vorausschauende Werkzeuganforderung und eine Werkzeugwechselvorrichtung ergänzten Maschinen, im zweiten Teil die Werkzeugmagazine mit Werkzeugwechselvorrichtung. Die Werkzeugwechselvorrichtung in Form eines Doppelgreifers, der beide Werkzeuge gleichzeitig erfaßt und in einer Operation miteinander vertauscht, bildet die Schnittstelle beider Systeme. Aufgrund seiner Arbeitsweise muß er eng mit beiden Systemen synchronisiert sein, d.h. er kann den Austausch nur vornehmen, wenn sich auf der einen Seite die Spindel in der Übergabeposition befindet und auf der anderen Seite das Magazin das Werkzeug bereitgestellt hat. Aus diesem Grunde enthalten beide Teilspezifikationen eine Beschreibung dieses Subsystems.

Der Kern der neuen Bearbeitungszentren, die Maschinen mit Werkzeugwechsler und vorausschauender Werkzeuganforderung, sind in Abbildung 4.14 spezifiziert. Innerhalb dieser Spezifikation sind die Transitionen „*wl*“ und „*wa*“ für die Generierung der

²⁴dieser Zusammenhang gilt nur für Modelle, deren Algebra direkt mit der Datentypkomponente der zugehörigen AN-Spezifikation gekoppelt ist, d.h. deren Datentypspezifikationsmorphismus (\rightarrow 2.2.1) die Identität ist

vorausschauenden Werkzeuganforderungen zuständig. Zunächst entfernt die Transition wl solange alle Operationen aus der auf pg vorliegenden Kopie des Bearbeitungsprogramms, bis sie auf eine Operation stößt, die ein Werkzeug benötigt, das sich nicht in der Spindel befindet. Danach wird wa aktiv und erzeugt die Werkzeuganforderung auf „ ma “. Um die Arbeit der Spindel für die Abfrage der Werkzeugnummer nicht unterbrechen zu müssen, erzeugen wir auf der Stelle „ wn “ eine Kopie dieser Nummer und führen die Vergleiche damit durch.

BAZ_V =

$$Paar_f(Nat \cup Werkstück_B) \cup Paar_g(Nat \cup Werkzeug) \cup$$

$$Paar_k(Nat \cup Liste(Operation)) \cup Paar(Nat)$$

$$Cond_{M_s} : ww \mapsto \emptyset$$

$$wl \mapsto \emptyset$$

$$wa \mapsto (x = n) = False$$

$$vf \mapsto \emptyset$$

$$st \mapsto (x = n) = False$$

$$ba \mapsto \emptyset$$

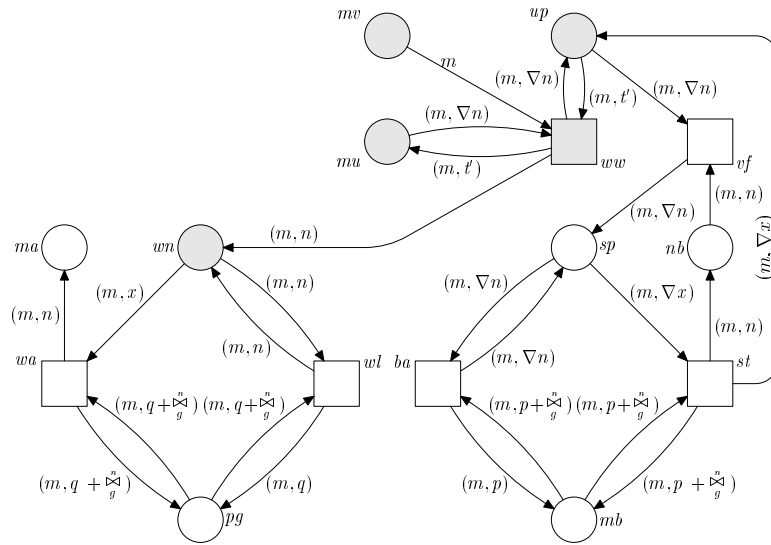


Abbildung 4.14: Spezifikation einer Maschine mit Vorausschau.

Die Werkzeugwechsellvorrichtung besteht aus der Transition „ ww “. Sie vertauscht in einem Arbeitsgang die Werkzeuge auf den Stellen „ up “ (Spindel in Übergabeposition) und „ mu “ (Übergabeplatz des Magazins), legt die Nummer des neuen Werkzeugs auf wn ab und setzt die magazinseitige Freigabe (mv) zurück, in dem sie die dort vermerkte Maschinenummer abzieht. Alle übrigen Bestandteile der Spezifikation in Abb. 4.14 wurden bereits an anderer Stelle (\rightarrow Abb. 4.1) vorgestellt.

Unsere Werkzeugmagazin-Spezifikation in Abb. 4.15 beschränkt sich auf das Notwendigste und qualifiziert sich damit als ein potentieller Kandidat für spätere Erweiterungen/Spezialisierungen. Trifft eine Anforderung in Form einer Werkzeugnummer auf

der Stelle „*ma*“ ein, kann „*mg*“ schalten und damit das entsprechende Werkzeug (*n*) bereitstellen, wenn es sich entweder schon auf der Übergabeposition ($z = n \wedge x = y$) oder im Werkzeugvorrat „*mv*“ befindet ($x = n \wedge y = z$). In letzterem Fall muß das Werkzeug auf der Übergabeposition des Magazins gegen das gesuchte Werkzeug im Vorrat ausgetauscht werden. Die Aktion wird in beiden Fällen durch eine entsprechende Belegung der Variablen *n*, *x*, *y*, *z* gesteuert. Darüber hinaus enthält die Spezifikation in Abbildung 4.15 die Bestandteile des schon besprochenen Werkzeugwechslers, der in Abbildung 4.16 noch einmal als separate Spezifikation zu sehen ist.

Werkzeugmagazin =

$$Paar_g(Nat \cup Werkzeug) \cup Paar(Nat)$$

$$Cond_M : mg \mapsto (z = n \wedge x = y) \vee (x = n \wedge y = z) = True$$

$$ww \mapsto \emptyset$$

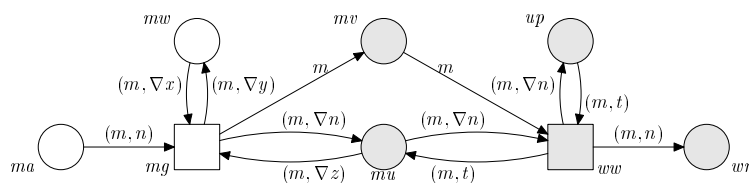


Abbildung 4.15: Spezifikation eines Magazins mit Werkzeugwechsler.

Werkzeugwechsler =

$$Paar_g(Nat \cup Werkzeug) \cup Paar(Nat)$$

$$Cond_W : ww \mapsto \emptyset$$

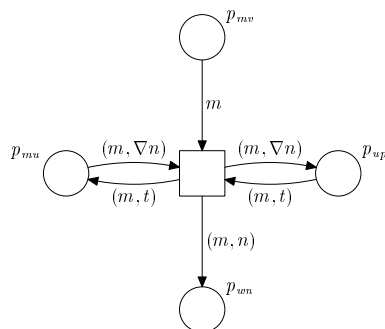


Abbildung 4.16: Spezifikation eines Werkzeugwechslers.

Die gesamte Spezifikation unserer Bearbeitungszentren neuen Typs ist als Ergebnis der Komposition von Maschinen und Werkzeugmagazinen über das Kompositionsdiagramm

$$BAZ_V \leftrightarrow Werkzeugwechsler \leftrightarrow Magazin$$

in Abbildung 4.17 zu sehen.

BAZ_VM =

$\text{Paar}_f(\text{Nat} \cup \text{Werkstück}_B) \cup \text{Paar}_g(\text{Nat} \cup \text{Werkzeug}) \cup$

$\text{Paar}_k(\text{Nat} \cup \text{Liste}(\text{Operation})) \cup \text{Paar}(\text{Nat})$

$\text{Cond}_B \cup \text{Cond}_M$

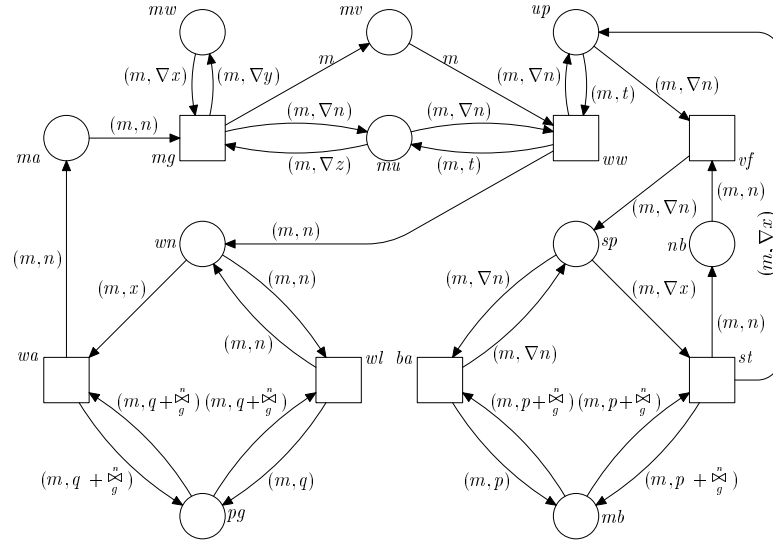


Abbildung 4.17: BAZ mit Vorausschau und lokalem Werkzeugvorrat

Wir haben die neuen, spezialisierteren Bearbeitungszentren als Erweiterung der im ersten Abschnitt spezifizierten Bearbeitungszentren (\rightarrow Abb. 4.1) entworfen. Da sie über eine Stelle verfügen, auf der Bearbeitungsprogramme abgelegt werden können (pg), erweitern sie auch den formalen FFS -Parameter BAZ^+ , dessen Spezifikation sich ansonsten nicht von der Spezifikation der ursprünglichen Bearbeitungszentren unterscheidet. Der sich aus diesem Zusammenhang ergebende Inklusionsmorphismus $BAZ^+ \hookrightarrow BAZ_VM$ qualifiziert BAZ_VM als zulässige Aktualisierung (\rightarrow 3.4.2) der Spezifikation $FFS(BAZ^+)$. Das Ergebnis dieser Aktualisierung, die Spezifikation $FFS(BAZ_VM)$, ist in den Abbildungen 4.18 und 4.19 zu sehen.

FFS(BAZ_VM) =

$\text{Systemzustand} \cup \text{Transportsystem} \cup$

$\text{Paar}_f(\text{Nat} \cup \text{Werkstück}) \cup \text{Paar}_g(\text{Nat} \cup \text{Werkzeug}) \cup \text{Paar}(\text{Nat})$

$\text{Cond}_B \cup \text{Cond}_M \cup \text{Cond}_T \cup \text{Cond}_S$

N_{Akt}

Abbildung 4.18: Aktualisiertes FFS ($N_{Akt} \rightarrow$ Abb. 4.19).

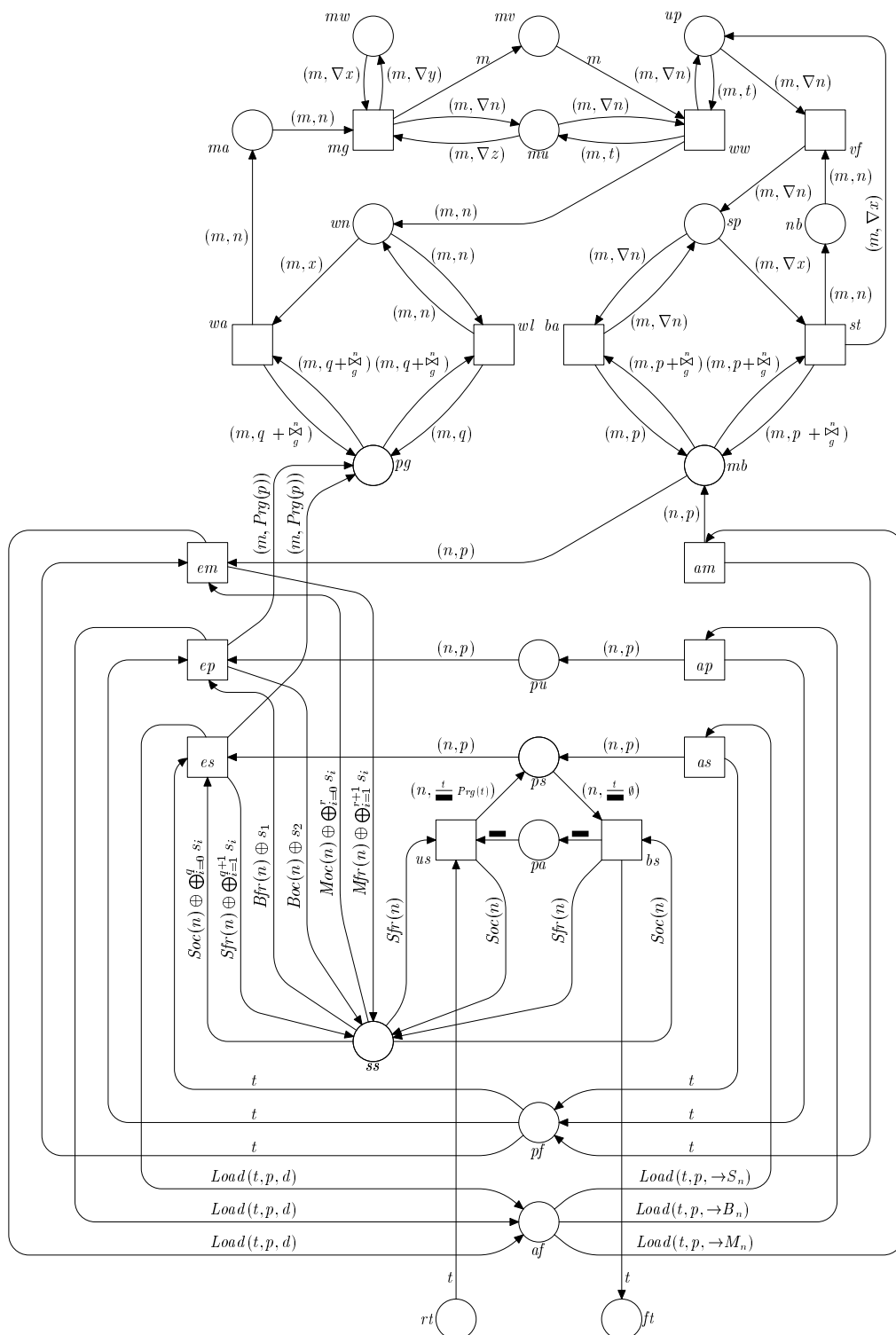


Abbildung 4.19: das CP/T-Netz N_{Act} der Spezifikation $FFS(BAZ_VM)$

Kategorielle Konstruktionen

In diesem Kapitel geben wir zunächst eine kurze Einführung in einige grundlegende Konstruktionen der Kategorientheorie. Dabei betrachten wir die Kategorientheorie als eine Art Typkonzept für mathematische Objekte. Ähnlich der objektorientierten Programmierung stehen dabei nicht die Definitionen der Objekte, sondern deren Eigenschaften, die durch die Beziehung zu anderen Objekten ausgedrückt werden, im Vordergrund. Die Vorteile dieses Ansatzes liegen in seiner Repräsentationsunabhängigkeit, die es erlaubt, Objekte mit vergleichbaren Eigenschaften innerhalb komplexer Strukturen auszutauschen, Ergebnisse zwischen Objekten zu übertragen und Objektkonstruktionen wieder zu verwenden.

Viele kategorielle Konzepte stellen Verallgemeinerungen bekannter mathematischer Konstruktionen dar. Während diese normalerweise nur für spezielle Objekte definiert sind, werden sie im Rahmen der Kategorientheorie getrennt von objektspezifischen Eigenheiten eingeführt. Konstruktionen und Resultate lassen sich so auf beliebige Objekttypen übertragen. Beispiele dafür bilden kartesische Produkte auf Mengen als spezielle Limites, die zu freien Funktoren korrespondierende Beziehung zwischen Mengen und darüber erzeugten Gruppen oder initiale Modelle in der Logik, die zu initialen Objekten innerhalb einer Modellkategorie verallgemeinert werden.

Aufbauend auf den Konstruktionen des ersten Abschnitts, führen wir die Kategorie der freien abelschen Gruppen sowie einige, durch verschiedene Einschränkungen daraus ableitbare, Unterkategorien ein und zeigen deren Eigenschaften soweit sie für diese Arbeit relevant sind. Gleichzeitig bildet dieser Abschnitt eine exemplarische Anwendung der bis dahin eingeführten kategoriellen Konzepte.

Freie abelsche Gruppen bilden eine Basis unserer Petrinetzdefinition und sind damit grundlegend für die Kapitel 1–3. Das trifft in besonderem Maße auf die Existenz und Berechnung von Limites endlicher Diagramme zu, deren duale Entsprechungen die Konstruktion von Colimites auf CP/T-Netzen und AN-Spezifikationen ermöglichen bzw. die Einschränkungen für deren Existenz bestimmen. In diesem Zusammenhang ist der enge Bezug von freien abelschen Gruppen zu Vektorräumen und Matrizen über den ganzen Zahlen hilfreich, der die für lineare Gleichungssysteme vorhandenen Lösungsverfahren nutzbar macht.

Der letzte Abschnitt schafft in Anlehnung an die entsprechenden Definitionen aus [EBO91] einen, von der Objektbeschreibung unabhängigen, kategoriellen Rahmen zur Verknüpfung von Spezifikationen mit ihren Modellen. Dadurch sind die in diesem Zusammenhang vorgestellten Ergebnisse auf eine ganze Reihe verschiedener Spezifikationstechniken und Modellbeschreibungen anwendbar, insbesondere auf die für

die übrigen Kapitel grundlegenden algebraischen Datentypspezifikationen mit loser Semantik (\rightarrow Kapitel 6) und auf AN-Spezifikationen mit CP/T-Netzmodellen (\rightarrow Kapitel 2).

5.1 Grundlagen

Im folgenden Abschnitt geben wir eine kurze Einführung in einige Grundlagen der Kategorientheorie. Die dabei getroffene Auswahl richtet sich nur nach der Relevanz für die vorliegende Arbeit und spiegelt nicht unbedingt die Bedeutung der Konstruktionen im Rahmen der Kategorientheorie wieder. Für eine ausführliche und weiterführende Betrachtung verweisen wir auf [HS73], aus dem die meisten der verwendeten Definitionen, Lemmata und Sätze entlehnt sind, sowie auf [ML70] und [BW90], auf deren Grundlage die übrigen Konstruktionen dieses Abschnitts entstanden sind.

„Kategorien“ beschreiben mathematische Objekttypen wie Mengen, Gruppen oder Vektorräume anhand der Beziehungen ihrer Objekte untereinander. Das bedeutet, die Struktur einer Kategorie, ausgedrückt durch „Morphismen“ über die die Objekte verbunden oder eben nicht verbunden sind, spielt eine bedeutendere Rolle als die Struktur ihrer Objekte. So werden Abbildungen zwischen Kategorien (\rightarrow 5.1.3) unabhängig von Objekten nur auf Basis der vorhandenen Morphismen definiert. Auch die Kategorien selbst könnten, anders als in 5.1.1, „objektfrei“ definiert werden (\rightarrow [HS73] § 2).

Die in 5.1.1 eingeführten Kategorien werden in der Literatur auch als „abstrakte Kategorien“ bezeichnet [HS73], im Gegensatz zu den um eine Abbildung auf Mengen und Funktionen ergänzten „konkreten Kategorien“. Da für viele der im weiteren verwendeten Kategorien diese als „unterliegendend“ bezeichnete Abbildung zwar möglich aber für uns von geringem Interesse ist, verwenden wir ausschließlich die allgemeinere Kategoriedefinition.

Ersetzt man in der folgenden Definition die Begriffe Objekt durch Knoten und Morphismus durch Kante erhält man reflexiv und transitiv abgeschlossene gerichtete Graphen. Umgekehrt kann jeder Graph durch seinen reflexiv-transitiven Abschluß in eine Kategorie transformiert werden. Wir werden von diesem Zusammenhang im Rahmen der formalen Beschreibung von Diagrammen des öfteren Gebrauch machen und von der durch einen Graphen generierten Kategorie reden.

5.1.1 Definition (Kategorie):

Eine „Kategorie“ ist ein Quintupel $\mathcal{C} = (\mathcal{O}, \mathcal{M}, dom, cod, \circ)$ worin

- (I) \mathcal{O} eine Klasse¹ bezeichnet, deren Elemente „ \mathcal{C} -Objekte“ genannt werden,
- (II) \mathcal{M} eine Klasse bezeichnet, deren Elemente „ \mathcal{C} -Morphismen“ genannt werden,

(III) dom und cod Abbildungen von \mathcal{M} nach \mathcal{O} bezeichnen, die „Domäne“ (dom) bzw. „Codomäne“ (cod) genannt werden,

(IV) \circ eine Abbildung von

$$D = \{ (f, g) \mid f, g \in \mathcal{M} \wedge dom(f) = cod(g) \}$$

nach \mathcal{M} bezeichnet, die „Kompositionsregel“ von \mathcal{C} genannt wird (wir notieren $\circ(f, g)$ durch $f \circ g$ und bezeichnen f, g als komponierbar, wenn $(f, g) \in D$);

so daß die folgenden Bedingungen erfüllt werden

(V) (**Verträglichkeit**) für alle komponierbaren Morphismen f, g gilt:
 $dom(f \circ g) = dom(g)$ und $cod(f \circ g) = cod(f)$;

(VI) (**Assoziativität**) für alle Morphismen f, g, h mit $(f, g), (h, f) \in D$ gilt:
 $h \circ (f \circ g) = (h \circ f) \circ g$;

(VII) (**Identitäten**) für jedes \mathcal{C} -Objekt A existiert ein \mathcal{C} -Morphismus id_A mit $dom(id_A) = A = cod(id_A)$, und für alle f, g aus \mathcal{M} mit $dom(f) = A = cod(g)$ gilt: $f \circ id_A = f$ und $id_A \circ g = g$;

(VIII) (**Morphismenmengen**): für alle \mathcal{C} -Objekte A, B bilden die Morphismen mit Domäne A und Codomäne B (notiert durch $Mor(\mathcal{C})[A, B]$) eine Menge.²

□

Für eine gegebenen Kategorie \mathcal{C} notieren wir die \mathcal{C} -Objekte durch $Obj(\mathcal{C})$ und die \mathcal{C} -Morphismen durch $Mor(\mathcal{C})$ bzw. durch $Mor(\mathcal{C})[A, B]$, wenn wir uns auf die Morphismen zwischen den Objekten A und B einschränken. Im übrigen werden wir Kategorien selten wirklich in Form von Quintupeln angeben. Häufig sind die Morphismen einer Kategorie durch Abbildungen gegeben, z.B. bei der Kategorie der freien abelschen Gruppen mit linearen Abbildungen als Morphismen, in diesen Fällen ergibt sich dom, cod für jeden Morphismus implizit aus der Domäne bzw. aus der Codomäne

¹Eine Klasse (\rightarrow [HS73]) ist eine unter Vereinigung, disjunkter Vereinigung, Schnitt und kartesischer Produktbildung abgeschlossene Kollektion von Mengen mit einer gemeinsamen Eigenschaft.

²Mengen verstehen wir hier im intuitiven Sinn wie sie beispielsweise im Rahmen der Mengentheorie betrachtet werden. Wir verzichten auf eine genauere Definition, die entweder unpräzise wäre oder den Rahmen dieser Arbeit sprengen würde und benennen anstellen dessen Eigenschaften, die erfüllt sein müssen. Bezüglich jeder Menge X und jeder Eigenschaft P , läßt sich die Menge „ $\{x \mid P(x)\}$ “ mit Elementen aus X bilden, die die Eigenschaft P erfüllen. Handelt es sich bei P um die Eigenschaft „ $x \neq x$ “ erhalten wir bzgl. jedem X die leere Menge, die wir mit „ \emptyset “ notieren. Für zwei beliebige Mengen X, Y existiert eine Menge „ $\{X, Y\}$ “, die genau X und Y enthält. Jede Menge X generiert eine Menge „ 2^X “, die alle Teilmengen von X enthält. Mengen sind abgeschlossen gegenüber Schnitt, Vereinigung, Komplementbildung bzgl. Teilmengen und dem kartesischen Produkt.

der entsprechenden Abbildung und die Komposition entspricht der üblichen Abbildungskomposition.

5.1.2 Definition (Kategorieeigenschaften):

Seien \mathcal{C} und \mathcal{D} Kategorien.

- (I) \mathcal{C} ist „*klein*“, wenn die \mathcal{C} -Morphismen eine Menge bilden.³
- (II) \mathcal{C} ist „*endlich*“, wenn \mathcal{C} klein und die Menge der \mathcal{C} -Morphismen endlich ist.
- (III) \mathcal{C} ist „*diskret*“, wenn die \mathcal{C} -Morphismen nur aus Identitäten bestehen.
- (IV) \mathcal{C} ist „*Unterkategorie*“ von \mathcal{D} , wenn gilt:

$$\text{Mor}(\mathcal{C}) \subseteq \text{Mor}(\mathcal{D}).^4$$

- (V) \mathcal{C} ist „*volle Unterkategorie*“ von \mathcal{D} , wenn \mathcal{C} eine Unterkategorie von \mathcal{D} ist und für alle \mathcal{C} -Objekte A, B gilt:

$$\text{Mor}(\mathcal{C})[A, B] = \text{Mor}(\mathcal{D})[A, B].$$

□

Im folgenden werden wir uns meist mit kleinen Kategorien, wie der Kategorie der freien abelschen Gruppen über endlichen Basismengen (\rightarrow 5.2.6) oder der Kategorie der Petrinetze mit individuellen Marken (\rightarrow 1.1.6), beschäftigen. Im Unterschied zu Kategorien im allgemeinen bilden die kleinen Kategorien zusammen mit strukturhaltenden Abbildungen (\rightarrow Funktoren 5.1.3) wiederum eine Kategorie. Endliche und diskrete Kategorien spielen eine wichtige Rolle als Muster bei der formalen Beschreibung von Diagrammen (\rightarrow S. 117). Diese „*Musterkategorien*“ sind von Graphen generiert und enthalten zumeist nur wenige Objekte und Morphismen ohne individuelle Eigenschaften.

„*Funktoren*“ zwischen Kategorien sind, ähnlich wie Gruppenmorphismen, strukturhaltende Abbildungen, d.h. sie sind mit den Identitäten und der Morphismenkomposition verträglich. Damit formt das Bild einer Kategorie unter einem beliebigen Funktor eine, nicht notwendigerweise volle, Unterkategorie seiner Codomäne. Da die Struktur von Kategorien ausschließlich von deren Morphismen bestimmt wird und die Objekte sich mittels ihrer Identitäten ebenfalls als Morphismen betrachten lassen reicht es aus, Funktoren als Abbildungen auf Morphismen zu definieren.

³Da die \mathcal{C} -Objekte nicht notwendigerweise eine Menge bilden, impliziert 5.1.1. VIII nicht das jede Kategorie klein ist.

⁴impliziert $\text{Obj}(\mathcal{C}) \subseteq \text{Obj}(\mathcal{D})$ (\rightarrow 5.1.1. VII)

5.1.3 Definition (Funktork):

Seien \mathcal{C} und \mathcal{D} Kategorien. Ein Funktor \mathcal{F} von \mathcal{C} nach \mathcal{D} , notiert $\mathcal{F}: \mathcal{C} \rightarrow \mathcal{D}$, ist eine Abbildung von $Mor(\mathcal{C})$ nach $Mor(\mathcal{D})$, die folgende Bedingungen erfüllt:

- (I) (**Identitäten bewahren**): für jede \mathcal{C} -Identität id_A ist $\mathcal{F}(id_A)$ eine \mathcal{D} -Identität und
- (II) (**Kompositionen bewahren**): für alle komponierbaren \mathcal{C} -Morphismen f, g gilt: $\mathcal{F}(f \circ g) = \mathcal{F}(f) \circ \mathcal{F}(g)$.

□

Die Identitäten jeder Kategorie \mathcal{C} stehen in bijektiven Zusammenhang zu den \mathcal{C} -Objekten. Dadurch induziert jeder Funktor \mathcal{F} von \mathcal{C} nach \mathcal{D} eine Abbildung der \mathcal{C} -Objekte in die \mathcal{D} -Objekte. Wir bezeichnen diese Abbildung wie üblich ebenfalls mit \mathcal{F} .

5.1.4 Definition (Funktoreigenschaften):

Ein Funktor $\mathcal{F}: \mathcal{C} \rightarrow \mathcal{D}$ ist „*treu*“, wenn seine Einschränkung

$$\mathcal{F} \begin{array}{l} \Big|_{Mor(\mathcal{D})[\mathcal{F}(A), \mathcal{F}(B)]} \\ \Big|_{Mor(\mathcal{C})[A, B]} \end{array}$$

für jedes Paar A, B aus \mathcal{C} -Objekten injektiv ist. Er ist „*voll*“, wenn die Einschränkungen surjektiv sind.

□

Ein im weiteren häufig vorkommendes Beispiel für Funktoren sind Diagramme. Um auf den umgangssprachlich als „*Diagramm*“ bezeichneten Objekten Limes- und Colimeskonstruktionen (\rightarrow 5.1.13, 5.1.15) anwenden zu können, muß man sie formal beschreiben. Dies geschieht mit Hilfe von Funktoren. Ein „*Diagrammfunktor*“ hat eine zumeist kleine (hier im wörtlichen Sinn), bei uns in jedem Falle jedoch endliche Kategorie \mathcal{S} (von „*Shape*“) als Domäne. Sie gibt das Muster des Diagramms wieder. Anstelle der Kategoriedefinition wird für „*Musterkategorien*“ zumeist der generierende Graph angegeben, z.B.

$$\begin{array}{ccc} & s_0 & \\ m_1 \swarrow & & \searrow m_2 \\ s_1 & & s_2 \end{array} \quad \text{oder} \quad s \begin{array}{c} \xrightarrow{m'} \\ \xrightarrow{m} \end{array} s'$$

(\rightarrow 5.1.14). Das Bild von \mathcal{S} unter dem Diagrammfunktor gibt die Belegung des durch \mathcal{S} vorgegebenen Musters in der Zielkategorie an. Diese Bilder lassen sich wie „normale Diagramme“ notieren

$$\begin{array}{ccc} & A_0 & \\ f_1 \swarrow & & \searrow f_2 \\ A_1 & & A_2 \end{array} \quad \text{bzw.} \quad A \begin{array}{c} \xrightarrow{f} \\ \xrightarrow{g} \end{array} B$$

und werden, soweit \mathcal{S} aus dem Kontext hervorgeht, ebenfalls als Diagramme bezeichnet.

Um mathematische Objekte wie die Kategorie aller Kategorien fassen zu können, muß die unter 5.1.1 gegebene Kategoriedefinition abgeschwächt werden. So bildet unter anderem die Gesamtheit aller Funktoren zwischen zwei Kategorien im allgemeinen, z.B. wenn es sich bei beiden Fällen um die Kategorie „ \mathcal{SET} “ der Mengen handelt, keine Menge und verletzt somit die Kategoriedefinition. Aus diesem Grund führen wir „*Quasikategorien*“ ein. Dieses Konzept erlaubt es, Kategorien und Funktoren zu einer Einheit, der Quasikategorie „ \mathcal{CATCAT} “ zusammenzufassen.

5.1.5 Definition (Quasikategorie):

Eine „*Quasikategorie*“ ist ein Quintupel $\mathcal{Q} = (\mathcal{O}, \mathcal{M}, dom, cod, \circ)$, dessen Definition sich nur in den folgenden Punkten von 5.1.1 unterscheidet:

- (I) \mathcal{O} ist ein Konglomerat⁵,
- (II) \mathcal{M} ist ein Konglomerat und
- (III) die Einschränkung auf Morphismenmengen zwischen je zwei Objekten entfällt.

□

Ähnlich wie Objekte über Morphismen und Kategorien über Funktoren lassen sich Funktoren über „*natürliche Transformationen*“ in Beziehung setzen. Sie übertragen die Bilder des einen von zwei Funktoren mit gleicher Domäne und gleicher Codomäne in Bilder des anderen, indem sie die Domänen und Codomänen der Morphismen in Beziehung setzen. Zu diesem Zweck wird jedem Objekt in der Domäne beider Funktoren ein Morphismus in deren Codomäne dergestalt zugeordnet, daß die beiden Bilder eines Morphismus aus der Domäne der Funktoren zu einem kommutativen Diagramm verknüpft werden.

5.1.6 Definition (Natürliche Transformation):

Seien $\mathcal{F}: \mathcal{C} \rightarrow \mathcal{D}$, $\mathcal{G}: \mathcal{C} \rightarrow \mathcal{D}$ zwei Funktoren. Eine „*natürliche Transformation*“ von \mathcal{F} nach \mathcal{G} ist ein Tripel $(\mathcal{F}, v, \mathcal{G})$ (notiert durch $v: \mathcal{F} \rightarrow \mathcal{G}$), bei dem

$$v: \text{Obj}(\mathcal{C}) \rightarrow \text{Mor}(\mathcal{D})$$

eine Abbildung ist, die folgende Bedingungen erfüllt:

- (I) für alle \mathcal{C} -Objekte A gilt $dom(v(A)) = \mathcal{F}(A)$ und $cod(v(A)) = \mathcal{G}(A)$ und

⁵Ein Konglomerat nach [HS73] ist eine Kollektion von Klassen oder Konglomeraten. Jede Klasse ist eine Kollektion. Konglomerate sind abgeschlossen unter den üblichen mengentheoretischen Konstruktionen.

(II) für alle \mathcal{C} -Morphismen $f: A \rightarrow A'$ kommutiert das folgende Diagramm

$$\begin{array}{ccccc}
 \mathcal{F}(A) & \xrightarrow{v(A)} & \mathcal{G}(A) & & A \\
 \mathcal{F}(f) \downarrow & & \downarrow \mathcal{G}(f) & & \downarrow f \\
 \mathcal{F}(A') & \xrightarrow{v(A')} & \mathcal{G}(A') & & A'
 \end{array}$$

□

Wir notieren die Bilder $v(A)$ einer natürlichen Transformation v in den meisten Fällen wie in der Literatur üblich durch v_A .

Ein Grundprinzip der Kategorientheorie ist das „*Dualitätsprinzip*“. Es ist sowohl auf Kategorien als auch auf Aussagen über Kategorien und auf Funktoren anwendbar. Dualitäten können zum einen dazu verwendet werden, aus vorhandenen Konzepten und Ergebnissen, die Kategorien im allgemeinen betreffen, neue Konzepte und Ergebnisse zu gewinnen, zum anderen erhält man damit zu jeder Eigenschaft einer speziellen Kategorie die entsprechende duale Eigenschaft der dualen Kategorie. Eine Reihe von Beispielen für derartige Dualitätsbeziehungen werden wir im weiteren betrachten.⁶ Besondere Bedeutung hat das Dualitätsprinzip für „*selbstduale Kategorien*“. Dabei handelt es sich um Kategorien, die äquivalent zu ihrer dualen Kategorie sind, wie beispielsweise die freien abelschen Gruppen (\rightarrow 5.2.9). Innerhalb einer selbstdualen Kategorie folgt aus jeder Eigenschaft sofort, d.h. ohne Beweis, auch deren duale Entsprechung.

5.1.7 Definition (Duale Kategorie):

Zu jeder Kategorie $\mathcal{C} = (\mathcal{O}, \mathcal{M}, \text{dom}, \text{cod}, \circ)$ bildet $\mathcal{C}^{op} = (\mathcal{O}, \mathcal{M}, \text{cod}, \text{dom}, \bullet)$ die duale Kategorie, wobei die Komposition in \mathcal{C}^{op} durch: $f \bullet g = g \circ f$ definiert ist.

□

5.1.8 Folgerung :

Für jede Kategorie \mathcal{C} gilt: $(\mathcal{C}^{op})^{op} = \mathcal{C}$.

□

Da viele Eigenschaften spezieller Kategorien wie beispielsweise Limites mit Hilfe von Funktoren beschrieben sind, werden zur Formulierung von deren dualen Entsprechungen „*duale Funktoren*“ eingeführt. Dabei ergibt sich zu jedem Funktor $\mathcal{F}: \mathcal{C} \rightarrow \mathcal{D}$ der duale Funktor $\mathcal{F}^{op}: \mathcal{C}^{op} \rightarrow \mathcal{D}^{op}$ aus $\mathcal{F}^{op}(f) = \mathcal{F}(f)$ für alle \mathcal{C}^{op} -Morphismen f . Da Diagramme als Funktoren definiert sind, ergibt sich auf diese Weise zu jedem Diagramm \mathcal{D} das duale Diagramm \mathcal{D}^{op} .

⁶die Konstruktion von Limites aus Produkten und Equalizern (\rightarrow 5.1.19) und dazu dual die Konstruktion von Colimites aus Coprodukten und Coequalizern für Kategorien allgemein, sowie initiale und terminale Objekte, Limites und Colimites oder universelle und couniverselle Abbildungen

Die Morphismenstruktur einer Kategorie gibt die Eigenschaften des entsprechenden Objekttyps wieder. In diesem Zusammenhang zeichnet sich das, wenn vorhanden bis auf Isomorphie eindeutige, „*initiale Objekt*“ einer Kategorie dadurch aus, daß in jedes andere Objekt genau⁷ ein Morphismus existiert. Interpretiert man Kategorien als Objekttypbeschreibungen, dann sind initiale Objekte diejenigen Objekte, die genau die Typeigenschaften besitzen und keine weiteren, d.h. sie sind in gewisser Weise minimal. In der Kategorie der endlichen Mengen „ \mathcal{SET}_f “ mit Mengenabbildungen als Morphismen ist die leere Menge initial, in der Kategorie der Ringe mit strukturerehaltenden Abbildungen als Morphismen, gilt dies für den Ring $(\mathbb{Z}, +, *)$.

5.1.9 Definition (Initiales Objekt):

Sei \mathcal{C} eine Kategorie, dann ist ein \mathcal{C} -Objekt I „*initial*“, wenn zu jedem \mathcal{C} -Objekt A die Menge $Mor(\mathcal{C})[I, A]$ mindestens einen Morphismus enthält (bezeichnet durch $eval_A$) und dieser Morphismus bis auf Isomorphie gleich allen anderen Morphismen in $Mor(\mathcal{C})[I, A]$ ist. □

In vielen Fällen läßt sich die Einbettung eines mathematischen Objekts, z.B. einer Menge in ein auf geeignete Weise vervollständigtes Objekt, z.B. der frei darüber erzeugten (abelschen) Gruppe (\rightarrow 5.2.10), als „*universelle Abbildung*“ interpretieren. Die Eindeutigkeit der universellen Abbildung impliziert das vervollständigte Objekt bis auf Isomorphie. Gleichzeitig läßt sich damit, wenn zwischen allen Objekten und ihren Vervollständigungen universelle Abbildungen existieren, die enge Beziehung zwischen der Kategorie der Objekte und der Kategorie ihrer Vervollständigungen zum Ausdruck bringen. Dieser Zusammenhang kann zu einem Funktor mit speziellen Eigenschaften, dem sogenannten „*freien Funktor*“ erweitert werden.

5.1.10 Definition (Universelle Abbildungen):

Sei $\mathcal{G}: \mathcal{C} \rightarrow \mathcal{D}$ ein Funktor und B ein \mathcal{D} -Objekt. Ein Paar (u, A) bestehend aus einem \mathcal{D} -Morphismus

$$u: B \rightarrow \mathcal{G}(A)$$

und einem \mathcal{C} -Objekt A bildet eine universelle Abbildung zu B bezüglich \mathcal{G} (oder eine \mathcal{G} -universelle Abbildung zu B), wenn für jedes \mathcal{C} -Objekt A' und jeden Morphismus $f: B \rightarrow \mathcal{G}(A')$ in \mathcal{D} eindeutig ein \mathcal{C} -Morphismus $f': A \rightarrow A'$ existiert, so daß das folgende Diagramm kommutiert:

$$\begin{array}{ccc}
 B & \xrightarrow{u} & \mathcal{G}(A) & & A \\
 & \searrow f & \downarrow \mathcal{G}(f') & & \downarrow f' \\
 & & \mathcal{G}(A') & & A'
 \end{array}$$

□

⁷bis auf Isomorphie

Objekte einer Kategorie, zu denen eine \mathcal{G} -universelle Abbildung existiert, bezeichnen wir als bezüglich \mathcal{G} „frei konstruiert“. Wir fordern dabei nicht, wie häufig in der Literatur (\rightarrow z.B. [HS73]), daß es sich bei \mathcal{G} um einen Vergißfunktorkomplex handelt. In den Fällen, in denen im vorliegenden Text freie Konstruktionen vorkommen, hat \mathcal{G} allerdings immer diese Eigenschaft; zumindest in dem Sinne, daß er Strukturinformationen seiner Domäne vergißt. Faßt man dagegen den Begriff „Vergißfunktorkomplex“ als „underlying functor“ auf (\rightarrow [HS73], [ML70]), der Objekte einer Kategorie in Mengen und Morphismen in Funktionen abbildet, so erfüllen die von uns verwendeten Funktoren diese Bedingung nicht in jedem Fall.

Bildet ein Funktor \mathcal{V} zwischen den Kategorien \mathcal{C} , \mathcal{D} wie \mathcal{G} in Definition 5.1.10 ab und existiert eine Objektabbildung F in Gegenrichtung, die jedem \mathcal{D} -Objekt A das \mathcal{C} -Objekt $F(A)$ zuordnet, so daß $(u, \mathcal{V}(F(A)))$ eine \mathcal{V} -universelle Abbildung zu A ist, dann bezeichnen wir diese Objektabbildung als „freie Konstruktion“ bzgl. \mathcal{V} . Freie Konstruktionen können mit Hilfe universeller Abbildungen eindeutig zu freien Funktoren erweitert werden. Dabei ergibt sich das Bild eines Morphismus f aufgrund der Eigenschaften universeller Abbildungen eindeutig aus der kommutativen Ergänzung des nachfolgenden Diagramms.

$$\begin{array}{ccc}
 A & \xrightarrow{u_A} & \mathcal{V}(\mathcal{F}(A)) & & \mathcal{F}(A) \\
 f \downarrow & & \downarrow \mathcal{V}(\mathcal{F}(f)) & & \downarrow \mathcal{F}(f) \\
 B & \xrightarrow{u_B} & \mathcal{V}(\mathcal{F}(B)) & & \mathcal{F}(B)
 \end{array}$$

In diesem Fall bildet $u: \mathcal{I}\mathcal{D}_{\mathcal{C}} \rightarrow \mathcal{V} \circ \mathcal{F}$ eine natürliche Transformation (\rightarrow 5.1.6).

Kommutative Ergänzungen von Diagrammen, bestehend aus einem Objekt und einer Morphismenfamilie von diesem Objekt in jedes Diagrammobjekt bzw. von den Diagrammobjekten in das Objekt, bezeichnen wir als „Quellen“ bzw. als „Senken“. Dabei müssen Objekt und Morphismenfamilie aus der Codomäne des Diagrammfunktors stammen. Jede Quelle eines Diagramms ist die Senke und jede Senke die Quelle des zugehörigen dualen Diagramms.

5.1.11 Definition (Quelle):

Sei $\mathcal{D}: \mathcal{S} \rightarrow \mathcal{C}$ ein Funktor, dann bildet ein Paar

$$(Q, (q_s)_{s \in \text{Obj}(\mathcal{S})})$$

bestehend aus einem \mathcal{C} -Objekt Q und einer Familie von \mathcal{C} -Morphismen mit Domäne Q eine „Quelle“ für \mathcal{D} in \mathcal{C} , wenn für jeden \mathcal{S} -Morphismus $m: s \rightarrow s'$ das folgende Diagramm kommutiert:

$$\begin{array}{ccc}
 \mathcal{D}(s) & \xrightarrow{\mathcal{D}(m)} & \mathcal{D}(s') \\
 & \searrow q_s & \uparrow q_{s'} \\
 & & Q
 \end{array}$$

□

5.1.12 Definition (Senke):

Sei $\mathcal{D}: \mathcal{S} \rightarrow \mathcal{C}$ ein Funktor, dann bildet ein Paar

$$((r_s)_{s \in \text{Obj}(\mathcal{S})}, R)$$

bestehend aus einem \mathcal{C} -Objekt R und einer Familie von \mathcal{C} -Morphismen mit Co-domäne R eine „Senke“ für \mathcal{D} in \mathcal{C} , wenn für jeden \mathcal{S} -Morphismus $m: s \rightarrow s'$ das folgende Diagramm kommutiert:

$$\begin{array}{ccc} \mathcal{D}(s) & \xrightarrow{\mathcal{D}(m)} & \mathcal{D}(s') \\ & \searrow r_s & \downarrow r_{s'} \\ & & R \end{array} \quad \circlearrowright$$

□

Faßt man alle Quellen (Senken) eines Diagramms als Objekte einer Kategorie auf, deren Morphismen von mit den Morphismenfamilien verträglichen Abbildungen zwischen den Objekten gebildet werden, dann ist der Limes (Colimes) gerade das initiale Objekt dieser Kategorie. Das heißt Limites und Colimites bilden eindeutige kommutative Ergänzungen von Diagrammen und zeichnen sich gegenüber anderen kommutativen Ergänzungen dadurch aus, daß ihre Objekte keine überflüssige Struktur haben, also bzgl. der Diagrammergänzung minimal sind.

5.1.13 Definition (Limes):

Sei $\mathcal{D}: \mathcal{S} \rightarrow \mathcal{C}$ ein Funktor, dann bildet eine Quelle $(L, (\ell_s)_{s \in \text{Obj}(\mathcal{S})})$ zu \mathcal{D} einen Limes, wenn sie die folgende, als „universell“ bezeichnete, Eigenschaft besitzt.

(Universelle Eigenschaft): Zu jeder Quelle $(Q, (q_s)_{s \in \text{Obj}(\mathcal{S})})$ für \mathcal{D} existiert genau ein \mathcal{C} -Morphismus $\hat{\ell}: Q \rightarrow L$ mit $\ell_s \circ \hat{\ell} = q_s$ für alle \mathcal{S} -Objekte s .

□

5.1.14 Beispiel (Produkte und Equalizer):

Ein im Rahmen von Limesbetrachtungen wichtiger Spezialfall sind Diagrammfunktoren, deren Domäne von einer endlichen diskreten Kategorie gebildet wird, d.h. von einer Kategorie, die nur Identitäten als Morphismen enthält. Die Limites dieser Diagramme werden als „endliche Produkte“ bezeichnet und spielen bei der Limesbildung endlicher Diagramme eine grundlegende Rolle (\rightarrow 5.1.19).

Neben den Produkten sind „Equalizer“ die andere grundlegende Limeskonstruktion. Dabei handelt es sich um Limites zu Diagrammen der Form⁸

⁸Die Musterkategorie zu diesem Diagramm bildet die vom rechten Graphen auf Seite 117 generierte Kategorie, und die Abbildungsvorschrift des Diagrammfunktors ergibt sich implizit aus dem Zusammenhang der Elemente beider Graphen. Die in jedem Diagramm eindeutigen Identitätsmorphismen sind weggelassen.

$$A \begin{array}{c} \xrightarrow{f} \\ \xrightarrow{g} \end{array} B$$

Wesentlicher Bestandteil jedes Equalizers ist ein eindeutiger Morphismus ℓ von einem Objekt E nach A mit der Eigenschaft

$$f \circ \ell = g \circ \ell.$$

Während das zugehörige Objekt durch die Domäne von ℓ festgelegt ist, ergibt sich der Morphismus von E nach B für jedes Diagramm aus der Komposition von ℓ mit einem der beiden Diagrammmorphismen. Aus diesem Grund notieren wir Equalizer abkürzend durch (E, ℓ) anstellen von $(E, (\ell_s)_{s \in \text{Obj}(\mathcal{S})})$.

In der Kategorie der Mengen \mathcal{SET} entspricht das Produkt eines Diagramms dem kartesischen Produkt der Diagrammobjekte (\rightarrow [ML70] S. 73–74). Das Objekt des Equalizers zweier Abbildungen f, g in \mathcal{SET} mit gleicher Domäne/Codomäne entspricht der Teilmenge der Domäne, deren Elemente von f und g gleich abgebildet werden (\rightarrow [ML70] S. 74). Die Inklusion bildet in diesem Fall den Equalizermorphismus. □

5.1.15 Definition (Colimes):

Sei $\mathcal{D}: \mathcal{S} \rightarrow \mathcal{C}$ ein Funktor, dann bildet eine Senke $((c_s)_{s \in \text{Obj}(\mathcal{S})}, C)$ zu \mathcal{D} einen Colimes, wenn sie die folgende, als „*co-universell*“ bezeichnete Eigenschaft besitzt.

(Co-universelle Eigenschaft): Zu jeder Senke $((r_s)_{s \in \text{Obj}(\mathcal{S})}, R)$ für \mathcal{D} existiert genau ein \mathcal{C} -Morphismus $\hat{c}: C \rightarrow R$ mit $\hat{c} \circ c_s = r_s$ für alle \mathcal{S} -Objekte s . □

5.1.16 Beispiel (Coproducte und Coequalizer):

Dual zu Produkten und Equalizern sind „*Coproducte*“ und „*Coequalizer*“. Dabei handelt es sich jeweils um die Colimites zu den in 5.1.14 beschriebenen Diagrammen. So ist der wesentliche Bestandteil jedes Coequalizers ein eindeutiger Morphismus $c: B \rightarrow C$ mit der Eigenschaft $c \circ f = c \circ g$. Ihnen kommt bei der Colimesbildung dieselbe grundlegende Rolle zu die Produkte und Equalizer bei der Limesbildung spielen.

In der Kategorie der Mengen \mathcal{SET} entspricht das Coprodukt eines Diagramms der disjunkten Vereinigung der Diagrammobjekte (\rightarrow [ML70] S. 66–67). Das Objekt des Coequalizers zweier Abbildungen f, g in \mathcal{SET} mit gleicher Domäne/Codomäne läßt sich aus der Faktorisierung der Codomäne mit der kleinsten Äquivalenzrelation, die alle Paare $(f(a), g(a))$ für die Elemente a aus der Codomäne enthält, gewinnen (\rightarrow [ML70] S. 69). Coequalizermorphismus ist in diesem Fall die Abbildung, die allen Elementen ihre Äquivalenzklasse zuordnet. □

Limites und Colimites sind zueinander duale Konstruktionen. Kennt man den Limes eines Diagramms \mathcal{D} in \mathcal{C} , so ist dadurch auch der Colimes des dualen Diagramms \mathcal{D}^{op} in \mathcal{C}^{op} gegeben und umgekehrt. Von besonderem Nutzen ist dieser Zusammenhang bei Kategorien, die äquivalent zu ihrer dualen Kategorie sind, wie beispielsweise die freien abelschen Gruppen (\rightarrow 5.2.6, 5.2.9). In diesem Fall braucht nur jeweils eine der beiden Konstruktionen durchgeführt werden, die andere ergibt sich durch Dualisierung.

5.1.17 Folgerung :

Sei $\mathcal{F}: \mathcal{S} \rightarrow \mathcal{C}$ ein Funktor, dann gilt:

- (I) ein Paar $(Q, (q_s)_{s \in \text{Obj}(\mathcal{S})})$ bildet eine Quelle zu \mathcal{D} , wenn $((q_s)_{s \in \text{Obj}(\mathcal{S})}, Q)$ eine Senke zu \mathcal{D}^{op} bildet und
- (II) ein Paar $(L, (\ell_s)_{s \in \text{Obj}(\mathcal{S})})$ bildet einen Limes zu \mathcal{D} , wenn $((\ell_s)_{s \in \text{Obj}(\mathcal{S})}, L)$ einen Colimes zu \mathcal{D}^{op} bildet.

□

Häufig verwendete Spezialfälle von Limes- bzw. Colimeskonstruktionen sind „Pullbacks“ und „Pushouts“, von denen wir nur letztere gesondert einführen. Pullbacks ergeben sich durch Dualisierung aus Pushouts. Im Unterschied zu der sonst üblichen Trennung von Diagramm und Colimes werden bei Pushouts deren Objekte und Morphismen zu einem Diagramm zusammengefaßt.

5.1.18 Definition (Pushout):

Sei \mathcal{S} die vom linken Graphen generierte Kategorie, dann bildet das rechte Diagramm

$$\begin{array}{ccc}
 & s_0 & \\
 m_1 \swarrow & & \searrow m_2 \\
 s_1 & & s_2
 \end{array}
 \qquad
 \begin{array}{ccc}
 A_0 & \xrightarrow{f_2} & A_2 \\
 f_1 \downarrow & \circlearrowleft & \downarrow c_{s_2} \\
 A_1 & \xrightarrow{c_{s_1}} & C
 \end{array}$$

einen Pushout wenn: $(C, (c_{s_i})_{i=0,1,2})$, $c_{s_0} = c_{s_1} \circ f_1$ einen Colimes zum Funktor \mathcal{D} mit Domäne \mathcal{S} , $\mathcal{D}(s_i) = A_i$, $i = 0, 1, 2$ und $\mathcal{D}(m_j) = f_j$, $j = 1, 2$ bildet.

□

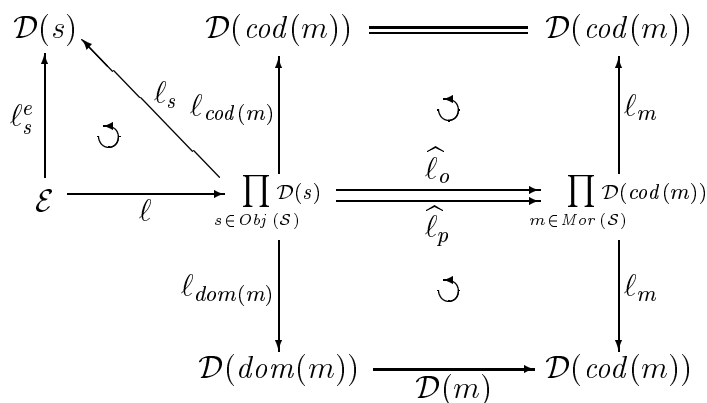
Zwei Konstruktionen sind grundlegend für jede Limesbildung, zum einen das Zusammenfügen der Diagrammobjekte zu einem Objekt und zum anderen dessen Faktorisierung anhand der durch die Diagrammorphismen gegebenen Struktur. Das Zusammenfügen von Objekten kann mit Hilfe von Produkten (\rightarrow 5.1.14) erfolgen, zum Faktorisieren lassen sich Equalizer nutzen (\rightarrow 5.1.14). Tatsächlich läßt sich der Limes jedes endlichen Diagramms unter ausschließlicher Verwendung dieser beiden Diagrammtypen mit den entsprechenden Limites ermitteln. Damit genügt es innerhalb einer beliebigen Kategorie, diese beiden konkreten und zumindest von der Struktur her einfachen Limeskonstruktionen zu beherrschen. Auf der anderen Seite manifestiert

sich schon in diesen beiden Konstruktionen die gesamte Problematik der Limesbildung.

Der nachfolgende Satz, der diese Konstruktion zum Inhalt hat, ist mit notationellen Modifikationen aus [ML70] (Satz 1, S. 117–118) entnommen. In diesem Zusammenhang notieren wird das Produkt eines Diagramms \mathcal{D} mit Domäne \mathcal{S} wie auch dessen Objekt durch $\prod_{s \in \text{Obj}(\mathcal{S})} \mathcal{D}(s)$.

5.1.19 Satz (Limites generiert aus Produkten und Equalizern):

Sei \mathcal{S} eine endliche Kategorie, \mathcal{C} eine Kategorie mit allen endlichen Produkten (bezeichnet mit \prod) und $\mathcal{D}: \mathcal{S} \rightarrow \mathcal{C}$ ein Funktor, dann ergibt sich der Limes von \mathcal{D} aus dem nachfolgenden Diagramm zu $(\mathcal{E}, (\ell_s^e)_{s \in \text{Obj}(\mathcal{S})})$



wenn der Equalizer (\mathcal{E}, ℓ) von $\widehat{\ell}_o, \widehat{\ell}_p$ existiert. Die Morphismen $\widehat{\ell}_o, \widehat{\ell}_p$ ergeben sich eindeutig aus den universellen Eigenschaft des rechts stehenden Produkts und den Quellen $(\prod_{s \in \text{Obj}(\mathcal{S})} \mathcal{D}(s), (\ell_{\text{cod}(m)})_{m \in \text{Mor}(\mathcal{S})}), (\prod_{s \in \text{Obj}(\mathcal{S})} \mathcal{D}(s), (\mathcal{D}(m) \circ \ell_{\text{dom}(m)})_{m \in \text{Mor}(\mathcal{S})})$.

□

(Beweis \rightarrow [ML70] Satz 1, S. 117–118)

5.2 Freie abelsche Gruppen

Bei den „freien abelschen Gruppen“ handelt es sich um additive Gruppen, deren Trägermenge sich mit Hilfe von Inversenbildung und der Gruppenoperation aus einer „Basis“ genannten Menge erzeugen läßt. Sie stehen in engem Zusammenhang zur Menge aller Funktionen von der Basis in die ganzen Zahlen und mit Vektorräumen über dem Ring $(\mathbb{Z}, +, *)$.⁹ Zusammen mit den linearen Abbildungen zwischen ihren Trägermengen, d.h. Abbildungen, die das neutrale Element bewahren und mit der Gruppenoperation verträglich sind, bilden die freien abelschen Gruppen die gleichnamige Kategorie.

⁹auch als \mathbb{Z} -Module bezeichnet

Im Rahmen dieser Arbeit spielen zwei Unterkategorien der freien abelschen Gruppen eine herausragende Rolle, zum einen die über einer endlichen Basis erzeugten Gruppen mit monotonen Morphismen, zum anderen die endlich erzeugten Gruppen, deren Morphismen kanonisch aus Funktionen zwischen den Basismengen erzeugt sind. Letztere entsprechen dem Bild eines freien Funktors von der Kategorie der endlichen Mengen in die Kategorie der freien abelschen Gruppen. Damit lassen sich insbesondere Colimites von den endlichen Mengen, wo sie für jedes endliche Diagramm existieren, auf die freien abelschen Gruppen übertragen.

Beim Übergang von der Unterkategorie der freien abelschen Gruppen mit funktionsgenerierten Morphismen auf die Unterkategorie, für deren Morphismen lediglich Monotonie gefordert ist, verliert man im allgemeinen die Möglichkeit, Limites von Diagrammen aus deren Entsprechung in der Kategorie der Mengen abzuleiten. Hier läßt sich aus dem Zusammenhang von freien abelschen Gruppen zu Vektorräumen und von linearen Abbildungen zu Matrizen in ausgezeichneten Fällen trotzdem eine, zu dem noch effizient berechenbare, Lösung ableiten.

5.2.1 Definition (Freie abelsche Gruppe):

Ein Paar (T, \oplus) bildet eine „freie abelsche Gruppe“ über der abzählbaren Menge A , bezeichnet mit A^{\otimes} , wenn T die kleinste bzgl. \oplus abgeschlossene Menge ist, für die gilt:

- (I) A ist eine Teilmenge von T ,
- (II) T enthält ein von A verschiedenes Element 0 , bezeichnet als „*neutrales Element*“,
- (III) für jedes a aus A enthält T ein von A verschiedenes Element \bar{a} , bezeichnet als „*inverses Element*“ zu a ,

und \oplus eine zweistellige Operation ist, die für jedes a in A , u, v, w in T die folgenden Bedingungen erfüllt:

- (IV) (**Neutralität**) $v \oplus 0 = v$,
- (V) (**Invertierbarkeit**) $a \oplus \bar{a} = 0$,
- (VI) (**Assoziativität**) $(u \oplus v) \oplus w = u \oplus (v \oplus w)$ und
- (VII) (**Kommutativität**) $v \oplus w = w \oplus v$.

□

Wenn Verwechslungen ausgeschlossen sind, bezeichnen wir die Trägermenge einer freien abelschen Gruppe über A , genau wie die Gruppe selbst mit A^{\otimes} und ihre Basismenge A durch $Base(A^{\otimes})$. Jedes Element aus A^{\otimes} , läßt sich als formale Summe

$$\bigoplus_{a \in A} x_a * a \quad \text{mit } x_a \in \mathbb{Z}$$

darstellen, dabei ist die Menge $\{a \in A \mid x_a \neq 0\}$ endlich, und $x_a * a$ steht als Abkürzung, für:

$$\bigoplus_1^{x_a} a \quad \text{wenn } x_a > 0, \quad \bigoplus_1^{-x_a} \bar{a} \quad \text{wenn } x_a < 0 \quad \text{oder} \quad 0 \quad \text{wenn } x_a = 0.$$

Alternativ können die Elemente der Gruppe auch als Funktionen von A in die ganzen Zahlen aufgefaßt werden. In diesem Fall notieren wir die Koeffizienten x_a aus den formalen Summen eines Elementes v durch $v(a)$.

5.2.2 Beispiel (Notationen für die Elemente freier abelscher Gruppen):

Sei A^\otimes die freie abelsche Gruppe über der Menge $\{a, b, c\}$, dann bezeichnen die folgenden drei Notationen

$$a \oplus b \oplus b \oplus b \oplus b \oplus b \oplus \bar{c} \oplus \bar{c}, \quad a \oplus 5 * b \oplus -2c, \quad v: \begin{array}{l} a \mapsto 1 \\ b \mapsto 5 \\ c \mapsto -2 \end{array}$$

jeweils das gleiche Element aus A^\otimes .

□

Mit der Gruppenstruktur verträgliche Abbildungen zwischen den Trägermengen freier abelscher Gruppen bezeichnen wir als Morphismen oder, um sie von Morphismen anderer Kategorien zu unterscheiden, als „*Morphismen in den freien abelschen Gruppen*“. Sie sind jeweils schon durch ihr Verhalten auf den Basiselementen eindeutig bestimmt, da sich jedes Element als formale Summe über Basiselementen beschreiben läßt und Summen bewahrt werden. Wir werden von dieser Möglichkeit zur endlichen Beschreibung von Morphismen in den freien abelschen Gruppen im weiteren häufig Gebrauch machen.

5.2.3 Definition (Morphismen):

Seien $A^\otimes = (T_A, \oplus)$, $B^\otimes = (T_B, \oplus)$, freie abelsche Gruppen, dann ist eine Abbildung f von T_A nach T_B ein Morphismus in den freien abelschen Gruppen, bezeichnet mit $f: A^\otimes \rightarrow B^\otimes$, wenn gilt:

(I) (**Verträglichkeit mit 0**): $f(0) = 0$ und

(II) (**Verträglichkeit mit \oplus**): $\forall a, a' \in A: f(a \oplus a') = f(a) \oplus f(a')$.

□

Morphismen in den freien abelschen Gruppen, die Basiselemente auf Summen von Basiselementen abbilden, d.h. bei denen in den Summen keine Inversen auftreten, bezeichnen wir als „*monoton*“. Definiert man für jede freie abelsche Gruppe A^\otimes eine partielle Ordnung „ \leq “ durch:

$$v \leq w \iff \forall a \in A: v(a) \leq w(a)$$

dann gilt für alle monotonen Morphismen f und Elemente v, w aus $\text{dom}(f)$:

$$v \leq w \implies f(v) \leq f(w).$$

Für Morphismen im allgemeinen gilt diese Implikation nicht, da die Bilder vergleichbarer Elemente nicht vergleichbar sein müssen.

5.2.4 Definition (Monotonie):

Sei $f: A^\otimes \rightarrow B^\otimes$ ein Morphismus in den freien abelschen Gruppen, dann ist f „*monoton*“, wenn für alle a aus A gilt:

$$f(a) = \bigoplus_{b \in B} x_b * b \implies \forall b \in B: x_b \geq 0.$$

□

Eine weitere wichtige Klasse innerhalb der freien abelschen Gruppen bilden die von Funktionen zwischen Mengen generierten Morphismen. Bei ihnen ist die Abbildung der Basiselemente durch die zugrundeliegende Funktion festgelegt, die Abbildung der übrigen Elemente ergibt sich aus der geforderten Verträglichkeit mit der Gruppenoperation und dem neutralen Element. Damit ist auch klar, daß diese Morphismen in jedem Fall monoton sind. Wenn Verwechslungen ausgeschlossen sind, bezeichnen wir diese Morphismen einfach als „*funktionsgenerierte Morphismen*“.

5.2.5 Definition (funktionsgenerierte Morphismen):

Sei $f: A \rightarrow B$ eine Abbildung zwischen Mengen, dann definieren wir einen f -generierten \mathcal{AG} -Morphismus $f^\otimes: A^\otimes \rightarrow B^\otimes$ als Morphismus in den freien abelschen Gruppen, der für alle a aus A die folgende Bedingung erfüllt:

$$f^\otimes(a) = f(a).$$

□

Zu jeder freien abelschen Gruppe bildet die Identitätsabbildung ihrer Trägermenge einen Morphismus im Sinne von Definition 5.2.3 und die sich aus der Funktionskomposition ergebende Komposition von Morphismen erfüllt die Verträglichkeitsbedingung von Kategorien (\rightarrow 5.1.1.V), so daß die freien abelschen Gruppen mit ihren Morphismen alle Bedingungen einer Kategorie erfüllen. Je nachdem, wie die eingeführten Einschränkungen auf den Objekten (endliche Basis) und auf den Morphismen (Monotonie, Funktionsgeneriertheit) miteinander kombiniert werden, erhält man fünf Unterkategorien der freien abelschen Gruppen, von denen die Kategorie der endlich erzeugten Gruppen mit monotonen Morphismen und die Kategorie der endlich erzeugten Gruppen mit generierten Morphismen im Rahmen dieser Arbeit von besonderem Interesse sind.

5.2.6 Definition (Kategorien freier abelscher Gruppen):

Wir definieren \mathcal{AG} als die Kategorie, die alle freien abelschen Gruppen und ihre Morphismen enthält. Identitäten und Komposition entsprechen den gleichnamigen Konstruktionen auf Abbildungen zwischen Mengen. Darüber hinaus definieren wir folgende Unterkategorien von \mathcal{AG} :

- (I) \mathcal{AG}_f als größte volle Unterkategorie von \mathcal{AG} , in der alle Objekte über endliche Basismengen verfügen,
- (II) \mathcal{AG}_m als größte Unterkategorie von \mathcal{AG} , in der alle Morphismen monoton sind,
- (III) \mathcal{AG}_g als größte Unterkategorie von \mathcal{AG} , in der alle Morphismen von Funktionen generiert sind,
- (IV) \mathcal{AG}_{fm} als größte Unterkategorie von \mathcal{AG}_f , in der alle Morphismen monoton sind und
- (V) \mathcal{AG}_{fg} als größte Unterkategorie von \mathcal{AG}_g , in der alle Objekte über endliche Basismengen verfügen.

□

In der zu \mathcal{AG}_f dualen Kategorie \mathcal{AG}_f^{op} (\rightarrow 5.1.7) ist jeder \mathcal{AG}_f -Morphismus enthalten, jedoch mit gegenüber den ursprünglichen Morphismen vertauschter Domäne und Codomäne. Um eine Beziehung zwischen den beiden Kategorien herzustellen, ordnen wir in der folgenden Definition jedem Morphismus f zwischen zwei freien abelschen Gruppen A^\otimes, B^\otimes den Morphismus f^{op} zu, der in entgegengesetzter Richtung abbildet. Er ergibt sich aus der Zuordnung der Basiselemente b aus B zu Summen derjenigen Elemente aus A , deren Bild unter f b enthält. Diese Konstruktion setzt die Endlichkeit von A voraus, da andernfalls unendliche Summen entstehen könnten und damit Bilder, die außerhalb von A^\otimes liegen.

5.2.7 Definition (Duale Morphismen):

Wir definieren die Abbildung $(-)^{op}: Mor(\mathcal{AG}_f) \rightarrow Mor(\mathcal{AG}_f)$ für jeden Morphismus $f: A^\otimes \rightarrow B^\otimes$ in \mathcal{AG}_f und b in B durch:

$$f^{op}: B^\otimes \rightarrow A^\otimes, \quad \text{mit} \quad f^{op}(b) = \bigoplus_{a \in A} f(a)(b) * a$$

□

5.2.8 Beispiel (Die Abbildung $(-)^{op}$):

$$\begin{array}{l} f: a_1 \mapsto 2 * b_1 \oplus 4 * b_2 \\ \quad a_2 \mapsto b_2 \\ \quad a_3 \mapsto b_3 \\ \quad \vdots \mapsto \vdots \end{array} \quad \Longrightarrow \quad \begin{array}{l} f^{op}: b_1 \mapsto 2 * a_1 \\ \quad b_2 \mapsto 4 * a_1 \oplus a_2 \\ \quad b_3 \mapsto a_3 \\ \quad \vdots \mapsto \vdots \end{array}$$

□

Da für jede Kategorie \mathcal{C} $\text{Mor}(\mathcal{C}) = \text{Mor}(\mathcal{C}^{op})$ gilt (\rightarrow 5.1.7), können wir $(-)^{op}$ als einen Funktor von \mathcal{AG}_f nach \mathcal{AG}_f^{op} auffassen, wodurch die zur dualen Kategorie korrespondierende Notation von $(-)^{op}$ motiviert ist. Von den Morphismuseigenschaften Generiertheit und Monotonie erhält $(-)^{op}$ nur die Monotonie. Lediglich die von injektiven Funktionen generierten Morphismen haben unter $(-)^{op}$ Bilder in \mathcal{AG}_{fg}^{op} .

5.2.9 Lemma (Eigenschaften von $(-)^{op}$):

Sei \mathcal{AG}_f^{op} die zu \mathcal{AG}_f und \mathcal{AG}_{fm}^{op} die zu \mathcal{AG}_{fm} duale Kategorie, dann gilt:

- (I) $(-)^{op}$ ist ein Funktor von \mathcal{AG}_f nach \mathcal{AG}_f^{op}
- (II) $\mathcal{AG}_f^{op} \equiv \mathcal{AG}_f$,
- (III) $\mathcal{AG}_{fm}^{op} \equiv \mathcal{AG}_{fm}$ und
- (IV) $(-)^{op} \circ (-)^{op} = \mathcal{ID}_{\mathcal{AG}_f}$.

□

(Beweis \rightarrow Anhang S. 258)

Kategorien, die wie die freien abelschen Gruppen äquivalent zu ihren dualen Entsprechungen sind, bezeichnet man als „*selbstdual*“. Aufgrund dieser Äquivalenz und der Tatsache, daß $(-)^{op}$ sowohl nach \mathcal{AG}_f^{op} als auch nach \mathcal{AG}_f abbildet, werden wir $(-)^{op}$ im weiteren, wenn nicht anders gesagt, als Funktor von \mathcal{AG}_f nach \mathcal{AG}_f betrachten.

Daß die freien abelschen Gruppen nicht nur dem Namen nach, sondern tatsächlich bezüglich ihrer Basis frei konstruiert sind, zeigt der nachfolgende Satz.¹⁰ Als Vergebefunktor in die Mengen fungiert dabei die Abbildung, die jedem Morphismus die zugrundeliegende Abbildung zwischen den Trägermengen zuordnet. Betrachtet man die generierten Morphismen (\rightarrow 5.2.5) als Bilder eines Funktors $(-)^{\circledast}$, so ergibt sich daraus ein „*freier Funktor*“ von den Mengen in die freien abelschen Gruppen.

5.2.10 Satz (Freie Konstruktion):

Sei die Abbildung $(-)^{\circledast}: \mathcal{SET} \rightarrow \mathcal{AG}$ auf allen Funktionen definiert wie in 5.2.5 und $\mathcal{V}: \mathcal{AG} \rightarrow \mathcal{SET}$ der Vergebefunktor, der den Morphismen die zugrundeliegende Abbildung zwischen Trägermengen zuordnet. Dann gilt:

- (I) für alle Mengen A ist A^{\circledast} bzgl. \mathcal{V} frei über A konstruiert und

¹⁰Aus algebraischer Sicht ist bekannt, daß es sich bei den freien abelschen Gruppen, betrachtet man sie als Vektorräume, um „*freie Module*“ handelt. Diese Eigenschaft entspricht wiederum der freien Konstruktion der Kategorientheorie. Da dieser Zusammenhang aber nicht so offensichtlich ist und darüber hinaus ein schönes Beispiel für eine freie Konstruktion liefert, haben wir Satz 5.2.10 hier aufgenommen.

(II) $(-)^{\otimes}$ ist ein freier Funktor bzgl. \mathcal{V} . □

(Beweis \rightarrow Anhang S. 260)

Als Spezialfälle von Linksadjunktionen zu den jeweiligen Vergißfunktoren bewahren freie Funktoren Colimites (\rightarrow [HS73] 27.5, 27.7). Daraus folgt mit der Existenz aller endlichen Colimites in der Kategorie der endlichen Mengen (\rightarrow 5.1.16) die Existenz aller Colimites in den endlich erzeugten freien abelschen Gruppen mit generierten Morphismen.

5.2.11 Korollar (Covollständigkeit):

Die Kategorie \mathcal{AG}_{fg} der endlich erzeugten freien abelschen Gruppen mit generierten Morphismen ist endlich covollständig. □

Die Kategorie der Vektorräume und Matrizen, die nachfolgend definiert wird, kann neben den Musterkategorien zu Diagrammen als ein weiteres Beispiel dafür betrachtet werden, daß Morphismen nicht notwendigerweise Abbildungen sein müssen. In dieser Kategorie bilden $m \times n$ -Matrizen über \mathbb{Z} die Morphismen, mit dem n -dimensionalen Vektorraum über \mathbb{Z} als Domäne und dem m -dimensionalen Vektorraum über \mathbb{Z} als Codomäne.

5.2.12 Definition (Kategorie der Vektorräume und Matrizen):

Die Kategorie der arithmetischen Vektorräume und Matrizen \mathcal{VR} enthält alle n -dimensionalen \mathbb{Z} -Vektorräume \mathbb{Z}^n (n aus \mathbb{N}) über dem Ring $(\mathbb{Z}, +, *)$ als Objekte und alle $m \times n$ -Matrizen über \mathbb{Z} ($m, n \in \mathbb{N}$) als Morphismen.

(I) **(Domäne)** Für jede $m \times n$ -Matrix M gilt: $dom(M) = \mathbb{Z}^n$.

(II) **(Codomäne)** Für jede $m \times n$ -Matrix M gilt: $cod(M) = \mathbb{Z}^m$.

(III) **(Identitäten)** Zu jedem n -dimensionalen Vektorraum \mathbb{Z}^n bildet die $n \times n$ -Einheitsmatrix, die wir mit E_n bezeichnen, die Identität.

(IV) **(Komposition)** Die Komposition in \mathcal{VR} ist durch die Matrixmultiplikation definiert und wird mit „ $*$ “ bezeichnet. □

Jede endliche Menge läßt sich total ordnen. Mit Hilfe dieser Ordnung kann jede endlich erzeugte freie abelsche Gruppe durch Zuordnung der Koeffizientenvektoren zu den formalen Summen

$$\bigoplus_{i=1}^n x_{a_i} * a_i \longmapsto \begin{pmatrix} x_{a_1} \\ \vdots \\ x_{a_n} \end{pmatrix},$$

in den n -dimensionalen Vektorraum über den ganzen Zahlen \mathbb{Z}^n abgebildet werden. Dabei steht n für die Anzahl der Elemente in der Basis der Gruppe. Auf der anderen Seite bildet der Zusammenhang von linearen Abbildungen zwischen Ringen zu Matrizen eine Grundlage linearer Algebra (\rightarrow z.B. [KM95] Kap. 3), und freie abelsche Gruppen lassen sich durch Hinzunahme der komponentweisen Multiplikation zu einem Ring erweitern.

5.2.13 Definition (Abbildung von Morphismen in Matrizen):

Wir definieren die Abbildung $(-)^{\mathcal{M}}: \text{Mor}(\mathcal{AG}_f) \rightarrow \text{Mor}(\mathcal{VR})$ durch:

$$(f)^{\mathcal{M}} \stackrel{\text{def}}{=} (f_{i,j} = f(a_j)(b_i))_{i=1,\dots,|B|; j=1,\dots,|A|}$$

für alle Morphismen $f: A^{\otimes} \rightarrow B^{\otimes}$ aus \mathcal{AG}_f .

□

Wenn im weiteren nicht anders gesagt, soll die Reihenfolge der Elemente einer durch Aufzählung definierten Menge, die mit „ $<$ “ bezeichnete Totalordnung bestimmen. Für die Menge $\{a, b, c\}$ folgt daraus beispielsweise $a < b < c$.

Obwohl alle freien abelschen Gruppen, deren Basen die gleiche Kardinalität haben, auf denselben Vektorraum abgebildet werden, geht dabei keine wesentliche Information verloren. Die Morphismen zwischen zwei Gruppen entsprechen in Anzahl und Eigenschaften den Matrizen zwischen den entsprechenden Vektorräumen, d.h. eingeschränkt auf diese Gruppen ist die Abbildung $(-)^{\mathcal{M}}$ bijektiv.

5.2.14 Satz (Funktor):

Die Abbildung $(-)^{\mathcal{M}}$ ist ein Funktor von \mathcal{AG}_f in \mathcal{VR} mit den Eigenschaften „treu“ und „voll“.

□

(Beweis \rightarrow Anhang S. 262)

Umgekehrt lassen sich auch Matrizen auf Morphismen in den freien abelschen Gruppen abbilden. Dabei muß außer der Matrix auch die Domäne und die Codomäne des resultierenden Morphismus als Parameter übergeben werden, da jeweils eine Matrix für eine Menge von Morphismen mit gleichem Verhalten, aber verschiedenen Anfangs- und Endpunkten steht. Die Abbildung von Morphismen in Matrizen und ihr Gegenstück verhalten sich invers zueinander, so daß ihre Hintereinanderausführung wieder den ursprünglichen Morphismus ergibt.

5.2.15 Definition (Abbildung von Matrizen in Morphismen):

Wir definieren eine partielle Abbildung

$$Mp: \text{Mor}(\mathcal{VR}) \times \text{Obj}(\mathcal{AG}_f) \times \text{Obj}(\mathcal{AG}_f) \rightarrow \text{Mor}(\mathcal{AG}_f)$$

für alle Mengen $A = \{a_1, \dots, a_n\}$, $B = \{b_1, \dots, b_m\}$, a_j aus A und $m \times n$ -Matrizen $M = (m_{i,j})$ durch:

$$Mp(M, A^\otimes, B^\otimes) \stackrel{\text{def}}{=} f: A^\otimes \rightarrow B^\otimes \quad \text{mit} \quad f(a_j) = \bigoplus_{i=1}^m m_{i,j} * b_i.$$

□

5.2.16 Korollar (Mp und $(-)^{\mathcal{M}}$):

Für alle freien abelschen Gruppen A^\otimes, B^\otimes , Gruppenmorphismen $f: A^\otimes \rightarrow B^\otimes$ und $|B| \times |A|$ -Matrizen M gilt:

$$Mp\left((f)^{\mathcal{M}}, A^\otimes, B^\otimes\right) = f \quad \text{und} \quad (Mp(M, A^\otimes, B^\otimes))^{\mathcal{M}} = M.$$

□

Die Kategorie der endlichen Mengen \mathcal{SET}_f ist, wie die Kategorie der Mengen allgemein, nicht selbstdual. Dementsprechend existieren mit dem kartesischen Produkt und der disjunkten Vereinigung zu jedem endlichen diskreten Diagramm zwei unterschiedliche Mengen als Produkt und Coprodukt (\rightarrow 5.1.14, 5.1.16). Im Gegensatz dazu fallen die Objekte dieser dualen Konstruktionen bei den selbstdualen freien abelschen Gruppen zusammen. Sie werden von der freien abelschen Gruppe über dem Coprodukt der Basismengen gebildet. Die Limes- bzw. Colimesmorphismen sind frei über den entsprechenden Mengenabbildungen konstruiert und zueinander dual.

5.2.17 Lemma (Produkte):

Sei \mathcal{S} eine endliche, diskrete Kategorie, \mathcal{C} eine der in 5.2.6 definierten Kategorien freier abelscher Gruppen und $\mathcal{D}: \mathcal{S} \rightarrow \mathcal{C}$ ein Funktor, dann bildet

$$\left(\left(\bigoplus_{s \in \text{Obj}(\mathcal{S})} \text{Base}(\mathcal{D}(s)) \right)^\otimes, (\ell_s)_{s \in \mathcal{S}} \right)$$

einen als „*endliches Produkt*“ bezeichneten Limes von \mathcal{D} , dessen universelle Abbildungen ℓ_s sich aus:

$$\ell_s(a_s) = a$$

für alle Objekte s in \mathcal{S} , a_s in der disjunkten Vereinigung und a in der Basis von $\mathcal{D}(s)$ ergeben.

□

(Beweis \rightarrow Anhang S. 263)

Da wir im weiteren nur endliche Produkte und Coprodukte betrachten, lassen wir, wenn Verwechslungen ausgeschlossen sind, das Adjektiv „endlich“ weg und reden nur noch von Produkten und Coprodukten.

Bei der Berechnung von Equalizern (\rightarrow 5.1.14) können wir uns den Zusammenhang von \mathcal{AG}_f -Morphismen zu Matrizen nutzbar machen. Überträgt man die Eigenschaft $f \circ \ell = g \circ \ell$ des gesuchten Morphismus ℓ im Equalizer zum Diagramm

$$A^{\otimes} \begin{array}{c} \xrightarrow{f} \\ \xrightarrow{g} \end{array} B^{\otimes}$$

mit Hilfe von $(-)^{\mathcal{M}}$ (\rightarrow 5.2.13) auf Matrizen, so erhält man

$$(f)^{\mathcal{M}} * (\ell)^{\mathcal{M}} = (g)^{\mathcal{M}} * (\ell)^{\mathcal{M}} \quad \text{oder} \quad \left((f)^{\mathcal{M}} - (g)^{\mathcal{M}} \right) * (\ell)^{\mathcal{M}} = 0.$$

Es wird demnach die Matrix gesucht, deren Spaltenvektoren die Basis des Kerns der Differenzmatrix $(f)^{\mathcal{M}} - (g)^{\mathcal{M}}$ bilden. Aufgrund der Eigenschaften von $(-)^{\mathcal{M}}$ (\rightarrow 5.2.14) existiert eine solche Basis genau dann, wenn ein Equalizer zum Diagramm existiert.

Die ganzen Zahlen \mathbb{Z} , über denen die Vektorräume und Matrizen in \mathcal{VR} gebildet sind, formen wegen des Fehlens von Inversen bezüglich der Multiplikation nur einen Ring und keinen Körper wie beispielsweise \mathbb{Q} . Aus diesem Grund existieren, anders als zu Vektorräumen über Körpern, Matrizen in \mathcal{VR} , deren Kern keine Basis über \mathbb{Z} besitzt, wie das folgenden Beispiel zeigt.

5.2.18 Beispiel (Kern von Matrizen über \mathbb{Z} ohne Basis):

Mit den folgenden drei Schritten läßt sich zu jeder Matrix M eine Lösungsmatrix L mit der Eigenschaft $M * L = 0$ und einem Rang, der der Anzahl der linear unabhängigen Lösungen entspricht, bestimmen. Die Spaltenvektoren von L bilden allerdings nicht in jedem Fall eine Basis für den Kern von M .

- (1) Konvertieren von M in eine Matrix über dem Körper \mathbb{Q} und Berechnen der Treppennormalform, z.B. nach Gauß.
- (2) Berechnen der Lösung über \mathbb{Q} , z.B. nach Gauß-Jordan.
- (3) Konvertieren der Lösungsmatrix nach \mathbb{Z} , durch Bildung des „kleinsten gemeinsamen Vielfachen“ über den Koordinaten der jeweiligen Spaltenvektoren.

$$M = \begin{pmatrix} 2 & -1 & -1 \end{pmatrix} \xrightarrow{(1)} \begin{pmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \end{pmatrix} \xrightarrow{(2)} \begin{pmatrix} \frac{1}{2} & \frac{1}{2} \\ 1 & 0 \\ 0 & 1 \end{pmatrix} \xrightarrow{(3)} \begin{pmatrix} 1 & 1 \\ 2 & 0 \\ 0 & 2 \end{pmatrix} = L$$

Die Spaltenvektoren der resultierenden Matrix L enthalten nur Koordinaten aus \mathbb{Z} und bilden eine Basis über \mathbb{Q} zum Kern von M . Betrachte man jedoch den Spaltenvektor v aus dem Kern, dessen Wert an allen Koordinaten Eins beträgt, dann existiert kein Zeilenvektor x in \mathbb{Z}^2 mit der Eigenschaft $L * x = v$, d.h. L ist keine Basis über \mathbb{Z} .

Die hier verwendete Matrix M entspricht unter anderem der Differenz $(f)^{\mathcal{M}} - (g)^{\mathcal{M}}$ der Matrizen zu den \mathcal{AG}_f -Morphismen

$$\begin{array}{ll} f: a_1 \mapsto 2 * c & g: a_1 \mapsto 0 \\ a_2 \mapsto 0 & a_2 \mapsto c \\ a_3 \mapsto 0 & a_3 \mapsto c \end{array} \quad \square$$

Das in 5.2.18 verwendete Verfahren ergibt für jede Matrix M aus \mathcal{VR} eine Lösungsmatrix L zur Matrixgleichung $M * L = 0$. Wie im Beispiel gezeigt, bilden deren Spaltenvektoren aber nicht notwendigerweise eine Basis zum Kern von M . Günstiger ist die Situation bei Lösungsmatrizen¹¹ deren Aufbau dem in Abb. 5.1 gezeigten Schema entspricht, hier läßt sich für jeden Vektor v der Vektor $x = (v_{j_1}, \dots, v_{j_r})$ mit

$$L = (l_{i,j}) = \begin{pmatrix} l_{1,1} & l_{2,1} & \cdots & l_{r,1} \\ \vdots & \vdots & \cdots & \vdots \\ l_{1,j_1-1} & l_{2,j_1-1} & \cdots & l_{r,j_1-1} \\ 1 & 0 & \cdots & 0 \\ l_{1,j_1+1} & l_{2,j_1+1} & \cdots & l_{r,j_1+1} \\ \vdots & \vdots & \cdots & \vdots \\ l_{1,j_2-1} & l_{2,j_2-1} & \cdots & l_{r,j_2-1} \\ 0 & 1 & \cdots & 0 \\ l_{1,j_2+1} & l_{2,j_2+1} & \cdots & l_{r,j_2+1} \\ l_{1,j_r-1} & l_{2,j_r-1} & \cdots & l_{r,j_r-1} \\ 0 & 0 & \cdots & 1 \\ l_{1,j_r+1} & l_{2,j_r+1} & \cdots & l_{r,j_r+1} \\ \vdots & \vdots & \cdots & \vdots \\ l_{1,n} & l_{2,n} & \cdots & l_{r,n} \end{pmatrix}$$

Abbildung 5.1: Schema einer Lösungsmatrix für die Matrixgleichung $M * L = 0$.

$L * x = v$ direkt aus den Koordinaten von v zusammensetzen.

Das Problem verschärft sich, wenn Equalizer in der Kategorie der freien abelschen Gruppen mit monotonen Morphismen \mathcal{AG}_{fm} gesucht werden. In diesem Fall muß die Lösungsmatrix, wenn sie denn eine Basis bildet, zusätzlich noch frei von negativen Einträgen sein.

¹¹Wie sie beispielsweise bei Morphismen auftreten, deren Bilder ausschließlich aus formalen Summen mit Koeffizienten größer oder gleich Eins bestehen.

5.2.19 Lemma (Existenz von Equalizern):

Sei \mathcal{S} die von dem Graphen $s \xrightleftharpoons[m]{m'} s'$ generierte Kategorie, $\mathcal{D}: \mathcal{S} \rightarrow \mathcal{AG}_f$ ein Funktor,

$$|Base(\mathcal{D}(s))| = n, \quad r = n - Rang \left((\mathcal{D}(m))^{\mathcal{M}} - (\mathcal{D}(m'))^{\mathcal{M}} \right),$$

$L = (l_{u,v})$ eine $n \times r$ -Matrix über \mathbb{Z} mit

$$\left((\mathcal{D}(m))^{\mathcal{M}} - (\mathcal{D}(m'))^{\mathcal{M}} \right) * L = 0$$

und es existiere eine Folge $1 \leq j_1 \leq \dots \leq j_r \leq n$, so daß für alle $i, p = 1, \dots, r$, $p \neq i$ gilt $l_{j_i, i} = 1$ und $l_{j_i, p} = 0$, dann bildet

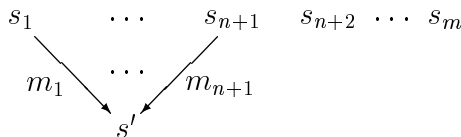
$$(E^{\otimes}, Mp(L, E^{\otimes}, \mathcal{D}(s)))^{12} \text{ mit } E = \{e_1, \dots, e_r\}$$

einen als „Equalizer“ bezeichneten Limes für \mathcal{D} in \mathcal{AG}_f .

Sind darüber hinaus $\mathcal{D}(m), \mathcal{D}(m')$ monoton und ist L eine Matrix über \mathbb{N} , dann bildet der Equalizer einen Limes für \mathcal{D} in \mathcal{AG}_{fm} . □

(Beweis \rightarrow Anhang S. 265)

Wir verwenden in den übrigen Kapiteln dieser Arbeit im Zusammenhang mit Limeskonstruktionen häufig Diagramme der Form



Sie erlauben es, die Objekte der oberen Reihe zusammenzufügen und das Ergebnis bezüglich der Verbindung zum unteren Objekt zu faktorisieren. Im Fall der freien abelschen Gruppen werden beim Faktorisieren die Elemente zusammengefaßt, die ein gemeinsames Bild in der unteren Gruppe haben. Satz 5.1.19 zeigt, wie diese Limes aus Produkten und Equalizern konstruiert werden können. Da wir aber in 5.2.18 schon gezeigt haben, daß speziell in \mathcal{AG}_{fm} nicht alle Equalizer existieren, kann man davon ausgehen, daß dies auch für Limes zu Diagrammen der gezeigten Form gilt. Das nachfolgende Beispiel belegt diese Annahme.

5.2.20 Beispiel (Diagramm ohne Limes in \mathcal{AG}_{fm}):

Seien $f: \{a_1, a_2\}^{\otimes} \rightarrow \{c\}^{\otimes}$, $g: \{b_1, b_2\}^{\otimes} \rightarrow \{c\}^{\otimes}$ zwei Morphismen in \mathcal{AG}_f mit

$$\begin{array}{ll}
 f: a_1 \mapsto c & g: b_1 \mapsto c \\
 a_2 \mapsto c & b_2 \mapsto c \\
 \vdots & \vdots
 \end{array}$$

¹²Da sich die universellen Abbildung $\ell_{s'}$ im Falle von Equalizern eindeutig aus ℓ_s ergibt, läßt man sie meist weg und schreibt (E, ℓ) anstelle von $(E, (\ell_s)_{s \in Obj(\mathcal{S})})$.

Zusammen bilden sie ein Diagramm der besprochenen Form. Das Produkt der Diagrammobjekte und das Produkt der Codomänen von f und g ¹³ ergibt sich nach Lemma (\rightarrow 5.2.17) zu $\{a_1, a_2, b_1, b_2, c\}^{\otimes}$ resp. zu $\{c_f, c_g\}^{\otimes}$.¹⁴

Damit sind auch die Morphismen

$$\begin{array}{l} \widehat{\ell}_p: a_1 \mapsto c_f \quad \widehat{\ell}_o: a_1 \mapsto 0 \\ a_2 \mapsto c_f \quad a_2 \mapsto 0 \\ b_1 \mapsto c_g \quad b_1 \mapsto 0 \\ b_2 \mapsto c_g \quad b_2 \mapsto 0 \\ c \mapsto 0 \quad c \mapsto c_f \oplus c_g \end{array} \quad \Longrightarrow \quad \left((\widehat{\ell}_p)^{\mathcal{M}} - (\widehat{\ell}_o)^{\mathcal{M}} \right) = \begin{pmatrix} 1 & 1 & 0 & 0 & -1 \\ 0 & 0 & 1 & 1 & -1 \end{pmatrix}$$

(\rightarrow 5.1.19) festgelegt, deren Equalizer sich in den Limes des Diagramms umformen läßt. Mit den im Beispiel 5.2.18 gezeigten Schritten erhält man dazu die Lösungsmatrix

$$\begin{pmatrix} -1 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & -1 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (\text{Lösungen zu } \left((\widehat{\ell}_p)^{\mathcal{M}} - (\widehat{\ell}_o)^{\mathcal{M}} \right) * x = 0)$$

die, da sie negative Einträge enthält, nicht in einen monotonen Morphismus übertragen werden kann. Durch Spaltentransformationen läßt sich dieser Mangel zwar beseitigen, aus den resultierenden Matrizen kann aber keine Folge mit den in 5.2.19 geforderten Eigenschaften abgeleitet werden.

Ersetzt man den Morphismus f durch $f': \{a_1\}^{\otimes} \rightarrow \{c\}^{\otimes}$ mit $f'(a_1) = c$, erhält man als Lösung die links stehende Matrix, die sich durch Spaltentransformation in die rechtsstehende Matrix umwandeln läßt.

$$\begin{pmatrix} 0 & 1 \\ -1 & 1 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} \quad \longrightarrow \quad \begin{pmatrix} 1 & 1 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{pmatrix}$$

Diese Matrix erfüllt alle Kriterien in 5.2.19 und kann in einen Equalizer umgewandelt werden. Daß der Equalizer in diesem Fall existiert, liegt daran, daß f' im Gegensatz zu f injektiv und generiert ist.

□

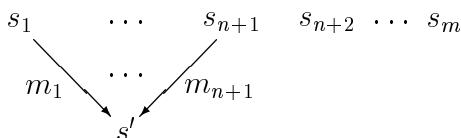
¹³Genau genommen müßten auch die Identitäten berücksichtigt werden. Da sich die dadurch entstehenden Elemente in den nachfolgenden Berechnungen gegeneinander aufheben, sind sie hier weggelassen.

¹⁴Da f, g die gleiche Codomäne haben, taucht $\{c\}^{\otimes}$ aufgrund der disjunkten Vereinigung der Basismengen über die Produkte gebildet werden (\rightarrow 5.2.17), im Produkt der Codomänen zweimal auf.

Da Diagramme der gezeigten Form im Rahmen dieser Arbeit eine besondere Rolle spielen, geben wir im nachfolgenden Lemma eine Bedingung an, unter der deren Limes existieren, und zeigen, wie sie gebildet werden. Anschließend veranschaulichen wir diese Konstruktion anhand eines Beispiels.

5.2.21 Lemma (Limes spezieller Diagramme):

Sei \mathcal{S} die von dem Graphen



$(n, m \in \mathbb{N})$ generierte Kategorie, $\mathcal{D}: \mathcal{S} \rightarrow \mathcal{AG}_{fm}$ ein Funktor mit

$$\mathcal{D}(s_i) = A_i^\otimes, \quad \mathcal{D}(s') = B^\otimes, \quad \mathcal{D}(m_i) = f_i$$

und $(g_i: A_i \setminus \text{Kern}(f_i) \rightarrow B)_{i=1, \dots, n}$ ¹⁵ eine Familie von Bijektionen für die

$$f_i \Big|_{\text{dom}(g_i^\otimes)}^B = g_i^\otimes$$

gilt, dann ergibt sich der Limes zu \mathcal{D} in \mathcal{AG}_{fm} aus der $((n+1) * |B| + r) \times r$ -Matrix

$$L \stackrel{\text{def}}{=} \begin{pmatrix} (f_{n+1})^{\mathcal{M}} & | & 0 \\ \vdots & & \\ (f_{n+1})^{\mathcal{M}} & | & 0 \\ \hline E_r \end{pmatrix} \quad \text{mit} \quad r = \sum_{i=1}^n |A_i \cap \text{Kern}(f_i)| + \sum_{i=n+1}^m |A_i|$$

wobei E_r die $r \times r$ -Einheitsmatrix und 0 die Nullmatrix bezeichnet, zu:

$$\left(E^\otimes, (\ell_s \circ \ell)_{s \in \text{Obj}(\mathcal{S})} \right) \quad \text{mit} \quad E \stackrel{\text{def}}{=} \{e_1, \dots, e_r\}, \quad \ell \stackrel{\text{def}}{=} Mp \left(L, E^\otimes, \prod_{s \in \text{Obj}(\mathcal{S})} \mathcal{D}(s) \right)$$

mit ℓ_s als universeller Abbildung vom Produkt

$$\begin{aligned}
 \prod_{s \in \text{Obj}(\mathcal{S})} \mathcal{D}(s) = & \{b_1, \dots, b_{|B|}, \\
 & a_1^1, \dots, a_{|B|}^1, \dots, a_1^n, \dots, a_{|B|}^n, \\
 & a_1^{n+1}, \dots, a_{|A_{n+1}|}^{n+1}, \dots, a_1^m, \dots, a_{|A_m|}^m, \\
 & a_{|B|+1}^1, \dots, a_{|A_1|}^1, \dots, a_{|B|+1}^n, \dots, a_{|A_n|}^n \}^\otimes
 \end{aligned}$$

in die Gruppe $\mathcal{D}(s)$. Darin bezeichnet b_j das j -te Element der Menge B und a_j^i das j -te Element der Menge A_i , so daß $g_i(a_j^i) = b_j$ für $j < |B|$ gilt. □

(Beweis \rightarrow Anhang S. 266)

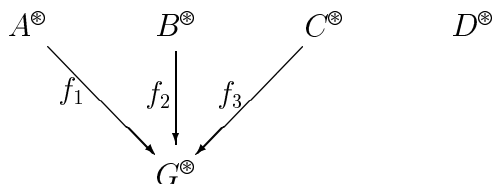
Lemma 5.2.21 stützt sich auf die in 5.1.19 beschriebene Limeskonstruktion und auf die in 5.2.19 angegebenen Bedingungen für die Existenz von Equalizern in \mathcal{AG}_{fm} . Schränkt man die f_i mit i kleiner n auf Elemente ein, die nicht zum Kern gehören, dann führt ihre Transformation in Matrizen jeweils zu einer $|B| \times |B|$ -Einheitsmatrix.¹⁶ Bei geeigneter Anordnung der Elemente in den beiden Produkten¹⁷ aus 5.1.19 ergibt sich daraus die unten links stehende Matrix als Differenz der Matrizen zu $\widehat{\ell}_p$ und $\widehat{\ell}_o$. Die Matrix unten rechts entspricht der Lösungsmatrix L aus 5.2.21.

$$\left(\begin{array}{c|c|c|c} \frac{-E_{|B|}}{\vdots} & E_{n*|B|} & 0 & 0 \\ \hline \frac{-E_{|B|}}{\vdots} & 0 & (f_{n+1})^{\mathcal{M}} & \end{array} \right) * \left(\begin{array}{c|c} (f_{n+1})^{\mathcal{M}} & 0 \\ \hline \vdots & \\ \hline (f_{n+1})^{\mathcal{M}} & 0 \\ \hline E_r & \end{array} \right) = 0$$

Die Multiplikation beider Matrizen ergibt offensichtlich die Nullmatrix. Darüber hinaus erfüllt L auch alle anderen Bedingungen aus 5.2.19, so daß ihre Transformation einen Equalizer zu $\widehat{\ell}_p, \widehat{\ell}_o$ ergibt. Nach 5.1.19 läßt sich in diesem Fall daraus der Limes des Diagramms gewinnen.

5.2.22 Beispiel (Limeskonstruktion):

Gegeben das folgende Diagramm \mathcal{D}



mit $A = \{a_1, a_2\}$, $B = \{b_1, b_2, b_3\}$, $C = \{c_1, c_2, c_3\}$, $D = \{d_1, d_2, d_3\}$, $G = \{g_1, g_2\}$ und

$$\begin{array}{lll} f_1: a_1 \mapsto g_1 & f_2: b_1 \mapsto g_1 & f_3: c_1 \mapsto 5 * g_1 \\ & a_2 \mapsto g_2 & b_2 \mapsto g_2 & c_2 \mapsto 4 * g_1 \oplus g_2 \\ & & b_3 \mapsto 0 & c_3 \mapsto 3 * g_1 \oplus 7 * g_2 \end{array}$$

dann bildet die linke der nachfolgenden freien abelschen Gruppen ein Produkt der Diagrammobjekte, dessen Elemente in der von Lemma 5.2.21 vorgegebenen Ordnung stehen, und die rechte Gruppe ein Produkt der f_i -Codomänen.¹⁸

$$\{g_1, g_2, a_1, a_2, b_1, b_2, c_1, c_2, c_3, d_1, d_2, d_3, b_3\}^{\otimes} \quad \{g_{1,1}, g_{1,2}, g_{1,3}, g_{2,1}, g_{2,2}, g_{2,3}\}^{\otimes}$$

¹⁵ $Kern(f)$ bezeichnet den Kern einer Funktion, d.h. im Falle von Gruppenmorphisimen, die Teilmenge der Domäne, die auf das neutrale Element abgebildet wird.

¹⁶Auf den Elementen außerhalb des Kerns gilt $f_i = g_i^{\otimes}$, d.h. f_i ist auf diesen Elementen bijektiv!

¹⁷die Anordnung des linken Produkts ist in 5.2.21 schon vorgegeben

¹⁸Darin steht $g_{i,j}$ für das i -te Element von G in der von der Codomäne des Morphismus f_j erzeugten Kopie.

Die Eindeutigkeit des rechten Produkts und die Morphismenfamilien

$$(f_i \circ \ell_{\text{dom}(m_i)})_{1 \leq i \leq 3}, \quad (\ell_{\text{cod}(m_i)})_{1 \leq i \leq 3}$$

(ℓ_{s_i} bezeichnet den universellen Morphismus vom linken Produkt in das entsprechende Diagrammobjekt), die zusammen mit dem linken Produkt eine Quelle (\rightarrow 5.1.11) für die Codomänen der Diagrammorphismen bilden, impliziert in diesem Fall zwei Morphismen $\widehat{\ell}_p, \widehat{\ell}_o$ vom linken ins rechte Produkt (\rightarrow Abb. 5.2). Die Differenz der

$$\begin{array}{ll} \widehat{\ell}_p: a_1 \mapsto g_{1,1} & \widehat{\ell}_o: a_1 \mapsto 0 \\ a_2 \mapsto g_{2,1} & a_2 \mapsto 0 \\ b_1 \mapsto g_{1,2} & b_1 \mapsto 0 \\ b_2 \mapsto g_{2,2} & b_2 \mapsto 0 \\ b_3 \mapsto 0 & b_3 \mapsto 0 \\ c_1 \mapsto 5 * g_{1,3} & c_1 \mapsto 0 \\ c_2 \mapsto 4 * g_{1,3} \oplus g_{2,3} & c_2 \mapsto 0 \\ c_3 \mapsto 3 * g_{1,3} \oplus 7 * g_{2,3} & c_3 \mapsto 0 \\ d_1 \mapsto 0 & d_1 \mapsto 0 \\ d_2 \mapsto 0 & d_2 \mapsto 0 \\ d_3 \mapsto 0 & d_3 \mapsto 0 \\ g_1 \mapsto 0 & g_1 \mapsto g_{1,1} \oplus g_{1,2} \oplus g_{1,3} \\ g_2 \mapsto 0 & g_2 \mapsto g_{2,1} \oplus g_{2,2} \oplus g_{2,3} \end{array}$$

Abbildung 5.2: Die Morphismen $\widehat{\ell}_p$ und $\widehat{\ell}_o$.

Matrizen zu $\widehat{\ell}_p$ und $\widehat{\ell}_o$ ergibt die in Abb. 5.3 links stehende Matrix, während die rechte Matrix eine Basis des Kerns der linken bildet. Wandelt man die rechte Matrix aus Abb. 5.3 unter Berücksichtigung der Ordnung auf den Elementen des Produkts der Diagrammobjekte in einen \mathcal{AG}_{fm} -Morphismus, erhält man ℓ mit

$$\begin{array}{l} \ell: e_1 \mapsto 5 * g_1 \oplus 5 * a_1 \oplus 5 * b_1 \oplus c_1 \\ e_2 \mapsto 4 * g_1 \oplus g_2 \oplus 4 * a_1 \oplus a_2 \oplus 4 * b_1 \oplus b_2 \oplus c_2 \\ e_3 \mapsto 3 * g_1 \oplus 7 * g_2 \oplus 3 * a_1 \oplus 7 * a_2 \oplus 3 * b_1 \oplus 7 * b_2 \oplus c_3 \\ e_4 \mapsto d_1 \\ e_5 \mapsto d_2 \\ e_6 \mapsto d_3 \\ e_7 \mapsto b_3 \end{array}$$

$$\left(\begin{array}{cc|cc|ccc|cccc} -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline -1 & 0 & 0 & 0 & 0 & 0 & 5 & 4 & 3 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 1 & 7 & 0 & 0 & 0 & 0 \end{array} \right) * \left(\begin{array}{ccc|cccc} 5 & 4 & 3 & 0 & 0 & 0 & 0 \\ 0 & 1 & 7 & 0 & 0 & 0 & 0 \\ \hline 5 & 4 & 3 & 0 & 0 & 0 & 0 \\ 0 & 1 & 7 & 0 & 0 & 0 & 0 \\ \hline 5 & 4 & 3 & 0 & 0 & 0 & 0 \\ 0 & 1 & 7 & 0 & 0 & 0 & 0 \\ \hline 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array} \right) = 0$$

Abbildung 5.3: Produkt aus der Matrix zu $\widehat{\ell}_p - \widehat{\ell}_o$ und der Basis des Kerns.

Mit Hilfe des Zusammenhangs $\ell_{s_i}^e = \ell_{s_i} \circ \ell$ lässt sich ℓ in die universellen Abbildungen zu den Diagrammobjekten zerlegen,

$$\begin{array}{lll} \ell_{s_1}^e : e_1 \mapsto 5 * a_1 & \ell_{s_4}^e : e_1 \mapsto 0 & \ell_{s'}^e : e_1 \mapsto 5 * g_1 \\ e_2 \mapsto 4 * a_1 \oplus a_2 & e_2 \mapsto 0 & e_2 \mapsto 4 * g_1 \oplus g_2 \\ e_3 \mapsto 3 * a_1 \oplus 7 * a_2 & e_3 \mapsto 0 & e_3 \mapsto 3 * g_1 \oplus 7 * g_2 \\ e_4 \mapsto 0 & \dots & e_4 \mapsto 0 \\ e_5 \mapsto 0 & e_5 \mapsto d_1 & e_5 \mapsto 0 \\ e_6 \mapsto 0 & e_6 \mapsto d_2 & e_6 \mapsto 0 \\ e_7 \mapsto 0 & e_7 \mapsto d_3 & e_7 \mapsto 0 \end{array}$$

die zusammen mit der Gruppe $\{e_1, \dots, e_7\}^{\otimes}$ den Limes zu \mathcal{D} in \mathcal{AG}_{fm} bilden.

□

Mit Hilfe dieses Zusammenhangs und der Dualität von Limes und Colimes, lassen sich durch Dualisierung aus den schon bekannten Limeskonstruktionen Colimes erzeugen. Dazu werden die Diagramme zunächst dualisiert, dann der Limes berechnet und anschließend die darin enthaltenen universellen Morphismen durch Dualisieren in die couniversellen Morphismen des Colimes verwandelt. Die Objekte in Limes und Colimes sind gleich. Alternativ ließe sich eine direkte Colimeskonstruktion aus der Dualisierung der Lemmata 5.2.19 und 5.2.21 ableiten. Da Colimes innerhalb der freien abelschen Gruppen in den Kapiteln 1–3 eine wesentliche Rolle spielen, gehen wir im folgenden auf einige Eigenschaften der resultierenden Objekte und Morphismen näher ein.

Alle Colimes zu endlichen Diagrammen haben zwei Eigenschaften gemeinsam, erstens ist jedes Element ihres Objekts von einem Element in einem der Diagrammobjekte generiert und zweitens existiert nur dann ein gemeinsames Objekt im Bild

zweier Colimesmorphismen, wenn die Urbilder über eine „Kette“ von Morphismen miteinander in Verbindung stehen. Diese Eigenschaften spielen in den übrigen Kapiteln eine wichtige Rolle bei der Übertragung anwendungsspezifischer Eigenschaften von den Diagrammobjekten auf das Colimesobjekt und werden daher im nachfolgenden Lemma noch einmal explizit formuliert. Ihre dualen Entsprechungen treffen auf alle Limites endlicher Diagramme zu.

5.2.23 Lemma (Colimeseigenschaften):

Sei \mathcal{S} eine endliche Kategorie, $\mathcal{D}: \mathcal{S} \rightarrow \mathcal{AG}_{fm}$ ein Funktor mit Colimes

$$((c_s)_{s \in \mathcal{S}}, C^{\otimes}),$$

A_s die Basis von $\mathcal{D}(s)$ für alle s in \mathcal{S} , P die disjunkte Vereinigung aller A_s ,
 $\rightarrow \subseteq P \times P$ eine Relation mit

$$\begin{aligned} \forall s, s' \in \text{Obj}(\mathcal{S}), a_s \in A_s, a'_{s'} \in A_{s'}: \\ a_s \rightarrow a'_{s'} \iff \exists m: s \rightarrow s' \in \text{Mor}(\mathcal{S}): \mathcal{D}(m)(a_s)(a'_{s'}) > 0 \end{aligned}$$

und \rightleftharpoons der symmetrische, transitive Abschluß von \rightarrow , dann gilt:

$$(I) \forall c \in C, \exists s \in \mathcal{S}, a_s \in A_s: c_s(a_s)(c) > 0$$

$$(II) \forall c \in C, s, s' \in \text{Obj}(\mathcal{S}), a_s \in A_s, a'_{s'} \in A_{s'}: \\ c_s(a_s)(c) > 0 \wedge c_{s'}(a'_{s'})(c) > 0 \implies a_s \rightleftharpoons a'_{s'}$$

□

(Beweis \rightarrow Anhang S. 274)

5.3 Spezifikationen und Modelle

Im folgenden Abschnitt führen wir einen kategoriellen Rahmen ein, in dem sich formale Objektbeschreibungen, im weiteren als „Spezifikationen“ bezeichnet, untereinander und mit ihren Semantiken in Beziehung setzen lassen. Die dabei verwendeten Definitionen sind, in notationell leicht abgewandelter Form, aus [EBO91]¹⁹ übernommen.

Innerhalb eines „Spezifikationsrahmens“ sind die Spezifikationen Objekte einer Kategorie, ihre Beziehungen werden über Morphismen ausgedrückt. Wie die Spezifikationen realisiert sind, wird nicht weiter festgelegt. Als Semantik ist jeder Spezifikation eine Kategorie zugeordnet, die wir als „Modellkategorie“ bezeichnen. Abgesehen davon unterliegt auch die Modellbildung keinerlei Einschränkungen.

Den Spezifikationsmorphismen entsprechen auf der Modellebene Funktoren zwischen Modellkategorien. Sie bilden die Modelle und Modellmorphismen zur Zielspezifikation

¹⁹Spezifikationsrahmen werden dort als Spezifikationslogiken bezeichnet.

eines Spezifikationsmorphisms in die Modellkategorie der Ausgangsspezifikation ab, d.h. der Funktor verläuft, verglichen mit dem ihn generierenden Spezifikationsmorphimus, in entgegengesetzter Richtung. Dahinter steht die Vorstellung, daß Spezifikationen, die Ziel eines Spezifikationsmorphisms sind, Objekte beschreiben, die neben den Eigenschaften der Ausgangsspezifikationen noch weitere Aspekte abdecken. Da die Modelle dieser größeren Spezifikationen alle darin geforderten Eigenschaften erfüllen, also auch die der Ausgangsspezifikationen, können sie mit deren Modellen in Beziehung gesetzt werden.

Schon auf diesem sehr abstrakten Niveau können mit Hilfe der in Abschnitt 5.1 beschriebenen Konstruktionen Ergebnisse erzielt werden. Diese Ergebnisse sind aufgrund der Abstraktion von allen Realisierungsdetails auf sehr unterschiedliche Spezifikationstechniken, wie Petrinetze (\rightarrow Kapitel 2, 3) oder algebraische Datentypspezifikationen (\rightarrow Kapitel 6) anwendbar.

5.3.1 Definition (Spezifikationsrahmen):

Ein „Spezifikationsrahmen“ \mathcal{SF} ist ein Paar $(\mathcal{C}, \mathcal{M})$ bestehend aus einer Kategorie \mathcal{C} (den Spezifikationen) und einem Funktor

$$\mathcal{M}: \mathcal{C}^{op} \rightarrow \mathcal{CATCAT}$$

der jedem \mathcal{C} -Morphismus $f: C \rightarrow C'$ einen Vergißfunktor

$$\mathcal{M}(f): \mathcal{M}(C') \rightarrow \mathcal{M}(C)$$

zuordnet. □

Für einen gegebenen Spezifikationsrahmen $(\mathcal{C}, \mathcal{M})$ notieren wir die von \mathcal{C} -Morphismen generierten Funktoren, indem wir \mathcal{M} mit dem Morphismenbezeichner indizieren, d.h. wir schreiben \mathcal{M}_h anstelle von $\mathcal{M}(h)$ für einen Morphismus h aus \mathcal{C} .

Ausgehend von einem Spezifikationsrahmen können die in den unterschiedlichen Modellkategorien separierten Modelle und Morphismen zu einer Kategorie aller Modelle zusammengefaßt werden, ohne das dabei Information verloren geht. Um dies zu gewährleisten bestehen die Morphismen dieser Kategorie aus einem Spezifikations- und einem Modellmorphismus. Ersterer setzt die Spezifikationen zu denen die Modelle gehören in Beziehung, während der Modellmorphismus das Modell in seinem Ursprung mit dem Bild des Zielmodells unter \mathcal{M} verknüpft, d.h. er gehört zur Modellkategorie der Ursprungsspezifikation.

Diese, die Grenzen der Modellkategorien überschreitende Form der Modellmorphisms, bezeichnen wir als „generalisierte Modellmorphisms“ oder einfach als generalisierte Morphisms und dementsprechend die von einem gegebenen Spezifikationsrahmen generierte Kategorie aller Modelle als „generalisierte Modellkategorie“. Modellmorphisms die innerhalb der Kategoriegrenzen abbilden, sind als Spezialfälle

generalisierter Morphismen mit Identitätsmorphismen als Spezifikationsabbildung in diesen Rahmen eingebettet.

5.3.2 Definition (Generalisierte Modellkategorie):

Sei $\mathcal{SF} = (\mathcal{C}, \mathcal{M})$ ein Spezifikationsrahmen, dann definieren wir die \mathcal{SF} -generierte generalisierte Modellkategorie \mathcal{GSF} durch:

$$(I) \text{ Obj}(\mathcal{GSF}) \stackrel{\text{def}}{=} \bigsqcup_{D \in \text{Obj}(\mathcal{C})} \text{Obj}(\mathcal{M}(D)),$$

$$(II) \forall A, B \in \text{Obj}(\mathcal{GSF}): \\ \text{Mor}(\mathcal{GSF})[A, B] \stackrel{\text{def}}{=} \{(h, f) \in \text{Mor}(\mathcal{C}) \times \text{Mor}(\mathcal{M}(\text{dom}(h))) \mid \\ \text{dom}(f) = A \wedge \text{cod}(f) = \mathcal{M}_h(B)\}$$

und

$$(III) \text{ f\u00fcr alle komponierbaren Morphismen } (h, f), (g, k) \text{ aus } \mathcal{GSF} \\ (h, f) \circ (g, k) \stackrel{\text{def}}{=} (h \circ g, \mathcal{M}_g(f) \circ k).$$

□

Das nachfolgende Beispiel veranschaulicht die Komposition $(h_2, f_2) \circ (h_1, f_1)$ zweier generalisierter Modellmorphisme. Dabei werden die enthaltenen Spezifikationsmorphisme auf die \u00fcbliche Weise, d.h. als Funktionen komponiert. Der Modellmorphimus f_2 mu\u00df dagegen zun\u00e4chst durch \mathcal{M}_{h_1} in einen C_1 -Modellmorphimus abgebildet werden, bevor eine Komposition mit f_1 in der Kategorie der C_1 -Modelle m\u00f6glich ist.

$$\begin{array}{ccccc}
 C_1 & \xrightarrow{h_1} & C_2 & \xrightarrow{h_2} & C_3 \\
 & & \mathcal{M}_{h_2 \circ h_1} & & \\
 & \swarrow & & \searrow & \\
 \mathcal{M}(C_1) & \xleftarrow{\mathcal{M}_{h_1}} & \mathcal{M}(C_2) & \xleftarrow{\mathcal{M}_{h_2}} & \mathcal{M}(C_3) \\
 & & (h_2 \circ h_1, \mathcal{M}_{h_1}(f_2) \circ f_1) & & \\
 & \swarrow & & \searrow & \\
 M_1 & \xrightarrow{(h_1, f_1)} & M_2 & \xrightarrow{(h_2, f_2)} & M_3 \\
 f_1 \downarrow & & f_2 \downarrow & & \\
 \mathcal{M}_{h_1}(M_2) & & \mathcal{M}_{h_2}(M_3) & & \\
 \mathcal{M}_{h_1}(f_2) \downarrow & & & & \\
 \mathcal{M}_{h_2 \circ h_1}(M_3) & & & &
 \end{array}$$

Eine derartige Komposition setzt voraus, daß \mathcal{M} die Identitäten und die Komposition der Spezifikationsmorphismen bewahrt, d.h. daß \mathcal{M} ein Funktor ist. Bezogen auf unser Beispiel muß die folgende Gleichung gelten:

$$\mathcal{M}_{h_2 \circ h_1}(M_3) = \mathcal{M}_{h_1}(\mathcal{M}_{h_2}(M_3)).$$

Initiale Objekte und freie Konstruktionen spielen eine wichtige Rolle bei der Betrachtung von Spezifikationen und Semantiken. Der Vorteil einer objektfreien Beschreibung dieser Eigenschaften im Rahmen der Kategorientheorie zeigt sich darin, daß wir ohne uns auf irgendwelche Notationen oder Kalküle einschränken zu müssen, an dieser Stelle Aussagen darüber treffen können, wie und unter welchen Bedingungen man Objekte (Modelle) mit diesen Eigenschaften erzeugen kann.

Innerhalb eines vorgegebenen Spezifikationsrahmens und der davon generierten generalisierten Modellkategorie existiert, unter der Voraussetzung daß die Modellkategorien initiale Objekte haben, zu jedem Spezifikationsmorphismus $h: C_1 \rightarrow C_2$ und jedem C_2 -Modell A , in Folge der Eindeutigkeit initialer Modelle (\rightarrow 5.1.9), genau ein generalisierter Morphismus

$$(h, eval): I_{C_1} \rightarrow A$$

vom initialen C_1 -Modell I_{C_1} nach A . Mit Hilfe dieser Morphismen und der Colimeskonstruktion, kann das initiale Objekt zum Colimes eines Diagramms in der Spezifikationskategorie, aus den initialen Objekten zu den im Diagramm vorkommenden Spezifikationen bestimmt werden.

5.3.3 Lemma (Colimites für initiale Modelle):

Sei $\mathcal{SF} = (\mathcal{C}, \mathcal{M})$ ein Spezifikationsrahmen, \mathcal{GSF} die \mathcal{SF} -generierte generalisierte Modellkategorie, I_C für jedes Objekt C aus \mathcal{C} das initiale Objekt in $\mathcal{M}(\mathcal{C})$, \mathcal{S} eine endliche Kategorie, $\mathcal{D}: \mathcal{S} \rightarrow \mathcal{C}$ ein Funktor mit Colimes

$$((c_s)_{s \in \mathcal{S}}, C)$$

und $\mathcal{G}: \mathcal{S} \rightarrow \mathcal{GSF}$ ein Funktor mit

$$\mathcal{G}(m) = (\mathcal{D}(m), eval): I_{\mathcal{D}(s)} \rightarrow I_{\mathcal{D}(s')}$$

für alle Morphismen $m: s \rightarrow s'$ aus \mathcal{S} , dann bildet

$$((c_s, eval), I_C)$$

einen Colimes von \mathcal{G} in \mathcal{GSF} . □

(Beweis \rightarrow Anhang S. 278)

Der zuvor schon angesprochene Zusammenhang von initialen Objekten und daraus resultierenden generalisierten Morphismen, spielt auch bei der Konstruktion freier Objekte eine zentrale Rolle. Dabei kann innerhalb eines vorgegebenen Spezifikationsrahmens, für jeden Spezifikationsmorphismus h und jedes Modell A zur Spezifikation

im Ursprung von h das bzgl. \mathcal{M}_h über A frei erzeugte Modell zur Zielspezifikation von h , mittels einer Colimeskonstruktion bestimmt werden.

5.3.4 Lemma (Pushouts und freie Konstruktionen):

Sei $\mathcal{SF} = (\mathcal{C}, \mathcal{M})$ ein Spezifikationsrahmen, \mathcal{GSF} die \mathcal{SF} -generierte generalisierte Modellkategorie, $E, D \in \mathcal{M}(C)$ mit initialen Objekten I_E bzw. I_D und A ein E -Objekt, dann bildet das Diagramm

$$\begin{array}{ccc}
 I_E & \xrightarrow{(h, eval)} & I_D \\
 (id_E, eval) \downarrow & \circlearrowright & \downarrow (id_D, eval) \\
 A & \xrightarrow{(h, u_A)} & C
 \end{array}$$

genau dann einen Pushout in \mathcal{GSF} , wenn das D -Objekt C bzgl. \mathcal{M}_h frei über A konstruiert ist. □

(Beweis \rightarrow Anhang S. 280)

Auch im nachfolgenden Lemma spielt der Zusammenhang von freien Konstruktionen und Colimites innerhalb generalisierter Modellkategorien, der sich aus der Verwandtschaft der couniversellen Eigenschaft von Colimites und der Eigenschaft universeller Abbildungen ergibt, eine zentrale Rolle. In diesem Fall werden Coequalizer in der generalisierten Modellkategorie aus universellen Abbildungen und Coequalizern in den Modellkategorien erzeugt, wo sie für konkrete Spezifikationsrahmen zumeist einfacher zu bilden sind.

5.3.5 Lemma (Konstruktion von Coequalizern):

Sei $\mathcal{SF} = (\mathcal{C}, \mathcal{M})$ ein Spezifikationsrahmen, \mathcal{GSF} die \mathcal{SF} -generierte generalisierte Modellkategorie, \mathcal{S} die von dem Graphen $s \rightrightarrows s'$ generierte Kategorie, $\mathcal{D}: \mathcal{S} \rightarrow \mathcal{GSF}$ ein Funktor mit

$$\mathcal{D}(s) = A, \quad \mathcal{D}(s') = A', \quad \mathcal{D}(m') = (h_1, f_1), \quad \mathcal{D}(m) = (h_2, f_2),$$

(c, C) der Coequalizer von h_1, h_2 in \mathcal{C} , $k = c \circ h_1 = c \circ h_2$, (u_A, B) eine \mathcal{M}_k -universelle Abbildung zu A , $(u_{A'}, B')$ eine \mathcal{M}_c -universelle Abbildung zu A' und (d, D) ein Coequalizer der Morphismen g_1, g_2 , die sich eindeutig aus dem nachfolgenden Diagramm ergeben

$$\begin{array}{ccc}
 A & \xrightarrow{u_A} & \mathcal{M}_k(B) & & B \\
 & \searrow & \downarrow \mathcal{M}_k(g_i) & & \downarrow g_i \\
 & & \mathcal{M}_k(B') & & B'
 \end{array}
 \quad i = 1, 2$$

dann bildet $((c, \mathcal{M}_c(d) \circ u_{A'}), D)$ einen Coequalizer für \mathcal{D} in \mathcal{GSF} . □

(Beweis \rightarrow Anhang S. 282)

Datentypspezifikationen

Ein wesentlicher Bestandteil jeder Spezifikations- und jeder Programmiersprache, egal ob vorgegeben oder vom Benutzer zu erzeugen, sind Datentypen. Entsprechend wichtig ist es, sie formal, d.h. vor allem mit einer formalen Semantik versehen, beschreiben zu können. Die in diesem Kapitel vorgestellten „*algebraischen Datentypspezifikationen*“, im weiteren einfach als „*algebraische Spezifikationen*“ bezeichnet, sind ein gut untersuchtes Mittel,¹ das diesen Zweck erfüllt, eine strukturierte Beschreibung komplexer Datentypen ermöglicht und darüber hinaus in einer Reihe rechnergestützter Werkzeuge² praktisch eingesetzt wird.

Wir geben im folgenden einen kurzen Überblick über grundlegende Aspekte algebraischer Spezifikationen, soweit sie im Rahmen dieser Arbeit von Interesse sind. Für ausführlichere und weiterführende Betrachtungen verweisen wir auf [EM85], [EM90] und [EBO91], aus denen die meisten, der in den ersten beiden Abschnitten dieses Kapitels eingeführten Konstruktionen, teilweise mit leichten Modifikationen, entnommen wurden. Dem gegenüber ist der Inhalt von Abschnitt 6.3 neu und speziell auf die Erfordernisse der Kapitel 1–3 zugeschnitten.

Algebraische Spezifikationen in der hier verwendeten Form sind ein formales Konstrukt, mit dem sich Datentypen bis auf Isomorphie eindeutig beschreiben lassen („*initiale Semantik*“), das aber gleichzeitig eine Charakterisierung aller Objekte erlaubt, die über die in der Spezifikation beschriebenen Eigenschaften verfügen („*lose Semantik*“). Diese Eigenschaften werden in Form von Gleichungen angegeben. Mehrsortige Algebren, verknüpft durch mit den Algebraoperationen verträgliche Abbildungen zwischen ihren Trägermengen, repräsentieren die Datentypen. Daraus ergibt sich zu jeder algebraischer Spezifikation eine Kategorie von Algebren und Algebraomorphismen als Semantik. Diese Kategorien enthalten jeweils ein initiales Objekt und sind gegen Komposition über Diagramme abgeschlossen.

Durch Komposition lassen sich algebraische Spezifikationen schrittweise zu komplexen Strukturen zusammenfügen und durch Parameterisierung an unterschiedliche Umgebungen anpassen. Zu diesem Zweck werden die Teilspezifikationen durch ihre interne Struktur und Gleichungen respektierende Spezifikationsmorphismen verknüpft. Ihnen entsprechen auf semantischer Ebene Vergißfunktoren, die Algebren und Algebraomorphismen ihrer Codomäne auf die für die Domäne des Spezifikationsmorphisms relevanten Teile reduzieren. In entgegengesetzter Richtung bilden die als Semantik

¹siehe [Hoa72], [GTW78], [Gro81], [Ehr89], [EGR94]

²unter anderem in [Kra87], [Int87], [Han88], [Sta90]

parameterisierter Spezifikationen verwendeten freien Funktoren ab. Sie betten die Parameteralgebren in die Algebren zur Rumpfspezifikation ein.

Das vorliegende Kapitel gliedert sich in drei Abschnitte. Im ersten werden „*Signaturen*“ als Teil algebraischer Spezifikationen eingeführt, sie beschreiben die Struktur der Datentypen. Spezifikationsmorphismen, -algebren und -algebromorphismen sind Spezialfälle der in diesem Abschnitt auf Signaturen definierten Objekte. Mit der Definition „*generalisierter Morphismen*“, die zwischen Algebren unterschiedlicher, über Morphismen in Beziehung stehender Signaturen verkehren, wird eine weitere Grundlage für die folgenden Abschnitte geschaffen.

Spezifikationen, parameterisierte Spezifikationen und deren Semantik sind Inhalt des zweiten Abschnitts. Einen Schwerpunkt in diesem Zusammenhang, bilden freie Konstruktionen bezüglich Spezifikationsmorphismen. Sie erlaubt es Algebren zur Ursprungsspezifikation des assoziierten Spezifikationsmorphismus in Algebren zu dessen Zielspezifikation einzubetten. Angewandt auf die Inklusionsbeziehung von Parameter und Spezifikationsrumpf bilden sie die Semantik parameterisierter Spezifikationen. Weitere Aspekte sind die Komposition von Spezifikationen und Algebren. Letztere können auch über die Grenzen der Semantikkategorien hinweg komponiert werden.

Aufbauend auf den Konstruktionen der ersten beiden Abschnitte, wird im dritten die Anreicherung von Spezifikationen und Algebren mit Variablen eingeführt. Sie erlaubt es Funktionen, die aus den Algebraoperationen zusammengesetzt sind, als Elemente der Algebra zu behandeln. Zu jeder Belegung der Funktionsvariablen mit Werten der ursprünglichen Algebra, ergibt sich dabei wiederum ein Wert in dieser Algebra. Die Konstruktionen dieses Abschnitts bilden eine Grundlage der Semantik für die in den Kapitel 2 und 3 als Spezifikationen nebenläufiger Systeme verwendeten AN-Spezifikationen.

6.1 Signaturen

Grundlegender Bestandteil algebraischer Spezifikationen sind „*Signaturen*“. Sie bestehen jeweils aus einer Menge von Sorten und einer nach Wertigkeit und Bildbereich geordneten Familie von Operationssymbolen und bestimmen den Aufbau des durch die Spezifikation beschriebenen Datentyps. Die Verwendung mehrsortiger Signaturen erlaubt zum einen die Spezifikation komplexer Datentypen wie „Stacks“ oder „markierte Graphen“, die sich aus mehreren einfachen Datentypen, z.B. einem Alphabet, Knoten und Kanten im Falle der markierten Graphen, zusammensetzen, zum anderen ist sie eine Voraussetzung für die Komposition von Signaturen.

Verknüpft werden die Signaturen durch *Signaturmorphismen*. Dabei handelt es sich um miteinander verträgliche Paare von Sorten- und Operationssymbolabbildungen. Gemeinsam mit den Signaturen formen sie die Kategorie STG , deren Identitäten und Komposition für jede der beiden Morphismuskomponenten separat in der Kategorie der Mengen gebildet werden. In diesem Zusammenhang vergessen wir die zusätzliche Struktur auf den Operationssymbolen und betrachten sie als Menge.

6.1.1 Definition (Signaturen und Signaturmorphismen):

- (I) Eine „Signatur“ $Sig = (SO, OP)$ ist ein Paar bestehend aus einer Menge SO genannt „Sorten“ und einer Familie

$$OP = (OP^{w,s})_{w \in SO^*, s \in SO} \quad ^3$$

genannt „Operationssymbole“. Für jede Sorte s in SO bezeichnen wir die Elemente aus $OP^{\lambda,s}$ ($\lambda =$ leeres Wort) als „Konstanten“.

- (II) Gegeben zwei Signaturen Sig_1, Sig_2 , dann bildet ein Paar von Abbildungen

$$h = (h^{SO}: SO_1 \rightarrow SO_2, h^{OP}: OP_1 \rightarrow OP_2)$$

einen „Signaturmorphismus“ (notiert $h: Sig_1 \rightarrow Sig_2$), wenn zu jeder Menge $OP_1^{s_1 \dots s_n, s}$ in OP_1 und jedem Operationssymbol op aus dieser Menge gilt:

$$h^{OP}(op) \in OP_2^{h^{SO}(s_1) \dots h^{SO}(s_n), h^{SO}(s)}.$$

- (III) Die Signaturen bilden zusammen mit den Signaturmorphismen die Kategorie STG . Deren Identitäten, Komposition, Domän- und Codomänabbildung werden aus Paaren der entsprechenden Konstrukte für Abbildungen zwischen Mengen gebildet.

□

Um den Zusammenhang eines Operationssymbols op aus $OP^{s_1 \dots s_n, s}$ zu seinen Argument- und Zielsorten hervorzuheben, notieren wir es durch $op: s_1 \dots s_n \rightarrow s$. Damit Konstanten bei Verwendung dieser Notation nicht gesondert behandeln werden müssen, notieren wir in diesem Rahmen das leer Wort λ über SO durch $s_1 \dots s_0$. Damit läßt sich eine Konstante op aus $OP^{\lambda,s}$ durch $op: s_1 \dots s_n \rightarrow s$ mit $n = 0$ darstellen.

6.1.2 Beispiel (Signaturen und Signaturmorphismen):

Die folgenden drei Signaturen beschreiben Trägermengen und Operationen von Mengen, booleschen Algebren und den natürlichen Zahlen.

$Set_{sig} =$	$Bool_{sig} =$	$Nat_{sig} =$
<u>sorts</u> SET	<u>sorts</u> BOOL	<u>sorts</u> NAT
	<u>opns</u> $T: \quad \rightarrow$ BOOL	<u>opns</u> $0 \quad : \quad \rightarrow$ NAT
	$F: \quad \rightarrow$ BOOL	Suc: NAT \rightarrow NAT
	$\neg: \text{BOOL} \rightarrow$ BOOL	$+$: NAT, NAT \rightarrow NAT
	$\wedge: \text{BOOL}, \text{BOOL} \rightarrow$ BOOL	$*$: NAT, NAT \rightarrow NAT
	$\vee: \text{BOOL}, \text{BOOL} \rightarrow$ BOOL	

³ SO^* bezeichnet die Menge aller endlichen Worte inklusive des leeren Worts über SO .

Von Set_{sig} existiert ein Signaturmorphismus sowohl nach $Bool_{sig}$ als auch nach Nat_{sig} . Dabei werden jeweils die Sorten aufeinander abgebildet, die Operationsabbildung ist leer. Darüber hinaus bildet das Abbildungspaar

$$\begin{aligned} h^{SO} : nat &\mapsto bool, & h^{OP} : 0 &\mapsto T \\ & & suc &\mapsto \neg \\ & & + &\mapsto \wedge \\ & & * &\mapsto \vee \end{aligned}$$

einen Signaturmorphismus von Nat_{sig} nach $Bool_{sig}$. □

Signaturen setzen sich im wesentlichen aus Mengen⁴ zusammen und Signaturmorphismen aus Abbildungen zwischen Mengen. Damit können wir Kompositionen von Signaturen komponentenweise aus Colimeskonstruktionen über Mengen bilden, wo entsprechende Colimites⁵ für jedes endliche Diagramm existieren (\rightarrow [ML70] S. 71). Die Strukturierung der Operationen in Mengenfamilien bleibt dabei gewahrt.

Mit Hilfe von Satz 5.1.19, der ein Verfahren zur Bildung endlicher Limites aus Produkten und Equalizern angibt, und der Dualität von Limites zu Colimites (\rightarrow 5.1.17), kann der Colimes jedes endlichen Diagramms aus Coprodukten und Coequalizern gebildet werden. Wir beschränken uns daher hier auf diese beiden Konstruktionen.

6.1.3 Lemma (Coprodukte in $SI\mathcal{G}$):

Sei \mathcal{S} eine endliche diskrete Kategorie, $\mathcal{D} : \mathcal{S} \rightarrow SI\mathcal{G}$ ein Funktor mit

$$\mathcal{D}(s) = (SO_s, OP_s)$$

für alle Objekte s aus \mathcal{S} , $((c_s^{SO})_{s \in Obj(\mathcal{S})}, \coprod_{s \in Obj(\mathcal{S})} SO_s)$ das Coprodukt⁶ der Sorten

$$OP_c = \left(OP_c^{c_s^{SO}(so_1) \cdots c_s^{SO}(so_n), c_s^{SO}(so)} = OP_s^{so_1 \cdots so_n, so} \right)_{s \in Obj(\mathcal{S}), so_1 \cdots so_n \in SO_s^\omega, so \in SO_s}$$

eine Familie von Operationssymbolen und $c_s^{OP} : OP_s \rightarrow OP_c$ für alle Objekte s in \mathcal{S} eine Abbildung mit

$$c_s^{OP}(op) \stackrel{\text{def}}{=} op : c_s^{SO}(so_1), \dots, c_s^{SO}(so_n) \rightarrow c_s^{SO}(so)$$

für alle Objekte s aus \mathcal{S} und Operationssymbole $op : so_1 \cdots so_n \rightarrow so$ aus OP_s , dann bildet

⁴Die Operationen bilden eine Mengenfamilie, d.h. sie enthalten zusätzliche Strukturinformationen.

⁵Beispiel 5.1.16 zeigt die Konstruktion von Coprodukten und Coequalizern. Mit Hilfe von Satz 5.1.19 können aus diesen beiden Colimites alle Colimites endlicher Diagramme konstruiert werden.

$$\left((c_s^{SO}, c_s^{OP})_{s \in \text{Obj}(\mathcal{S})}, \left(\coprod_{s \in \text{Obj}(\mathcal{S})} SO_s, OP_c \right) \right)$$

ein Coprodukt für \mathcal{D} .

□

(Beweis → Anhang S. 284)

6.1.4 Lemma (Coequalizer in STG):

Sei \mathcal{S} die von dem Graphen $s \rightrightarrows s'$ generierte Kategorie,

$$\begin{aligned} \mathcal{G}: \mathcal{S} &\rightarrow \mathcal{SET} && \text{ein Funktor mit Coequalizer } (c, SO), \\ \mathcal{R}: \mathcal{S} &\rightarrow \mathcal{SET} && \text{ein Funktor mit Coequalizer } (d, C), \end{aligned}$$

$\mathcal{D}: \mathcal{S} \rightarrow STG$ ein Funktor mit $\mathcal{D}(m) = (\mathcal{G}(m), \mathcal{R}(m))$ für Morphismen m und $\mathcal{D}(s) = (SO_s, OP_s)$ für Objekte s in \mathcal{S} und

$$OP \stackrel{\text{def}}{=} \{ d(op): c(so_1) \cdots c(so_n) \rightarrow c(so) \mid op \in OP_{s'}^{so_1 \cdots so_n, so} \}$$

eine Familie von Operationssymbolen, dann bildet $((c, d), (SO, OP))$ einen Coequalizer für \mathcal{D} .

□

(Beweis → Anhang S. 286)

6.1.5 Folgerung (Colimites in STG):

In STG existiert zu jedem endlichen Diagramm ein Colimes.

□

Signaturen bestimmen lediglich die Struktur der zugehörigen Datentypen, das Verhalten der Datentypoperationen wird dabei nicht festgelegt. Unter einem Datentyp verstehen wir in diesem Zusammenhang eine „Algebra“, die eine Trägermenge zu jeder Sorte und eine Operation mit passender Domäne und Codomäne zu jedem Operationssymbol der Signatur enthält. Sie stehen, über mit den Operationen verträglichen Abbildungsfamilien zwischen ihren Trägermengen, miteinander in Beziehung. Algebren und die als „Algebromorphismen“ bezeichneten Abbildungen zu einer Signatur ergeben deren Semantik in Form einer Kategorie.

6.1.6 Definition (Signaturalgebren und Signaturalgebromorphismen):

Sei $Sig = (SO, OP)$ eine Signatur.

- (I) Eine „Sig-Algebra“ $A = (SO_A, OP_A)$ besteht aus einer Familie von Trägermengen $SO_A = (A_s)_{s \in SO}$ und einer Familie von Funktionen

$$OP_A = (op_A: A_{s_1} \times \cdots \times A_{s_n} \rightarrow A_s)_{op \in OP^{s_1, \dots, s_n, s}}$$

genannt „Operationen“.

⁶dessen Objekt entspricht bei Mengen der disjunkten Vereinigung

(II) Gegeben zwei *Sig*-Algebren A, B , dann bildet eine Familie von Abbildungen

$$h = (h_s: A_s \rightarrow B_s)_{s \in SO}$$

einen „*Sig*-Algebromorphismus“ (notiert $h: A \rightarrow B$), wenn für alle Operationen $op_A: A_{s_1} \times \cdots \times A_{s_n} \rightarrow A_s$ und Elemente a_i aus A_{s_i} gilt:

$$h_s(op_A(a_1, \dots, a_n)) = op_B(h_{s_1}(a_1), \dots, h_{s_n}(a_n)).$$

(III) Die *Sig*-Algebren formen zusammen mit den *Sig*-Algebromorphismen die Kategorie $\mathcal{ALG}(Sig)$. Deren Identitäten, Komposition, Domän- und Codomänabbildung entsprechen der kanonischen Fortsetzung, der entsprechenden Konstrukte für Abbildungen zwischen Mengen, auf Abbildungsfamilien.

□

Eine wichtige Rolle im Zusammenhang mit Signaturen und Signaturalgebren spielen Variablen und ihre Belegungen. Bei den Elementen der im folgenden definierten Variablenfamilien handelt es sich um getypte Variablen, d.h. ihnen ist eine Sorte der Signatur als Typ zugeordnet. Dabei ist eine „Überladung“ der Variablenbezeichner⁷ möglich. Die Belegungen der Variablen mit den Werten einer Signaturalgebra bewahrt die auf beiden Seiten der Abbildung definierte Sortenzuordnung.

6.1.7 Definition (Variablenfamilien und Variablenbelegungen):

Sei $Sig = (SO, OP)$ eine Signatur und A eine *Sig*-Algebra, dann bildet

(I) jede SO -sortierte Mengenfamilie

$$X = (X_s)_{s \in SO}$$

eine „*Variablenfamilie zu Sig*“ (kurz „*Sig*-*Variablenfamilie*“) und

(II) jede Abbildungsfamilie

$$ass = (ass_s: X_s \rightarrow A_s)_{s \in SO},$$

notiert $ass: X \rightarrow A$, eine „*Variablenbelegung*“.

□

Zu jeder Signatur $Sig = (SO, OP)$ existiert mit der von ihr induzierten „*Termalgebra*“ $T_{OP} \rightarrow [EM85] 3.2$ ein initiales Objekt in $\mathcal{ALG}(Sig)$. Dessen Trägermengen T_{OP_s} enthalten für jede Sorte s aus SO alle Konstanten zu s und zu jedem Operationssymbol op aus $OP^{s_1 \cdots s_n, s}$ und allen Termen t_i in $T_{OP_{s_i}}$ den Term $op(t_1, \dots, t_n)$. Die jedem

⁷Variablen mit gleichem Namen zu unterschiedlichen Sorten

Operationssymbol op aus $OP^{s_1 \dots s_n, s}$ zugeordnete Funktion op_{TOP} liefert, angewandt auf die Argumente t_1, \dots, t_n , den Term $op(t_1, \dots, t_n)$ als Ergebnis.

Läßt man bei der Termbildung zusätzlich zu den Operationssymbolen Variablen einer *Sig*-Variablenfamilie X zu, ergibt sich daraus die *Sig*-induzierte „Termalgebra mit Variablen“ $T_{OP}(X)$. Im Unterschied zu den in Abschnitt 6.3 vorkommenden Termalgebren $T_{OP(X)}$ zu erweiterten Signaturen sind die Variablen in $T_{OP}(X)$ nicht funktionsgeneriert, d.h. es existieren keine Funktionen $x: \rightarrow T_{OP_s}(X)$ zu Variablen x aus X_s . Wenn Verwechslungen ausgeschlossen sind, bezeichnen wir die Menge der Terme bzw. die Menge der Terme mit Variablen über *Sig*, wie die entsprechenden Algebren mit T_{OP} bzw. mit $T_{OP}(X)$.

Ähnlich wie Signaturen notieren wir auch Algebren in tabellarischer Form. Dabei ergibt sich die Zuordnung von Domänen zu Sorten und von Operationen zu Operationssymbolen aus deren Reihenfolge in Signaturen und Algebra. Konstanten geben wir innerhalb der Algebra durch ihren Wert an, d.h. an der entsprechenden Stelle steht ein Element der Domäne in die die Konstante abbildet.

6.1.8 Beispiel (Signaturalgebren und Signaturalgebrenmorphisamen):

Jede Menge ist eine Algebra – und jede Abbildung zwischen Mengen ein Morphismus zur Signatur Set_{Sig} (\rightarrow 6.1.2). Ein Objekt aus $\mathcal{ALG}(Bool_{Sig})$ bildet die folgende Algebra

$B =$

doms 0, 1;

fkts 1, 0, *not*, *eq*, *or*;

mit

$$not(a) = \begin{cases} 1 & ; a = 0 \\ 0 & ; a = 1 \end{cases}, \quad eq(a, b) = \begin{cases} 1 & ; a = b \\ 0 & ; \text{sonst} \end{cases}, \quad or(a, b) = \begin{cases} 1 & ; a + b > 0 \\ 0 & ; \text{sonst} \end{cases}$$

die bis auf die Operation *eq* anstelle des logischen Und der üblichen booleschen Algebra entspricht. Die Algebren $A = (SO_B, OP_B \setminus \{1\})$ und

$\mathbb{N}_{Alg} =$

doms \mathbb{N} ;

fkts 0, *next*, *add*, *mult*;

mit

$$next(n) = n + 1, \quad add(n, m) = n + m, \quad mult(n, m) = n * m.$$

liegen beide in $\mathcal{ALG}(Nat_{Sig})$ und sind dort über den Signaturalgebrenmorphismus

$$f: \mathbb{N}_{Alg} \rightarrow A \quad \text{mit} \quad f(x) = \begin{cases} 1, & x = 0 \vee x \text{ ist gerade;} \\ 0, & \text{sonst.} \end{cases}$$

verknüpft. Hätten wir in A die Operation eq durch ein normales logisches Und ersetzt, wäre f wegen

$$1 = f(2) = f(+ (1, 1)) \neq \text{and}(f(1), f(1)) = \text{and}(0, 0) = 0$$

nicht mit den Operationen verträglich und damit kein Signaturalgebromorphismus. Die Abbildung

$$\text{eval}(t) = \begin{cases} 0, & t = 0; \\ \text{next}(\text{eval}(t')), & t = \text{suc}(t'); \\ \text{add}(\text{eval}(t_1), \text{eval}(t_2)), & t = +(t_1, t_2); \\ \text{mult}(\text{eval}(t_1), \text{eval}(t_2)), & t = *(t_1, t_2); \end{cases}$$

ist ein NAT_{Sig} -Algebromorphismus von der NAT_{Sig} -Termalgebra nach \mathbb{N}_{Alg} . □

Da alle Elemente der Trägermengen einer Sig -Termalgebra von Sig -Operationssymbolen generiert sind und jede Sig -Algebra A zu jedem Operationssymbol eine passende Operation enthält, läßt sich daraus in naheliegender Weise ein Signatormorphismus von der Termalgebra nach A konstruieren. Diese als „*Evaluierungsfunktion*“ bezeichnete Abbildung ist für jede Sig -Algebra eindeutig, d.h. alle Sig -Algebromorphismen von der Termalgebra in eine Sig -Algebra A sind bis auf ihren Bezeichner gleich. Die Existenz und Eindeutigkeit dieser Evaluierungsfunktionen ist ein sowohl hinreichendes als auch notwendiges Kriterium für die Initialität (\rightarrow 5.1.9) der Termalgebra in $\mathcal{ALG}(\text{Sig})$.

6.1.9 Definition (Evaluierungsfunktion):

Sei $\text{Sig} = (SO, OP)$ eine Signatur, T_{OP} die Sig -Termalgebra und A eine Sig -Algebra, dann definieren wir

(I) den Sig -Algebromorphismus $\text{eval}_A: T_{OP} \rightarrow A$ durch:

$$\text{eval}_{A_s}(\text{op}(t_1, \dots, t_n)) = \text{op}_A(\text{eval}_A(t_1), \dots, \text{eval}_A(t_n))$$

für alle Sorten s aus SO und Terme $\text{op}(t_1, \dots, t_n)$ aus T_{OP_s} und bezeichnen ihn als „*Evaluierungsfunktion*“, und

(II) zu jeder Variablenfamilie X und jeder Variablenbelegung $\text{ass}: X \rightarrow A$, den Sig -Algebromorphismus $\text{eval}_A^{\text{ass}}: T_{OP}(X) \rightarrow A$ durch:

$$\text{eval}_{A_s}^{\text{ass}}(u) = \begin{cases} \text{ass}(u), & u \in X; \\ \text{op}_A(\text{eval}_A^{\text{ass}}(t_1), \dots, \text{eval}_A^{\text{ass}}(t_n)), & u = \text{op}(t_1, \dots, t_n); \end{cases}$$

für alle Sorten s aus SO und Terme u aus $T_{OP_s}(X)$. □

Wenn die Codomäne einer Evaluierungsfunktion aus dem Kontext hervorgeht, lassen wir ihren Index weg und schreiben $eval: T_{OP} \rightarrow A$ für $eval_A: T_{OP} \rightarrow A$. Ein Beispiel für eine Evaluierungsfunktion bildet der Algebromorphismus $eval$ in 6.1.8.

6.1.10 Lemma (Initiales Objekt):

Zu jeder Signatur Sig , ist die Sig -Termalgebra initiales Objekt in $\mathcal{ALG}(Sig)$. \square

Beweis: Die Behauptung folgt aus [EM85] Theorem 3.3.

Reduziert man bezüglich eines Signaturmorphisms $h: Sig_1 \rightarrow Sig_2$ eine Sig_2 -Algebra auf Trägermengen und Operationen zu Sorten und Operationssymbolen im Bild von h , erhält man bei injektiven Morphismen eine $Spec_1$ -Algebra. Andernfalls müssen neben der Reduktion Trägermengen und Operationen so oft kopiert werden, wie ihre Sig_2 -Entsprechungen Sig_1 -Urbilder unter h haben.

Ein Beispiel für diese Konstruktion bildet die Algebra Nat_{sig} -Algebra A in Beispiel 6.1.8, die sich durch die beschriebene Reduktion (die Konstante 1 wurde weggelassen) bezüglich $(h^{SO}, h^{OP}): Nat_{sig} \rightarrow Bool_{sig}$ (\rightarrow 6.1.2) aus der $Bool_{sig}$ -Algebra B im gleichen Beispiel ergibt.

Diese Konstruktion läßt sich auf Algebromorphismen fortsetzen, sodaß man zu jedem Signaturmorphisms $h: Sig_1 \rightarrow Sig_2$ einen Funktor von $\mathcal{ALG}(Sig_2)$ nach $\mathcal{ALG}(Sig_1)$ erhält. Diesen Funktor bezeichnen wir als den von „ h -generierten Vergißfunktors“ oder, wenn der Signaturmorphisms sich aus dem Kontext ergibt, einfach als *Vergißfunktors*.

6.1.11 Definition (Vergißfunktors):

Sei $h: Sig_1 \rightarrow Sig_2$ ein Signaturmorphisms, dann definieren wir den Funktor

$$\mathcal{V}_h: \mathcal{ALG}(Sig_2) \rightarrow \mathcal{ALG}(Sig_1),$$

genannt *h -generierter Vergißfunktors*, für alle $\mathcal{ALG}(Sig_2)$ -Morphismen $f: A_1 \rightarrow A_2$ durch:

$$(I) \quad \mathcal{V}_h(f) \stackrel{\text{def}}{=} (f_{h^{SO}(s)})_{s \in SO_1}: \mathcal{V}_h(A_1) \rightarrow \mathcal{V}_h(A_2) \text{ mit}$$

$$(II) \quad \mathcal{V}_h(A_i) \stackrel{\text{def}}{=} \left(\left(A_{i_{h^{SO}(s)}} \right)_{s \in SO_1}, \left(op_{\mathcal{V}_h(A_i)} \right)_{op \in OP_1} \right) \text{ und}$$

$$op_{\mathcal{V}_h(A_i)}(a_1, \dots, a_n) = h^{OP}(op)(a_1, \dots, a_n)$$

für $i = 1, 2, j = 1, \dots, n$ alle $op \in OP_1^{s_1 \dots s_n, s}$ und $a_j \in A_{i_{s_j}}$. \square

Die Signaturen als formale Datentypbeschreibungen und die Signaturalgebrakategorien als deren Semantik lassen sich zu einer als „Spezifikationsrahmen“ (\rightarrow 5.3.1)

bezeichneten Einheit zusammenfassen. Dabei entsprechen den Signaturmorphismen auf der Semantikebene die jeweiligen Vergißfunktoren. Die Identitäten und die Komposition in SIG werden im Rahmen dieser Zuordnung in Identitätsfunktoren und die Funktorkomposition umgesetzt. Dieses abstrakte Konzept bietet die Grundlage für eine ganze Reihe von Konstruktionen und Ergebnissen (\rightarrow 5.3).

6.1.12 Lemma (Spezifikationsrahmen):

Das Paar $(SIG, \mathcal{M}: SIG^{op} \rightarrow CATCAT)$ ⁸ mit

(I) $\mathcal{M}(Sig) \stackrel{\text{def}}{=} \mathcal{ALG}(Sig)$ für alle SIG -Objekte Sig und

(II) $\mathcal{M}(h) \stackrel{\text{def}}{=} \mathcal{V}_h$ für alle SIG -Morphismen $h: Sig_1 \rightarrow Sig_2$

bildet einen mit „ $SIG\mathcal{F}$ “ bezeichneten „Spezifikationsrahmen“ (\rightarrow 5.3.1). □

Beweis: Die Behauptung folgt aus [EBO91]. Darin bilden die algebraischen Spezifikationen zusammen mit dem Funktor MOD einen „specification frame“. Die Zuordnung von Vergißfunktoren zu Spezifikationsmorphismen entspricht der Konstruktion in 6.1.11, sodaß die Funktoreigenschaften von MOD die Funktoreigenschaften dieser Konstruktion bedingen.

Jeder Spezifikationsrahmen generiert eine Kategorie, die die Modelle und Modellmorphismen aller Signaturen zusammenfaßt. Zusätzlich enthält diese als *generalisierte Modellkategorie* (\rightarrow 5.3.2) bezeichnete Struktur Morphismen, die die Grenzen der ursprünglichen Modellkategorien überschreiten. Sie verknüpfen Modelle, deren Spezifikationen miteinander in Verbindung stehen.

6.1.13 Beispiel (Vergißfunktoren und generalisierter Morphismus):

Die NAT_{sig} -Algebra A aus Beispiel 6.1.8 entspricht dem Bild der $Bool_{sig}$ -Algebra B unter dem Vergißfunktoren \mathcal{V}_h zum Signaturmorphismus $h = (h^{SO}, h^{OP})$ aus 6.1.8 und ist Ziel des NAT_{sig} -Algebromorphismus $f: \mathbb{N}_{Alg} \rightarrow A$. Damit bildet das Paar (h, f) einen generalisierten Morphismus von \mathbb{N}_{Alg} nach B . □

Ein Bestandteil der Signaturalgebren in der von $SIG\mathcal{F}$ -generierten (generalisierten) Modellkategorie $\mathcal{GSIG-ALG}$, sind Mengenfamilien, die sich über die disjunkte Vereinigung auf Mengen abbilden lassen. Dementsprechend lassen sich die Abbildungsfamilien der Algebromorphismen in Abbildungen zwischen Mengen transformieren. Mit Hilfe des Signaturmorphismus als erster Komponente eines generalisierten Algebromorphismus läßt sich die Sortenindizierung der Bilder des Algebromorphismus

⁸ $CATCAT$ (\rightarrow S 117) bezeichnet die Quasikategorie aller Kategorien mit Funktoren als Morphismen

(zweite Komponente) auf die Sorten der Zielsignatur anpassen. Das Ergebnis dieser Konstruktion bildet ein Funktor von $\mathcal{GSIG}\text{-}\mathcal{ALG}$ in die Kategorie der Mengen \mathcal{SET} .

6.1.14 Definition (Funktor von Algebren auf Mengen):

Sei $\mathcal{GSIG}\text{-}\mathcal{ALG}$ die generalisierte Modellkategorie (\rightarrow 5.3.2) zu \mathcal{SIGF} , dann definieren wir den Funktor $\|_-\|: \mathcal{GSIG}\text{-}\mathcal{ALG} \rightarrow \mathcal{SET}$ durch:

$$\|(h, f)\|: \left(\bigsqcup_{s \in SO_{dom(h)}} A_s \right) \rightarrow \left(\bigsqcup_{s \in SO_{cod(h)}} B_s \right) \quad \text{mit} \quad \|(h, f)\| (a_s) \stackrel{\text{def}}{=} f_s(a)_{h^{SO}(s)}$$

für alle generalisierten Morphismen $(h, f): A \rightarrow B$, s aus $SO_{dom(h)}$ und a aus A_s . \square

6.2 Spezifikationen

Mit den im letzten Abschnitt eingeführten Signaturen läßt sich die Struktur von Datentypen, d.h. die Anzahl der Trägermengen und die Domäne respektive die Codomäne ihrer Operationen, festlegen. Was bisher fehlt, ist die Möglichkeit, das Verhalten der Operationen und damit die Struktur der Trägermengen, soweit deren Elemente von Operationen generiert sind, zu beschreiben. Sie wird in *algebraischen Spezifikationen* durch die Zusicherung von Operationseigenschaften in Form von Gleichungen ergänzt. Da algebraische Spezifikationen in diesem Kapitel die einzige Spezifikationsform darstellen, bezeichnen wir sie hier einfach als Spezifikationen und ihren Signaturanteil als die „*unterliegende Signatur*“.

Gleichungen zu einer gegebenen Signatur, ihre Erfüllbarkeit in Bezug auf Algebren und ihre aus Signaturmorphismen generierten Transformationen sind in diesem Abschnitt von zentraler Bedeutung. Aus diesem Grunde geben wir im folgenden eine formale Definitionen dieser im allgemeinen intuitiv verstandenen Begriffe.

Mit den darin vorkommenden Variablen enthalten Gleichungen Bestandteile, die nicht zur Signatur gehören. Aus diesem Grunde gehen sie neben den beiden Termen, die für die linke und rechte Seite einer Gleichung stehen, als dritte Komponente in die Gleichungsdefinition ein. Bezogen auf eine Signaturalgebra A ist eine derartige Gleichung erfüllt, wenn die Interpretation (\rightarrow 6.1.9) beider Terme für jede Belegung der Variablen mit Werten aus A jeweils das gleiche Element in A ergibt.

6.2.1 Definition (Gleichungen und Erfüllbarkeit):

Sei $Sig = (SO, OP)$ eine Signatur und A eine Sig -Algebra, dann

- (I) bildet ein Tripel (X, l, r) eine „*Gleichung*“ zur Sorte s , wenn X eine Sig -Variablenfamilie ist und l, r Terme aus $T_{OP_s}(X)$ sind und

(II) eine Gleichung (X, l, r) ist „erfüllt in A “, wenn für alle Variablenbelegungen $ass: X \rightarrow A$ gilt:

$$eval_A^{ass}(l) = eval_A^{ass}(r).$$

□

6.2.2 Beispiel (Erfüllbarkeit von Gleichungen):

Sei B die $Bool_{sig}$ -Algebra aus Beispiel 6.1.8 und $ass: \{a\} \rightarrow B$ eine Variablenbelegung mit $ass(a) = 0$, dann ist die Gleichung $(\{a\}, \wedge(F, a), F)$ wegen

$$eval_B^{ass}(\wedge(F, a)) = eq(0, 0) = 1 \neq 0 = eval_B^{ass}(F)$$

in B nicht erfüllt.

□

Um Gleichungen von einer Signatur auf eine andere, mit der ursprünglichen durch einen Signatormorphismus verknüpfte Signatur zu übertragen, müssen zum einen die Variablenfamilien an die neuen Sorten angepaßt werden und zum anderen muß aus dem Signatormorphismus eine Abbildung zwischen den jeweiligen Termalgebren mit Variablen konstruiert werden. Die Anpassung der Variablenmengen erfordert die Vereinigung aller Variablen zu Sorten mit einem gemeinsamen Bild unter dem Signatormorphismus, während sich die Termabbildung direkt aus der Abbildung der Operationssymbole ergibt.

6.2.3 Definition (Transformation von Variablenfamilien und Termen):

Sei $h: Sig_1 \rightarrow Sig_2$ ein Signatormorphismus, X eine Sig_1 -Variablenfamilie und EQ_i die Menge aller Gleichungen zu Sig_i , dann definieren wir

(I) die Sig_2 -Variablenfamilie $X^h = (X_s^h)_{s \in SO_2}$ durch:

$$X_s^h = \bigsqcup_{s' \in \{s'' \in SO_1 \mid h^{SO}(s'') = s\}} X_{s'}$$

für alle Sorten s aus SO_2 ,

(II) die Abbildung $h^t: T_{OP_1}(X) \rightarrow T_{OP_2}(X^h)$ durch:

$$h^t(u) = \begin{cases} u, & u \in X; \\ h^{OP}(op)(h^t(t_1), \dots, h^t(t_n)), & u = op(t_1, \dots, t_n) \wedge op \in OP_1; \end{cases}$$

für alle $T_{OP_1}(X)$ -Terme u und

(III) die Abbildung $h: EQ_1 \rightarrow EQ_2$ durch:

$$h((X, l, r)) \stackrel{\text{def}}{=} (X^h, h^t(l), h^t(r))$$

für alle Gleichungen (X, l, r) in EQ_1 . □

Algebraische Spezifikationen werden durch die Gleichungen respektierende, Signaturmorphismen miteinander in Beziehung gesetzt. Damit bleibt neben der Struktur auch das in den Gleichungen festgelegte Verhalten der Operationen in der Zielspezifikation eines derartigen, als „Spezifikationsmorphismus“ bezeichneten Signaturmorphismus erhalten. Zu diesem Zweck muß die vom Spezifikationsmorphismus generierte Transformation einer Gleichung der Ursprungsspezifikation nicht notwendigerweise Element der Gleichungsmenge zur Zielspezifikation sein, es reicht aus, wenn sie daraus ableitbar ist.

6.2.4 Definition (Spezifikationen und Spezifikationsmorphismen):

- (I) Eine algebraische Spezifikation $Spec = (SO, OP, EQ)$ besteht aus einer Signatur (SO, OP) und einer Menge von Gleichungen EQ zu dieser Signatur.
- (II) Seien $Spec_i = (SO_i, OP_i, EQ_i)$, $i = 1, 2$, algebraische Spezifikationen und $h: (SO_1, OP_1) \rightarrow (SO_2, OP_2)$ ein Signaturmorphismus, dann bildet

$$h: Spec_1 \rightarrow Spec_2$$

einen Spezifikationsmorphismus, wenn die Transformation $(X^h, h^t(l), h^t(r))$ jeder Gleichung (X, l, r) in EQ_1 aus EQ_2 folgt.⁹

- (III) Die Spezifikationen bilden zusammen mit den Spezifikationsmorphismen die Kategorie \mathcal{SPEC} . Deren Identitäten, Komposition, Domän- und Codomänenabbildung entsprechen den Konstrukten in \mathcal{SIG} . □

6.2.5 Beispiel (Spezifikationen und Spezifikationsmorphismen):

Die nachfolgenden Spezifikationen $Bool$ mit unterliegender Signatur $Bool_{sig}$ und Nat mit unterliegender Signatur Nat_{sig} (\rightarrow 6.1.2) beschreiben die aus Programmiersprachen bekannten Datentypen „Boolean“ und „Integer“.

$$Bool = Bool_{sig} \cup$$

$$\underline{\text{eqns}} \quad a, b : \text{BOOL};$$

$$\neg(T) = F$$

$$\neg(\neg(a)) = a$$

$$\neg(\wedge(a, b)) = \vee(\neg(a), \neg(b))$$

$$\neg(\vee(a, b)) = \wedge(\neg(a), \neg(b))$$

$$\wedge(a, b) = \wedge(b, a)$$

$$\wedge(F, a) = F$$

$$\wedge(T, a) = a$$

$$Nat = Nat_{sig} \cup$$

$$\underline{\text{eqns}} \quad n, m : \text{NAT};$$

$$+(n, m) = +(m, n)$$

$$+(n, 0) = n$$

$$+(suc(n), m) = suc(+(n, m))$$

$$*(n, m) = *(m, n)$$

$$*(n, 0) = 0$$

$$*(suc(n), m) = +(*(n, m), m)$$

⁹In [EM85] wird ein korrekter und vollständiger Ableitungskalkül für Gleichungen eingeführt.

Der Signaturmorphismus $(h^{SO}, h^{OP}): Nat_{sig} \rightarrow Bool_{sig}$ aus Beispiel 6.1.2, der die unterliegenden Signaturen der beiden Spezifikationen verbindet, bildet in diesem Fall keinen Spezifikationsmorphismus, da die Transformation

$$(\{n, m\}, \wedge(\neg(n), m), \neg(\wedge(n, m)))$$

der *Nat*-Gleichung $(\{n, m\}, +(suc(n), m), suc(+ (n, m)))$ im Widerspruch zu den *Bool*-Gleichungen steht. Erweitert man *Nat* zur Spezifikation Nat_R

$$\begin{aligned} Nat_R &= Nat \cup Bool \cup \\ \underline{\text{opns}} &=: \text{NAT}, \text{NAT} \rightarrow \text{BOOL} \\ &<: \text{NAT}, \text{NAT} \rightarrow \text{BOOL} \\ \underline{\text{eqns}} & \quad n, m : \text{NAT}; \\ &= (n, m) \quad \quad \quad = = (m, n) \\ &= (n, n) \quad \quad \quad = T \\ &= (n, Suc(n)) \quad \quad = F \\ &< (0, Suc(n)) \quad \quad = T \\ &< (Suc(n), 0) \quad \quad = F \\ &< (Suc(n), Suc(m)) = < (n, m) \end{aligned}$$

dann bildet die Inklusion von *Bool* in Nat_R einen Spezifikationsmorphismus. □

Die Komposition von Spezifikationen über Diagramme und Colimites (\rightarrow S. 117, 5.1.15), läßt sich aus der Komposition der unterliegenden Signaturen ableiten. Dabei ergibt sich die Gleichungsmenge der komponierten Spezifikation aus der Vereinigung der Gleichungsmengen aller Komponenten, die durch Transformation (\rightarrow 6.2.3) mit Hilfe der Colimesmorphismen an die komponierte Signatur angepaßt werden.

6.2.6 Lemma (Colimites in $SP\mathcal{E}\mathcal{C}$):

Sei \mathcal{S} eine endliche Kategorie, $\mathcal{G}: \mathcal{S} \rightarrow \mathcal{SIG}$ ein Funktor mit Colimes

$$\left((c_s)_{s \in \text{Obj}(\mathcal{S})}, \text{Sig} \right),$$

$\mathcal{D}: \mathcal{S} \rightarrow \mathcal{SP\mathcal{E}\mathcal{C}}$ ein Funktor mit

$$\mathcal{D}(m) \Big|_{\mathcal{G}(s)}^{\mathcal{G}(s')} = \mathcal{G}(m)$$

für alle \mathcal{S} -Morphismen $m: s \rightarrow s'$, EQ_s die Gleichungsmenge von $\mathcal{D}(s)$ für alle \mathcal{S} -Objekte s und

$$EQ = \left\{ \left(X^{c_s}, c_s^t(l), c_s^t(r) \right) \mid s \in \text{Obj}(\mathcal{S}), (X, l, r) \in EQ_s \right\}$$

dann bildet

$$\left((c_s)_{s \in \text{Obj}(S)}, (Sig, EQ) \right)$$

einen Colimes für \mathcal{D} .

□

(Beweis → Anhang S. 287)

6.2.7 Folgerung (Covollständigkeit):

In der Kategorie $SP\mathcal{E}C$ existiert zu jedem endlichen Diagramm ein Colimes.

□

Wie den Signaturen wird auch Spezifikationen eine Kategorie von Algebren und Algebromorphismen als Semantik zugeordnet. Sie enthält für jede Spezifikation alle Algebren zur unterliegenden Signatur, die die Gleichungen der Spezifikation erfüllen (→ 6.2.1, 6.2.2) sowie alle zwischen diesen Objekten verkehrenden Signaturalgebromorphismen und bildet damit eine volle Unterkategorie (→ 5.1.2) der Signatursemantik.

6.2.8 Definition (Algebren und Algebromorphismen):

Sei $Spec = (SO, OP, EQ)$ eine Spezifikation mit unterliegender Signatur Sig , dann:

- (I) bildet eine Sig -Algebra A eine Algebra zu $Spec$, wenn alle Gleichungen aus EQ in A erfüllt sind,
- (II) bildet ein Sig -Algebromorphismus einen Algebromorphismus zu $Spec$, wenn sowohl seine Domäne als auch seine Codomäne $Spec$ -Algebren sind und
- (III) formen die $Spec$ -Algebren zusammen mit den $Spec$ -Algebromorphismen eine volle Unterkategorie von $\mathcal{ALG}(Sig)$, die wir mit $\mathcal{ALG}(Spec)$ bezeichnen.

□

6.2.9 Beispiel (Spezifikationsalgebren und Spezifikationsalgebromorphismen):

Die $Bool_{sig}$ -Algebra B (→ 6.1.8) ist keine Algebra zur Spezifikation $Bool$, da sie die Gleichung $(\{a\}, \wedge(F, a), F)$ (für $a = 0$) nicht erfüllt (→ 6.2.2). Aus demselben Grund handelt es sich bei der Nat_{sig} -Algebra A (→ 6.1.8) um keine Nat -Algebra, hier wird die Gleichung $(\{n, m\}, *(suc(n), m), +(*n, m), m)$ (z.B. für $n = 0$ und $m = 1$) nicht erfüllt.

Im Unterschied dazu erfüllt die Nat_{sig} -Algebra \mathbb{N}_{Alg} alle Nat -Gleichungen und ist damit ebenso eine Nat -Algebra, wie

$$\begin{aligned} \perp &= \\ \underline{\text{doms}} &\perp; \\ \underline{\text{fkts}} &\perp, s, a, m; \end{aligned}$$

Von jeder *Nat*-Algebra existiert, bis auf Umbenennung, genau ein *Nat*-Algebromorphismus nach \perp , dem terminalen Objekt¹⁰ in $\mathcal{ALG}(Nat)$. Ein derartig konstruiertes Objekt existiert zu jeder Spezifikation und wird jeweils mit \perp bezeichnet. □

Wie die Signaturalgebrakategorien enthalten auch die Kategorien der Spezifikationsalgebren jeweils ein initiales Objekt, die „*Quotiententermalgebra*“ (\rightarrow [EM85] 2.4–2.9). Sie ergibt sich aus der Termalgebra zur unterliegenden Signatur durch Faktorisierung bezüglich der von den Gleichungen der Spezifikation generierten Kongruenzrelation. Dabei handelt es sich um die kleinste Kongruenzrelation, die zu jeder Gleichung (X, l, r) und jeder Variablenbelegung *ass* der Variablen die Paare $(\overline{ass}(l), \overline{ass}(r))$ enthält. Dabei bezeichnet \overline{ass} die kanonische Erweiterung von *ass* auf Terme mit Variablen, die die enthaltenen Variablen wie *ass* abbildet und den Rest unverändert läßt.

Wir bezeichnen die Quotiententermalgebra zu einer Spezifikation *Spec* mit T_{Spec} und deren Elemente, d.h. die Kongruenzklassen, durch $[t]_{\equiv}$ für die Terme *t* aus T_{OP} . Wenn vom Kontext her klar ist, daß es sich um die von den *Spec*-Gleichungen induzierten Kongruenzklassen handelt, lassen wir den Index weg und schreiben einfach $[t]$. Als weitere notationelle Vereinfachung bezeichnen wir die Termalgebra T_{OP} zu der *Spec* unterliegenden Signatur als „*Spec*-Termalgebra“.

6.2.10 Lemma (Initiales Objekt):

Zu jeder Spezifikation *Spec* ist die „*Spec*-Quotiententermalgebra“, bezeichnet mit T_{Spec} , initiales Objekt in $\mathcal{ALG}(Spec)$. □

Beweis: Die Behauptung folgt aus [EM85] Theorem 3.17.

Da alle Algebren zu einer gegebenen Spezifikation *Spec* gleichzeitig Algebren zu der ihr unterliegenden Signatur *Sig* sind, gibt es eine Evaluierungsfunktion (\rightarrow 6.1.9) von der Termalgebra T_{OP} in die Quotiententermalgebra T_{Spec} . Aus deren Verträglichkeit mit den Operationen der Quotiententermalgebra folgt, daß dabei jeder Term auf seine Kongruenzklasse abgebildet wird. Mit der Initialität von T_{OP} in $\mathcal{ALG}(Sig)$ ergibt sich daraus für jede *Spec*-Algebra *A* der im folgenden Diagramm dargestellte kommutative Zusammenhang in $\mathcal{ALG}(Sig)$.

$$\begin{array}{ccc}
 T_{OP} & \xrightarrow{eval} & T_{Spec} \\
 & \searrow eval & \downarrow eval \\
 & & A
 \end{array}$$

¹⁰dual zum initialen Objekt

Damit lässt sich die Evaluierungsfunktion von T_{Spec} nach A aus der Evaluierungsfunktion von T_{OP} nach A konstruieren, indem jeder Kongruenzklasse $[t]$ das Algebraelement $eval(t)$ zugeordnet wird. Welcher Repräsentant der Kongruenzklasse dabei gewählt wird, ist für das Ergebnis unerheblich.

6.2.11 Definition (Evaluierungsfunktion):

Sei $Spec = (SO, OP, EQ)$ eine Spezifikation, T_{Spec} die $Spec$ -Quotiententermalgebra und A eine $Spec$ -Algebra, dann definieren wir den $Spec$ -Algebromorphismus $eval_A: T_{Spec} \rightarrow A$, genannt „Evaluierungsfunktion“, durch:

$$eval_{A_s}([t]) = (eval_{A_s}: T_{OP_s} \rightarrow A)(t)$$

für alle Sorten s in SO und Äquivalenzklassen $[t]$ in T_{Spec_s} .

□

Für jede $Spec$ -Algebra A bezeichnen wir sowohl die Evaluierungsfunktion von der Termalgebra als auch die Evaluierungsfunktion von der Quotiententermalgebra nach A mit $eval_A$, da es sich in beiden Fällen um Morphismen von initialen Objekten nach A handelt. Wenn aus dem jeweiligen Kontext nicht klar hervorgeht, welcher der beiden Morphismen gemeint ist, geben wir entweder den kompletten Bezeichner, z.B. $eval_A: T_{OP} \rightarrow A$, oder die Domäne an.

Ein durch eine Inklusionsbeziehung verbundenes Paar von Spezifikationen (Par, Bod) (kurz $Bod(Par)$), nennen wir eine „parameterisierte Spezifikation“ mit Parameter Par oder einfach parameterisierte Spezifikation. Prinzipiell wäre die Einschränkung auf Inklusionsbeziehungen für die weiteren Konstruktionen nicht notwendig, man könnte an deren Stelle beliebige Spezifikationsmorphismen zulassen, sie ist aber in vielen Fällen einfacher zu handhaben und für praktische Zwecke ausreichend.

6.2.12 Definition (Parameterisierte Spezifikationen):

Ein Spezifikationspaar (Par, Bod) bildet eine „parameterisierte Spezifikation“, notiert durch $Bod(Par)$, wenn Par in Bod enthalten ist.

□

6.2.13 Beispiel (Parameterisierte Spezifikation der freien abelschen Gruppen):

Die Spezifikation der Mengen Set (\rightarrow 6.2.5) ist in der nachfolgenden Spezifikation der freien abelschen Gruppen enthalten. Zusammen bilden sie die parameterisierte

Spezifikation $(Set)^\otimes$ mit Parameter Set .

$$\begin{array}{l}
 (Set)^\otimes = Set \cup \\
 \text{sorts } set^\otimes \\
 \text{opns } 0 : \quad \quad \quad \rightarrow set^\otimes \\
 \quad \quad + : set \quad \quad \rightarrow set^\otimes \\
 \quad \quad - : set \quad \quad \rightarrow set^\otimes \\
 \quad \quad \oplus : set^\otimes, set^\otimes \rightarrow set^\otimes \\
 \text{eqns } d : set; m_1, m_2, m_3 : set^\otimes; \\
 \quad \quad \oplus(+d, -d) \quad \quad = 0 \\
 \quad \quad \oplus(m_1, 0) \quad \quad = m_1 \\
 \quad \quad \oplus(m_1, m_2) \quad \quad = \oplus(m_2, m_1) \\
 \quad \quad \oplus(m_1, \oplus(m_2, m_3)) = \oplus(\oplus(m_1, m_2), m_3)
 \end{array}$$

Die Operation „+“ bettet die Elemente des Parameters in die Trägermenge der Gruppe ein, während „-“ die inversen Elemente erzeugt. „0“ steht für das neutrale Element. Die eigentliche Gruppenoperation wird durch „ \oplus “ spezifiziert.

□

Ein wesentliches Merkmal parameterisierter Spezifikationen ist die Möglichkeit sie mit Hilfe der Ersetzung des Parameters durch eine als „Aktualisierung“ bezeichnete Spezifikation an unterschiedliche Erfordernisse anzupassen. Damit ihre Eigenschaften dabei erhalten bleiben, lassen wir als Aktualisierungen nur Spezifikationen zu, die mit dem Parameter über einen Spezifikationsmorphismus in Beziehung stehen. Das Ergebnis einer derartigen als „Parameterübergabe“ bezeichneten Ersetzung ergibt sich aus der Komposition des Spezifikationsrumpfes mit der Aktualisierung. Die zu verschmelzenden Teile werden im Rahmen dieser Komposition durch den Parameter festgelegt, der mit beiden Komponenten über Spezifikationsmorphisms in Beziehung steht.

6.2.14 Definition (Parameterübergabe):

Sei $Bod(Par)$ eine parameterisierte Spezifikation und Act eine Spezifikation, dann bezeichnen wir:

(I) einen Spezifikationsmorphismus

$$h: Par \rightarrow Act$$

als „Parameterübergabemorphismus“ und

(II) die Spezifikation $Par_h(Act)$ im Pushout

$$\begin{array}{ccc}
 Par & \xrightarrow{\hookrightarrow} & Bod \\
 \downarrow h & (PO) & \downarrow c_1 \\
 Act & \xrightarrow{c_2} & Par_h(Act)
 \end{array}$$

als Ergebnis der von h induzierten Parameterübergabe.

□

Korrespondierend zur Einbettung des Parameters einer Spezifikation in den Spezifikationsrumpf lassen sich dessen Algebren in die Kategorie der Algebren zur Rumpfspezifikation einbetten. Dabei werden Trägermengen und Operationen zu Sorten u. Operationssymbolen der Rumpfspezifikation, die keine Entsprechung im Parameter haben, durch entsprechende Teile der Quotiententermalgebra zur Rumpfspezifikation ergänzt.

Da wir die zur Einbettung notwendige Erweiterung der Algebren (\rightarrow 6.2.15) und die darauf aufbauende „freie Konstruktion“ (\rightarrow 6.2.18) nicht nur zur Definition der Semantik parameterisierter Spezifikationen, sondern auch zur Konstruktion von Coequalizern (\rightarrow 5.3.5) benötigen, verallgemeinern wir die in [EM85] für Inklusionen angegebene Definition auf beliebige Spezifikationsmorphismen.

Die Konstruktion der „ A -Quotiententermalgebra“ (\rightarrow [EM85] 3.12) genannten Erweiterung einer Algebra A zur Domäne eines Spezifikationsmorphisms $h: Spec_1 \rightarrow Spec_2$ in eine Algebra zu seiner Codomäne erfolgt in drei Schritten. Zunächst werden die Signaturen von $Spec_1$ und $Spec_2$ jeweils um die Elemente der A -Trägermengen als zusätzliche Konstantensymbole erweitert. Im nächsten Schritt wird jedes Termpaar zur erweiterten $Spec_1$ -Signatur, dessen Terme sich zum gleichen Element in A auswerten lassen, in eine Gleichung zur erweiterten $Spec_2$ -Signatur transformiert (\rightarrow 6.2.3). Diese Gleichungen spiegeln insbesondere den Effekt der Funktionen in A wieder. Enthält A beispielsweise korrespondierend zum Operationssymbol op eine Funktion f mit $f(a) = a'$, dann werden die Terme $op(a)$ und a' durch die Evaluierungsfunktion auf a' abgebildet. Daraus ergibt sich durch Transformation die Gleichung $(\emptyset, h^{OP}(op)(a), a')$. Die erweiterte $Spec_2$ -Signatur zusammen mit den ursprünglichen und den neuen $Spec_2$ -Gleichungen ergibt die erweiterte $Spec_2$ -Spezifikation $Spec_2(A)$, aus deren Quotiententermalgebra im letzten Schritt nur noch die Konstanten zu den A -Elementen entfernt werden müssen. Die dadurch entstandene $Spec_2$ -Algebra ist im Unterschied zur $Spec_2(A)$ -Quotiententermalgebra immer dann nicht vollständig operationsgeneriert, wenn A diese Bedingung nicht erfüllt.

6.2.15 Definition (Erweiterung von Algebren):

Seien $Spec_i = (SO_i, OP_i, EQ_i)$ Spezifikationen mit unterliegenden Signaturen Sig_i ($i = 1, 2$), $h = (h^{SO}, h^{OP}): Spec_1 \rightarrow Spec_2$ ein Spezifikationsmorphismus und A eine $Spec_1$ -Algebra, dann definieren wir:

(I) zwei Familien von Konstanten

$$\begin{aligned} Const_1(A) &\stackrel{\text{def}}{=} \{ a: \rightarrow s \mid s \in SO_1 \wedge a \in A_s \}, \\ Const_2(A) &\stackrel{\text{def}}{=} \left\{ a: \rightarrow h^{SO}(s) \mid s \in SO_1 \wedge a \in A_s \right\}, \end{aligned}$$

(II) die „ A -Erweiterungen von Sig_1 und Sig_2 “

$$Sig_i(A) \stackrel{\text{def}}{=} (SO_i, OP_i \uplus Const_i(A))$$

mit Termalgebren $T_{OP_i(A)}$ für $i = 1, 2$,

(III) die „ A -Erweiterungen von h “

$$\begin{aligned} h^A: Sig_1(A) \rightarrow Sig_2(A) &\stackrel{\text{def}}{=} (h^{SO}, h^{OPA}) \\ \text{mit } h^{OPA}(op) &= \begin{cases} h^{OP}(op), & op \in OP_1; \\ a: \rightarrow h^{SO}(s), & op = a: \rightarrow s \wedge a \in A_s; \end{cases} \end{aligned}$$

für alle $Sig_1(A)$ -Operationen op ,

(IV) eine Menge von Gleichungen

$$Eqns(A) \stackrel{\text{def}}{=} \left\{ \left(\emptyset, h^{A^t}(t_1), h^{A^t}(t_2) \right) \mid \begin{array}{l} t_1, t_2 \in T_{OP_1(A)} \wedge \\ eval_A(t_1) = eval_A(t_2) \end{array} \right\}$$

wobei $h^{A^t}: T_{OP_1(A)} \rightarrow T_{OP_2(A)}$ die h^A -generierte Abbildung zwischen Termen bezeichnet (\rightarrow 6.2.3) und $eval_A: T_{OP_1(A)} \rightarrow A$ die Evaluierungsfunktion in $\mathcal{ALG}(Sig_1(A))$,

(V) die „ A -Erweiterung von $Spec_2$ “

$$Spec_2(A) \stackrel{\text{def}}{=} (Sig_2(A), EQ_2 \uplus Eqns(A))$$

(VI) und die „ A -Quotiententermalgebra“ in $\mathcal{ALG}(Spec_2)$

$$T_{Spec_2(A)} \stackrel{\text{def}}{=} \mathcal{V}(T_{Spec_2(A)})$$

wobei \mathcal{V} den, von der Inklusion $Spec_2 \subset Spec_2(A)$ generierten, Vergißfunktorkomplex bezeichnet.

□

6.2.16 Beispiel (Erweiterung der *Bool*-Algebra A):

Sei $Bool(X) = Bool \uplus (\emptyset, \{x : \rightarrow bool \mid x \in X\}, \emptyset)$ die Spezifikation, die sich aus der Erweiterung von $Bool$ mit einer Menge $X = x_1, \dots, x_n$ von Konstanten ergibt. In diesem Fall bildet die Inklusion von $Bool$ in $Bool(X)$ einen Spezifikationsmorphismus, bzgl. dessen die rechte der nachfolgenden Algebren die A -Erweiterung der links stehenden $Bool$ -Algebra A bildet.

$A =$ <u>doms</u> 0, 1; <u>fkts</u> 1, 0, <i>not</i> , <i>and</i> , <i>or</i> ; mit $not(x) = \begin{cases} 1 & ; x = 0 \\ 0 & ; x = 1 \end{cases},$ $and(x, y) = x * y,$ $or(x, y) = not(and(not(x), not(y)))$	$T_{Bool(X)}(A) =$ <u>doms</u> [0], [1], [x ₁], ..., [x _n], $[\wedge(x_1, 1)], \dots;$ <u>fkts</u> [1], [0], <i>not</i> , <i>and</i> , <i>or</i> ; mit $not(x) = [\neg(x)],$ $and(x, y) = [\wedge(x, y)],$ $or(a, b) = [\vee(x, y)]$
---	---

Im Unterschied zur zweielementigen Domäne von A enthält die $A(X)$ -Domäne unendlich viele Äquivalenzklassen, die ihrerseits unendlich viele Terme aus $T_{OP(X,A)}$, der Termalgebra zur Signatur der A -Erweiterung von $Bool(X)$, enthalten. So sind z.B. die Terme „1“, „T“, „ $\vee(x_1, 1)$ “, „ $\vee(x_1, \vee(1, x_2))$ “ Elemente in „[1]“. Diese Form der Erweiterung von Algebren um eine Menge von Konstanten spielt eine zentrale Rolle in Abschnitt 6.3 dieses Kapitels. □

Jede Algebra A ist mit ihrer bezüglich eines Spezifikationsmorphismus h gebildeten A -Quotiententermalgebra über eine Abbildung verbunden, die jedem Element seine Kongruenzklasse zuordnet. Schränkt man die Codomäne dieser Abbildung auf das Vergißbild der A -Quotiententermalgebra bezüglich h ein, erhält man einen Algebromorphismus mit universellen Eigenschaften (\rightarrow 5.1.10), der die A -Quotiententermalgebra bis auf Isomorphie eindeutig charakterisiert.

6.2.17 Satz (Universelle Abbildung zwischen Algebren):

Sei $h: Spec_1 \rightarrow Spec_2$ ein Spezifikationsmorphismus, A eine $Spec_1$ -Algebra und

$$u_A: A \rightarrow \mathcal{V}_h(T_{Spec_2}(A)) \quad \text{mit} \quad u_{A_s}(a) \stackrel{\text{def}}{=} [a]$$

für alle s aus SO_1 und a aus A_s , dann bildet das Paar $(u_A, T_{Spec_2}(A))$ eine \mathcal{V}_h -universelle Abbildung (\rightarrow 5.1.10). □

Beweis: Die Behauptung folgt aus [EM85] Theorem 7.16.

Existiert in der Codomäne eines Vergißfunktors \mathcal{V} zu jedem Objekt A eine universelle Abbildung (u_A, B) , spricht man von einer *freien Konstruktion* bezüglich \mathcal{V} . Freie

Konstruktionen können unter Ausnutzung der Eigenschaften universeller Abbildungen in eindeutiger Weise zu einem *freien Funktor* \mathcal{F} erweitert werden (\rightarrow S. 121). Die universellen Abbildungen bilden in diesem Fall eine natürliche Transformation von der Identität zur Komposition $\mathcal{V} \circ \mathcal{F}$.

6.2.18 Folgerung (Freier Funktor):

Sei $h: \text{Spec}_1 \rightarrow \text{Spec}_2$ ein Spezifikationsmorphismus, dann ist der Funktor

$$\mathcal{F}_h: \mathcal{ALG}(\text{Spec}_1) \rightarrow \mathcal{ALG}(\text{Spec}_2)$$

definiert durch:

$$\begin{array}{ccccc} A & \xrightarrow{u_A} & \mathcal{V}_h(T_{\text{Spec}_2}(A)) & & T_{\text{Spec}_2}(A) \\ f \downarrow & & \downarrow \mathcal{V}_h(\mathcal{F}_h(f)) & & \downarrow \mathcal{F}_h(f) \\ B & \xrightarrow{u_B} & \mathcal{V}_h(T_{\text{Spec}_2}(B)) & & T_{\text{Spec}_2}(B) \end{array}$$

für alle Morphismen $f: A \rightarrow B$ aus $\mathcal{ALG}(\text{Spec}_1)$, ein \mathcal{V}_h -freier Funktor. □

Zu jedem Spezifikationsmorphismus h ist jedes Element der Menge der \mathcal{V}_h -freien Funktoren bis auf Isomorphie gleich \mathcal{F}_h . Handelt es sich bei h um den Inklusionsmorphismus, der den Parameter Par einer parameterisierten Spezifikation $Bod(Par)$ mit dem Spezifikationsrumpf Bod verbindet, dann bildet diese Menge die Semantik von $Bod(Par)$.

6.2.19 Beispiel (Semantik der parameterisierten Spezifikation $(Set)^\circledast$):

Die Kategorie der Algebren zum Parameter Set , der parameterisierten Spezifikation $(Set)^\circledast$ (\rightarrow 6.2.13) ist isomorph zur Kategorie der Mengen \mathcal{SET} . Der freie Funktor $\mathcal{F}_\hookrightarrow$ zur Inklusion \hookrightarrow von Set in $(Set)^\circledast$ ordnet jeder Menge A eine Algebra zu, deren Trägermenge zur Sorte set^\circledast alle formalen Summen

$$\bigoplus_{a \in A} x_a * [a], \quad x_a \in \mathbb{Z}$$

enthält und damit zusammen mit der Operation „ \oplus “ eine frei über A erzeugte abelsche Gruppe (\rightarrow 5.2.1) mit neutralem Element $[0]$ bildet.

Die Kategorie der $(Set)^\circledast$ -Algebren ist isomorph zur Kategorie der freien abelschen Gruppen \mathcal{AG} (\rightarrow 5.2.6), während die Bilder von Funktionen unter $\mathcal{F}_\hookrightarrow$ den funktionsgenerierten Morphismen (\rightarrow 5.2.5) in \mathcal{AG} entsprechen. □

Sowohl Spezifikationsmorphismen als auch Spezifikationsalgebren und Spezifikationsalgramorphismen sind Spezialfälle der entsprechenden Konstruktionen für Signaturen. Aus diesem Grund müssen Vergißfunktoren zu Spezifikationsmorphismen nicht

gesondert erzeugt werden, die für Signaturmorphismen definierten Vergebefunktoren (\rightarrow 6.1.11) sind darauf gleichermaßen anwendbar. Daß dabei Spezifikationsalgebren auf Spezifikationsalgebren abgebildet werden, folgt aus der Bewahrung der Spezifikationsgleichungen durch die Spezifikationsmorphismen (\rightarrow 6.2.4). Damit lassen sich ebenso wie Signaturen und Signaturalgebren auch Spezifikationen und Spezifikationsalgebren zu einem *Spezifikationsrahmen* (\rightarrow 5.3.1) zusammenfassen.

6.2.20 Lemma (Spezifikationsrahmen):

Das Paar $(\mathcal{SPEC}, \mathcal{M}: \mathcal{SPEC}^{op} \rightarrow \mathcal{CATCAT})$ mit

(I) $\mathcal{M}(Spec) \stackrel{\text{def}}{=} \mathcal{ALG}(Spec)$ für alle Objekte $Spec$ aus \mathcal{SPEC} und

(II) $\mathcal{M}(h) \stackrel{\text{def}}{=} \mathcal{V}_h: \mathcal{ALG}(Spec_2) \rightarrow \mathcal{ALG}(Spec_1)$ für alle Morphismen $h: Spec_1 \rightarrow Spec_2$ aus \mathcal{SPEC}

bildet einen mit „ $\mathcal{SPEC}\mathcal{F}$ “ bezeichneten *Spezifikationsrahmen* (\rightarrow 5.3.1). □

(\rightarrow [EGR91] Beispiel 2.2.1)

Colimites innerhalb von Algebrakategorien einzelner Spezifikationen ergeben sich als Spezialfälle der entsprechenden Colimites in $\mathcal{GSPEC-ALG}$, bei denen die Spezifikationskomponente der generalisierten Morphismen jeweils von einer Identität gebildet wird. Aus diesem Grund konstruieren wir Coprodukte und Coequalizer im folgenden innerhalb der $\mathcal{SPEC}\mathcal{F}$ -generierten generalisierten Algebrakategorie $\mathcal{GSPEC-ALG}$ (\rightarrow 5.3.2).

Ähnlich Signaturen und Signaturmorphismen sind auch Algebren und Algebromorphismen aus Mengen und Abbildungen zwischen Mengen aufgebaut. Ein wichtiger Unterschied besteht darin, daß die Elemente der Trägermengen von Algebren zusätzlich über Operationen miteinander verknüpft sind. Bei der Konstruktion von Colimites diskreter Kategorien in $\mathcal{GSPEC-ALG}$ spielt dieser Unterschied keine Rolle, da die Trägermengen der Algebren bei der disjunkten Vereinigung (Coprodukt bei Mengen (\rightarrow 5.1.16)) unverändert¹¹ in das Colimesobjekt übernommen werden.

6.2.21 Lemma (Coprodukte für Algebren):

Sei \mathcal{S} eine endliche diskrete Kategorie, $\mathcal{G}: \mathcal{S} \rightarrow \mathcal{SPEC}$ ein Funktor mit

$$\mathcal{G}(s) = (SO_s, OP_s, EQ_s)$$

für alle Objekte s in \mathcal{S} und Coprodukt

$$\left((c_s)_{s \in \text{Obj}(\mathcal{S})}, (SO_c, OP_c, EQ_c) \right),$$

¹¹bis auf Isomorphie, da die evtl. die Namen der Elemente geändert werden müssen

$\mathcal{D}: \mathcal{S} \rightarrow \mathcal{GSP\mathcal{E}C}\text{-}\mathcal{ALG}$ ein Funktor mit

$$\mathcal{D}(s) = ((A_{so}^s)_{so \in SO_s}, OP_{A_s}) \in \mathcal{ALG}(\mathcal{G}(s))$$

für alle Objekte s in \mathcal{S} ,

$$A_{c_s^{SO}(so)}^c \stackrel{\text{def}}{=} A_{so}^s$$

für alle Objekte s aus \mathcal{S} und Sorten so aus SO_s Trägermengen und

$$c_s^{OP}(op)_{A_c}(a_1, \dots, a_n) \stackrel{\text{def}}{=} op_{A_s}(a_1, \dots, a_n)$$

für alle Objekte s aus \mathcal{S} , Operationssymbole op aus OP_s und a_i aus A_{so_i} , dann bildet

$$\left((c_s, id)_{s \in \text{Obj}(\mathcal{S})}, ((A_{so}^c)_{so \in SO_c}, (op_{A_c}: A_{so_1}^c \times \dots \times A_{so_n}^c \rightarrow A_{so}^c)_{op \in OP_c^{so_1 \dots so_n, so}}) \right)$$

ein Coprodukt für \mathcal{D} .

□

(Beweis \rightarrow Anhang S. 288)

Anders ist die Situation bei der Konstruktion von Coequalizern¹². In der Kategorie der Mengen \mathcal{SET} wird dabei die Codomäne der beiden Diagrammorphismen nach der kleinsten Äquivalenzrelation faktorisiert, die alle Paare $(f(a), g(a))$ für Diagrammorphismen f, g und Elemente a aus deren gemeinsamer Domäne enthält (\rightarrow 5.1.16). Um der zusätzlichen Verknüpfung durch die Operationen Rechnung zu tragen, wählt man bei Algebren anstelle der kleinsten Äquivalenzrelation die kleinste Kongruenzrelation, sodaß für alle Operationen op_B aus der Codomäne von f bzw. g und Elemente b_i bzw. b'_i mit $b_i \equiv b'_i$ ($i = 1, \dots, n$) aus den entsprechenden Trägermengen $op_B(b_1, \dots, b_n) \equiv op_B(b'_1, \dots, b'_n)$ gilt.

Darüber hinaus genügt es, Coequalizer für jede Spezifikation separat auf deren Algebrakategorie zu definieren, da bezüglich jedes Vergißfunktors \mathcal{V}_h ($h \in \text{Mor}(\mathcal{SPEC})$) und jedes Objekts A seiner Codomäne eine \mathcal{V}_h -universelle Abbildung zu A (\rightarrow 6.2.17) existiert. Damit ist Lemma 5.3.5 auf $\mathcal{GSP\mathcal{E}C}\text{-}\mathcal{ALG}$ anwendbar, das es gestattet, Coequalizer in generalisierten Modellkategorien aus universellen Abbildungen und lokalen Coequalizern¹³ zu bilden.

¹²Colimites zu Diagrammen der Form $s \rightrightarrows s'$

¹³Coequalizern innerhalb der einzelnen Modellkategorien

6.2.22 Lemma (Coequalizer für Algebren):

Sei $Spec = (SO, OP, EQ)$ eine Spezifikation, \mathcal{S} die von dem Graphen $s \rightrightarrows s'$ generierte Kategorie, $\mathcal{D}: \mathcal{S} \rightarrow \mathcal{ALG}(Spec)$ ein Funktor mit

$$\mathcal{D}(s) = A, \quad \mathcal{D}(s') = B, \quad \{ \mathcal{D}(m) \mid m: s \rightarrow s' \in \mathcal{S} \} = \{f, g\}$$

und $\equiv \subset (B_s)_{s \in SO} \times (B_s)_{s \in SO}$ die kleinste Kongruenzrelation, die für alle $Spec$ -Sorten s und A_s -Elemente a das Paar $(f_s(a), g_s(a))$ enthält, dann bildet

$$\left(c, \left((C_s)_{s \in SO}, (op_C)_{op \in OP} \right) \right) \quad \text{mit} \quad C_s = B_s / \equiv$$

für alle $Spec$ -Sorten s einen Coequalizer für \mathcal{D} in $\mathcal{ALG}(Spec)$. Dessen Morphismus $c = (c_s)_{s \in SO}$ ordnet allen B -Elementen ihre Kongruenzklasse zu. Die Funktionen

$$op_C: C_{s_1} \times \cdots \times C_{s_n} \rightarrow C_s$$

zu den Operationssymbolen aus $OP^{s_1 \cdots s_n, s}$ ergeben sich als kommutative Ergänzung des folgenden Diagramms

$$\begin{array}{ccc} B_{s_1} \times \cdots \times B_{s_n} & \xrightarrow{op_B} & B_s \\ \downarrow c_{s_1} \times \cdots \times c_{s_n} & \curvearrowright & \downarrow c_s \\ C_{s_1} \times \cdots \times C_{s_n} & \xrightarrow{op_C} & C_s \end{array}$$

□

(Beweis \rightarrow Anhang S. 289)

Jeder Colimes eines endlichen Diagramms läßt sich unter ausschließlicher Verwendung zweier Coprodukte und eines Coequalizers konstruieren (\rightarrow 5.1.17, 5.1.19). Da in $\mathcal{GSP\mathcal{E}C-ALG}$ beide Colimeskonstruktionen für jedes Diagramm der entsprechenden Form existieren, lassen sich in dieser Kategorie für alle endlichen Diagramme Colimites bilden.

6.2.23 Satz (Colimites in $\mathcal{GSP\mathcal{E}C-ALG}$):

In der von $\mathcal{SPEC\mathcal{F}}$ generierten, generalisierten Algebrakategorie $\mathcal{GSP\mathcal{E}C-ALG}$ existiert zu jedem endlichen Diagramm ein Colimes.

□

Beweis: Die Behauptung folgt mit Lemma 6.2.22 aus Lemma 5.3.5.

6.3 Spezifikationen mit Variablen

Jedem Term über den Operationen einer *Spec*-Algebra A ist ein Wert in A zugeordnet, z.B. dem \mathbb{N}_{Alg} -Term¹⁴ $add(suc(0), mult(suc^2(0), suc^4(0)))$ (\rightarrow 6.1.8) der Wert 9. In bestimmten Fällen möchten wir jedoch in der Lage sein Terme anzugeben, beispielsweise $add(x, y)$ oder $mult(suc^2(0), z)$, die je nach Substitution ausgezeichneter Symbole (hier x, y, z), für eine ganze Menge von Werten aus A stehen. Zu diesem Zweck erweitern wir die Algebra um diese „Variablensymbole“ als nullstellige Operationen.

Die Terme über den erweiterten Algebren haben quasi zwei Interpretationen, zum einen die Kongruenzklasse, die den Term selbst enthält, und zum anderen für jede Variablenbelegung die Interpretation des entsprechend substituierten Terms in A . Ein beabsichtigter Nebeneffekt dieser Konstruktion ist die Entstehung neuer zusammengesetzter Operationen. So läßt sich beispielsweise der Term $mult(suc^2(0), z)$ zusammen mit seinen Variablenbelegungen als Operation $2 * z : \mathbb{N} \rightarrow \mathbb{N}$ interpretieren.

Die Erweiterung einer *Spec*-Algebra A erfolgt in zwei Schritten, zunächst wird *Spec* um die Variablen als zusätzliche Konstanten zu $Spec(X)$ erweitert. Anschließend kann mit Hilfe des freien Funktors \mathcal{F} zur Inklusion von *Spec* in $Spec(X)$ aus A die erweiterte Algebra $\mathcal{F}(A)$ erzeugt werden. Passend zu den Erweiterungen der Spezifikationen und Algebren definieren wir im weiteren entsprechende Erweiterungen von Spezifikationsmorphismen und von generalisierten Algebromorphismen.

Zur Vereinfachung der folgenden Konstruktionen nehmen wir an, daß die Variablenfamilie X für alle Signaturen im wesentlichen gleich ist. Wir erhalten eine derartige Menge durch Indizierung einer fixen Variablenmenge X_{fix} mit den Sorten der jeweiligen Signatur, d.h.

$$X = (X_{fix,s})_{s \in SO} .^{15}$$

Im Rahmen dieser Annahme notieren wir im folgenden die Erweiterung einer Spezifikation $Spec = (SO, OP, EQ)$ um die Variablen aus X mit $Spec(X)$, d.h.

$$Spec(X) \stackrel{\text{def}}{=} (SO, OP \uplus X, EQ),$$

und die über einer *Spec*-Algebra A bezüglich der Inklusion von *Spec* in $Spec(X)$ frei konstruierte $Spec(X)$ -Algebra $\mathcal{F}(A)$ mit $A(X)$.

Die Elemente in den Trägermengen einer $Spec(X)$ -Algebra $A(X)$ sind Kongruenzklassen über den Elementen der Termalgebra zur A -Erweiterung von $Spec(X)$ (\rightarrow 6.2.15). Sie enthalten im allgemeinen Terme mit einer unterschiedlichen Anzahl von Variablen. So existiert beispielsweise in der $Bool(X)$ -Algebra $A(X)$ (\rightarrow 6.2.16) für jede Teilmenge X' von X ein Term in der Kongruenzklasse „[1]“, der alle Variablen aus

¹⁴ $suc^n(0)$ ist eine abkürzenden Schreibweise für die n -fache Schachtelung $suc(\dots(suc(0))\dots)$.

¹⁵ Die Variablen in X stehen in keinem Zusammenhang zu den Variablen in den Gleichungen einer Spezifikation, sie müssen vielmehr von diesen Variablen wohlunterscheidbar sein.

X' enthält („ $\vee(x_1, T)$ “ für $\{x_1\}$ oder „ $\vee(\vee(x_3, \wedge(x_5, x_{19})), T)$ “ für $\{x_3, x_5, x_{19}\}$). Aus diesem Grund müssen wir, um den $A(X)$ -Elementen die Menge ihrer Variablen zu zuordnen, für jede Kongruenzklasse einen eindeutigen Repräsentanten bestimmen. Dazu nehmen wir an, daß für jede Spezifikation eine beliebige aber feste Ordnung auf den Termen ihrer Signatur existiert, sodaß jede Termmenge ein kleinstes Element hat.

Da das Alphabet über dem die Elemente einer Termalgebra erzeugt sind, d.h. die Familie der Operationssymbole, in eine abzählbare Menge abbildbar ist, existiert dafür immer eine Totalordnung. Diese Ordnung läßt sich auf die Menge der Terme fortsetzen. Für praktische Zwecke wählt man die Termordnung am besten so, daß sie die Terme zunächst nach der Anzahl enthaltener Variablen sortiert. Diese Annahme ist jedoch keine Voraussetzung für die folgenden Definitionen.

Im folgenden notieren wir für eine Spezifikation $Spec$ und eine $Spec$ -Algebra A , die A -Erweiterung von $Spec(X)$ (\rightarrow 6.2.15) mit $Spec(X, A)$.

6.3.1 Definition (Projektion auf Variablen):

Sei $Spec$ eine Spezifikation, A eine $Spec$ -Algebra und $Spec(X, A)$ die A -Erweiterung von $Spec(X)$, dann definieren wir die Abbildung $var: A(X) \rightarrow 2^X$ durch:

$$var(a) = \begin{cases} \{x\}, & a = [x] \wedge x \in X; \\ \emptyset, & a = op \in OP; \\ \emptyset, & a = [a'] \wedge a' \in A; \\ var([t_1]) \cup \dots \cup var([t_n]), & min(a) = op(t_1, \dots, t_n); \end{cases}$$

für alle Elemente a aus $A(X)$, wobei min das, bezüglich einer beliebigen aber festen Ordnung auf $T_{OP(X,A)}$ -Termen, kleinste Element einer Termmenge liefert. \square

Mit Hilfe der Termordnung läßt sich die, sonst nur für Terme definierte, Variablen-substitution auf entsprechende Kongruenzklassen erweitern. Dazu wird die Substitution auf dem jeweils kleinsten Term einer Klasse durchgeführt und anschließend darüber die Kongruenzklasse gebildet. Der Weg über die Substitution des kleinsten Terms dient in diesem Zusammenhang lediglich der Definitionsvereinfachung, da die der jeweiligen Variablenmenge angepaßte Substitution jedes Terms innerhalb einer Kongruenzklasse zum gleichen Ergebnis führt.¹⁶

¹⁶In den Gleichungen der erweiterten Spezifikationen tauchen keine Variablen aus X auf. Damit können in einer Situation, in der zwei äquivalente Terme verschiedene Variablen oder an Stelle von Variablen Terme enthalten, diese Variablen durch beliebige Terme ersetzt werden. Als Ergebnis erhält man in jedem Fall wieder zu einander äquivalente Terme.

6.3.2 Definition (Variablensubstitution):

Sei $Spec$ eine Spezifikation, A eine $Spec$ -Algebra, X' eine Teilmenge von X , $Spec(X, A)$ die A -Erweiterung von $Spec(X)$ und

$$ass: X' \rightarrow A(X)$$

eine Variablenbelegung, dann definieren wir die Abbildung $\widetilde{ass}: A(X) \rightarrow A(X)$ durch:

$$\widetilde{ass}(a) = \begin{cases} ass(x), & a = [x] \wedge x \in X'; \\ a, & a = [x] \wedge x \in X \setminus X'; \\ a, & a = op \in OP; \\ a, & a = [a'] \wedge a' \in A; \\ op_{A(X)}(\widetilde{ass}([t_1]), \dots, \widetilde{ass}([t_n])), & min(a) = op(t_1, \dots, t_n); \end{cases}$$

für alle Elemente a aus $A(X)$, wobei min das, bezüglich einer beliebigen aber festen Ordnung auf $T_{OP(X,A)}$ -Termen, kleinste Element einer Termmenge liefert. \square

6.3.3 Beispiel (Variablensubstitution):

Setzt man für die $Bool(X)$ -Spezifikation (\rightarrow 6.2.16) eine Ordnung voraus, die die Terme zunächst nach der Anzahl enthaltener Variablen sortiert, ergibt $var([1])$ die leere Menge und $var([\wedge(x_1, \vee(x_7, 1))])$ die Menge $\{x_1\}$, da in $[\wedge(x_1, \vee(x_7, 1))]$ der bzgl. der Termordnung kleinere Term x_1 ¹⁷ enthalten ist. In diesem Fall führt die Variablenbelegung

$$ass: var([\wedge(x_1, \vee(x_7, 1))]) \rightarrow A(X) \quad \text{mit} \quad ass: x_1 \mapsto [1]$$

zur Substitution

$$\widetilde{ass}([\wedge(x_1, \vee(x_7, 1))]) = \widetilde{ass}([x_1]) = ass(x_1) = [1].$$

Würde man anstelle von x_1 den äquivalenten Term $\wedge(x_1, \vee(x_7, 1))$ als bezüglich der Termordnung kleinsten Term wählen, gelangt man zum selben Ergebnis

$$\begin{aligned} \widetilde{ass}([\wedge(x_1, \vee(x_7, 1))]) &= \wedge_{A(X)}(\widetilde{ass}([x_1]), \widetilde{ass}([\vee(x_7, 1)])) \\ &= \wedge_{A(X)}(ass(x_1), \vee_{A(X)}(\widetilde{ass}([x_7]), \widetilde{ass}([1]))) \\ &= \wedge_{A(X)}([1], \vee_{A(X)}([x_7], [1])) \\ &= [\wedge(1, 1)] \\ &= [1]. \end{aligned}$$

\square

¹⁷ $\vee(x_7, 1) = 1 \implies x_1 \in [\wedge(x_1, \vee(x_7, 1))]$

Da alle Spezifikationen im wesentlichen mit den gleichen Variablen erweitert werden, genügt es für eine entsprechende Erweiterung der Spezifikationsmorphismen, deren Abbildung der Operationssymbole so zu ergänzen, daß die den Variablen entsprechenden Konstantensymbole auf gleichnamige Symbole abgebildet werden.

6.3.4 Definition (Spezifikationstransformation):

Wir definieren die Abbildung $\mathcal{T}^S: Mor(\mathcal{SPEC}) \rightarrow Mor(\mathcal{SPEC})$ durch:

$$\mathcal{T}^S(h) \stackrel{\text{def}}{=} (h^{SO}, h^{OP}): Spec_1(X) \rightarrow Spec_2(X)$$

für alle Spezifikationsmorphismen $h = (h^{SO}, h^{OP}): Spec_1 \rightarrow Spec_2$ mit

$$h^{OP}(op) = \begin{cases} h^{OP}(op), & op \in OP; \\ x: \rightarrow h^{SO}(s), & op = x: \rightarrow s \wedge x \in X. \end{cases}$$

□

Die Erweiterungen von Spezifikationen und Spezifikationsmorphismen sind in ihrem Resultat nur von den Sorten der betreffenden Spezifikation(en) abhängig, die dabei selbst unverändert bleiben. Aus diesem Grund spielt es keine Rolle, ob der Colimes eines endlichen Diagramms zunächst auf den „normalen Spezifikationen“ konstruiert und dann erweitert oder ob er gleich für die erweiterten Spezifikationen gebildet wird. Die Sorten der jeweiligen Colimesobjekte und damit die durch die Erweiterung hinzukommenden Konstanten sind in beiden Fällen dieselben.

6.3.5 Lemma (Transformation von Colimites):

\mathcal{T}^S ist ein treuer Funktor, der endliche Colimites bewahrt.

□

(Beweis → Anhang S. 291)

Generalisierte Algebromorphismen, bestehend aus einer Spezifikations- und einer Algebrakomponente, verknüpfen die Algebren der generalisierten Modellkategorie $\mathcal{GSPEC-ALG}$ (→ 5.3.2). Die den Erweiterungen der Algebren entsprechenden Erweiterungen dieser Morphismen wollen wir im nächsten Schritt einführen. Dabei genügt es nicht, die Komponenten eines generalisierten Morphismus einzeln zu erweitern, da das erweiterte Vergebild¹⁸ seiner Codomäne (= Codomäne der erweiterten Algebrakomponente) nicht dem Vergebild¹⁹ der erweiterten Codomäne (= Codomäne der Algebrakomponente in der Erweiterung) entspricht. Wir werden diesen Sachverhalt im weiteren näher erläutern und die beiden Algebren durch einen von der Spezifikationskomponente generierten Algebromorphismus verknüpfen.

Für einen generalisierten Algebromorphismus $(h, f): A \rightarrow B$ ergibt sich die Spezifikationskomponente h' seiner Erweiterung $(h', f'): A(X) \rightarrow B(X)$ aus der Transformation von h mit \mathcal{T}^S . Die naheliegende Erweiterung von $f: A \rightarrow \mathcal{V}_h(B)$ mit dem

¹⁸unter der Spezifikationskomponente

¹⁹unter der erweiterten Spezifikationskomponente

freien Funktor \mathcal{F} zur Inklusion von $\text{dom}(h)$ in $\text{dom}(h' = \mathcal{T}^S(h))$ ergibt dagegen nicht die notwendige Algebrakomponente $f': A(X) \rightarrow \mathcal{V}_{h'}(B(X))$. Deren Codomäne $\mathcal{V}_{h'}(B(X))$ unterscheidet sich im allgemeinen von $\mathcal{V}_h(B)(X)$, der Codomäne von $\mathcal{F}(f)$, wie Beispiel 6.3.6 exemplarisch belegt. Diese Lücke wollen wir durch einen $\text{dom}(h)$ -Algebramorphismus schließen, der wie α im nachfolgenden Diagramm, $\mathcal{F}(f)$ zu f' ergänzt.

$$\begin{array}{ccccc}
 & & A(X) & \xrightarrow{f'} & \mathcal{V}_{h'}(B(X)) \\
 & & \downarrow \mathcal{F}(f) & \nearrow \alpha & \\
 A & & & & \\
 \downarrow f & & & & \\
 \mathcal{V}_h(B) & & \mathcal{V}_h(B)(X) & &
 \end{array}$$

6.3.6 Beispiel ($\mathcal{V}_h(A)(X) \neq \mathcal{V}_{\mathcal{T}^S(h)}(A(X))$):

Sei h die Inklusion von $Bool$ in Nat_R (\rightarrow 6.2.5), und $\mathcal{F}(T_{Nat_R}) = T_{Nat_R}(X)$ die Erweiterung von T_{Nat_R} mit Variablen. In diesem Fall enthält die $T_{Nat_R}(X)$ -Trägermenge zur Sorte $bool$ das Element $[< (x_1, x_2)]$. Da der Vergißfunktorkomplex $\mathcal{V}_{\mathcal{T}^S(h)}$ diese Trägermenge bewahrt, ist $[< (x_1, x_2)]$ auch Element der $\mathcal{V}_{\mathcal{T}^S(h)}(T_{Nat_R}(X))$ -Trägermenge zur Sorte $bool$.

Im Gegensatz dazu ist $[< (x_1, x_2)]$ in der $\mathcal{V}_h(T_{Nat_R})(X)$ -Trägermenge²⁰ nicht enthalten. Die Spezifikation $Bool(X)$ kennt kein Operationssymbol „ $<$ “, daher ist $< (x_1, x_2)$ kein Element der Termalgebra zur $\mathcal{V}_h(T_{Nat_R})$ -Erweiterung von $Bool(X)$, über der die Elemente von $\mathcal{V}_h(T_{Nat_R})(X)$ gebildet sind (\rightarrow 6.2.15).

□

Zu jedem Spezifikationsmorphismus $h: Spec_1 \rightarrow Spec_2$ existiert ein generalisierter Algebramorphismus $(h, eval)$ zwischen den Quotiententermalgebren T_{Spec_1} und T_{Spec_2} . Diesen Zusammenhang können wir bei der Konstruktion des Algebramorphismus von $\mathcal{V}_h(B)(X) = \mathcal{V}(T_{Spec_1(X, \mathcal{V}_h(B))})$ ²¹ nach $\mathcal{V}_{\mathcal{T}^S(h)}(B(X)) = \mathcal{V}_{\mathcal{T}^S(h)}(\mathcal{V}(T_{Spec_2(X, B)}))$ ²² nutzen. Dazu konstruieren wir zunächst aus h einen Spezifikationsmorphismus h^e zwischen der $\mathcal{V}_h(B)$ -Erweiterung von $Spec_1(X)$ und der B -Erweiterung von $Spec_2(X)$ (\rightarrow 6.2.15), der das nachfolgende Diagramm kommutativ ergänzt.

$$\begin{array}{ccccc}
 Spec_1 & \xrightarrow{\hookrightarrow} & Spec_1(X) & \xrightarrow{\hookrightarrow} & Spec_1(X, \mathcal{V}_h(B)) \\
 \downarrow h & \circlearrowright & \downarrow \mathcal{T}^S(h) & \circlearrowright & \downarrow h^e \\
 Spec_2 & \xrightarrow{\hookrightarrow} & Spec_2(X) & \xrightarrow{\hookrightarrow} & Spec_2(X, B)
 \end{array}$$

²⁰ $\mathcal{V}_h(T_{Nat_R})(X)$ ist isomorph zu $B(X)$ aus 6.2.16

²¹ $T_{Spec_1(X, \mathcal{V}_h(B))}$ bezeichnet die Quotiententermalgebra zur $\mathcal{V}_h(B)$ -Erweiterung von $Spec_1(X)$ und \mathcal{V} den Vergißfunktorkomplex zur Inklusion von $Spec_1(X)$ in $Spec_1(X, \mathcal{V}_h(B))$.

²² $T_{Spec_2(X, B)}$ bezeichnet die Quotiententermalgebra zur B -Erweiterung von $Spec_2(X)$ und \mathcal{V} den Vergißfunktorkomplex zur Inklusion von $Spec_2(X)$ in $Spec_2(X, B)$.

Mit Hilfe der Algebrakomponente des von h^e generierten Morphismus ($h^e, eval$) läßt sich durch Vergessen der hinzugefügten Konstanten in Domäne und Codomäne, d.h. durch Abbildung von $eval$ mit dem entsprechende Vergißfaktor, der gesuchte Algebramorphismus erzeugen. Dabei spielt der kommutative Zusammenhang zwischen den im vorangehenden Diagramm gezeigten Spezifikationsmorphisms und deren dualer Zusammenhang zu den Vergißfaktoren (\rightarrow 5.3.1) eine wesentliche Rolle. Sie implizieren $\mathcal{V}_{\hookrightarrow} \circ \mathcal{V}_{h^e} = \mathcal{V}_{\mathcal{T}^S(h)} \circ \mathcal{V}_{\hookrightarrow}$ und damit

$$\mathcal{V}_{\hookrightarrow}(\mathcal{V}_{h^e}(T_{Spec_1(X,B)})) = \mathcal{V}_{\mathcal{T}^S(h)}(\mathcal{V}_{\hookrightarrow}(T_{Spec_1(X,B)})) = \mathcal{V}_{\mathcal{T}^S(h)}(B(X)).$$

Das nachfolgende Diagramm, indem die geschweiften Klammern die „Grenzen“ der jeweiligen Algebrakategorien symbolisieren, veranschaulicht die beschriebene Konstruktion.

$$\begin{array}{ccccc}
 \underbrace{\mathcal{ALG}(Spec_1(X))} & \xleftarrow{\mathcal{V}_{\hookrightarrow}} & \underbrace{\mathcal{ALG}(Spec_1(X, \mathcal{V}_h(B)))} & \xleftarrow{\mathcal{V}_{h^e}} & \underbrace{\mathcal{ALG}(Spec_2(X, B))} \\
 \mathcal{V}_h(B)(X) & \xrightarrow{(\hookrightarrow, id)} & T_{Spec_1(X, \mathcal{V}_h(B))} & & \\
 \downarrow \mathcal{V}_{\hookrightarrow}(eval) & & \downarrow eval & & \\
 \mathcal{V}_{\mathcal{T}^S(h)}(B(X)) & \xrightarrow{(\hookrightarrow, id)} & \mathcal{V}_{h^e}(T_{Spec_2(X,B)}) & \xrightarrow{(h^e, id)} & T_{Spec_2(X,B)}
 \end{array}$$

6.3.7 Definition (Generierte Morphismen):

Sei $h: Spec_1 \rightarrow Spec_2$ ein Spezifikationsmorphismus, $Spec_i = (SO_i, OP_i, EQ_i)$ für $i = 1, 2$ und A eine $Spec_2$ -Algebra, dann definieren wir

(I) den Spezifikationsmorphismus $h^e: Spec_1(X, \mathcal{V}_h(A)) \rightarrow Spec_2(X, A)$, durch:
 $h^e \stackrel{\text{def}}{=} (h^{SO}, h^{OP_e})$ mit

$$h^{OP_e}(op) = \begin{cases} h^{OP}(op), & op \in OP_1; \\ x: \rightarrow h^{SO}(s), & op = x: \rightarrow s \wedge x \in X; \\ a: \rightarrow h^{SO}(s), & op = a: \rightarrow s \wedge a \in \mathcal{V}_h(A). \end{cases}$$

(II) den $Spec_1(X)$ -Algebramorphismus $h_A^{\mathcal{V}}: \mathcal{V}_h(A)(X) \rightarrow \mathcal{V}_{\mathcal{T}^S(h)}(A(X))$ durch:

$$h_A^{\mathcal{V}} \stackrel{\text{def}}{=} \mathcal{V}(eval: T_{Spec_1(X, \mathcal{V}_h(A))} \rightarrow \mathcal{V}_{h^e}(T_{Spec_2(X,A)}))$$

wobei \mathcal{V} den Vergißfaktor zur Inklusion von $Spec_1(X)$ in $Spec_1(X, \mathcal{V}_h(A))$ bezeichnet.

□

Die in 6.3.7 bezüglich eines Spezifikationsmorphisms h und einer $\text{cod}(h)$ -Algebra erzeugten Morphismen transformieren, wie im nachfolgenden Diagramm veranschaulicht, Algebromorphismen im Bild von $\mathcal{F} \circ \mathcal{V}_h$ in Bilder von $\mathcal{V}_{\mathcal{T}^S(h)} \circ \mathcal{F}$, wobei \mathcal{F} jeweils den freien Funktor bezüglich der Inklusion einer Spezifikation in ihre Erweiterung bezeichnet.

$$\begin{array}{ccccc}
 \mathcal{V}_h(A_1)(X) & \xrightarrow{h_{A_1}^{X_V}} & \mathcal{V}_{\mathcal{T}^S(h)}(A_1(X)) & & A_1 \\
 \mathcal{F} \circ \mathcal{V}_h(f) \downarrow & & \curvearrowright & & \downarrow f \\
 \mathcal{V}_h(A_2)(X) & \xrightarrow{h_{A_2}^{X_V}} & \mathcal{V}_{\mathcal{T}^S(h)}(A_2(X)) & & A_2 \\
 & & \downarrow \mathcal{V}_{\mathcal{T}^S(h)} \circ \mathcal{F}(f) & & \\
 & & & &
 \end{array}$$

Aufgrund dieser Eigenschaft bilden die h_A^V eine natürliche Transformation zwischen den Funktoren $\mathcal{F} \circ \mathcal{V}_h$ und $\mathcal{V}_{\mathcal{T}^S(h)} \circ \mathcal{F}$, die die $\text{cod}(h)$ -Algebrakategorie in die Algebrakategorie zu $\mathcal{T}^S(\text{dom}(h))$ abbilden.

6.3.8 Lemma (Natürliche Transformation):

Sei $h: \text{Spec}_1 \rightarrow \text{Spec}_2$ ein Spezifikationsmorphimus und \mathcal{F}_i der freie Funktor zur Inklusion von Spec_i in $\text{Spec}_i(X)$, dann bildet

$$h^V: \mathcal{F}_1 \circ \mathcal{V}_h \rightarrow \mathcal{V}_{\mathcal{T}^S(h)} \circ \mathcal{F}_2 \quad \text{mit} \quad h^V(A) = h_A^V$$

für alle Spec_2 -Algebren A , eine natürliche Transformation. □

(Beweis \rightarrow Anhang S. 296)

Mit der Transformation \mathcal{T}^S , den durch die Inklusionen von Spezifikationen in ihre Erweiterungen generierten freien Funktoren und den Ergänzungen ihrer Bilder mit Hilfe der natürlichen Transformation h^V können wir beide Komponenten generalisierter Algebromorphismen so ergänzen, daß wir zu jedem generalisierten Morphismus von A nach B eine erweiterte Version von $A(X)$ nach $B(X)$ erhalten.

6.3.9 Definition (Transformation generalisierter Algebromorphismen):

Wir definieren die Abbildung $\mathcal{T}^A: \text{Mor}(\mathcal{G}\text{SPEC-ALG}) \rightarrow \text{Mor}(\mathcal{G}\text{SPEC-ALG})$ durch:

$$\mathcal{T}^A((h, f)) \stackrel{\text{def}}{=} (\mathcal{T}^S(h), h_B^V \circ \mathcal{F}(f)): A(X) \rightarrow B(X)$$

für alle generalisierten Morphismen $(h, f): A \rightarrow B$ in $\mathcal{G}\text{SPEC-ALG}$, wobei \mathcal{F} den freien Funktor zur Inklusion von $\text{dom}(h)$ in $\mathcal{T}^S(\text{dom}(h))$ bezeichnet. □

Die Transformation \mathcal{T}^A ist mit den Identitäten und der Komposition in $\mathcal{G}\text{SPEC-ALG}$ verträglich. Darüber hinaus bildet sie Colimites zu endlichen Diagrammen wieder auf

Colimites zu Diagrammen der gleichen Form mit erweiterten Objekten und Morphismen ab, d.h. Colimites können auf „normalen Spezifikationen“ berechnet und im nachhinein mit Variablen erweitert werden.

6.3.10 Lemma (Eigenschaften von \mathcal{T}^A):

\mathcal{T}^A ist ein Funktor, der endliche Colimites bewahrt.

□

(Beweis → Anhang S. 298)

Zusammenfassung und Ausblick

Die vorliegende Arbeit erweitert eine der beiden, zum gegenwärtigen Zeitpunkt für praktische Anwendungen am geeignetsten erscheinenden Petrinetzklassen, die „Algebraischen High-Level-Netze“¹ und bringt sie, durch die Einführung von Kompositions- und Parameterisierungsoperationen, dem Einsatz als Spezifikationsmethode für nebenläufige Systeme einen wesentlichen Schritt näher. Durch den Verzicht auf praxisferne Einschränkungen bei Operationen und Netzen, etwa auf „Free Choice“ oder „State Machines“ und durch die Verknüpfung mit einer aussagekräftigen Semantik wird dieser Anspruch unterstrichen.

In Hinblick auf die Erweiterung der bestehenden Theorie höherer Petrinetze sind zwei Punkte von herausragender Bedeutung

- zum einen die „lose Modellsemantik“, deren Modelle sowohl in Hinblick auf Nebenläufigkeit als auch auf die erreichbaren Zustände, mindestens das spezifizierte Verhalten aufweisen und mit den initialen Modellen über eindeutige Referenzobjekte verfügen,
- zum anderen die „Prozeßtermsemantik“, die jedem Modell eine den Prozessen für B/E-Systeme vergleichbare und mit Hilfe zweier Operationen algebraisch komponierbare „true concurrency“ Semantik auf der Ablaufebene zuordnet und darüber hinaus sowohl unterschiedliche Sichten eines Ablaufs als auch Klassen von Abläufen repräsentierende abstrakte Abläufe zuläßt.

Dabei vermeidet die Modellsemantik die bisher übliche Asymmetrie zwischen Datentypen und Nebenläufigkeit bei der Modellbildung und ermöglicht damit die Unterscheidung von notwendigem und implementierungsabhängigem Verhalten für beide Aspekte einer Systemspezifikation.

Im Unterschied zum überwiegenden Teil der in Arbeiten aus dem Bereich der Petrinetztheorie verwendeten Beispiele, sind Systeme in der Praxis meist deutlich komplexer und erfordern daher neben einer formal eindeutigen Spezifikation, Möglichkeiten zur deren strukturierter Erstellung. Die im Rahmen dieser Arbeit eingeführten Kompositions- und Parameterisierungsoperationen, mit ihren bis auf Isomorphie

¹bei der anderen, aus unsere Sicht geeigneten Netzklasse handelt es sich um die „Colored Nets“ mit in „ML“ beschriebenen Datentypen von Kurt Jensen [Jen92, Jen94]

eindeutigen Ergebnissen, schaffen dafür die Grundlagen, indem sie ein Zusammen-setzen großer Spezifikationen aus überschaubaren Teilen und deren Wiederverwen-dung, durch Anpassung umgebungsabhängiger Bestandteile an sich ändernde Kon-texte, ermöglichen.

Den Strukturierungsmöglichkeiten auf Spezifikationsebene stehen entsprechende Ei-genschaften auf Seiten der Modell- und Prozeßtermsemantik gegenüber. Sie ermögli-chen eine zur Strukturierung der Systembeschreibungen analoge Strukturierung bei der Analyse von Modelleigenschaften, ohne die der Nutzen von Kompositions- und Parameterisierungsoperationen in der Praxis stark eingeschränkt wäre. Die wichtig-ten semantischen Eigenschaften von Komposition und Parameterisierung sind

- die „eindeutige Dekomponierbarkeit“ von Modellen mit deren Hilfe sich kom-ponentspezifische Sichten erzeugen und kompositionsbedingte Veränderungen an den Modellschnittstellen analysieren lassen,
- die „eindeutige Komponierbarkeit“ aller Modelle komponierter Spezifikationen aus Modellen der Komponenten die belegt, daß jedes Ereignis und jeder Zustand im Modell einer komponierten Spezifikation seinen Ursprung in einem Komponent-modell hat und
- die „freie Konstruktion“ der Modelle parameterisierter Spezifikationen über den Parametermodellen, die alle möglichen Effekte von Parameteraktualisierungen be-schreibt und Nebeneffekte auf Spezifikationsteile außerhalb des Parameters aus-schließt.
- Darüberhinaus gelten für Parameteraktualisierungen als Spezialfälle von Kompo-sitionen deren semantische Eigenschaften, d.h. aktualisierte Spezifikationen lassen sich in parameter-, rumpf- und schnittstellenspezifische Bestandteile zerlegen bzw. sich aus ihnen zusammensetzen.

Alle im Zusammenhang mit Komposition und Parameterisierung genannten Eigen-schaften sind formal auf der Basis von „generalisierten Modellmorphismen“ formuliert und können, mit Hilfe der Umsetzung dieser, die Modellgrenzen überschreitenden Ab-bildungen, in Zusammenhänge zwischen den korrespondierenden Prozeßtermseman-tiken übertragen werden.

Die Verwendung und praktische Bedeutung aller wichtigen Konstruktionen, inklusive der semantischen, demonstriert die Fallstudie eines „Flexiblen Fertigungssystems“. Darüber hinaus stellt dieses realitätsnahen Beispiel aus der Produktionstechnik die Notwendigkeit von Strukturierungsoperationen unter Beweis, da ohne die Zerlegung in Teilprobleme zumindest seine nächste Ausbaustufe nicht mehr handhabbar und ohne Parameterisierung eine Anpassung an Systeme mit unterschiedlichen Werkzeug-versorgungen unmöglich wäre.

Trotz des praktischen Bezugs der vorliegenden Arbeit — alle Konstruktionen werden in diesem Zusammenhang begründet — hat sie eher den Status einer Grundlagen-arbeit, als den einer ingenieursmäßig verwendbaren Methode. Das Schließen dieser

Lücke bildet einen Schwerpunkt zukünftiger Weiterentwicklungen. Dabei spielt, neben der notwendigen Integration in bestehende oder neu zu schaffende graphische Netzwerkzeuge, die Schaffung einer übergeordneten Spezifikationsebene, durch Zusammenführung der vorhandenen Strukturierungsmöglichkeiten zu einem einheitlichen Modulkonzept mit festgelegten Input/Output-Schnittstellen und Parameteranteil, eine wichtige Rolle. Für algebraische Datentypspezifikationen existiert eine entsprechende Modultheorie (\rightarrow [EM90]) seit längerem.

Auf eine Vereinheitlichung der Beschreibung und eine Unterstützung bei der Analyse von Systemeigenschaften zielt die Verknüpfung von Petrinetztheorie und einer, die Halbordnung der Ereignisse respektierenden, temporalen Logik ab. Für B/E-Systeme ohne weitere Strukturierungsmöglichkeiten wurden diesbezügliche Ergebnisse in den Arbeiten der Gruppe um W. Reisig (\rightarrow [Rei92, DGK⁺92]) vorgestellt. Eine auf höhere Netze übertragene und mit Komposition und Parameterisierung kompatible Verbreiterung unseres Ansatzes in diese Richtung wäre eine ideale Ergänzung des angesprochenen Modulkonzepts.

Ein weiterer, aus Sicht der Petrinetztheorie wichtiger Ansatzpunkt zukünftiger Arbeiten, ist die Entwicklung „abstrakter Invarianten“. Darunter verstehen wir Invarianten auf Spezifikationsebene, mit deren Hilfe sich quantifizierbare Eigenschaften aller Zustände ausdrücken und eventuell auch berechnen lassen, die für alle Modelle einer Spezifikation gelten. Die Anwendbarkeit der existierenden, auf den Arbeiten von W. Reisig [Rei91] basierenden Ansätze, ist stark von der Symmetrie der untersuchten Netzstrukturen abhängig. Die dabei verwendeten Konstruktionen sind nur in Ausnahmefällen in der Lage alle Invarianten zu beschreiben.

Die Einführung von internen Zuständen und Ereignissen, ähnlich den „ τ -Transitionen“ bei Prozeßalgebren, wäre eine aus theoretischer Sicht interessante Erweiterung der vorliegenden Arbeit. Dabei setzen die Spezifikationsmorphismen nur nach außen hin sichtbare Komponenten miteinander in Beziehung, als intern gekennzeichnete Stellen und Transitionen bleiben unberücksichtigt. Auf diese Weise lassen sich Spezifikationsverfeinerungen beschreiben, indem die Implementierungsdetails einer Ebene auf allen höheren Abstraktionsebenen als interne Komponenten ausgeblendet werden. Ein weiterer positiver Effekt liegt in dem erhöhten Freiheitsgrad bei der Zuordnung von Aktualisierungen zum formalen Parameter im Rahmen parameterisierter Spezifikationen. Allerdings verläßt die dazu notwendige Erweiterung der Netzabbildungen den Rahmen der linearen Algebra und erfordert dadurch die Entwicklung neuer Verfahren zur Berechnung von Colimites, als Grundlage von Parameterisierung und Komposition.

Anhang

Beweise zu Kapitel 1

Beweis (1.1.2 Funktoreigenschaft von \otimes):

Aus der Definition von $\|_-\|$ (\rightarrow 6.1.14) folgt für jede freie abelsche Gruppe G und jede Algebra A : $G \times \|A\|$ ist eine Menge. Damit ist die darüber gebildete freie abelsche Gruppe $(G \times \|A\|)^\otimes$ ein Element in \mathcal{AG}_m (\rightarrow 5.2.6).

Wenn $f: G_1^\otimes \rightarrow G_2^\otimes$ ein monotoner Morphismus in den freien Abelschen Gruppen ist und $k: A_1 \rightarrow A_2$ ein generalisierter Morphismus in $\mathcal{G}S\mathcal{I}\mathcal{G}-\mathcal{AL}\mathcal{G}$, dann ist $f \otimes k$ ein Morphismus in den freien Abelschen Gruppen; da er jedes Paar (g, a) aus $(G_1 \times \|A_1\|)$ auf eine formale Summe über $(G_2 \times \|A_2\|)$ abbildet und sich die übrigen Abbildungen als lineare Erweiterung daraus ergeben. Er ist monoton weil für jedes Element (g, a) aus $(G_1 \times \|A_1\|)$ gilt:

$$f \otimes k (g, a) = \bigoplus_{g' \in G_2} f(g)(g') * (g', k(a))$$

$$\implies f \otimes k \text{ ist monoton wenn } \forall g' \in G_2: f(g)(g') \geq 0 \quad 5.2.3$$

$$\implies f \otimes k \text{ ist monoton} \quad f \text{ ist monoton}$$

Damit sind die Bilder aller Paare aus einem monotonen Morphismus in den freien abelschen Gruppen und einem generalisierte Algebromorphismus, monotone Morphismen in den freien abelschen Gruppen. Darüberhinaus muß gezeigt werden (\rightarrow 5.1.3):

- \otimes bewahrt die Identitäten und
- \otimes bewahrt die Komposition.

(\otimes bewahrt Identitäten): Für alle \mathcal{AG}_m -Objekte G^\otimes , Algebren A , Elemente g aus G^\otimes und Elemente a aus $\|A\|$ gilt:

$$id_{G^\otimes} \otimes id_A (g, a) = \bigoplus_{g' \in G} id_{G^\otimes}(g)(g') * (g', id_A(a)) \quad 1.1.1$$

$$= 1 * (g, a) = (g, a)$$

(\otimes **bewahrt Kompositionen**): Wir nehmen an, daß $f_1: G_1^{\otimes} \rightarrow G_2^{\otimes}$, $f_2: G_2^{\otimes} \rightarrow G_3^{\otimes}$ monotone Morphismen in den freien Abelschen Gruppen sind und das es sich bei $k_1: A_1 \rightarrow A_2$, $k_2: A_2 \rightarrow A_3$ um generalisierte Morphismen in $\mathcal{G}S\mathcal{I}\mathcal{G}-\mathcal{A}\mathcal{L}\mathcal{G}$ handelt. Dann gilt für alle g aus G^{\otimes} und a aus $\|A\|$:

$$\begin{aligned}
& f_2 \circ f_1 \otimes k_2 \circ k_1 (g, a) \\
&= \bigoplus_{g_3 \in G_3} f_2(f_1(g)) (g_3) * (g_3, k_2 \circ k_1(a)) && 1.1.1 \\
&= \bigoplus_{g_3 \in G_3} f_2 \left(\bigoplus_{g_2 \in G_2} f_1(g)(g_2) * g_2 \right) (g_3) * (g_3, k_2 \circ k_1(a)) \\
&= \bigoplus_{g_3 \in G_3} \bigoplus_{g_2 \in G_2} f_1(g)(g_2) * f_2(g_2)(g_3) * (g_3, k_2(k_1(a))) && f_2 \text{ bewahrt } \oplus \\
&= \bigoplus_{g_2 \in G_2} \bigoplus_{g_3 \in G_3} f_1(g)(g_2) * f_2(g_2)(g_3) * (g_3, k_2(k_1(a))) && \oplus \text{ ist kommutativ} \\
&= \bigoplus_{g_2 \in G_2} f_1(g)(g_2) * \bigoplus_{g_3 \in G_3} f_2(g_2)(g_3) * (g_3, k_2(k_1(a))) && \oplus \text{ ist kommutativ} \\
&= \bigoplus_{g_2 \in G_2} f_1(g)(g_2) * f_2 \otimes k_2 (g_2, k_1(a)) && 1.1.1 \\
&= f_2 \otimes k_2 \left(\bigoplus_{g_2 \in G_2} f_1(g)(g_2) * (g_2, k_1(a)) \right) && f_2 \otimes k_2 \text{ bewahrt } \oplus \\
&= f_2 \otimes k_2 \circ f_1 \otimes k_1 (g, a) && 1.1.1
\end{aligned}$$

□

Beweis (1.2.2 Komponentweise Konstruktion von Colimites):

Das es sich bei

$$T_C^{\otimes} \begin{array}{c} \xrightarrow{Pre_C} \\ \xrightarrow{Post_C} \end{array} P_C^{\otimes} \otimes A$$

um ein CP/T-Netz und bei

$$(c_s^T, c_s^P, c_s^A)$$

um eine Familie von CP/T-Netzmorphismen handelt, folgt direkt aus den entsprechenden Definitionen in 1.2.2.

(**Verträglichkeit mit Diagrammorphismen**): Für alle \mathcal{S} -Morphismen $m: s \rightarrow s'$ gilt:

$$\begin{aligned}
& (c_s^T, c_s^P, c_s^A) \\
&= (\mathcal{D}^T(m) \circ c_{s'}^T, \mathcal{D}^P(m) \circ c_{s'}^P, \mathcal{D}^A(m) \circ c_{s'}^A) && \text{Eigensch. lok. Col.} \\
&= \mathcal{D}(m) \circ (c_{s'}^T, c_{s'}^P, c_{s'}^A) && 1.2.2
\end{aligned}$$

(**Couniverselle Eigenschaft**): Sei

$$((\bar{c}_s^T, \bar{c}_s^P, \bar{c}_s^A)_{s \in \mathcal{S}}, \bar{T}^\otimes \xrightleftharpoons[Post_{\bar{C}}]{Pre_{\bar{C}}} \bar{P}^\otimes \otimes \bar{A})$$

eine Senke für \mathcal{D} .

(Existenz): Da die Morphismen in \mathcal{CPTN} komponentweise gebildet werden, lassen sich daraus die Senken:

$$\begin{aligned} ((\bar{c}_s^T)_{s \in \mathcal{S}}, \bar{T}^\otimes), ((Pre_{\bar{C}} \circ \bar{c}_s^T)_{s \in \mathcal{S}}, \bar{P}^\otimes \otimes \bar{A}) & \text{ für } \mathcal{D}^T, \\ ((\bar{c}_s^P)_{s \in \mathcal{S}}, \bar{P}^\otimes) & \text{ für } \mathcal{D}^P \text{ und} \\ ((\bar{c}_s^A)_{s \in \mathcal{S}}, \bar{A}) & \text{ für } \mathcal{D}^A \end{aligned}$$

bilden. Das impliziert wegen der Eindeutigkeit der Colimites für \mathcal{D}^T , \mathcal{D}^P und \mathcal{D}^A die Existenz und Eindeutigkeit der folgenden Morphismen:

$$\begin{aligned} \hat{c}^T: T^\otimes &\rightarrow \bar{T}^\otimes, & \text{ mit } \hat{c}^T \circ c_s^T &= \bar{c}_s^T & \text{ für alle } s \text{ in } \mathcal{S} \\ \tilde{c}^T: T^\otimes &\rightarrow \bar{P}^\otimes \otimes \bar{A}, & \text{ mit } \tilde{c}^T \circ c_s^T &= Pre_{\bar{C}} \circ \bar{c}_s^T & \text{ für alle } s \text{ in } \mathcal{S} \\ \hat{c}^P: P^\otimes &\rightarrow \bar{P}^\otimes, & \text{ mit } \hat{c}^P \circ c_s^P &= \bar{c}_s^P & \text{ für alle } s \text{ in } \mathcal{S} \\ \hat{c}^A: A &\rightarrow \bar{A}, & \text{ mit } \hat{c}^A \circ c_s^A &= \bar{c}_s^A & \text{ für alle } s \text{ in } \mathcal{S} \end{aligned}$$

Diese Morphismen lassen sich zu einer Abbildung

$$(\hat{c}^T, \hat{c}^P, \hat{c}^A): T^\otimes \xrightleftharpoons[Post_C]{Pre_C} P^\otimes \otimes A \rightarrow \bar{T}^\otimes \xrightleftharpoons[Post_{\bar{C}}]{Pre_{\bar{C}}} \bar{P}^\otimes \otimes \bar{A},$$

zusammenfügen.

Da \bar{c}_s^T und \bar{c}_s^P als Komponenten von CP/T-Netzmorphismen monoton sein müssen und \hat{c}^T , \hat{c}^P mit diesen Morphismen verträgliche Morphismen in den freien abelschen Gruppen sind folgt, daß auch sie monoton sind. \hat{c}^A ist ein generalisierter Morphismus in $\mathcal{GSIG-ALG}$. Damit fehlt nur noch der Nachweis der Verträglichkeit mit den Markenflußmorphismen, d.h.

$$Pre_{\bar{C}} \circ \hat{c}^T = \hat{c}^P \otimes \hat{c}^A \circ Pre_C \quad \text{und} \quad Post_{\bar{C}} \circ \hat{c}^T = \hat{c}^P \otimes \hat{c}^A \circ Post_C.$$

$$\begin{aligned} \forall s \in \mathcal{S}: Pre_{\bar{C}} \circ \bar{c}_s^T &= Pre_{\bar{C}} \circ \hat{c}^T \circ c_s^T & \bar{c}_s^T &= c_s^T \circ \hat{c}^T \\ \implies \tilde{c}^T &= Pre_{\bar{C}} \circ \hat{c}^T & & \text{Eindeutigkeit v. } \tilde{c}^T \end{aligned}$$

und

$$\begin{aligned} \forall s \in \mathcal{S}: Pre_{\bar{C}} \circ \bar{c}_s^T & \\ &= \bar{c}_s^P \otimes \bar{c}_s^A \circ Pre_s & \text{Morph. Eigenschaft} \\ &= \hat{c}^P \circ c_s^P \otimes \hat{c}^A \circ c_s^A \circ Pre_s & \text{Eigenschaft v. } \hat{c}^P, \hat{c}^A \end{aligned}$$

$$\begin{aligned}
&= \widehat{c}^P \otimes \widehat{c}^A \circ c_s^P \otimes c_s^A \circ Pre_s && \text{Funktoreigensch. v. } \otimes \\
&= \widehat{c}^P \otimes \widehat{c}^A \circ Pre_C \circ c_s^T && 1.2.2 \\
&\implies \widetilde{c}^T = \widehat{c}^P \otimes \widehat{c}^A \circ Pre_C && \text{Eindeutigkeit v. } \widetilde{c}^T \\
&\implies \\
&Pre_{\overline{c}} \circ \widetilde{c}^T = \widetilde{c}^T = \widehat{c}^P \otimes \widehat{c}^A \circ Pre_C
\end{aligned}$$

Auf die gleiche Weise erfolgt der Beweis für $Post_C$.

Die Kompatibilität von $(\widetilde{c}^T, \widehat{c}^P, \widehat{c}^A)$ mit den Morphismen der Senke, d.h.

$$(\widetilde{c}^T, \widehat{c}^P, \widehat{c}^A) \circ (c_s^T, c_s^P, c_s^A) = (\overline{c}_s^T, \overline{c}_s^P, \overline{c}_s^A)$$

folgt aus der Verträglichkeit von \widehat{c}^T mit \overline{c}_s^T , \widehat{c}^P mit \overline{c}_s^P und \widehat{c}^A mit \overline{c}_s^A für alle s aus \mathcal{S} , zusammen mit der komponentweisen Komposition von Morphismen in \mathcal{CPTN} .

(Eindeutigkeit): Die Eindeutigkeit von $(\widetilde{c}^T, \widehat{c}^P, \widehat{c}^A)$ mit der Eigenschaft verträglich mit den Morphismen $(\overline{c}_s^T, \overline{c}_s^P, \overline{c}_s^A)$ zu sein, folgt aus der Eindeutigkeit seiner Komponenten, die mit den entsprechenden Teilen von $(\overline{c}_s^T, \overline{c}_s^P, \overline{c}_s^A)$ verträglich sein müssen.

□

Beweis (1.2.5 Zerlegungen bewahren die Netzstruktur):

Wir gehen im weiteren davon aus, daß N ein CP/T-Netz ist, $h: N_0 \rightarrow N$ ein CP/T-Netzmorphimus und \mathcal{DC}_h die Zerlegung von N durch h .

Notation: Um die Notation intuitiver zu gestalten, notieren wir die Teile $\mathcal{DC}_h(s)$ der Zerlegung durch N_s für alle s in \mathcal{S} .

(**Senke**): Die drei couniversellen Abbildungen $\hookrightarrow_{N_{is}, N}$, $\hookrightarrow_{N_{im}, N}$ und $\hookrightarrow_{N_{co}, N}$ enthalten jeweils Inklusionen als Stellen- und Transitionskomponente und die Identität als Algebraabbildung. Inklusionen sind monotone Morphismen in den freien abelschen Gruppen und die Identitäten sind Morphismen in $\mathcal{GSIG-ALG}$. Damit muß nur noch die Verträglichkeit der Netzabbildungen mit den Markenflußmorphismen gezeigt werden.

(Verträglichkeit v. $\hookrightarrow_{N_{is}, N}$): N_{is} enthält keine Transitionen und $\hookrightarrow_{T_{is}^{\otimes}, T^{\otimes}}$, $\hookrightarrow_{P_{is}^{\otimes}, P^{\otimes}} \otimes id$, Pre_{is} , $Post_{is}$, Pre , $Post$ sind Morphismen in den freien abelschen Gruppen, d.h. sie bilden das neutrale Element jeder freien abelschen Gruppe wieder auf das neutrale Element ab. Zusammen mit der Tatsache das die freie abelsche Gruppe über der leeren Menge nur aus dem neutralen Element besteht, folgt daraus:

$$\hookrightarrow_{P_{is}^{\otimes}, P^{\otimes}} \otimes id \circ Pre_{is} = 0: \{0\}^{\otimes} \rightarrow \{0\}^{\otimes} = Pre \circ \hookrightarrow_{T_{is}^{\otimes}, T^{\otimes}}$$

und die selbe Eigenschaft für $Post$.

(Verträglichkeit v. $\hookrightarrow_{N_{im},N}$ u. $\hookrightarrow_{N_{co},N}$): $Pre_{im}, Post_{im}, Pre_{co}, Post_{co}$ sind Einschränkungen der entsprechenden Markenflußmorphisme in N . Die Definition der Stellenvektoren P_{im}, P_{co} und die Übernahme der Algebra aus N in N_{im}, N_{is} stellt sicher das die Bilder dieser Morphisme definiert sind. Daraus folgt:

$$\begin{aligned} \hookrightarrow_{P_{im}^{\otimes}, P^{\otimes}} \otimes id \circ Pre_{im} &= Pre \circ \hookrightarrow_{T_{im}^{\otimes}, T^{\otimes}} \quad , \\ \hookrightarrow_{P_{im}^{\otimes}, P^{\otimes}} \otimes id \circ Post_{im} &= Post \circ \hookrightarrow_{T_{im}^{\otimes}, T^{\otimes}} \quad , \\ \hookrightarrow_{P_{co}^{\otimes}, P^{\otimes}} \otimes id \circ Pre_{co} &= Pre \circ \hookrightarrow_{T_{co}^{\otimes}, T^{\otimes}} \quad , \\ \hookrightarrow_{P_{co}^{\otimes}, P^{\otimes}} \otimes id \circ Post_{co} &= Post \circ \hookrightarrow_{T_{co}^{\otimes}, T^{\otimes}} \quad . \end{aligned}$$

(Verträglichkeit mit Diagrammorphisme): Ausgenommen die Identitäten, gibt es zwei Morphisme in \mathcal{S} :

$$\begin{aligned} m_1: is \rightarrow im \quad \text{mit} \quad \mathcal{DC}_h(m_1) &= \hookrightarrow_{N_{is}, N_{im}} \quad \text{und} \\ m_2: is \rightarrow co \quad \text{mit} \quad \mathcal{DC}_h(m_2) &= \hookrightarrow_{N_{is}, N_{co}} \quad . \end{aligned}$$

Wir zeigen die Verträglichkeit der Colimesmorphisme mit den Diagrammorphisme nur für den Ersten, der Beweis für den zweiten Morphismus ergibt sich durch Umbenennung.

$$\begin{aligned} &\hookrightarrow_{N_{im}, N} \circ \hookrightarrow_{N_{is}, N_{im}} \\ &= (\hookrightarrow_{T_{im}^{\otimes}, T^{\otimes}} \circ \hookrightarrow_{T_{is}^{\otimes}, T_{im}^{\otimes}}, \hookrightarrow_{P_{im}^{\otimes}, P^{\otimes}} \circ \hookrightarrow_{P_{is}^{\otimes}, P_{im}^{\otimes}}, id \circ id) \quad 1.1.6 \\ &= (\hookrightarrow_{T_{is}, T}, \hookrightarrow_{P_{is}, P}, id) \\ &= \hookrightarrow_{N_{is}, N} \end{aligned}$$

(Couniverselle Eigenschaft): Sei $((\bar{c}_s)_{s \in \mathcal{S}}, \bar{N})$ eine weitere Senke für \mathcal{DC}_h . Wir zeigen, daß in diesem Fall eindeutig ein Morphismus

$$\hat{c}: N \rightarrow \bar{N} \quad \text{mit} \quad \bar{c}_s = \hat{c} \circ \hookrightarrow_{\mathcal{DC}_h(s), N}$$

für alle s in \mathcal{S} existiert.

Wir definieren \hat{c} durch:

$$\hat{c} \stackrel{\text{def}}{=} \bar{c}_{im} \cup \bar{c}_{co}.$$

Das ist eine gültige Definition, weil folgende Gleichungen gelten:

$$N_{im} \cup N_{co} = N, \quad N_{im} \cap N_{co} = N_{is}$$

und durch die Verträglichkeit der Morphisme der Senke mit den Diagrammorphisme, sichergestellt ist, daß gleiche Elemente im Ursprung von \bar{c}_{im} und \bar{c}_{co} gleich

abgebildet werden. Aus dieser Definition von \hat{c} läßt sich direkt für alle s in \mathcal{S} die Gleichung: $\bar{c}_s = \hat{c} \circ \hookrightarrow_{\mathcal{DC}_h(s), N}$ ableiten.

Die Eindeutigkeit von \hat{c} folgt aus der vollständigen Überdeckung von N mit den Teilen der Zerlegung und der Injektivität der Colimesmorphisimen.

□

Beweis (1.3.5 Eigenschaften des Paralleloperators):

(I):

Wir nehmen an, daß N ein CP/T-Netz ist und zeigen:

- die Bilder von \parallel_N sind in \mathcal{ST}_N enthalten,
- \parallel_N bewahrt die Identitäten und
- \parallel_N bewahrt Komposition.

(Die Bilder von \parallel_N sind in \mathcal{ST}_N enthalten): Für alle \mathcal{ST}_N -Morphismenpaare f, g gilt:

$$\exists n \in \mathbb{N}: f = [f'_1]; \dots; [f'_n] \quad \text{und} \quad g = [g'_1]; \dots; [g'_n]$$

$$\text{mit } f'_i, g'_i \in (T \cup (P_\times \parallel A \parallel))^\otimes \quad \text{für } i = 1, \dots, n.$$

Es genügt eine Konstante (n) für beide Morphismen, da man den kürzeren von beiden durch Komposition mit der passenden Identität beliebig verlängern kann ohne dabei seinen Wert zu ändern.

Wir zeigen zunächst durch Induktion über n ,

(a):
$$(f_1; \dots; f_n) \parallel (g_1; \dots; g_n) = f_1 \parallel g_1; \dots; f_n \parallel g_n$$

für $f_i = [f'_i]$ und $g_i = [g'_i]$ mit $f'_i, g'_i \in (T \cup (P_\times \parallel A \parallel))^\otimes$ und $i = 1, \dots, n$. Dabei machen wir von der Assoziativität der Morphismenkomposition Gebrauch. Aus dieser Eigenschaft folgt:

(b):
$$f; (g; h) = (f; g); h.$$

(Induktionsbasis): $n = 2$.

Mit der Definition des Paralleloperators (\rightarrow 1.3.3) gilt:

$$(f_1; f_2) \parallel (g_1; g_2) = f_1 \parallel g_1; f_2 \parallel g_2$$

(Induktionsschritt): $n + 1$.

$$(f_1; \dots; f_n; f_{n+1}) \parallel (g_1; \dots; g_n; g_{n+1})$$

$$= \left((f_1; \dots; f_n) \parallel (g_1; \dots; g_n) \right); f_{n+1} \parallel g_{n+1} \quad 1.3.3, \text{ (b)}$$

$$= f_1 \parallel g_1; \dots; f_n \parallel g_n; f_{n+1} \parallel g_{n+1} \quad \text{Induktionsannahme}$$

Wir zeigen die eigentliche Behauptung durch Induktion über n (\rightarrow s.o.).

(Induktionsbasis): $n = 1$.

Daraus folgt: $f \parallel g = [f'_1] \parallel [g'_1] = [f'_1 \oplus g'_1] \in \mathcal{ST}_f$ (\rightarrow 1.3.1, 1.3.3).

(Induktionsschritt): $n + 1$.

Aus (a) folgt:

$$f \parallel g = f_1 \parallel g_1; \dots; f_n \parallel g_n; f_{n+1} \parallel g_{n+1}.$$

Aufgrund der Induktionsannahme sind

$$f_1 \parallel g_1; \dots; f_n \parallel g_n \quad \text{und} \quad f_{n+1} \parallel g_{n+1}$$

Morphismen in \mathcal{ST}_N und \mathcal{ST}_N ist gegenüber der Komposition von Morphismen abgeschlossen also auch gegenüber der sequentiellen Komposition über „;“.

Mit unseren Annahmen gilt: $\text{cod}(f_i) = \text{dom}(f_{i+1})$, $\text{cod}(g_i) = \text{dom}(g_{i+1})$ und damit

$$\text{cod}(f_i \parallel g_i) = \text{dom}(f_{i+1} \parallel g_{i+1})$$

für $1 \leq i \leq n$. Daraus folgt:

$$\text{cod}(f_1 \parallel g_1; \dots; f_n \parallel g_n) = \text{cod}(f_n \parallel g_n) = \text{dom}(f_{n+1} \parallel g_{n+1}),$$

d.h. die gezeigten Morphismen sind komponierbar und

$$f \parallel g = f_1 \parallel g_1; \dots; f_n \parallel g_n; f_{n+1} \parallel g_{n+1}$$

ist ein Morphismus in \mathcal{ST}_N .

(\parallel_N **bewahrt die Identitäten**): Für alle \mathcal{ST}_N -Objekte v, w gilt:

$$\text{id}_v \parallel \text{id}_w = [v] \parallel [w] = [v \oplus w] = \text{id}_{v \oplus w}$$

(\rightarrow 1.3.1(V), 1.3.3).

(\parallel_N **bewahrt die Komposition**): Für alle komponierbaren Morphismen $[f], [g]$ und $[r], [u]$ gilt:

$$[g] \circ [f] \parallel [u] \circ [r] = [f; g] \parallel [r; u] = [g] \parallel [u] \circ [f] \parallel [r]$$

(\rightarrow 1.3.1(VI), 1.3.3). □

(II):

Wir nehmen o.B.d.A. an:

$$f = [f'_1; \dots; f'_n] \quad \text{und} \quad g = [g'_1; \dots; g'_n]$$

$$\text{mit } f'_i, g'_i \in (T \cup (P_{\times} \parallel A \parallel))^\circ \quad \text{für } i = 1, \dots, n$$

und zeigen durch Induktion über n

$$f \parallel g = g \parallel f.$$

(Induktionsbasis): $n = 1$.

Daraus folgt: $f \parallel g = [f'_1 \oplus g'_1] = [g'_1 \oplus f'_1] = g \parallel f$ (\rightarrow 1.3.3, 5.2.1).

(Induktionsschritt): $n + 1$.

$$\begin{aligned} f \parallel g &= (f'_1; \dots; f'_{n+1}) \parallel (g'_1; \dots; g'_{n+1}) \\ &= f'_1 \parallel g'_1; \dots; f'_{n+1} \parallel g'_{n+1} && \text{(a)} \\ &= g'_1 \parallel f'_1; \dots; g'_{n+1} \parallel f'_{n+1} && \text{Induktionsannahme} \\ &= (g'_1; \dots; g'_{n+1}) \parallel (f'_1; \dots; f'_{n+1}) && \text{(a)} \\ &= g \parallel f \end{aligned}$$

□

□

Beweis (1.3.8 Eigenschaften der Schrittsemantikabbildung):

(I):

Wir nehmen an, daß $f: N_1 \rightarrow N_2$ ein CP/T-Netzmorphismus ist und zeigen:

- die Definition von \mathcal{ST}_f ist deterministisch, d.h. unabhängig von der Wahl der Repräsentanten der Äquivalenzklassen,
- das Bild von \mathcal{ST}_f ist in \mathcal{ST}_{N_2} enthalten,
- \mathcal{ST}_f bewahrt die Identitäten und
- \mathcal{ST}_f bewahrt die Komposition.

(Die Definition von \mathcal{ST}_f ist deterministisch): Es muß für alle \mathcal{ST}_{N_1} -Morphismen $[v], [w]$ gezeigt werden:

$$[v] = [w] \implies \mathcal{ST}_f([v]) = \mathcal{ST}_f([w]).$$

Die Elemente der Äquivalenzklassen, sind die Element aus Mor , d.h. sie haben die Form

$$v_1; \dots; v_n \quad \text{mit} \quad v_i \in (T_1 \cup (P_1 \times \|A_1\|))^{\otimes}.$$

Damit können wir unsere Behauptung durch strukturelle Induktion über den Aufbau der Mor -Terme führen.

Zunächst zeigen wir, ebenfalls durch strukturelle Induktion,

$$(a): \quad \widehat{f}(dom(v)) = dom(\widehat{f}(v))$$

für alle v aus $(T_1 \cup (P_1 \times \|A_1\|))^{\otimes}$.

(Induktionsbasis): $v \in (T_1 \cup (P_1 \times \|A_1\|))$

Daraus folgt: $v \in T_1$ oder $v \in (P_1 \times \|A_1\|)$.

$v \in T_1$

Annahme

\implies

$$\widehat{f}(dom(v))$$

$$= f^P \otimes f^A(Pre_1(v))$$

1.3.7, 1.3.1(I.b)

$$= Pre_2(f^T(v))$$

1.1.4

$$= dom(f^T(v))$$

1.3.1(I.b)

$$= dom(\widehat{f}(v))$$

1.3.7

$v \in P_1 \times \|A_1\|$

Annahme

\implies

$$\widehat{f}(dom(v))$$

$$= f^P \otimes f^A(v)$$

1.3.7, 1.3.1(I.a)

$$= dom(f^P \otimes f^A(v))$$

1.3.7, 1.3.1(I.b)

$$= dom(\widehat{f}(v))$$

1.3.7

(Induktionsschritt): $v = r \oplus u$

$$\widehat{f}(dom(v))$$

$$= \widehat{f}(dom(r \oplus u))$$

$$= \widehat{f}(dom(r) \oplus dom(u))$$

1.3.1(I.c)

$$= \widehat{f}(dom(r)) \oplus \widehat{f}(dom(u))$$

$\widehat{f} \in \mathcal{AG}$

$$\begin{aligned}
&= \text{dom}(\widehat{f}(r)) \oplus \text{dom}(\widehat{f}(u)) && \text{Induktionsvor.} \\
&= \text{dom}(\widehat{f}(r) \oplus \widehat{f}(u)) && 1.3.1(\text{I.c}) \\
&= \text{dom}(\widehat{f}(r \oplus u)) && \widehat{f} \in \mathcal{AG} \\
&= \text{dom}(\widehat{f}(v))
\end{aligned}$$

Damit gilt (a) und wir können die eigentliche Behauptung beweisen. Dabei gehen wir von

$$[v] = [w] \quad \text{und} \quad v \neq w$$

aus und zeigen, daß daraus

$$\mathcal{ST}_f([v]) = \mathcal{ST}_f([w])$$

folgt.

Unsere Annahme impliziert $v \approx w$, d.h. $\exists r_1, \dots, r_n \in \text{Mor}: v \sim r_1 \sim \dots \sim r_n \sim w$.
Damit genügt es zu zeigen:

$$v \sim w \implies \mathcal{ST}_f([v]) = \mathcal{ST}_f([w]).$$

Wir führen den Beweis durch strukturelle Induktion über den Aufbau von v .

(Induktionsbasis): $v \in (T_1 \cup (P_1 \times \|A_1\|))^\circ$.

Daraus folgt: $v = u \oplus r$ und $w = u \oplus \text{dom}(r)$; $r \oplus \text{cod}(u)$ (\rightarrow 1.3.1(II)).

$$\begin{aligned}
&\mathcal{ST}_f([v]) \\
&= \mathcal{ST}_f([u \oplus r]) \\
&= [\widehat{f}(u \oplus r)] && 1.3.7 \\
&= [(\widehat{f}(u) \oplus \widehat{f}(r))] && \widehat{f} \in \mathcal{AG} \\
&= [\widehat{f}(u) \oplus \text{dom}(\widehat{f}(r)); \widehat{f}(r) \oplus \text{cod}(\widehat{f}(u))] && 1.3.1(\text{II.a}) \\
&= [\widehat{f}(u) \oplus \widehat{f}(\text{dom}(r)); \widehat{f}(r) \oplus \widehat{f}(\text{cod}(u))] && (\text{a}) \\
&= [\widehat{f}(u \oplus \text{dom}(r)); \widehat{f}(r \oplus \text{cod}(u))] && \widehat{f} \in \mathcal{AG} \\
&= [\widehat{f}(r \oplus \text{cod}(u))] \circ [\widehat{f}(u \oplus \text{dom}(r))] && 1.3.1(\text{VI}) \\
&= \mathcal{ST}_f([r \oplus \text{cod}(u)]) \circ \mathcal{ST}_f([u \oplus \text{dom}(r)]) && 1.3.7 \\
&= \mathcal{ST}_f([u \oplus \text{dom}(r); r \oplus \text{cod}(u)]) && 1.3.7 \\
&= \mathcal{ST}_f([w])
\end{aligned}$$

(Induktionsschritt): $v = u; r$.

Daraus folgt: $w = u'; r', u \sim u'$ und $r \sim r'$ (\rightarrow 1.3.1(II)).

$$\begin{aligned}
& \mathcal{ST}_f([v]) \\
&= \mathcal{ST}_f([u; r]) \\
&= \mathcal{ST}_f([r]) \circ \mathcal{ST}_f([u]) && 1.3.7 \\
&= \mathcal{ST}_f([r']) \circ \mathcal{ST}_f([u']) && \text{Induktionsannahme} \\
&= \mathcal{ST}_f([u'; r']) && 1.3.7 \\
& \mathcal{ST}_f([w])
\end{aligned}$$

(Das Bild von \mathcal{ST}_f ist in \mathcal{ST}_{N_2} enthalten): Jeder Morphismus in \mathcal{ST}_{N_1} ist eine Sequenz über „atomare“ Morphismen, d.h. Morphismen der Form $[v]$ mit v in $(T_1 \cup (P_1 \times \|A\|))^\otimes$. Diese Morphismen werden von (\rightarrow 1.3.7) \mathcal{ST}_f auf eine Äquivalenzklasse abgebildet, deren Repräsentant eine formale Summe über Transitionen und Markierungen zu N_2 ist. Diese Äquivalenzklasse ist ein atomarer Morphismus in \mathcal{ST}_{N_2} . Die Sequenz zweier Morphismen wird auf die Komposition ihrer Bilder unter \mathcal{ST}_f abgebildet und damit letztendlich, wenn die Rekursion bei der sequentiellen Komposition zweier atomarer Morphismen anlangt, wieder auf eine Sequenz von atomaren Morphismen in \mathcal{ST}_{N_2} .

(\mathcal{ST}_f bewahrt Identitäten): Die Identitäten in \mathcal{ST}_{N_1} sind Äquivalenzklassen mit Repräsentanten in der freien abelschen Gruppe $P_1^\otimes \otimes A_1$. Sie werden von \mathcal{ST}_f auf die Äquivalenzklassen abgebildet, die formale Summen über Markierungen zu N_2 als Repräsentanten haben und damit auf Identitäten in \mathcal{ST}_{N_2} .

(\mathcal{ST}_f bewahrt die Komposition): Für alle komponierbaren Morphismen $[v]$, $[w]$ in \mathcal{ST}_{N_1} gilt:

$$\mathcal{ST}_f([w] \circ [v]) = \mathcal{ST}_f([v; w]) = \mathcal{ST}_f([w]) \circ \mathcal{ST}_f([v]) \quad .$$

□

(II):

Wir führen den Beweis als Induktion. Die dafür notwendige totale Ordnung auf den Morphismuspaaren entsteht, indem wir jedem der beiden Morphismen die Anzahl seiner atomaren Morphismen zuordnen, d.h. Morphismen der Form $[v]$ mit v aus $(T_1 \cup (P_1 \times \|A\|))^\otimes$, aus denen er durch Komposition gebildet ist und darüber das Maximum bilden. Identitäten werden dabei nicht berücksichtigt. Diese Größe bezeichnen wir als Länge des Morphismuspaars. Eine derartige Zuordnung ist eindeutig, da in \mathcal{ST}_{N_1} alle atomaren Morphismen paarweise verschieden von den komponierten Morphismen² sind.

²abgesehen von der Komposition mit Identitäten

(Induktionsbasis): Die Länge der Paare $([g], [k])$ ist 0.

In diesem Fall müssen g und k Elemente aus $(T_1 \cup (P_1 \times \parallel A \parallel))^\circledast$ sein (\rightarrow 1.3.1). Daraus folgt (\rightarrow 1.3.3, 1.3.7):

$$\begin{aligned}
& \mathcal{ST}_f([g] \parallel [k]) \\
&= \mathcal{ST}_f([g \oplus k]) = [\widehat{f}(g \oplus k)] = [\widehat{f}(g) \oplus \widehat{f}(k)] && 1.3.3, 1.3.7, \widehat{f} \in \mathcal{AG} \\
&= [\widehat{f}(g)] \parallel [\widehat{f}(k)] && 1.3.3 \\
&= \mathcal{ST}_f([g]) \parallel \mathcal{ST}_f([k]) && 1.3.3
\end{aligned}$$

(Induktionsannahme): Unsere Behauptung gilt für alle Paare deren Länge kleiner oder gleich n ist.

(Induktionsschritt): Die Länge von $([g], [k])$ beträgt $n + 1$.

In diesem Fall gilt:

$$g = g' ; g'' \quad \vee \quad k = k' ; k''.$$

Wir gehen zunächst davon aus, daß sowohl g als auch k sequentiell komponiert sind.

$$\begin{aligned}
& g = g' ; g'' \wedge k = k' ; k'' && \text{Annahme} \\
& \mathcal{ST}_f([g' ; g''] \parallel [k' ; k'']) \\
&= \mathcal{ST}_f([g''] \parallel [k''] \circ [g'] \parallel [k']) && 1.3.3 \\
&= \mathcal{ST}_f([g''] \parallel [k'']) \circ \mathcal{ST}_f([g'] \parallel [k']) && \text{Funktoreigenschaft } \mathcal{ST}_f \\
&= \mathcal{ST}_f([g'']) \parallel \mathcal{ST}_f([k'']) \circ \mathcal{ST}_f([g']) \parallel \mathcal{ST}_f([k']) && \text{Induktionsannahme} \\
&= \mathcal{ST}_f([g'']) \circ \mathcal{ST}_f([g']) \parallel \mathcal{ST}_f([k'']) \circ \mathcal{ST}_f([k']) && \text{Funktoreigenschaft } \parallel \\
&= \mathcal{ST}_f([g''] \circ [g']) \parallel \mathcal{ST}_f([k''] \circ [k']) && \text{Funktoreigenschaft } \mathcal{ST}_f \\
&= \mathcal{ST}_f([g' ; g'']) \parallel \mathcal{ST}_f([k' ; k'']) && 1.3.1
\end{aligned}$$

In den Fällen wo entweder g oder k atomar sind, komponieren wir den atomaren Morphismus mit der entsprechenden Identität, z.B.:

$$[g] = [g] \circ id_{dom([g])} = [dom(g) ; g] \quad \text{wenn } g \text{ atomar ist.}$$

Damit lassen sich diese Fälle auf den schon bewiesenen Fall zurückführen. \square

\square

Beweise zu Kapitel 2

Beweis (2.2.6 Initiale Modelle):

Der Beweis gliedert sich in drei Unterpunkte, in denen gezeigt wird,

- $I = (ANS, id, T_{Spec}, N_I^o, eval_I)$ ist ein *ANS*-Modell,
- zu jedem anderen *ANS*-Modell existiert ein Modellmorphismus und
- dieser Modellmorphismus ist eindeutig.

(I ist ein *ANS*-Modell):

Mit der Definition der *ANS*-Modelle (\rightarrow 2.2.3) muß gezeigt werden:

- (1) *ANS* ist eine AN-Spezifikation,
- (2) $id: Spec \rightarrow Spec$ ist ein Spezifikationsmorphismus,
- (3) T_{Spec} ist eine *Spec*-Algebra,
- (4) N_I^o ist ein CP/T-Netz mit einer endlichen Stellen- und Transitionsmenge, das die Gleichung:

$$\mathcal{P}^A(N_I^o) = T_{Spec(X)}$$

erfüllt,

- (5) $eval_I: N \rightarrow N_I^o$ ist ein CP/T-Netzmorphismus der die Gleichung:

$$\mathcal{P}^A(eval_I) = (id, eval_{\mathcal{V}(T_{Spec(X)})})$$

erfüllt,

- (6) für alle Transitionen t', t'' aus N und t aus N_I^o gilt:

$$eval_I^T(t')(t) > 0 \wedge eval_I T(t'')(t) > 0 \implies Cond(t') = Cond(t'')$$

- (7) und für alle Transitionen t aus N_I^o und Variablen $x_s, y_{s'}$ aus $var(t)$ gilt:

$$s \neq s' \implies x \neq y.$$

(.1): Folgt direkt aus der Definition.

(.2): Die algebraischen Spezifikationen, bilden die Kategorie \mathcal{SPEC} . Damit ist die Identität für jede Spezifikation ein Morphismus in \mathcal{SPEC} .

(.3): Die Quotiententalgebra T_{Spec} ist die initiale *Spec*-Algebra.

(.4): Die Stellen- und Transitionskomponenten von N_I^o sind mit den entsprechenden Komponenten in der Netzstruktur von *ANS* identisch und daher definitionsgemäß (\rightarrow 2.1.3) freie abelsche Gruppen mit endlichen Basen.

Die Algebrakomponente $T_{Spec(X)}$ von N_I^o ist als Transformation von T_{Spec} eine Algebra mit Variablen zur Signatur von $Spec(X)$. Daraus folgt mit der Definition der Projektion \mathcal{P}^A (\rightarrow 1.2.1)

$$\mathcal{P}^A(N_I^o) = T_{Spec(X)}.$$

Die Markenflußmorphisme $Pre_I, Post_I$ sind Kompositionen von monotonen Morphismen in den freien abelschen Gruppen. Sie verbinden die Gruppen T^\otimes und $P^\otimes \otimes T_{Spec(X)}$.

(.5): Die Identitäten $id: T^\otimes \rightarrow T^\otimes$, $id: P^\otimes \rightarrow P^\otimes$ sind Teil der Kategorie der freien abelschen Gruppen und $(id, eval_{T_{Spec(X)}}): T_{OP(X)} \rightarrow T_{Spec(X)}$ ist ein generalisierter Algebromorphismus, mit dem Evaluierungsmorphismus $eval_{T_{Spec(X)}}$ als Algebrakomponente. Daraus folgt:

$$\mathcal{P}^A(eval_I) = (id, eval_{T_{Spec(X)}}).$$

Verbleibt noch der Nachweis der Verträglichkeit von $eval_I$ mit den Markenflußmorphismen. Der ergibt sich aus der Definition von Pre_I mit

$$Pre_I \circ id_{T^\otimes} = Pre_I = id_{P^\otimes} \otimes eval_{T_{Spec(X)}} \circ Pre.$$

Für $Post_I$ gilt der selbe Zusammenhang.

(.6): Folgt mit $eval_I^T = id$ direkt aus der Definition.

(.7): Die Evaluierungsfunktion $eval_{T_{Spec(X)}}$ bewahrt die Variablen aus X . Damit und mit der Definition von $eval_I$ folgt die Gleichheit der Variablenmengen zu den Transitionen in N und N_I^o . Mit der Definition der AN-Spezifikationen (\rightarrow 2.1.3), für deren Netzstrukturen die zu zeigende Eigenschaft per Definition gilt, folgt sie auch für N_I^o . \square

(Zu jedem anderen ANS-Modell existiert ein Modellmorphismus):

Wir nehmen im folgenden o.B.d.A. an

- $AN = (ANS, g, A, N^o, eval)$
- $N^o = T_M^\otimes \xrightarrow[Post_M]{Pre_M} P_M^\otimes \otimes A(X)$
- $eval = (e^T, e^P, (g^X, eval_{\mathcal{V}(A(X))}))$

und konstruieren daraufhin die Abbildung $eval_{AN}: I \rightarrow AN$ durch:

$$eval_{AN} \stackrel{\text{def}}{=} ((g, eval_{\mathcal{V}(A)}), (e^T, e^P, \mathcal{T}^A(g, eval_{\mathcal{V}(A)}))).$$

Um zu zeigen, daß es sich bei $eval_{AN}$ um einen ANS-Modellmorphismus handelt, muß mit deren Definition (\rightarrow 2.2.3) nachgewiesen werden:

- (1) $(g, eval_{\mathcal{V}(A)})$ ist ein generalisierter Algebromorphismus von T_{Spec} nach A ,

- (2) $(e^T, e^P, \mathcal{T}^A(g, eval_{\mathcal{V}(A)}))$ ist ein CP/T-Netzmorphismus von N_I^o nach N^o ,
- (3) $\mathcal{P}^A((e^T, e^P, \mathcal{T}^A(g, eval_{\mathcal{V}(A)}))) = \mathcal{T}^A(g, eval_{\mathcal{V}(A)})$,
- (4) $(e^T, e^P, \mathcal{T}^A(g, eval_{\mathcal{V}(A)})) \circ eval_I = eval$
- (5) $(e^T, e^P, \mathcal{T}^A(g, eval_{\mathcal{V}(A)})) \Big|_{\mathcal{DC}_{eval_I}(co)}^{\mathcal{DC}_{eval}(co)} \in Mor(\mathcal{CPTN})[\mathcal{DC}_{eval_I}(co), \mathcal{DC}_{eval}(co)]$.

(.1): Aus der Definition der A-Netze (\rightarrow 2.2.1) folgt: $g: Spec \rightarrow Spec'$ ist ein Spezifikationsmorphismus. $eval_{\mathcal{V}(A)}$ ist der eindeutige Evaluierungsmorphismus von der initialen Spec-Algebra T_{Spec} nach $\mathcal{V}_g(A(X))$. Daraus ergibt sich der generalisierte Algebromorphismus $(g, eval_{\mathcal{V}(A(X))})$ von T_{Spec} nach A .

(.2): Aufgrund der A-Netz-Definition (\rightarrow 2.2.1) sind $e^T: T^{\otimes} \rightarrow T_M^{\otimes}$, $e^P: P^{\otimes} \rightarrow P_M^{\otimes}$ monotone Morphismen in den freien abelschen Gruppen. Die Transformation des generalisierten Algebromorphismus $(g, eval_{\mathcal{V}(A)})$ mittels \mathcal{T}^A (\rightarrow 6.3.9), führt zu einem generalisierten Algebromorphismus von $T_{Spec(X)}$ nach $A(X)$. Damit sind alle Komponenten vom richtigen Typ und es verbleibt der Nachweis der Verträglichkeit mit den Markenflußmorphismen.

Aufgrund der Initialität der Termalgebra $T_{OP(X)}$ in der Kategorie der Sig-Algebren und mit der Definition der Transformation \mathcal{T}^A (\rightarrow 6.3.9) folgt das Kommutieren des nachfolgenden Diagramms.

(a):

$$\begin{array}{ccc}
 T_{OP(X)} & \xrightarrow{eval_{T_{Spec}(X)}} & T_{Spec}(X) \\
 & \searrow (g^X, eval_{\mathcal{V}(A(X))}) & \downarrow \mathcal{T}^A(g, eval_{\mathcal{V}(A)}) \\
 & & A(X)
 \end{array}$$

Damit können wir zeigen:

$$\begin{aligned}
 Pre_M \circ e^T &= e^P \otimes (g^X, eval_{\mathcal{V}(A(X))}) \circ Pre && eval \in \mathcal{CPTN} \\
 &= e^P \otimes \mathcal{T}^A(g, eval_{\mathcal{V}(A)}) \circ id \otimes (id, eval_{T_{Spec}(X)}) \circ Pre && (a) \\
 &= e^P \otimes \mathcal{T}^A(g, eval_{\mathcal{V}(A)}) \circ Pre_I && \text{Def. } Pre_I
 \end{aligned}$$

Der Nachweis der Kompatibilität mit $Post_M$ verläuft entsprechend.

(.3): Folgt direkt aus der Definition von $eval_{AN}$.

(.4):

$$\begin{aligned}
 &(e^T, e^P, \mathcal{T}^A(g, eval_A)) \circ eval_I \\
 &= (e^T \circ id, e^P \circ id, \mathcal{T}^A(g, eval_A) \circ (id, eval_{T_{Spec}(X)})) && \text{Def. } eval_{AN}
 \end{aligned}$$

$$\begin{aligned}
&= (e^T, e^P, (g^x, eval_{\mathcal{V}(A(X))})) && \text{(a)} \\
&= eval
\end{aligned}$$

(.5): Alle Transitionen aus N_I^o liegen im Bild von $eval_I$. Damit ist $\mathcal{DC}_{eval_I}(co)$ das leere Netz und die zu zeigenden Eigenschaft gilt. \square

(Die Modellmorphismen zu anderen ANS-Modellen sind eindeutig.):

Sei $((f^S, f^A), (f^T, f^P, f^{G^X}))$ ein weiterer ANS-Modellmorphismus von I nach AN , dann gilt aufgrund von Definition 2.2.3

$$\text{(b):} \quad (f^T, f^P, f^{G^X}) \circ eval_I = eval \quad \text{und} \quad \mathcal{T}^A(f^S, f^A) = f^{G^X}.$$

Mit den Definitionen von \mathcal{T}^A und \mathcal{T}^S (\rightarrow 6.3.9, 6.3.4), der Komposition generalisierter Morphismen und der Definition von $eval_I$ folgt daraus

$$\mathcal{T}^S(f^S) \circ id_{Spec(X)} = \mathcal{T}^S(f^S) = g^x = \mathcal{T}^S(g) \implies f^S = g$$

und mit der Initialität der Quotiententermalgebra T_{Spec} in der Kategorie der *Spec*-Algebren

$$f^A = eval_{\mathcal{V}(A)} \quad \text{und} \quad f^{G^X} = \mathcal{T}^A(g, eval_{\mathcal{V}(A)}).$$

Darüberhinaus folgt aus (b) und der Definition von $eval_I$

$$f^T = e^T \quad \text{und} \quad f^P = e^P.$$

Damit ist die Gleichheit von $((f^S, f^A), (f^T, f^P, f^{G^X}))$ und $eval_{AN}$ auf allen Komponenten gezeigt. \square

\square

Beweise zu Kapitel 3

6.3.11 Lemma (Eigenschaften der Colimites in \mathcal{CPTN}):

Sei \mathcal{S} eine endliche Kategorie, $\mathcal{D}^T: \mathcal{S} \rightarrow \mathcal{AG}$, $\mathcal{D}^P: \mathcal{S} \rightarrow \mathcal{AG}$, $\mathcal{D}^G: \mathcal{S} \rightarrow \mathcal{GSPEC-ALG}$ Funktoren und $\mathcal{D}: \mathcal{S} \rightarrow \mathcal{CPTN}$ ein Funktor mit

$$\mathcal{D}(m) = (\mathcal{D}^T(m), \mathcal{D}^P(m), \mathcal{D}^G(m))$$

für alle Morphismen m in \mathcal{S} und Colimes

$$\left(\left((c_s^T, c_s^P, c_s^G) \right)_{s \in Obj(\mathcal{S})}, T_c^\otimes \xrightarrow[Post_c]{Pre_c} P_c^\otimes \otimes A_c \right)$$

dann gilt:

- (I) $((c_s^T)_{s \in \text{Obj}(\mathcal{S})}, T_c^\otimes)$ ist ein Colimes für \mathcal{D}^T ,
- (II) $((c_s^P)_{s \in \text{Obj}(\mathcal{S})}, P_c^\otimes)$ ist ein Colimes für \mathcal{D}^P und
- (III) $((c_s^G)_{s \in \text{Obj}(\mathcal{S})}, A_c)$ ist ein Colimes für \mathcal{D}^G .

□

Beweis:**(I):**

Aus der komponentweisen Komposition der CP/T-Netzmorphismen folgt:

$$((c_s^T)_{s \in \text{Obj}(\mathcal{S})}, T_c^\otimes) \text{ ist eine Senke für } \mathcal{D}^T.$$

Verbleibt der Nachweis der couniversellen Eigenschaft dieser Senke. Dazu nehmen wir an:

$$((\bar{c}_s^T)_{s \in \text{Obj}(\mathcal{S})}, T_{\bar{c}}^\otimes) \text{ ist eine Senke für } \mathcal{D}^T.$$

In diesem Fall bildet

$$\left(\left((\bar{c}_s^T, \bar{c}_s^P, c_s^G) \right)_{s \in \text{Obj}(\mathcal{S})}, T_c^\otimes \begin{array}{c} \xrightarrow{\text{Pre}_c} \\ \xrightarrow{\text{Post}_c} \end{array} \emptyset \otimes A_c \right)$$

eine Senke für \mathcal{D}^3 und es existiert ein eindeutiger Morphismus

$$\hat{c}: T_c^\otimes \begin{array}{c} \xrightarrow{\text{Pre}_c} \\ \xrightarrow{\text{Post}_c} \end{array} P_c^\otimes \otimes A_c \rightarrow T_{\bar{c}}^\otimes \begin{array}{c} \xrightarrow{\text{Pre}_{\bar{c}}} \\ \xrightarrow{\text{Post}_{\bar{c}}} \end{array} \emptyset \otimes A_c$$

mit der Eigenschaft

$$\hat{c} \circ (c_s^T, c_s^P, c_s^G) = (\bar{c}_s^T, \bar{c}_s^P, c_s^G)$$

für alle Objekte s in \mathcal{S} .Sei $\hat{c}^T: T_c^\otimes \rightarrow T_{\bar{c}}^\otimes$ die Transitionskomponente von \hat{c} , dann gilt:

$$\hat{c}^T \circ c_s^T = \bar{c}_s^T$$

für alle Objekte s in \mathcal{S} .Jeder andere monotone Morphismus $\tilde{c}^T: T_c^\otimes \rightarrow T_{\bar{c}}^\otimes$ in den freien abelschen Gruppen mit

$$\tilde{c}^T \circ c_s^T = \bar{c}_s^T$$

für alle Objekte s in \mathcal{S} , läßt sich zu dem CP/T-Netzmorphismus

$$(\tilde{c}^T, \tilde{c}^P, \tilde{c}^G): T_c^\otimes \begin{array}{c} \xrightarrow{\text{Pre}_c} \\ \xrightarrow{\text{Post}_c} \end{array} P_c^\otimes \otimes A_c \rightarrow T_{\bar{c}}^\otimes \begin{array}{c} \xrightarrow{\text{Pre}_{\bar{c}}} \\ \xrightarrow{\text{Post}_{\bar{c}}} \end{array} \emptyset \otimes A_c$$

³ $\text{Pre}_c, \text{Post}_c$ und \bar{c}_s^P für alle Objekte s aus \mathcal{S} , sind durch Definition des Operators $_ \otimes _$ (\rightarrow 1.1.1) und die terminale abelsche Gruppe \emptyset eindeutig bestimmt

erweitern, wobei \widehat{c}^P und \widehat{c}^G die Stellen- resp. die Algebrakomponente von \widehat{c} bezeichnen. Es gilt

$$(\widehat{c}^T, \widehat{c}^P, \widehat{c}^G) \circ (c_s^T, c_s^P, c_s^G) = (\overline{c}_s^T, \overline{c}_s^P, c_s^G)$$

für alle Objekte s in \mathcal{S} und damit wegen der Eindeutigkeit von \widehat{c} mit dieser Eigenschaft

$$\widehat{c} = (\overline{c}_s^T, \overline{c}_s^P, c_s^G).$$

Daraus folgt $\widehat{c}^T = \overline{c}^T$. □

Die Behauptungen **(II)** und **(III)** lassen sich nach dem gleichen Muster beweisen. □

Beweis (3.1.7 Vergißfunktork):

Wir nehmen im folgenden, für $1 \leq i \leq 3$, $1 \leq j \leq 2$, o.B.d.A. an:

- $ANS_i = (Spec_i, Cond_i, N_i)$ ist eine AN-Spezifikation mit
- $N_i = T_{i,s}^{\otimes} \xrightarrow[Post_{i,s}]{Pre_{i,s}} P_{i,s}^{\otimes}$,
- $h: ANS_1 \rightarrow ANS_2$ ist ein AN-Spezifikationsmorphismus,
- $AN_i = (ANS_2, g_i, A_i, N_i^o, eval_i)$ ist ein ANS_2 -Modell mit
- $g_i: Spec_i \rightarrow Spec'_i$ und $eval_i = (e_i^T, e_i^P, (g_i^X, eval_{\mathcal{V}(A_i(X))}))$,
- $f_j = (f_j^G, f_j^N): AN_j \rightarrow AN_{j+1}$ ist ein ANS_2 -Modellmorphismus mit
- $f_j^N = (f_j^T, f_j^P, f_j^{GX})$.

und zeigen:

- $\mathcal{V}_h(AN_i) \in \mathcal{MOD}(ANS_1)$
- $\mathcal{V}_h(f_j): \mathcal{V}_h(AN_j) \rightarrow \mathcal{V}_h(AN_{j+1}) \in \mathcal{MOD}(ANS_1)$
- $\mathcal{V}_h(id_{AN_i}) = id_{\mathcal{V}_h(AN_i)}$
- $\mathcal{V}_h(f_2 \circ f_1) = \mathcal{V}_h(f_2) \circ \mathcal{V}_h(f_1)$

($\mathcal{V}_h(AN_i) \in \mathcal{MOD}(ANS_1)$): Wir notieren im folgenden $\mathcal{T}^S(g_i)$ durch g_i^X und $\mathcal{T}^S(h^S)$ durch h^{S_x} .

Aus der Definition des Vergißfunktors \mathcal{V}_h (\rightarrow 3.1.4) folgt:

- $\mathcal{V}_h(AN_i) = (ANS_1, g_i \circ h^S, A_i, N_{i,\mathcal{V}}^o, eval_{i,\mathcal{V}})$ mit
- $N_{i,\mathcal{V}}^o = T_{i,\mathcal{V}}^{\otimes} \xrightarrow[Post_{i,\mathcal{V}}]{Pre_{i,\mathcal{V}}} P_{i,\mathcal{V}}^{\otimes} \otimes A_i(X)$ und
- $eval_{i,\mathcal{V}} = \left(e_i^T \Big|_{T_{i,\mathcal{V}}^{\otimes}} \circ h^T, e_i^P \Big|_{P_{i,\mathcal{V}}^{\otimes}} \circ h^P, (g_i^X \circ h^{S_x}, eval_{\mathcal{V}(A_i(X))}) \right)$ ⁴

⁴ \mathcal{T}^S ist ein Funktor, damit gilt: $\mathcal{T}^S(g_i \circ h^S) = \mathcal{T}^S(g_i) \circ \mathcal{T}^S(h^S)$.

Es muß gezeigt werden (\rightarrow 2.2.1):

- (1) ANS_I ist eine AN-Spezifikation,
- (2) $g_i \circ h^S$ ist ein Spezifikationsmorphismus,
- (3) A_i ist eine $Spec'_i$ -Algebra,
- (4) $N_{i,\mathcal{V}}^o$ ist ein CP/T-Netz mit endlichen Mengen $P_{i,\mathcal{V}}, T_{i,\mathcal{V}}$ und es gilt:

$$\mathcal{P}^A(N_{i,\mathcal{V}}^o) = A_i(X),$$

- (5) $eval_{i,\mathcal{V}}$ ist ein CP/T-Netzmorphimus von N_1 nach $N_{i,\mathcal{V}}^o$ und es gilt:

$$\mathcal{P}^A(eval_{i,\mathcal{V}}) = (g_i^x \circ h^{S^x}, eval_{\mathcal{V}(A_i(X))}),$$

- (6) für alle Transitionen t', t'' aus N_i^o , t aus N :

$$e_{i,\mathcal{V}}^T(t')(t) > 0 \wedge e_{i,\mathcal{V}}^T(t'')(t) > 0 \implies g_i^* \circ h^{S^*}(Cond_i(t')) = g_i^* \circ h^{S^*}(Cond_i(t'')).$$

(.1): Gilt laut Annahme.

(.2): Da es sich bei g_i und h^S um Spezifikationsmorphimusen handelt, ergibt ihre Komposition ebenfalls einen Spezifikationsmorphismus.

(.3): AN_i mit den Komponenten $g_i: Spec_i \rightarrow Spec'_i$ und A_i ist gemäß unseren Annahmen ein ANS_2 -Modell (\rightarrow 2.2.1), daraus folgt $A_i \in \mathcal{ALG}(Spec'_i)$.

(.4): Aus der Definition von $N_{i,\mathcal{V}}^o$ folgt:

- $P_{i,\mathcal{V}} \subseteq P_i$ und $T_{i,\mathcal{V}} \subseteq T_i$, da sich beide Mengen jeweils aus Teilmengen der Stellen-/Transitionsmenge von N_i^o zusammensetzen. Damit ergibt sich die Endlichkeit von $P_{i,\mathcal{V}}/T_{i,\mathcal{V}}$ aus der Endlichkeit von P_i/T_i und $P_{i,\mathcal{V}}^\otimes/T_{i,\mathcal{V}}^\otimes$ sind definitionsgemäß, die darüber frei erzeugten abelschen Gruppen.
- $A_i(X)$ ist eine $Spec_i$ -Algebra
- $Pre_{i,\mathcal{V}}/Post_{i,\mathcal{V}}$ sind Restriktionen der monotonen Markenflußmorphimusen aus N_i^o auf die Transitions- und Stellengruppen in $N_{i,\mathcal{V}}^o$. Sie sind damit ebenfalls monotone Morphimusen in den freien abelschen Gruppen.

Damit ist $N_{i,\mathcal{V}}^o$ nach Definition 1.1.3 ein CP/T-Netz mit endlicher Stellen- und Transitionsmenge. Aus der Definition von \mathcal{P}^A (\rightarrow 1.2.1) folgt: $\mathcal{P}^A(N_{i,\mathcal{V}}^o) = A_i(X)$.

(.5): h^N und $eval_i$ sind definitionsgemäß CP/T-Netzmorphimusen (\rightarrow 2.2.1, 3.1.1). Die durch die Bilder von $e_i^T \circ h^T$ bzw. $e_i^P \circ h^P$ in T_i^\otimes bzw. P_i^\otimes ausgezeichneten Untergruppen sind Teil von $N_{i,\mathcal{V}}^o$, da sie die Transitions- und Stellengruppen

in $\mathcal{DC}_{eval_i \circ h^N}(im)^5$ ausmachen. Damit erfüllen die Transitions- und die Stellenkomponente von $eval_{i,\mathcal{V}}$ die notwendigen Bedingungen (\rightarrow 1.1.4 I, II).

Aufgrund der Initialität von $T_{OP_i(X)}$ folgt

$$\begin{aligned} & (g_i^X, eval_{\mathcal{V}(A_i(X))}) \circ (h^{S_x}, h^t) \\ &= (g_i^X \circ h^{S_x}, \mathcal{V}_{h^{S_x}}(eval_{\mathcal{V}(A_i(X))}) \circ h^t) \\ &= (g_i^X \circ h^{S_x}, eval_{\mathcal{V}(A_i(X))}) \end{aligned} \quad T_{OP_i(X)} \text{ ist initial}$$

Damit ist gleichzeitig gezeigt, daß $(g_i^X \circ h^{S_x}, eval_{\mathcal{V}(A_i(X))})$ (Komposition zweier generalisierter Algebromorphismen) ein generalisierter Algebromorphismus ist.

Aus den CP/T-Netzmorphismus-Eigenschaften von h^N und $eval_i$ und den Definitionen von $T_{i,\mathcal{V}}^{\otimes}$, $P_{i,\mathcal{V}}^{\otimes}$ und $Pre_{i,\mathcal{V}}$ folgt

$$\begin{array}{ccc} T_{1,s}^{\otimes} & \xrightarrow{Pre_{1,s}} & P_{1,s}^{\otimes} \otimes T_{OP_1(X)} \\ \downarrow h^T & \curvearrowright & \downarrow h^P \otimes (h^{S_x}, h^t) \\ T_{2,s}^{\otimes} & \xrightarrow{Pre_{2,s}} & P_{2,s}^{\otimes} \otimes T_{OP_2(X)} \\ \downarrow e_i^T | T_{i,\mathcal{V}}^{\otimes} & \curvearrowright & \downarrow e_i^P | P_{i,\mathcal{V}}^{\otimes} \otimes (g_i^X, eval_{\mathcal{V}(A_i(X))}) \\ T_{i,\mathcal{V}}^{\otimes} & \xrightarrow{Pre_{i,\mathcal{V}}} & P_{i,\mathcal{V}}^{\otimes} \otimes A_i(X) \end{array}$$

Zusammen mit der zuvor gezeigten Eigenschaft der Algebrakomponente von $eval_{i,\mathcal{V}}$, folgt daraus dessen Verträglichkeit mit $Pre_{i,\mathcal{V}}$. Ersetzt man im vorausgehenden Diagramm Pre durch $Post$ ergibt sich die Verträglichkeit von $eval_{i,\mathcal{V}}$ mit $Post_{i,\mathcal{V}}$. Damit erfüllt $eval_{i,\mathcal{V}}$ alle Bedingungen für CP/T-Netzmorphismen und mit der Definition der Projektion \mathcal{P}^A (\rightarrow 3.2.1) gilt:

$$\mathcal{P}^A(eval_{i,\mathcal{V}}) = (g_i^X \circ h^{S_x}, eval_{\mathcal{V}(A_i(X))}).$$

(.6): Wir zeigen, daß die Annahme: t'_1, t''_1 seien zwei Transitionen in N_1 und es existiert eine Transition t aus $N_{i,\mathcal{V}}^o$ mit:

$$e_{i,\mathcal{V}}^T(t'_1)(t) > 0 \wedge e_{i,\mathcal{V}}^T(t''_1)(t) > 0,$$

die Gültigkeit der Gleichung $g_i^* \circ h^{S^*}(Cond_1(t'_1)) = g_i^* \circ h^{S^*}(Cond_1(t''_1))$ impliziert.

$$e_{i,\mathcal{V}}^T(t'_1)(t) > 0 \wedge e_{i,\mathcal{V}}^T(t''_1)(t) > 0 \quad \text{Annahmen}$$

⁵Teil der Zerlegung von N_i^o über $eval_i \circ h^N$ (\rightarrow 3.1.4)

$$\implies e_i^T \circ h^T(t'_1)(t) > 0 \wedge e_i^T \circ h^T(t''_1)(t) > 0 \quad 3.1.4$$

$$\implies \exists t'_2, t''_2 \in N_2:$$

$$h^T(t'_1)(t'_2) > 0 \wedge h^T(t'_1)(t''_2) > 0 \wedge$$

$$e_i^T(t'_2)(t) > 0 \wedge e_i^T(t''_2)(t) > 0$$

$$\implies \exists t'_2, t''_2 \in N_2:$$

3.1.1, 2.2.1

$$h^{S^*}(Cond_1(t'_1)) = Cond_2(t'_2) \wedge$$

$$h^{S^*}(Cond_1(t''_1)) = Cond_2(t''_2) \wedge$$

$$g_i^*(Cond_2(t'_2)) = g_i^*(Cond_2(t''_2))$$

$$\implies g_i^* \circ h^{S^*}(Cond_1(t'_1)) = g_i^* \circ h^{S^*}(Cond_1(t''_1))$$

$(\mathcal{V}_h(f_j): \mathcal{V}_h(AN_j) \rightarrow \mathcal{V}_h(AN_{j+1}) \in \mathcal{MOD}(ANS_I))$: Aus der Definition von $\mathcal{V}_h(f_j)$ (\rightarrow 3.1.5) folgt:

$$\mathcal{V}_h(f_j) = (f_j^G, f_{j,\nu}^N) \quad \text{mit} \quad f_{j,\nu}^N = (f_{j,\nu}^T, f_{j,\nu}^P, f_j^{GX})$$

(\rightarrow 3.1.4), wobei $f_{j,\nu}^T$ resp. $f_{j,\nu}^P$ die Einschränkungen von f_j^T resp. f_j^P auf $T_{j,\nu}^\otimes$ resp. $P_{j,\nu}^\otimes$ bezeichnen. Unter Berücksichtigung der Definition der A-Netzmorphismen (\rightarrow 2.2.1) muß gezeigt werden:

(1) $f_j^G: A_j \rightarrow A_{j+1}$ ist ein generalisierter Algebromorphismus

(2) $f_{j,\nu}^N: N_{j,\nu}^o \rightarrow N_{j+1,\nu}^o \in \mathcal{CPTN}$

(3) $\mathcal{P}^A(f_{j,\nu}^N) = f_j^{GX}$

(4) $f_{j,\nu}^N \circ eval_{j,\nu} = eval_{j+1,\nu}$

(5) $f_{j,\nu}^N \Big|_{\mathcal{DC}_{eval_{j,\nu}}(co)} \in [\mathcal{DC}_{eval_{j,\nu}}(co) \longrightarrow \mathcal{DC}_{eval_{j+1,\nu}}(co)]$

(.1): Folgt aus der Annahme, daß f_j ein A-Netzmorphismus mit Algebrakomponente f_j^G ist.

(.2): Wir zeigen zunächst, daß die Einschränkung der Transitionskomponente von f_j^N konsistent ist, d.h.

$$\forall t \in T_{j,\nu}, t' \in T_{j+1}: f_j^T(t)(t') > 0 \implies t' \in T_{j+1,\nu}.$$

Der Beweis der entsprechenden Eigenschaft für die Stellenkomponente verläuft analog.

$$t \in T_{j,\nu}, t' \in T_{j+1}, f_j^T(t)(t') > 0$$

Annahmen

$$\implies t \in \mathcal{P}^T(\mathcal{DC}_{eval_{j \circ h^N}}(im)) \vee t \in \mathcal{P}^T(\mathcal{DC}_{eval_j}(co))$$

3.1.4

$t \in \mathcal{P}^T(\mathcal{DC}_{eval_j \circ h^N}(im))$	Annahme
$\implies \exists t'' \in N_j: e_j^T \circ h^T(t'')(t) > 0$	1.2.3
$\implies \exists t'' \in N_j: f_j^T \circ e_j^T \circ h^T(t'')(t) > 0$	
$\implies \exists t'' \in N_j: e_{j+1}^T \circ h^T(t'')(t) > 0$	2.2.3
$\implies t' \in \mathcal{P}^T(\mathcal{DC}_{eval_{j+1} \circ h^N}(im))$	1.2.3
$\implies t' \in T_{j+1, \mathcal{V}}$	3.1.4
$t \in \mathcal{P}^T(\mathcal{DC}_{eval_j}(co))$	Annahme
$\implies t' \in \mathcal{P}^T(\mathcal{DC}_{eval_{j+1}}(co))$	2.2.3
$\implies t' \in T_{j+1, \mathcal{V}}$	3.1.4

Da f_j^P/f_j^T monotone Morphismen in den freien abelschen Gruppen sind, gilt dies auch für ihre Einschränkungen auf Untergruppen. Die Verknüpfung des Algebromorphismus $f_j^{G^X}$ mit der Stellenkomponente von $\mathcal{V}_h(f_j^N)$ über \otimes (\rightarrow 1.1.1) ergibt den monotonen Morphismus

$$f_j^P \Big|_{P_{j, \mathcal{V}}^{\otimes}}^{P_{j+1, \mathcal{V}}^{\otimes}} \otimes f_j^{G^X}: P_{j, \mathcal{V}}^{\otimes} \otimes A_j(X) \rightarrow P_{j+1, \mathcal{V}}^{\otimes} \otimes A_{j+1}(X).$$

in den freien abelschen Gruppen. Damit haben die Stellen-, die Transitions- und die Algebrakomponente von $\mathcal{V}_h(f_j^N)$ die in 2.2.3 geforderten Eigenschaften.

Da die Vergißbilder der A-Netze ebenso in den ursprünglichen A-Netzen enthalten sind wie die Vergißbilder der A-Netzmorphismen in den ursprünglichen A-Netzmorphismen, erhält die Vergißkonstruktion die Kompatibilität der A-Netzmorphismen mit den Markenflußmorphismen der Netze.

(.3): Folgt aus der Definition von $\mathcal{V}_h(f)$ zusammen mit der Definition des Vergißfunktors \mathcal{P}^A (\rightarrow 1.2.1).

(.4): Im Zuge des Beweise der Behauptung: $eval_{i, \mathcal{V}}$ ist ein CP/T-Netzmorphismus von N_1 nach $N_{i, \mathcal{V}}^o$, haben wir gezeigt:

$$(g_i^X \circ h^{S_x}, eval_{\mathcal{V}(A_i(X))}) = (g_i^X, eval_{\mathcal{V}(A_i(X))}) \circ (h^{S_x}, h^t).$$

Mit der Definition des Vergißfunktors \mathcal{P}^A , folgt daraus

$$\mathbf{(a):} \quad \mathcal{P}^A(eval_{i, \mathcal{V}}) = \mathcal{P}^A(eval_i) \circ \mathcal{P}^A(h^N)$$

und damit

$$f_{j, \mathcal{V}}^N \circ eval_{j, \mathcal{V}}$$

$$\begin{aligned}
&= (f_{j,\nu}^T \circ e_j^T \Big|_{T_{j,\nu}^{\otimes}} \circ h^T, f_{j,\nu}^P \circ e_j^P \Big|_{P_{j,\nu}^{\otimes}} \circ h^P, & (a) \\
&\quad \mathcal{P}^A(f_j^N) \circ \mathcal{P}^A(\text{eval}_j) \circ \mathcal{P}^A(h^N)) \\
&= (f_j^T \Big|_{T_{j+1,\nu}^{\otimes}} \circ e_j^T \circ h^T, f_j^P \Big|_{P_{j+1,\nu}^{\otimes}} \circ e_j^P \circ h^P, & f_j^N \circ \text{eval}_j = \text{eval}_{j+1} \\
&\quad \mathcal{P}^A(\text{eval}_{j+1}) \circ \mathcal{P}^A(h^N)) \\
&= (e_{j+1}^T \Big|_{T_{j+1,\nu}^{\otimes}} \circ h^T, e_{j+1}^P \Big|_{P_{j+1,\nu}^{\otimes}} \circ h^P, & f_j^N \circ \text{eval}_j = \text{eval}_{j+1} \\
&\quad \mathcal{P}^A(\text{eval}_{j+1}) \circ \mathcal{P}^A(h^N)) \\
&= \text{eval}_{j+1,\nu} & (a)
\end{aligned}$$

(.5): In (.2) haben wir bereits gezeigt, daß die Bilder von Elementen aus $\mathcal{DC}_{\text{eval}_{j,\nu}}(co)$ unter $f_{j,\nu}^P$ und $f_{j,\nu}^T$ in $\mathcal{DC}_{\text{eval}_j}(co)$ liegen. Mit der Definition der Zerlegung $\mathcal{DC}_{\text{eval}_{j+1,\nu}}$ (\rightarrow 1.2.3) und 3.1.4 gilt außerdem

$$\mathcal{DC}_{\text{eval}_{j+1,\nu}}(co) = \mathcal{DC}_{\text{eval}_{j+1}}(co).$$

Beide Folgerungen zusammen implizieren unsere Behauptung.

$$(\mathcal{V}_h(\text{id}_{AN_i}) = \text{id}_{\mathcal{V}_h(AN_i)}):$$

$$\begin{aligned}
&\mathcal{V}_h(\text{id}_{AN_i}) \\
&= \mathcal{V}_h((\text{id}_{A_i}, (\text{id}_{T_i}, \text{id}_{P_i}, \text{id}_{A_i(X)}))) \\
&= (\text{id}_{A_i}, (\text{id}_{T_i^{\otimes}} \Big|_{T_{i,\nu}^{\otimes}}, \text{id}_{P_i^{\otimes}} \Big|_{P_{i,\nu}^{\otimes}}, \text{id}_{A_i(X)})) \\
&= (\text{id}_{A_i}, (\text{id}_{T_{i,\nu}^{\otimes}}, \text{id}_{P_{i,\nu}^{\otimes}}, \text{id}_{A_i(X)})) \\
&= (\text{id}_{A_i}, \text{id}_{N_{i,\nu}^{\circ}})
\end{aligned}$$

$$(\mathcal{V}_h(f_2 \circ f_1) = \mathcal{V}_h(f_2) \circ \mathcal{V}_h(f_1)):$$

$$\begin{aligned}
&\mathcal{V}_h(f_2 \circ f_1) \\
&= \mathcal{V}_h((f_2^G \circ f_1^G, f_2^N \circ f_1^N)) \\
&= \left(f_2^G \circ f_1^G, & 3.1.4 \\
&\quad \left(f_2^T \circ f_1^T \Big|_{T_{1,\nu}^{\otimes}} \Big|_{T_{3,\nu}^{\otimes}}, f_2^P \circ f_1^P \Big|_{P_{1,\nu}^{\otimes}} \Big|_{P_{3,\nu}^{\otimes}}, f_2^{GX} \circ f_1^{GX} \right) \right) \\
&= \left(f_2^G \circ f_1^G, & f_j^T \Big|_{T_{j,\nu}^{\otimes}} \Big|_{T_{j+1,\nu}^{\otimes}} \in \mathcal{V}_h(f_j^N) \\
&\quad \left(f_2^T \Big|_{T_{2,\nu}^{\otimes}} \Big|_{T_{3,\nu}^{\otimes}} \circ f_1^T \Big|_{T_{1,\nu}^{\otimes}} \Big|_{T_{2,\nu}^{\otimes}}, f_2^P \Big|_{P_{2,\nu}^{\otimes}} \Big|_{P_{3,\nu}^{\otimes}} \circ f_1^P \Big|_{P_{1,\nu}^{\otimes}} \Big|_{P_{2,\nu}^{\otimes}}, f_2^{GX} \circ f_1^{GX} \right) \right)
\end{aligned}$$

$$= \mathcal{V}_h(f_2^G, f_2^N) \circ \mathcal{V}_h(f_1^G, f_1^N)$$

□

Beweis (3.1.8 Spezifikationsrahmen):

$\mathcal{AHLN}\text{-SPEC}$ ist eine Kategorie (\rightarrow 3.1.3). Damit muß noch gezeigt werden, daß \mathcal{MOD} ein Funktor von $\mathcal{AHLN}\text{-SPEC}^{op}$ nach \mathcal{CATCAT} ist.

Da die Domäne und die Codomäne von \mathcal{MOD} den genannten Kategorien entsprechen, folgt unmittelbar aus den Definitionen von \mathcal{MOD} und den Vergißfunktoren \mathcal{V}_h für h aus $\mathcal{Mor}(\mathcal{AHLN}\text{-SPEC})$. Verbleibt zu Zeigen:

- (I) $\forall ANS \in \mathcal{AHLN}\text{-SPEC}: \mathcal{MOD}(id_{ANS}) = \mathcal{ID}_{\mathcal{MOD}(ANS)}$ und
- (II) $\forall h_2, h_1 \in \mathcal{AHLN}\text{-SPEC}: \mathcal{MOD}(h_2 \circ h_1) = \mathcal{MOD}(h_2) \circ \mathcal{MOD}(h_1)$.

Dazu nehmen wir im folgenden, für $1 \leq i \leq 3$, $1 \leq j \leq 2$, o.B.d.A. an:

- $ANS_i = (Spec_i, Cond_i, N_i)$ ist eine AN-Spezifikation,
- $AN_i = (ANS_3, g_i, A_i, N_i^o, eval_i)$ ist ein ANS_3 -Modell, mit
- $N_i^o = T_i^{\otimes} \xrightarrow[Post_i]{Pre_i} P_i^{\otimes} \otimes A_i(X)$ und $eval_i = (e_i^T, e_i^P, (g_i^X, eval_{A_i(X)}))$,
- $f = (f^G, (f^T, f^P, f^{G_X})): AN_1 \rightarrow AN_2$ ist ein ANS_3 -Modellmorphismus und
- $h_j: ANS_j \rightarrow ANS_{j+1}$ ist ein AN-Spezifikationsmorphismus.

(I):

Die Behauptung gilt, wenn folgende Teilresultate gelten:

- $\mathcal{V}_{id}(AN_i) = AN_i$ und
- $\mathcal{V}_{id}(f) = f$.

$(\mathcal{V}_{id}(AN_i) = AN_i)$: Aus der Definition der Vergißfunktoren (\rightarrow 3.1.4) folgt:

$$\mathcal{V}_{id}(AN_i) = (ANS_3, g_i \circ id, A_i, N_{i,3,3}^o, eval_{i,3,3}).$$

Damit muß gezeigt werden:

$$N_{i,3,3}^o = N_i^o \quad \text{und} \quad eval_{i,3,3} = eval_i$$

$(N_{i,3,3}^o = N_i^o)$: CP/T-Netze sind gleich wenn ihre Komponenten gleich sind. Für die Algebrakomponente beider Netze gilt per Definition (\rightarrow 2.2.1):

$$\mathcal{P}^A(N_{i,3,3}^o) = A_i(X) = \mathcal{P}^A(N_i^o).$$

Für die Transitionsmengen folgt das mit 3.1.4 und 1.2.3 aus:

$$\begin{aligned} T_{i,3,3} &= \mathcal{P}^T(\mathcal{DC}_{eval_i,oid}(im)) \cup \mathcal{P}^T(\mathcal{DC}_{eval_i}(co)) \\ &= \mathcal{P}^T(\mathcal{DC}_{eval_i}(im)) \cup \mathcal{P}^T(\mathcal{DC}_{eval_i}(co)) \\ &= T_i \end{aligned}$$

Die Gleichheit der Stellenmengen läßt sich auf die gleiche Weise zeigen.

Aus der Definition der Vergißfunktoren (\rightarrow 3.1.4) und der Gleichheit der Transitionsmengen folgt wegen:

$$Pre_{i,3,3} = Pre_i \Big|_{T_{i,3,3}^\otimes} = Pre_i \quad \text{und} \quad Post_{i,3,3} = Post_i \Big|_{T_{i,3,3}^\otimes} = Post_i$$

die Gleichheit der Markenflußmorphismen.

($eval_{i,3,3} = eval_i$):

$$\begin{aligned} &eval_{i,3,3} \\ &= \left(e_i^T \Big|_{T_{i,3,3}^\otimes} \circ id, e_i^P \Big|_{P_{i,3,3}^\otimes} \circ id, (g_i^X \circ id, eval_{\mathcal{V}(A_i(X))}) \right) \quad 3.1.4 \\ &= (e_i^T, e_i^P, (g_i^X, eval_{\mathcal{V}(A_i(X))})) = eval_i \quad \begin{array}{l} T_{i,3,3}^\otimes = T_i^\otimes, \\ P_{i,3,3}^\otimes = P_i^\otimes \end{array} \end{aligned}$$

($\mathcal{V}_{id}(f) = f$):

$$\begin{aligned} &\mathcal{V}_{id}(f) \\ &= (f^G, (f^T \Big|_{T_{1,3,3}^\otimes}^{T_{2,3,3}^\otimes}, f^P \Big|_{P_{1,3,3}^\otimes}^{P_{2,3,3}^\otimes}, f^{G_X})) \quad 3.1.5 \\ &= (f^G, (f^T, f^P, f^{G_X})) = f \quad \begin{array}{l} T_{i,3,3}^\otimes = T_i^\otimes \\ P_{i,3,3}^\otimes = P_i^\otimes \end{array} \end{aligned}$$

□

(II):

Die Behauptung gilt, wenn folgende Teilresultate gelten:

- $\mathcal{V}_{h_2 \circ h_1}(AN_i) = \mathcal{V}_{h_1} \circ \mathcal{V}_{h_2}(AN_i)$
- $\mathcal{V}_{h_2 \circ h_1}(f) = \mathcal{V}_{h_1} \circ \mathcal{V}_{h_2}(f)$.

($\mathcal{V}_{h_2 \circ h_1}(AN_i) = \mathcal{V}_{h_1} \circ \mathcal{V}_{h_2}(AN_i)$): Aus Definition 3.1.4 folgt:

$$\mathcal{V}_{h_2 \circ h_1}(AN_i) = (ANS_1, g_i \circ h_2^S \circ h_1^S, A_i, N_{i,3,1}^o, eval_{i,3,1}).$$

Mit $\mathcal{V}_{h_2}(AN_i) = (ANS_2, g_i \circ h_2^S, A_i, N_{i,3,2}^o, eval_{i,3,2})$ und daraus folgend:

$$\mathcal{V}_{h_1} \circ \mathcal{V}_{h_2}(AN_i) = (ANS_1, g_i \circ h_2^S \circ h_1^S, A_i, N_{i,3,2,1}^o, eval_{i,3,2,1})$$

bleibt zu Zeigen:

$$N_{i,3,1}^o = N_{i,3,2,1}^o \quad \text{und} \quad eval_{i,3,1} = eval_{i,3,2,1}$$

($N_{i,3,1}^o = N_{i,3,2,1}^o$): CP/T-Netze sind gleich wenn ihre Komponenten gleich sind. Für die Algebrakomponente beider Netze gilt per Definition (\rightarrow 2.2.1):

$$\mathcal{P}^A(N_{i,3,1}^o) = A_i(X) = \mathcal{P}^A(N_{i,3,2,1}^o).$$

Für die Transitionsmengen folgt das mit 3.1.4 und 1.2.3 aus:

$$\begin{aligned} & T_{i,3,1} \\ &= \mathcal{P}^T(\mathcal{DC}_{eval_{i,3,2}^o \circ h_2^N \circ h_1^N}(im)) \cup \mathcal{P}^T(\mathcal{DC}_{eval_i}(co)) \end{aligned} \quad 3.1.4$$

$$= \mathcal{P}^T(\mathcal{DC}_{eval_{i,3,2}^o \circ h_1^N}(im)) \cup \mathcal{P}^T(\mathcal{DC}_{eval_{i,3,2}}(co)) \quad 3.1.4, 1.2.3$$

$$= T_{i,3,2,1} \quad 3.1.4$$

Die Gleichheit der Stellenmengen läßt sich auf die gleiche Weise zeigen.

Aus der Definition der Vergißfunktoren (\rightarrow 3.1.4) und der Gleichheit der Transitionsmengen folgt die Gleichheit der Markenflußmorphisamen wegen:

$$Pre_{i,3,1} = Pre_i \Big|_{T_{i,3,1}^\otimes} = Pre_i \Big|_{T_{i,3,2,1}^\otimes} = Pre_{i,3,2,1}$$

und

$$Post_{i,3,1} = Post_i \Big|_{T_{i,3,1}^\otimes} = Post_i \Big|_{T_{i,3,2,1}^\otimes} = Post_{i,3,2,1}.$$

($eval_{i,3,1} = eval_{i,3,2,1}$):

$$\begin{aligned} & eval_{i,3,1} \\ &= \left(e_i^T \Big|_{T_{i,3,1}^\otimes} \circ h_2^T \circ h_1^T, e_i^P \Big|_{P_{i,3,1}^\otimes} \circ h_2^P \circ h_1^P, \right. \\ & \quad \left. (g_i^X \circ h_2^S \circ h_1^S, eval_{\mathcal{V}(A_i(X))}) \right) \end{aligned} \quad 3.1.4$$

$$\begin{aligned} &= \left(e_{i,3,2}^T \Big|_{T_{i,3,2,1}^\otimes} \circ h_1^T, e_{i,3,2}^P \Big|_{P_{i,3,2,1}^\otimes} \circ h_1^P, \right. \\ & \quad \left. (g_i^X \circ h_2^S \circ h_1^S, eval_{\mathcal{V}(A_i(X))}) \right) \end{aligned} \quad \begin{array}{l} 3.1.4, 3.1.7, \\ T_{i,3,1}^\otimes = T_{i,3,2,1}^\otimes, \\ P_{i,3,1}^\otimes = P_{i,3,2,1}^\otimes \end{array}$$

$$= eval_{i,3,2,1}$$

$$(\mathcal{V}_{h_2 \circ h_1}(f) = \mathcal{V}_{h_1} \circ \mathcal{V}_{h_2}(f)):$$

$$\mathcal{V}_{h_1} \circ \mathcal{V}_{h_2}(f)$$

$$= \mathcal{V}_{h_1}((f^G, (f^T \Big|_{T_{1,3,2}^\otimes}^{T_{2,3,2}^\otimes}, f^P \Big|_{P_{1,3,2}^\otimes}^{P_{2,3,2}^\otimes}, f^{G_X}))) \quad 3.1.5$$

$$= (f^G, (f^T \Big|_{T_{1,3,2,1}^\otimes}^{T_{2,3,2,1}^\otimes}, f^P \Big|_{P_{1,3,2,1}^\otimes}^{P_{2,3,2,1}^\otimes}, f^{G_X})) \quad \begin{array}{l} T_{i,3,2,1}^\otimes \subseteq T_{i,3,2}^\otimes \\ P_{i,3,2,1}^\otimes \subseteq P_{i,3,2}^\otimes \end{array}$$

$$= (f^G, (f^T \Big|_{T_{1,3,1}^\otimes}^{T_{2,3,1}^\otimes}, f^P \Big|_{P_{1,3,1}^\otimes}^{P_{2,3,1}^\otimes}, f^{G_X})) \quad \begin{array}{l} T_{i,3,1}^\otimes = T_{i,3,2,1}^\otimes \\ P_{i,3,1}^\otimes = P_{i,3,2,1}^\otimes \end{array}$$

$$= \mathcal{V}_{h_2 \circ h_1}(f) \quad 3.1.5$$

□

□

Beweis (3.2.2 Projektionsfunktork):

Wir nehmen im folgenden für $1 \leq i \leq 3$, $1 \leq j \leq 2$, o.B.d.A. an:

- $ANS_i = (Spec_i, Cond_i, N_i)$ ist eine AN-Spezifikation,
- $AN_i = (ANS_i, g_i, A_i, N_i^o, eval_i)$ ist ein ANS_i -Modell, mit
- $N_i^o = T_i^\otimes \xrightarrow[Post_i]{Pre_i} P_i^\otimes \otimes A_i(X)$ und $eval_i = (e_i^T, e_i^P, (g_i^X, eval_{A_i(X)}))$,
- $k_j = (h_j, (f_j^G, f_j^N))$: $AN_j \rightarrow AN_{j+1}$ ist ein Morphismus in $\mathcal{GAHL}\text{-NETS}$ mit
- $f_j^N = (f_j^T, f_j^P, f_j^{G_X})$,
- $\mathcal{V}_{h_j}(AN_{j+1}) = (ANS_j, g_{j+1} \circ h_j^S, A_{j+1}, N_{j+1,j+1,j}^o, eval_{j+1,j+1,j})$,
- $\mathcal{V}_{h_2 \circ h_1}(AN_3) = \mathcal{V}_{h_1}(\mathcal{V}_{h_2}(AN_3)) = (ANS_1, g_3 \circ h_2^S \circ h_1^S, A_3, N_{3,3,1}^o, eval_{3,3,1})$,
- $f = (f^A, (f^T, f^P, f^{G_X}))$ und
- $\mathcal{V}_h(AN_2) = (ANS_1, g_2 \circ h^S, A_2, N_{2,2,1}^o, eval_{2,2,1})$.

(I):

Die Behauptung gilt, wenn folgende Teilresultate gelten:

- (1) $\mathcal{P}^N(AN_i) \in \mathcal{CPTN}$,
- (2) $\mathcal{P}^N(k_j): N_j^o \rightarrow N_{j+1}^o \in \mathcal{CPTN}$,
- (3) $\mathcal{P}^N((id, (id_A, id_{N^o}))) = id_{N^o}$ und
- (4) $\mathcal{P}^N(k_2) \circ \mathcal{P}^N(k_1) = \mathcal{P}^N(k_2 \circ k_1)$.

(I.1): Folgt direkt aus den Definitionen von \mathcal{P}^N (\rightarrow 3.2.1) und A-Netzen (\rightarrow 2.2.1).

(I.2): Nach Definition 3.2.1 gilt:

$$\begin{aligned}\mathcal{P}^N(k_j) &= (\hookrightarrow_{T_{j+1,j+1,j}^\otimes, T_{j+1}^\otimes}, \hookrightarrow_{P_{j+1,j+1,j}^\otimes, P_{j+1}^\otimes}, id_{A_{j+1}(X)}) \circ f_j^N \\ &= \mathcal{P}^N((h_j, id_{\mathcal{V}_{h_j}(AN_{j+1})})) \circ f_j^N\end{aligned}$$

Da es sich bei $f_j^N: N_j^o \rightarrow N_{j+1,j}^o$ bereits um einen CP/T-Netzmorphismus handelt (\rightarrow 2.2.3), \mathcal{CPTN} bzgl. Komposition abgeschlossen ist und die $\mathcal{P}^N((h_j, id))$ -Komponenten, $N_{j+1,j+1,j}^o$ -Komponenten auf N_j^o -Komponenten abbilden, gilt unsere Behauptung, wenn $\mathcal{P}^N((h_j, id))$ ein Morphismus in \mathcal{CPTN} ist.

Die Inklusionen von Untergruppen in freie Abelsche Gruppen sind monotone Gruppomorphismen und $id_{A_{j+1}(X)}$ ist ein Algebromorphismus. Da es sich bei $N_{j+1,j+1,j}^o$ um ein CP/T-Netz handelt (\rightarrow 3.1.4), dessen Transitions- und Stellenmengen in den entsprechenden Mengen von N_{j+1}^o enthalten sind, dessen Algebra mit der Algebra von N_{j+1}^o übereinstimmt und dessen Markenflußmorphismen, den Einschränkungen der Stellen- und Transitions Mengen entsprechende Einschränkungen der Morphismen in N_{j+1}^o sind, ist $\mathcal{P}^N(h_j, id)$ mit den Markenflußmorphismen von $N_{j+1,j+1,j}^o$ und N_{j+1}^o verträglich.

(I.3): \mathcal{P}^N bildet Identitätsmorphismen $((id_{Spec}, id_N), (id_A, id_{N^o}))$ in $\mathcal{GAHL-NETS}$ auf

$$(\hookrightarrow_{T^\otimes, T^\otimes}, \hookrightarrow_{P^\otimes, P^\otimes}, id_{A(X)}) \circ id_{N^o} = (id_{T^\otimes}, id_{P^\otimes}, id_{A(X)}) \quad ^6$$

ab (\rightarrow 3.2.1), dem korrespondierenden Identitätsmorphismus in \mathcal{CPTN} .

(I.4):

$$\begin{aligned}\mathcal{P}^N(k_2 \circ k_1) &= \mathcal{P}^N((h_2 \circ h_1, \mathcal{V}_{h_1}((f_2^G, f_2^N)) \circ (f_1^G, f_1^N))) \\ &= \mathcal{P}^N(h_2^S \circ h_1^S, id) \circ \left(f_2^T \Big|_{T_{2,2,1}^\otimes}^{T_{3,3,1}^\otimes}, f_2^P \Big|_{P_{2,2,1}^\otimes}^{P_{3,3,1}^\otimes}, f_2^{GX} \right) \circ f_1^N \quad \begin{array}{l} 3.1.5, 3.1.8 \\ cod(f_2^T) = T_{3,3,2}^\otimes, \\ cod(f_2^P) = P_{3,3,2}^\otimes \end{array} \\ &= \left(\hookrightarrow_{T_{3,3,1}^\otimes, T_3^\otimes} \circ f_2^T \Big|_{T_{2,2,1}^\otimes}^{T_{3,3,1}^\otimes} \circ \hookrightarrow_{T_{2,2,1}^\otimes, T_2^\otimes} \circ f_1^T, \right. \quad \begin{array}{l} 3.1.5, 3.2.1, \\ cod(f_1^T) = T_{2,2,1}^\otimes, \\ cod(f_1^P) = P_{2,2,1}^\otimes \end{array} \\ &\quad \left. \hookrightarrow_{P_{3,3,1}^\otimes, P_3^\otimes} \circ f_2^P \Big|_{P_{2,2,1}^\otimes}^{P_{3,3,1}^\otimes} \circ \hookrightarrow_{P_{2,2,1}^\otimes, P_2^\otimes} \circ f_1^P, f_2^{GX} \circ f_1^{GX} \right)\end{aligned}$$

⁶ $\mathcal{V}_{id}(AN_i) = AN_i$ (\rightarrow 3.1.7)

$$\begin{aligned}
&= (\hookrightarrow_{T_{3,3,2}, T_3^\otimes} \circ f_2^T, \hookrightarrow_{P_{3,3,2}, P_3^\otimes} \circ f_2^P, f_2^{G_X}) \circ && 3.1.4, \\
&\quad (\hookrightarrow_{T_{2,2,1}, T_2^\otimes} \circ f_1^T, \hookrightarrow_{P_{2,2,1}, P_2^\otimes} \circ f_1^P, f_1^{G_X}) && T_{3,3,1}^\otimes \subseteq T_{3,3,2}^\otimes, \\
&&& P_{3,3,1}^\otimes \subseteq P_{3,3,2}^\otimes \\
&= \mathcal{P}^N(h_2, id) \circ (f_2^T, f_2^P, f_2^{G_X}) \circ \mathcal{P}^N(h_1, id) \circ (f_1^T, f_1^P, f_1^{G_X}) \\
&= \mathcal{P}^N(k_2) \circ \mathcal{P}^N(k_1) && 3.2.1
\end{aligned}$$

□

(II):

Aus der Definition der Vergißfunktoren (\rightarrow 3.1.4) und der Definition der Zerlegung von A-Netzen (\rightarrow 1.2.3) folgt:

$$(a): \quad \forall t \in N_1, t' \in N_2: e_2^T \circ h^T(t)(t') > 0 \implies t' \in T_{2,2,1}^\otimes$$

Damit gilt:

$$\begin{aligned}
&eval_2 \circ h^N \\
&= (e_2^T \circ h^T, e_2^P \circ h^P, (g_2^X, eval_{A_2(X)})) \circ (h^{S_x}, h^t) \\
&= (\hookrightarrow_{T_{2,2,1}, T_2^\otimes} \circ e_2^T \Big|_{T_{2,2,1}^\otimes} \circ h^T, \hookrightarrow_{P_{2,2,1}, P_2^\otimes} \circ e_2^P \Big|_{P_{2,2,1}^\otimes} \circ h^P, && 3.1.4, (a) \\
&\quad (g_2^X \circ h^{S_x}, \mathcal{V}_{h^{S_x}}(eval_{\mathcal{V}(A_2(X))}) \circ h^t)) \\
&= (\hookrightarrow_{T_{2,2,1}, T_2^\otimes} \circ e_{2,2,1}^T, \hookrightarrow_{P_{2,2,1}, P_2^\otimes} \circ e_{2,2,1}^P, && 3.1.4, \\
&\quad (g_2^X \circ h^{S_x}, eval_{\mathcal{V}(A_2(X))})) && T_{OP_1(X)} \text{ initial} \\
&= (\hookrightarrow_{T_{2,2,1}, T_2^\otimes} \circ f^T \circ e_1^T, \hookrightarrow_{P_{2,2,1}, P_2^\otimes} \circ f^P \circ e_1^P, && 2.2.3, \\
&\quad f^{G_X} \circ (g_1^X, eval_{A_1(X)})) && f^N \circ eval_1 = eval_{2,2,1} \\
&= \mathcal{P}^N((h, f)) \circ eval_1
\end{aligned}$$

□

□

Beweis (3.2.3 Komponentweise Colimites in $\mathcal{GAHL}\text{-NETS}$):

Der Beweis gliedert sich in drei Unterpunkte, in denen gezeigt wird,

- die Definitionen in 3.2.3 sind korrekt,
- $\left(\left((c_s^S, c_s^N), ((d_s^S, d_s^A), d_s^N) \right)_{s \in S}, AN_c \right)$ bildet eine Senke für \mathcal{D} und
- die Senke ist ein Colimes für \mathcal{D} .

Dazu nehmen wir im Folgenden (o.B.d.A.) an:

- $\mathcal{D}(s) = AN_s = (ANS_s, g_s, A_s, N_s^o, eval_s)$ mit
 $ANS_s = (Spec_s, Cond_s, N_s)$, $N_s^o = T_s^{\otimes} \xrightarrow[Post_s]{Pre_s} P_s^{\otimes} \otimes A_s(X)$ und
 $eval_s = (e_s^T, e_s^P, (g_s^X, eval_{\mathcal{V}(A_s(X))}))$,
- $\mathcal{D}(m: s \rightarrow s') = (h_m, (f_m^G, f_m^N)): AN_s \rightarrow AN_{s'}$ mit
 $h_m = (h_m^S, h_m^N): ANS_s \rightarrow ANS_{s'}$ und $f_m^N = (f_m^T, f_m^P, f_m^{GX}): N_s^o \rightarrow N_{s',s}^o$,
- $\mathcal{V}_{h_m}(AN_{s'}) = (ANS_s, g_{s'} \circ h_m^S, A_{s'}, N_{s',s}^o, eval_{s',s})$,
- $ANS_c = (Spec_c, Cond_c, N_c)$,
- $c_s = (c_s^S, c_s^N)$, $d_s^G = (d_s^S, d_s^A)$, $\mathcal{T}^A((d_s^S, d_s^A)) = (d_s^{SX}, d_s^{AX})$ und
- $\mathcal{V}_{c_s}(AN_c) = (ANS_s, g_c \circ c_s^S, A_c, N_{c,s}^o, eval_{c,s})$ mit
 $N_{c,s}^o = T_{c,s}^{\otimes} \xrightarrow[Post_{c,s}]{Pre_{c,s}} P_{c,s}^{\otimes} \otimes A_c(X)$.

(Die Definitionen in 3.2.3 sind korrekt.):

Im einzelnen muß gezeigt werden:

- $((d_s^S \circ g_s)_{s \in \mathcal{S}}, Spec'_c)$ ist eine Senke für \mathcal{D}^S ,
- $((c_s^N \circ eval_s)_{s \in \mathcal{S}}, N_c^o)$ ist eine Senke für \mathcal{D}^N und
- für alle Objekte s aus \mathcal{S} existiert d_s^N in 3.2.3(IV) eindeutig.

$((d_s^S \circ g_s)_{s \in \mathcal{S}}, Spec'_c)$ ist eine Senke für \mathcal{D}^S : Aufgrund der Morphismuseigenschaften der (f_m^G, f_m^N) (\rightarrow 2.2.3) für alle $m: s \rightarrow s'$ in \mathcal{S} gilt:

$$f_m^N \circ eval_s = eval_{s',s} \quad 2.2.3$$

$$\implies \mathcal{P}^A(f_m^N) \circ \mathcal{P}^A(eval_s) = \mathcal{P}^A(eval_{s',s}) \quad 3.1.4$$

$$\implies \mathcal{T}^A(f_m^G) \circ (\mathcal{T}^S(g_s), eval) = (\mathcal{T}^S(g_{s'} \circ h_m^S), eval) \quad 2.2.3, 3.1.4$$

$$\implies \mathcal{T}^S(f_m^S \circ g_s) = \mathcal{T}^S(g_{s'} \circ h_m^S) \quad f_s^G = (f_s^S, f_s^A), 6.3.9$$

$$\implies f_m^S \circ g_s = g_{s'} \circ h_m^S \quad \mathcal{T}^S \text{ faithful, 6.3.5}$$

Damit kommutiert das äußere Rechteck im Diagramm zur Definition von g_c . Die beiden enthaltenen Dreiecke sind jeweils Teile der entsprechenden Colimites und kommutieren aus diesem Grund. Damit läßt sich unsere Behauptung direkt im Diagramm ablesen. Aus dem Colimes zu \mathcal{D}^S folgt dann die Existenz und Eindeutigkeit von g_c mit der Eigenschaft:

$$(a): \quad g_c \circ c_s^S = d_s^S \circ g_s \quad \text{für alle } s \text{ in } \mathcal{S}.$$

$((\dot{c}_s^N \circ eval_s)_{s \in \mathcal{S}}, N_c^o)$ ist eine Senke für \mathcal{D}^N): Für alle $m: s \rightarrow s'$ in \mathcal{S} gilt:

$$\dot{c}_{s'}^N \circ eval_{s'} \circ \mathcal{D}^N(m)$$

$$= \dot{c}_{s'}^N \circ \mathcal{P}^N(\mathcal{D}(m)) \circ eval_s \quad 3.2.2$$

$$= \dot{c}_s^N \circ eval_s \quad \text{Colimes für } \mathcal{P}^N \circ \mathcal{D}$$

Damit bildet $((\dot{c}_s^N \circ eval_s)_{s \in \mathcal{S}}, N_c^o)$ eine Senke zu \mathcal{D}^N und es existiert aufgrund des Colimites $((c_s^N)_{s \in \mathcal{S}}, N_c)$ ein eindeutiger Morphismus $(eval_c)$ von N_c nach N_c^o mit der Eigenschaft:

$$(b): \quad \dot{c}_s^N \circ eval_s = eval_c \circ c_s^N \quad \text{für alle } s \text{ in } \mathcal{S}.$$

(Für alle Objekte s aus \mathcal{S} existiert d_s^N in 3.2.3(IV) eindeutig): Mit

$$\mathcal{P}^N((c_s, id_{\mathcal{V}_{c_s}(AN_c)})) = (\hookrightarrow_{T_{c,s}^\otimes, T_c^\otimes}, \hookrightarrow_{P_{c,s}^\otimes, P_c^\otimes}, id_{\mathcal{V}_{c_s X}(A_c(X))})$$

(\rightarrow 3.2.1) für alle Objekte s in \mathcal{S} , folgt unter der Annahme:

$$\dot{c}_s^T \Big|_{T_{c,s}^\otimes} \quad \text{und} \quad \dot{c}_s^P \Big|_{P_{c,s}^\otimes} \quad \text{sind Morphismen in } \mathcal{AG}_f,$$

deren Gültigkeit wir nachfolgend zeigen und $d_s^N = (d_s^T, d_s^P, d_s^{GX})$

$$\dot{c}_s^T = \hookrightarrow_{T_{c,s}^\otimes, T_c^\otimes} \circ d_s^T, \quad \dot{c}_s^P = \hookrightarrow_{P_{c,s}^\otimes, P_c^\otimes} \circ d_s^P \quad \text{und} \quad d_s^{GX} = \mathcal{T}^A((d_s^S, d_s^A)).$$

Daraus ergibt sich

$$(c): \quad d_s^T = \dot{c}_s^T \Big|_{T_{c,s}^\otimes}, \quad d_s^P = \dot{c}_s^P \Big|_{P_{c,s}^\otimes} \quad \text{und} \quad d_s^{GX} = \mathcal{T}^A((d_s^S, d_s^A))$$

eindeutig, da es sich sowohl bei den Inklusionen als auch der Identität um Monomorphismen handelt und d_s^N komplettiert das Diagramm in 3.2.3(IV) kommutativ.

(Die Einschränkungen von \dot{c}_s^T und \dot{c}_s^P sind korrekt.): Aufgrund der Vergißfunktordefinition (\rightarrow 3.1.4) gelten für alle s in \mathcal{S} die Inklusionen:

$$T_{c,s}^\otimes \subseteq T_c^\otimes \quad \text{und} \quad P_{c,s}^\otimes \subseteq P_c^\otimes.$$

Damit muß für alle Transitionen t_s in N_s^o und t_c in N_c^o gezeigt werden:

$$\dot{c}_s^T(t_s)(t_c) > 0 \implies t_c \in T_{c,s}.$$

Alle Transitionen t_s in N_s^o sind entweder Elemente in $\mathcal{DC}_{eval_s}(im)$, d.h. sie liegen in der Codomäne von $eval_s$ oder sie sind Elemente in $\mathcal{DC}_{eval_s}(co)$, d.h. sie liegen nicht in der Codomäne (\rightarrow 1.2.3). Dementsprechend gliedert sich der Beweis in zwei Teile.

$$\begin{aligned}
t_s &\in \mathcal{DC}_{eval_s}(im) \wedge \dot{c}_s^T(t_s)(t_c) > 0 && \text{Annahme} \\
\implies \exists t'_s \in N_s: \dot{c}_s^T \circ eval_s(t'_s)(t_c) > 0 &&& 1.2.3 \\
\implies \exists t'_s \in N_s: eval_c \circ c_s^T(t'_s)(t_c) > 0 &&& (b) \\
\implies t_c \in \mathcal{DC}_{eval_c \circ c_s^N}(im) &&& 1.2.3 \\
\implies t_c \in T_{c,s} \wedge t_c \in \mathcal{DC}_{eval_c}(im) &&& 3.1.4
\end{aligned}$$

Aus den Eigenschaften der A-Netzmorphismen (\rightarrow 2.2.3(V),(VI)) und der Vergißfunktorddefinition (\rightarrow 3.1.4) läßt sich der Zusammenhang

$$\begin{aligned}
(\mathbf{d}): \quad \forall m: s \rightarrow s', t \in N_s^o, t' \in N_{s'}^o: \\
\quad \hookrightarrow_{T_{s',s}^{\otimes}, T_{s'}^{\otimes}} \circ f_m^T(t)(t') > 0 \\
\quad \implies (t \in \mathcal{DC}_{eval_s}(im) \iff t' \in \mathcal{DC}_{eval_{s'}}(im)) \quad 7
\end{aligned}$$

ableiten.

Mit dem Colimes zu $\mathcal{P}^N \circ \mathcal{D}$, Lemma 6.3.11,

$$\mathcal{P}^N \left((h_m, (f_m^G, f_m^N)) \right) = (\hookrightarrow_{T_{s',s}^{\otimes}, T_{s'}^{\otimes}} \circ f_m^T, \hookrightarrow_{P_{s',s}^{\otimes}, P_{s'}^{\otimes}} \circ f_m^P, f_m^{GX})$$

und den Definitionen der Relation ' \rightarrow ' und ihrem symmetrischen und transitiven Abschluß ' \rightleftharpoons ', (\rightarrow 5.2.23) folgt daraus:

$$\begin{aligned}
\forall s, s' \in \text{Obj}(\mathcal{S}), t \in N_s^o, t' \in N_{s'}^o: \\
t \rightarrow t' \implies (t \in \mathcal{DC}_{eval_s}(im) \iff t' \in \mathcal{DC}_{eval_{s'}}(im))
\end{aligned}$$

und damit auch

$$\begin{aligned}
(\mathbf{e}): \quad \forall s, s' \in \text{Obj}(\mathcal{S}), t \in N_s^o, t' \in N_{s'}^o: \\
t \rightleftharpoons t' \implies (t \in \mathcal{DC}_{eval_s}(im) \iff t' \in \mathcal{DC}_{eval_{s'}}(im)).
\end{aligned}$$

Die zu zeigende Eigenschaft gilt, wenn alle Transitionen aus $\mathcal{DC}_{eval_s}(co)$ von \dot{c}_s^T in $\mathcal{DC}_{eval_c}(co)$ abgebildet werden. Wir beweisen diesen Teil indem wir die folgende Annahme zum Widerspruch führen:

$$\begin{aligned}
t_s \in \mathcal{DC}_{eval_s}(co), t'_c \in N_c, eval_c(t'_c)(t_c) > 0, \dot{c}_s^T(t_s)(t_c) > 0 &&& \text{Widerspruchsannahme} \\
\implies \exists s' \in \mathcal{S}, t'_{s'} \in N_{s'}: c_{s'}^T(t'_{s'})(t'_c) > 0 &&& 5.2.23, \text{Colimes } \mathcal{P}^N \circ \mathcal{D}, \\
&&& 6.3.11 \\
\implies \exists s' \in \mathcal{S}, t'_{s'} \in N_{s'}: eval_c \circ c_{s'}^T(t'_{s'})(t_c) > 0 &&& \\
\implies \exists s' \in \mathcal{S}, t'_{s'} \in N_{s'}: \dot{c}_{s'}^T \circ eval_{s'}(t'_{s'})(t_c) > 0 &&& (b)
\end{aligned}$$

$${}^7 \forall t \in N_{s',s}^o: t \in \mathcal{DC}_{eval_{s',s}}(co) \iff t \in \mathcal{DC}_{eval_{s'}}(co) \quad (\rightarrow 3.1.4)$$

$\implies \exists s' \in \mathcal{S}, t'_{s'} \in N_{s'}, \hat{t}_{s'} \in N_{s'}^o:$

$$eval_{s'}(t'_{s'}) (\hat{t}_{s'}) > 0 \wedge$$

$$\dot{c}_{s'}^T(\hat{t}_{s'})(t_c) > 0 \wedge \dot{c}_s^T(t_s)(t_c) > 0$$

$\implies \exists s' \in \mathcal{S}, t'_{s'} \in N_{s'}, \hat{t}_{s'} \in N_{s'}^o:$

$$\hat{t}_{s'} \in \mathcal{DC}_{eval_{s'}}(im) \wedge t_s \rightleftharpoons \hat{t}_{s'}$$

5.2.23, 1.2.2,
Colimes $\mathcal{P}^N \circ \mathcal{D}$, 6.3.11

$\implies t_s \in \mathcal{DC}_{eval_s}(im)$

(e)

\implies Widerspruch zur Annahme $t_s \in \mathcal{DC}_{eval_s}(co)$

Auf die gleiche Weise läßt sich die Eigenschaft für \dot{c}_s^P zeigen. □

$((c_s, (d_s^G, d_s^N))_{s \in \mathcal{S}}, AN_c)$ ist eine Senke für \mathcal{D} :

Dazu muß gezeigt werden:

- AN_c ist ein ANS_c -Modell,
- $(c_s, (d_s^G, d_s^N))$ ist ein Morphismus in $\mathcal{GAHL-NETS}$ und
- die Kommutativität.

$(AN_c$ ist ein ANS_c -Modell): Die erste Komponente von AN_c ist das Colimes-Objekt zu \mathcal{D}^{AS} und damit eine AN-Spezifikation. g_c ist ein Spezifikationsmorphismus von $Spec_c$ nach $Spec'_c$ (\rightarrow 3.2.3). A_c ist das Colimesobjekt zu \mathcal{D}^A und damit eine $Spec'_c$ -Algebra.⁸

N_c^o ist Objekt des Colimites zu $\mathcal{P}^N \circ \mathcal{D}$ in \mathcal{CPTN} und damit ein CP/T-Netz. Die Stellen- und Transitionsmengen von N_c^o sind Objekte von Colimites (\rightarrow 1.2.2) in den freien abelschen Gruppen. Sie sind endlich, da die Netze im Bild von $\mathcal{P}^N \circ \mathcal{D}$ A-Netz-Komponenten sind, diese Netze definitionsgemäß (\rightarrow 2.2.1) über endliche Stellen- und Transitionsmengen verfügen und die Basis von Objekten endlicher Colimites über freien abelschen Gruppen mit endlichen Basen endlich ist.

Gemäß unseren Annahmen ist $(d_s^{Sx}, d_s^{Ax}): A_s(X) \rightarrow A_c(X)$ die Algebrakomponente von $\dot{c}_s^N: N_s^o \rightarrow N_c^o$ für alle s in \mathcal{S} . Daraus folgt das $A_c(X)$ die Algebrakomponente von N_c^o ist und damit

$$\mathcal{P}^A(N_c^o) = A_c(X).$$

$eval_c$, die letzte Komponente von AN_c ist per Konstruktion ein CP/T-Netzmorphimus von N_c nach N_c^o (\rightarrow 3.2.3). Damit ist

$$\mathcal{P}^A(eval_c): T_{OP_c(X)} \rightarrow A_c(X)$$

⁸ $cod(d_s^S) = Spec'_c$ für s in $Obj(\mathcal{S})$.

ein generalisierter Algebromorphismus (\rightarrow 1.1.4). Gemäß unseren Annahmen gilt mit Lemma 6.3.5:

(f): $((c_s^{Sx})_{s \in \mathcal{S}}, \text{Spec}_c(X))$ ist Colimes zu $\mathcal{T}^S \circ \mathcal{D}^S$.

Aus der Senke $((d_s^S \circ g_s)_{s \in \mathcal{S}}, \text{Spec}'_c(X))$ zur Spezifikationskomponente von \mathcal{D}^A ergibt sich die Senke:

$$((d_s^{Sx} \circ g_s^x)_{s \in \mathcal{S}}, \text{Spec}'_c(X))$$

zur deren Transformation mittels \mathcal{T}^S . Damit existiert ein eindeutiger Spezifikationsmorphismus

$$\mathcal{T}^S(g_c) = g_c^x : \text{Spec}_c(X) \rightarrow \text{Spec}'_c(X) \quad \text{mit} \quad \forall s \in \mathcal{S}: g_c^x \circ c_s^{Sx} = d_s^{Sx} \circ g_s^x. \quad 9$$

Aus (b) folgt:

$$\mathcal{P}^A(c_s^N) \circ \mathcal{P}^A(\text{eval}_s) = \mathcal{P}^A(\text{eval}_c) \circ \mathcal{P}^A(c_s^N).$$

Damit erhält man unter der Annahme: $\mathcal{P}^A(\text{eval}_c) = (k, l)$ für jedes s aus \mathcal{S} das nachstehende kommutierende Diagramm

$$\begin{array}{ccc} T_{OP_s(X)} & \xrightarrow{(c_s^{Sx}, c_s^t)} & T_{OP_c(X)} \\ \downarrow (g_s^x, \text{eval}_{\mathcal{V}(A_s(X))}) & \curvearrowright & \downarrow (k, l) \\ A_s(X) & \xrightarrow{(d_s^{Sx}, d_s^{Ax})} & A_c(X) \end{array}$$

Daraus ergibt sich:

$$\forall s \in \mathcal{S}: k \circ c_s^{Sx} = d_s^{Sx} \circ g_s^x$$

$$\implies k = g_c^x$$

Eindeutigkeit von g_c^x

$$\implies l = \text{eval}_{\mathcal{V}(A_c(X))}$$

Initialität v. $T_{OP_c(X)}$

$$\implies \mathcal{P}^A(\text{eval}_c) = (k, l) = (g_c^x, \text{eval}_{\mathcal{V}(A_c(X))})$$

Als nächstes muß für alle Transitionen t'_c, t''_c aus N_c , die jeweils die Transition t aus N_c^o in ihrem Bild unter eval_c haben, gezeigt werden:

$$g_c^*(\text{Cond}_c(t'_c)) = g_c^*(\text{Cond}_c(t''_c)).$$

Mit dem Colimes zu $\mathcal{P}^N \circ \mathcal{D}$, Lemma 6.3.11 und der Definition der Relation „ \rightarrow “ und ihrem symmetrischen und transitiven Abschluß „ \rightleftharpoons “ (\rightarrow 5.2.23) und

$$\mathcal{P}^N(\mathcal{D}(m)) = \mathcal{P}^N(h_m, (f_m^G, f_m^N)) = (\hookrightarrow_{T_{s',s}^\otimes, T_{s'}^\otimes} \circ f_m^T, \hookrightarrow_{P_{s',s}^\otimes, P_{s'}^\otimes} \circ f_m^P, f_m^{GX})$$

⁹ $\mathcal{T}^S(g_c) = g_c^x$ folgt zusammen mit 3.2.3(II) aus der Eindeutigkeit von g_c^x mit der genannten Eigenschaft und

für alle Morphismen m in \mathcal{S} , zeigen wir zunächst:

$$\begin{aligned} \text{(g): } \quad & \forall t_s \in N_s^o, t_{s'} \in N_{s'}^o, t'_s \in N_s, t'_{s'} \in N_{s'}: \\ & t_s \rightharpoonup t_{s'} \wedge e_s^T(t'_s)(t_s) > 0 \wedge e_{s'}^T(t'_{s'})(t_{s'}) > 0 \\ & \implies c_s^{S^*}(\text{Cond}_s(t'_s)) = c_{s'}^{S^*}(\text{Cond}_{s'}(t'_{s'})) \end{aligned}$$

$$t_s \in N_s^o, t_{s'} \in N_{s'}^o, t'_s \in N_s, t'_{s'} \in N_{s'} \quad \text{Annahmen}$$

$$t_s \rightharpoonup t_{s'}, e_s^T(t'_s)(t_s) > 0, e_{s'}^T(t'_{s'})(t_{s'}) > 0$$

$$\implies \exists m: s \rightarrow s' \in \text{Mor}(\mathcal{S}): \hookrightarrow_{T_{s',s}^{\otimes}, T_{s'}^{\otimes}} \circ f_m^T(t_s)(t_{s'}) > 0 \quad 5.2.23$$

$$m: s \rightarrow s' \in \text{Mor}(\mathcal{S}), \hookrightarrow_{T_{s',s}^{\otimes}, T_{s'}^{\otimes}} \circ f_m^T(t_s)(t_{s'}) > 0 \quad \text{Zusatzannahme}$$

$$\implies \hookrightarrow \circ f_m^T \circ e_s^T(t'_s)(t_{s'}) > 0$$

$$\implies e_{s'}^T \circ h_m^T(t'_s)(t_{s'}) > 0 \quad 3.2.2$$

$$\implies \exists t''_{s'} \in N_{s'}: \quad e_{s'}^T(t''_{s'})(t_{s'}) > 0$$

$$h_m^T(t'_s)(t''_{s'}) > 0 \wedge e_{s'}^T(t''_{s'})(t_{s'}) > 0 \wedge \quad 2.2.1$$

$$\text{Cond}_{s'}(t'_s) = \text{Cond}_{s'}(t''_{s'})$$

$$\implies \exists t''_{s'} \in N_{s'}: \quad 3.1.1, 2.2.1$$

$$h_m^{S^*}(\text{Cond}_s(t'_s)) = \text{Cond}_{s'}(t''_{s'}) = \text{Cond}_{s'}(t'_s)$$

\implies

$$= c_s^{S^*}(\text{Cond}_s(t'_s))$$

$$= c_{s'}^{S^*} \circ h_m^{S^*}(\text{Cond}_s(t'_s)) \quad \text{Colim. } \mathcal{D}^S$$

$$= c_{s'}^{S^*}(\text{Cond}_{s'}(t'_s))$$

Mit Hilfe dieses Ergebnisses läßt sich

$$\text{(h): } \quad \forall s_1, s_n \in \text{Obj}(\mathcal{S}), t_1 \in N_{s_1}^o, t_n \in N_{s_n}^o, t'_1 \in N_{s_1}, t'_n \in N_{s_n}:$$

$$t_1 \rightrightarrows t_n \wedge e_{s_1}^T(t'_1)(t_1) > 0 \wedge e_{s_n}^T(t'_n)(t_n) > 0$$

$$\implies c_{s_1}^{S^*}(\text{Cond}_{s_1}(t'_1)) = c_{s_n}^{S^*}(\text{Cond}_{s_n}(t'_n))$$

wie folgt zeigen:

$$s_1, s_n \in \text{Obj}(\mathcal{S}), t_1 \in N_{s_1}^o, t_n \in N_{s_n}^o, t'_1 \in N_{s_1}, t'_n \in N_{s_n} \quad \text{Annahmen}$$

$$t_1 \rightrightarrows t_n \wedge e_{s_1}^T(t'_1)(t_1) > 0 \wedge e_{s_n}^T(t'_n)(t_n) > 0$$

$$t_1 \rightharpoonup t_n \vee t_n \rightharpoonup t_1 \quad \text{Fallunterscheidung}$$

$$\implies c_{s_1}^{S^*}(\text{Cond}_{s_1}(t'_1)) = c_{s_n}^{S^*}(\text{Cond}_{s_n}(t'_n)) \quad \text{(g)}$$

$$t_1 \not\rightarrow t_n \wedge t_n \not\rightarrow t_1 \quad \text{Fallunterscheidung}$$

$$\begin{aligned}
&\implies \exists s_2, \dots, s_{n-1} \in \text{Obj}(\mathcal{S}), && 5.2.23, 3.2.2 \\
&\quad t_2 \in N_{s_2}^o, \dots, t_{n-1} \in N_{s_{n-1}}^o, && e_{s_1}^T(t'_1)(t_1) > 0 \\
&\quad t'_2 \in N_{s_2}, \dots, t'_{n-1} \in N_{s_{n-1}}: \\
&\quad (t_i \multimap t_{i+1} \vee t_{i+1} \multimap t_i); \quad i = 1, \dots, n-1 \quad \wedge \\
&\quad (e_{s_i}^T(t'_i)(t_i) > 0); \quad i = 1, \dots, n
\end{aligned}$$

$$\begin{aligned}
&\implies \exists s_2, \dots, s_{n-1} \in \text{Obj}(\mathcal{S}), && (g) \\
&\quad t_2 \in N_{s_2}^o, \dots, t_{n-1} \in N_{s_{n-1}}^o, \\
&\quad t'_2 \in N_{s_2}, \dots, t'_{n-1} \in N_{s_{n-1}}: \\
&\quad c_{s_1}^{S^*}(\text{Cond}_{s_1}(t'_1)) = \dots = c_{s_n}^{S^*}(\text{Cond}_{s_n}(t'_n))
\end{aligned}$$

Damit können wir unsere eigentliche Behauptung zeigen.

$$\begin{aligned}
&\exists s, s' \in \text{Obj}(\mathcal{S}), t'_s \in N_s, t''_{s'} \in N_{s'}: && 5.2.23, \text{Colimes } \mathcal{P}^N \circ \mathcal{D}, \\
&\quad c_s^T(t'_s)(t'_c) > 0 \wedge c_{s'}^T(t''_{s'})(t''_c) > 0 && 6.3.11
\end{aligned}$$

$$\begin{aligned}
&\implies e_c^T \circ c_s^T(t'_s)(t) > 0 \wedge e_c^T \circ c_{s'}^T(t''_{s'})(t) > 0 \\
&\implies \check{c}_s^T \circ e_s^T(t'_s)(t) > 0 \wedge \check{c}_{s'}^T \circ e_{s'}^T(t''_{s'})(t) > 0 && (b)
\end{aligned}$$

$$\begin{aligned}
&\implies \exists t_s \in N_s, t_{s'} \in N_{s'}: \\
&\quad e_s^T(t'_s)(t_s) > 0 \wedge \check{c}_s^T(t_s)(t) > 0 \wedge \\
&\quad e_{s'}^T(t''_{s'})(t_{s'}) > 0 \wedge \check{c}_{s'}^T(t_{s'})(t) > 0 \\
&\implies t_s \multimap t_{s'} \wedge e_s^T(t'_s)(t_s) > 0 \wedge e_{s'}^T(t''_{s'})(t_{s'}) > 0 && 5.2.23, \text{Colimes } \mathcal{P}^N \circ \mathcal{D}, \\
&&& 6.3.11
\end{aligned}$$

\implies

$$\begin{aligned}
&\text{Cond}_c(t'_c) \\
&= c_s^{S^*}(\text{Cond}_s(t'_s)) && 3.1.1 \\
&= c_{s'}^{S^*}(\text{Cond}_{s'}(t''_{s'})) && (h) \\
&= \text{Cond}_c(t''_c) && 3.1.1 \\
&\implies g_c^*(\text{Cond}_c(t'_c)) = g_c^*(\text{Cond}_c(t''_c))
\end{aligned}$$

Abschließend muß für alle Transitionen t aus T_c gezeigt werden:

$$\forall x_{s_1}, y_{s_2} \in \text{var}_c(t): s_1 \neq s_2 \implies x \neq y$$

$$t \in T_c, x_{s_1}, y_{s_2} \in \text{var}_c(t) \quad \text{Annahme}$$

$$\implies \exists s \in \text{Obj}(\mathcal{S}), t_s \in T_s: \check{c}_s^T(t_s)(t) > 0 \quad 5.2.23, \text{Colimes } \mathcal{P}^N \circ \mathcal{D}, 6.3.11$$

$$\begin{aligned}
&\implies \exists s \in \text{Obj}(\mathcal{S}), t_s \in T_s, \forall (p, a) \in P_c \times A_c(X): && \dot{c}_s^N \in \mathcal{CPTN}, \\
&\quad \text{Pre}_c(t)(p, a) > 0 && \text{Kompat. mit } \text{Pre}_c, \text{Post}_c \\
&\quad \implies \dot{c}_s^P \otimes (d_s^{Sx}, d_s^{Ax})(\text{Pre}_s(t_s))(p, a) > 0 \wedge \\
&\quad \text{Post}_c(t)(p, a) > 0 && \\
&\quad \implies \dot{c}_s^P \otimes (d_s^{Sx}, d_s^{Ax})(\text{Post}_s(t_s))(p, a) > 0 \\
&\implies \exists s \in \text{Obj}(\mathcal{S}), t_s \in T_s, x_{s'_1}, y_{s'_2} \in \text{var}_s(t_s): && d_s^{Ax} \text{ bewahrt Var.} \\
&\quad (d_s^{Sx}, d_s^{Ax})(x_{s'_1}) = x_{s_1} \wedge (d_s^{Sx}, d_s^{Ax})(y_{s'_1}) = y_{s_1} \\
&\implies s_1 \neq s_2 \implies s'_1 \neq s'_2 \implies x \neq y && \mathcal{D}(s) \in \text{MOD}(\mathcal{D}^{AS}(s)), \\
& && 2.2.1
\end{aligned}$$

(($c_s, (d_s^G, d_s^N)$) ist ein Morphismus in $\mathcal{GAHL}\text{-NETS}$): Für alle s in \mathcal{S} ist c_s ein Morphismus von ANS_s nach ANS_c im Colimes zu \mathcal{D}^{AS} . Damit verbleibt zu zeigen, daß (d_s^G, d_s^N) ein A-Netzmorphismus von AN_s nach $\mathcal{V}_{c_s}(AN_c)$ ist, d.h. mit Definition 2.2.3:

- (1) d_s^G ist ein generalisierter Algebromorphismus von A_s nach A_c ,
- (2) d_s^N ist ein CP/T-Netzmorphismus von N_s^o nach $N_{c,s}^o$,
- (3) $\mathcal{P}^A(d_s^N) = \mathcal{T}^A(d_s^G)$,
- (4) $d_s^N \circ \text{eval}_s = \text{eval}_{c,s}$ und
- (5) $d_s^N \Big|_{\mathcal{DC}_{\text{eval}_s}(co)} \in [\mathcal{DC}_{\text{eval}_s}(co) \longrightarrow \mathcal{DC}_{\text{eval}_{c,s}}]$.

(.1): $d_s^G: A_s \rightarrow A_c$ ist nach den Voraussetzungen ein Morphismus im Colimes zu \mathcal{D}^A in $\mathcal{GSP\mathcal{E}C}\text{-ALG}$.

(.2): d_s^N ist als Einschränkung des CP/T-Netzmorphismus \dot{c}_s^N auf Stellen und Transitionen aus $N_{c,s}^o$ definiert und wir haben bereits gezeigt, daß diese Einschränkung korrekt ist. Damit bildet die Stellen-/Transitionskomponente von d_s^N einen monotonen Morphismus in den freien abelschen Gruppen. Die Algebrakomponente von d_s^N ist ein generalisierter Morphismus von $A_c(X)$ nach $A_{\bar{c}}(X)$ und identisch mit der Algebrakomponente von \dot{c}_s^N . Für jede Transition und jedes Paar aus einer Stelle und einem Algebraelement aus N_s^o , ist der Effekt von d_s^N mit dem Effekt von \dot{c}_s^N identisch, daher folgt aus der Verträglichkeit von \dot{c}_s^N mit den Markenflußmorphisms von N_s^o und N_c^o die Verträglichkeit von d_s^N .

(.3): Folgt direkt aus der Definition von d_s^N .

(.4):

$$d_s^N \circ \text{eval}_s$$

$$\begin{aligned}
&= \left(d_s^T \circ e_s^T, d_s^P \circ e_s^P, (d_s^{SX}, d_s^{AX}) \circ (g_s^X, eval_{\mathcal{V}(A_s(X))}) \right) \\
&= \left((\dot{c}_s^T \circ e_s^T) \Big|_{T_{c,s}^\otimes}, (\dot{c}_s^P \circ e_s^P) \Big|_{P_{c,s}^\otimes}, \right. \\
&\quad \left. (d_s^{SX}, d_s^{AX}) \circ (g_s^X, eval_{\mathcal{V}(A_s(X))}) \right) \tag{c} \\
&= \left((e_c^T \circ c_s^T) \Big|_{T_{c,s}^\otimes}, (e_c^P \circ c_s^P) \Big|_{P_{c,s}^\otimes}, \right. \\
&\quad \left. (g_c^X, eval_{\mathcal{V}(A_c(X))}) \circ (c_s^{SX}, c_s^t) \right) \tag{b} \\
&= \left(e_{c,s}^T, e_{c,s}^P, (g_c^X \circ c_s^{SX}, \mathcal{V}_{c_s^{SX}}(eval_{\mathcal{V}_{g_c^X}(A_c(X))}) \circ c_s^t) \right) \tag{3.1.4} \\
&= eval_{c,s} \tag{3.1.4, } T_{OP_s(X)} \text{ is initial}
\end{aligned}$$

(.5): Im ersten Beweisschritt haben wir nachgewiesen, daß \dot{c}_s^N Stellen und Transitionen aus $\mathcal{DC}_{eval_s}(co)$ auf die entsprechenden Elemente in $\mathcal{DC}_{eval_c}(co)$ abbildet. Aus der Definition der Vergißfunktoren (\rightarrow 3.1.4) folgt, daß alle Stellen und Transitionen aus $\mathcal{DC}_{eval_c}(co)$ in $\mathcal{DC}_{eval_{c,s}}(co)$ enthalten sind, sodaß mit der Definition der Stellen- und Transitionskomponente von d_s^N , als Einschränkung der entsprechenden Komponente von \dot{c}_s^N , unsere Behauptung für diese Teile von d_s^N bereits gezeigt ist. Für die Algebraelemente gilt sie ebenfalls, da in jedem Teil einer \mathcal{CPTN} -Zerlegung die gesamte Algebra enthalten ist.

(Kommutativität): Für alle Morphismen $m: s \rightarrow s'$ in \mathcal{S} gilt

$$\mathcal{P}^N((h_m, (f_m^G, f_m^N))) = (\hookrightarrow_{T_{s',s}^\otimes, T_s^\otimes} \circ f_m^T, \hookrightarrow_{P_{s',s}^\otimes, P_s^\otimes} \circ f_m^P, f_m^{GX})$$

(\rightarrow 3.2.1). Aus den Funktoreigenschaften der Vergißfunktorkonstruktion (\rightarrow 3.1.8) folgt

$$\mathcal{V}_{h_m} \circ \mathcal{V}_{c_{s'}} = \mathcal{V}_{c_{s'} \circ h_m} = \mathcal{V}_{c_s}$$

und damit $T_{c_{s'},s}^\otimes = T_{c,s}^\otimes$ bzw. $P_{c_{s'},s}^\otimes = P_{c,s}^\otimes$ wobei $T_{c_{s'},s}^\otimes / P_{c_{s'},s}^\otimes$ die Transitions-/Stellengruppen in $N_{c_{s'},s}^o$ der Netzstruktur von $\mathcal{V}_{h_m}(\mathcal{V}_{c_{s'}}(AN_c))$ bezeichnen.

Wir zeigen für alle m in \mathcal{S} :

$$\begin{aligned}
&(c_{s'}, (d_{s'}^G, d_{s'}^N)) \circ \mathcal{D}(m) \\
&= (c_{s'}, (d_{s'}^G, d_{s'}^N)) \circ (h_m, (f_m^G, f_m^N)) \\
&= (c_{s'} \circ h_m, \mathcal{V}_{h_m}((d_{s'}^G, d_{s'}^N)) \circ (f_m^G, f_m^N)) \\
&= \left(c_s, \left(d_{s'}^G \circ f_m^G, \right. \right. \\
&\quad \left. \left(d_{s'}^T \Big|_{T_{c,s}^\otimes} \circ f_m^T, d_{s'}^P \Big|_{P_{c,s}^\otimes} \circ f_m^P, \right. \right. \\
&\quad \left. \left. d_{s'}^{GX} \circ f_m^{GX} \right) \right) \tag{Colim. zu } \mathcal{D}^{AS}, \text{ 3.1.4}
\end{aligned}$$

$$\begin{aligned}
&= \left(c_s, \left(d_s^G, \right. \right. && \text{Colim. zu } \mathcal{D}^A, 3.2.3 \\
&\quad \left. \left((\dot{c}_{s'}^T \circ \hookrightarrow \circ f_m^T) \Big|_{T_{c,s}^\otimes}, (\dot{c}_{s'}^P \circ \hookrightarrow \circ f_m^P) \Big|_{P_{c,s}^\otimes}, \right. \right. \\
&\quad \left. \left. d_{s'}^{GX} \circ f_m^{GX} \right) \right) \\
&= \left(c_s, (d_s^G, (\dot{c}_s^T \Big|_{T_{c,s}^\otimes}, \dot{c}_s^P \Big|_{P_{c,s}^\otimes}, d_s^{GX})) \right) && \text{Colim. zu } \mathcal{P}^N \circ \mathcal{D}, 3.2.1 \\
&= (c_s, (d_s^G, d_s^N)) && 3.2.3
\end{aligned}$$

□

(Couniverselle Eigenschaft):

Wir nehmen im weiteren an, daß $((\bar{c}_s, (\bar{d}_s^G, \bar{d}_s^N)))_{s \in \mathcal{S}}, AN_{\bar{c}}$ mit

- $\bar{c}_s = (\bar{c}_s^S, \bar{c}_s^N)$,
- $\bar{d}_s^N = (\bar{d}_s^T, \bar{d}_s^P, \bar{d}_s^{GX})$,
- $AN_{\bar{c}} = (ANS_{\bar{c}}, g_{\bar{c}}, A_{\bar{c}}, N_{\bar{c}}^o, eval_{\bar{c}})$ und
- $\mathcal{V}_{\bar{c}_s}(AN_{\bar{c}}) = (ANS_s, g_{\bar{c}} \circ \bar{c}_s^S, A_{\bar{c}}, N_{\bar{c},s}^o, eval_{\bar{c},s})$ mit

$$N_{\bar{c},s}^o = T_{\bar{c},s}^\otimes \xrightarrow[Post_{\bar{c},s}]{Pre_{\bar{c},s}} P_{\bar{c},s}^\otimes \otimes A_{\bar{c}}(X)$$

eine Senke für \mathcal{D} in $\mathcal{GAHLN-NESTS}$ ist und zeigen:

- es existiert ein Morphismus von AN_c nach $AN_{\bar{c}}$ in $\mathcal{GAHLN-NESTS}$,
- dieser Morphismus ist mit den $(c_s, (d_s^G, d_s^N))$ für alle s in \mathcal{S} verträglich und
- er ist eindeutig mit dieser Eigenschaft.

(Es existiert ein Morphismus von AN_c nach $AN_{\bar{c}}$): Im folgenden konstruieren wir den generalisierten Morphismus:

$$(\hat{c}, (\hat{d}^G, \hat{d}^N)): AN_c \rightarrow AN_{\bar{c}}.$$

Aus der neuen Senke in unserer Annahme ergibt sich:

- eine Senke $((\bar{c}_s)_{s \in \mathcal{S}}, ANS_{\bar{c}})$ für \mathcal{D}^{AS} in $\mathcal{AHLN-SPEC}$ und damit aufgrund des Colimites zu \mathcal{D}^{AS} ein eindeutiger $\mathcal{AHLN-SPEC}$ -Morphismus

$$(i): \quad \hat{c} = (\hat{c}^S, \hat{c}^N): ANS_c \rightarrow ANS_{\bar{c}} \quad \text{mit} \quad \hat{c} \circ c_s = \bar{c}_s \quad \text{für alle } s \text{ in } \mathcal{S};$$

- eine Senke $((\bar{d}_s^G)_{s \in \mathcal{S}}, A_{\bar{c}})$ für \mathcal{D}^A in $\mathcal{GSPEC-ALG}$ und damit aufgrund des Colimites zu \mathcal{D}^A ein eindeutiger $\mathcal{GSPEC-ALG}$ -Morphismus

$$(j): \quad \hat{d}^G = (\hat{d}^S, \hat{d}^A): A_c \rightarrow A_{\bar{c}} \quad \text{mit} \quad \hat{d}^G \circ d_s^G = \bar{d}_s^G \quad \text{für alle } s \text{ in } \mathcal{S};$$

– eine Senke $(\mathcal{P}^N((\bar{c}_s, (\bar{d}_s^G, \bar{d}_s^N))))_{s \in \mathcal{S}}, N_{\bar{c}}^o)$ für $\mathcal{P}^N \circ \mathcal{D}$ in \mathcal{CPTN} und damit aufgrund des Colimites zu $\mathcal{P}^N \circ \mathcal{D}$ ein eindeutiger \mathcal{CPTN} -Morphismus

$$(k): \quad \hat{c}\bar{c}: N_{\bar{c}}^o \rightarrow N_{\bar{c}}^o \quad \text{mit} \quad \hat{c}\bar{c} \circ \hat{c}_s^N = \mathcal{P}^N((\bar{c}_s, (\bar{d}_s^G, \bar{d}_s^N))) \quad \text{für alle } s \text{ in } \mathcal{S};$$

– eine Senke $(\mathcal{T}^A(\bar{d}_s^G)_{s \in \mathcal{S}}, \mathcal{T}^A(A_{\bar{c}}))$ für $\mathcal{T}^A \circ \mathcal{D}^A$ in $\mathcal{GSP\mathcal{E}C-ALG}$ und damit aufgrund des Colimites $((\mathcal{T}^A(d_s^S, d_s^A))_{s \in \text{Obj}(\mathcal{S})}, \mathcal{T}^A(A_c))$ zu $\mathcal{T}^A \circ \mathcal{D}^A$, der aus der Bewahrung von Colimites durch \mathcal{T}^A (\rightarrow 6.3.10) und dem Colimites zu \mathcal{D}^A folgt, ein eindeutiger \mathcal{CPTN} -Morphismus

$$(l): \quad \tilde{c}: \mathcal{T}^A(A_c) \rightarrow \mathcal{T}^A(A_{\bar{c}}) \quad \text{mit} \quad \tilde{c} \circ \mathcal{T}^A(d_s^G) = \mathcal{T}^A(\bar{d}_s^G) \quad \text{für alle } s \text{ in } \mathcal{S}.$$

Wir zeigen zunächst mit Hilfe von (l): $\mathcal{P}^A(\hat{c}\bar{c}) = \tilde{c} = \mathcal{T}^A(\hat{d}^G)$.

$\forall s \in \text{Obj}(\mathcal{S})$:

$$\mathcal{P}^A(\hat{c}\bar{c}) \circ \mathcal{P}^A(\hat{c}_s^N) = \mathcal{P}^A(\mathcal{P}^N((\bar{c}_s, (\bar{d}_s^G, \bar{d}_s^N)))) \quad (k)$$

$$\implies \mathcal{P}^A(\hat{c}\bar{c}) \circ \mathcal{T}^A(d_s^G) = \mathcal{T}^A(\bar{d}_s^G) \quad 3.2.3, 2.2.3$$

$$\implies \mathcal{P}^A(\hat{c}\bar{c}) = \tilde{c} \quad \text{Eindeutigkeit } \tilde{c}$$

$\forall s \in \text{Obj}(\mathcal{S})$:

$$\mathcal{T}^A(\hat{d}^G) \circ \mathcal{T}^A(d_s^G) = \mathcal{T}^A(\bar{d}_s^{Gx}) \quad (j)$$

$$\implies \mathcal{T}^A(\hat{d}^G) = \tilde{c} = \mathcal{P}^A(\hat{c}\bar{c}) \quad \text{Eindeutigkeit } \tilde{c}$$

Mit der Definition der Vergißfunktoren (\rightarrow 3.1.4) erhalten wir

$$\mathcal{V}_{\hat{c}}(AN_{\bar{c}}) = (ANS_c, g_{\bar{c}} \circ \hat{c}^{Sx}, A_{\bar{c}}, N_{\bar{c},c}^o, \text{eval}_{\bar{c},c})$$

$$\text{mit} \quad N_{\bar{c},c}^o = T_{\bar{c},c}^{\otimes} \xrightarrow[\text{Post}_{\bar{c},c}]{\text{Pre}_{\bar{c},c}} P_{\bar{c},c}^{\otimes} \otimes A_{\bar{c}}(X)$$

als Vergißbild von $AN_{\bar{c}}$ unter dem von \hat{c} generierten Vergißfunktore $\mathcal{V}_{\hat{c}}$ und definieren einen Morphismus $\hat{d}^N: N_c^o \rightarrow N_{\bar{c},c}^o$ durch:

$$\hat{d}^N \stackrel{\text{def}}{=} (\hat{d}^T, \hat{d}^P, \hat{d}^{Gx}) \quad \text{mit} \quad \hat{d}^T = \hat{c}\bar{c}^T \Big|_{T_{\bar{c},c}^{\otimes}}, \quad \hat{d}^P = \hat{c}\bar{c}^P \Big|_{P_{\bar{c},c}^{\otimes}} \quad \text{und} \quad \hat{d}^{Gx} = \mathcal{T}^A(\hat{d}^G).$$

Damit sind alle Komponenten des generalisierten Morphismus definiert und es muß gezeigt werden:

- die Definition von \hat{d}^N ist korrekt,
- (\hat{d}^G, \hat{d}^N) ist ein ANS_c -Modellmorphismus und
- die Kompatibilität von $(\hat{c}, (\hat{d}^G, \hat{d}^N))$ mit den Morphismen Senke.

(Die Definition von \widehat{d}^N ist korrekt): Wir zeigen, daß die Einschränkungen von \widehat{c}^T , \widehat{c}^P auf die Stellen- bzw. Transitionen aus $N_{\bar{c},c}^o$ zu Morphismen in den freien abelschen Gruppen führen. Dazu nutzen wir den Zusammenhang

$$\widehat{c} \circ c_s = \bar{c}_s \implies \mathcal{V}_{c_s}(\mathcal{V}_{\widehat{c}}(AN_{\bar{c}})) = \mathcal{V}_{\bar{c}_s}(AN_{\bar{c}})$$

aus dem mit der Vergißfunktordefinition (\rightarrow 3.1.4) $T_{\bar{c},s} \subseteq T_{\bar{c},c}$ und $P_{\bar{c},s} \subseteq P_{\bar{c},c}$ folgt.

$$\begin{aligned} t_c \in T_c, t_{\bar{c}} \in T_{\bar{c}}, \widehat{c}^T(t_c)(t_{\bar{c}}) > 0 & \quad \text{Annahmen} \\ \implies \exists s \in \mathcal{S}, t_s \in T_s: \dot{c}_s^T(t_s)(t_c) > 0 & \quad 5.2.23, \text{ Colimes } \mathcal{P}^N \circ \mathcal{D}, \\ & \quad 6.3.11 \\ \implies \exists s \in \mathcal{S}, t_s \in T_s: \widehat{c}^T \circ \dot{c}_s^T(t_s)(t_{\bar{c}}) > 0 & \\ \implies \exists s \in \mathcal{S}, t_s \in T_s: \hookrightarrow_{T_{\bar{c},s}^\otimes, T_{\bar{c}}^\otimes} \circ \bar{c}_s^T(t_s)(t_{\bar{c}}) > 0 & \quad (\text{k}), 3.2.1 \\ \implies t_{\bar{c}} \in T_{\bar{c},s} \subseteq T_{\bar{c},c} & \quad \text{Codom. } \bar{c}_s^T \text{ ist } T_{\bar{c},s}^\otimes \end{aligned}$$

Der Beweis für Stellen entspricht dem für Transitionen.

($(\widehat{d}^G, \widehat{d}^N)$ ist ein ANS_c -Modellmorphismus.): Es muß gezeigt werden (\rightarrow 2.2.3):

- (1) $\widehat{d}^G: A_c \rightarrow A_{\bar{c}}$ ist ein generalisierter Algebromorphismus,
- (2) $\widehat{d}^N: N_c^o \rightarrow N_{\bar{c},c}^o$ ist ein CP/T-Netzmorphismus,
- (3) $\mathcal{P}^A(\widehat{d}^N) = \widehat{d}^G x$,
- (4) $\widehat{d}^N \circ eval_c = eval_{\bar{c},c}$ und
- (5) $\widehat{d}^N \Big|_{eval_c(co)} \in [eval_c(co) \longrightarrow eval_{\bar{c},c}(co)]$.

(.1): (\rightarrow (j))

(.2): \widehat{d}^N ist als Einschränkung des CP/T-Netzmorphismus $\widehat{c}c$ auf Stellen und Transitionen aus $N_{\bar{c},c}^o$ definiert und wir haben bereits gezeigt, daß diese Einschränkung korrekt ist. Damit bildet die Stellen-/Transitionskomponente von \widehat{d}^N einen monotonen Morphismus in den freien abelschen Gruppen. Die Algebrakomponente ist mit der von $\widehat{c}c$ identisch, d.h. ein generalisierter Morphismus von $A_c(X)$ nach $A_{\bar{c}}(X)$. Für jede Transition und jedes Paar aus einer Stelle und einem Algebraelement aus N_c^o , ist der Effekt von \widehat{d}^N mit dem Effekt von $\widehat{c}c$ identisch, daher folgt aus der Verträglichkeit von $\widehat{c}c$ mit den Markenflußmorphismen von N_c^o und $N_{\bar{c},c}^o$ die Verträglichkeit von \widehat{d}^N .

(.3): Folgt direkt aus der Definition von \widehat{d}^N .

(.4): Aus 3.2.2 und unserer Annahme, dass $((\bar{c}_s, (\bar{d}_s^G, \bar{d}_s^N))_{s \in \mathcal{S}}, AN_{\bar{\tau}})$ eine Senke für \mathcal{D} bildet, folgt für alle $m: s \rightarrow s'$:

$$\begin{array}{ccccc}
 N_s & \xrightarrow{h_m^N} & N_{s'} & & \\
 \downarrow \text{eval}_s & \searrow \bar{c}_s^N & \swarrow \bar{c}_{s'}^N & \circlearrowright & \downarrow \text{eval}_{s'} \\
 & & N_{\bar{c}} & & \\
 & \circlearrowleft & \downarrow \text{eval}_{\bar{\tau}} & \circlearrowright & \\
 N_s^o & \xrightarrow{\mathcal{P}^N(\bar{c}_s, (\bar{d}_s^G, \bar{d}_s^N))} & N_{\bar{c}}^o & \xleftarrow{\mathcal{P}^N(\bar{c}_{s'}, (\bar{d}_{s'}^G, \bar{d}_{s'}^N))} & N_{s'}^o
 \end{array}$$

Damit ist

$$((\mathcal{P}^N(\bar{c}_s, (\bar{d}_s^G, \bar{d}_s^N)) \circ \text{eval}_s)_{s \in \mathcal{S}}, N_{\bar{c}}^o)$$

eine Senke für \mathcal{D}^N und aufgrund der Eindeutigkeit von Colimites existiert ein eindeutiger CP/T-Netzmorphimus

$$\widehat{c}_n: N_c \rightarrow N_{\bar{c}}^o \quad \text{mit} \quad \widehat{c}_n \circ c_s^N = \mathcal{P}^N(\bar{c}_s, (\bar{d}_s^G, \bar{d}_s^N)) \circ \text{eval}_s$$

für alle s in $\text{Obj}(\mathcal{S})$. Im folgenden zeigen wir mit Hilfe der Eindeutigkeit von \widehat{c}_n :

$$\widehat{c}_c \circ \text{eval}_c = \widehat{c}_n = \text{eval}_{\bar{\tau}} \circ \widehat{c}^N.$$

$$\widehat{c}_c \circ \text{eval}_c \circ c_s^N$$

$$= \widehat{c}_c \circ c_s^N \circ \text{eval}_s \tag{b)}$$

$$= \mathcal{P}^N(\bar{c}_s, (\bar{d}_s^A, \bar{d}_s^N)) \circ \text{eval}_s \tag{k)}$$

$$\implies \widehat{c}_c \circ \text{eval}_c = \widehat{c}_n \tag{\text{Eindeutigkeit v. } \widehat{c}_n}$$

$$\text{eval}_{\bar{\tau}} \circ \widehat{c}^N \circ c_s^N$$

$$= \text{eval}_{\bar{\tau}} \circ \bar{c}_s^N \tag{i)}$$

$$= \mathcal{P}^N(\bar{c}_s, (\bar{d}_s^A, \bar{d}_s^N)) \circ \text{eval}_s \tag{3.2.2}$$

$$\implies \widehat{c}_n = \text{eval}_{\bar{\tau}} \circ \widehat{c}^N \tag{\text{Eindeutigkeit v. } \widehat{c}_n}$$

Damit kommutiert das Diagramm

$$\begin{array}{ccc}
 N_c & \xrightarrow{\widehat{c}^N} & N_{\bar{c}} \\
 \text{eval}_c \downarrow & \circlearrowright & \downarrow \text{eval}_{\bar{\tau}} \\
 N_c^o & \xrightarrow{\widehat{c}_c} & N_{\bar{c}}^o
 \end{array}$$

in \mathcal{CPTN} und wir können zeigen:

$$\begin{aligned}
& \widehat{d}^N \circ eval_c \\
&= \left(\widehat{d}^T \circ e_c^T, \widehat{d}^P \circ e_c^P, \widehat{d}^{GX} \circ (g_c^X, eval_{\mathcal{V}(A_c(X))}) \right) \\
&= \left(\widehat{c}^T \Big|_{T_{\widehat{c},c}^{\otimes}} \circ e_c^T, \widehat{c}^P \Big|_{P_{\widehat{c},c}^{\otimes}} \circ e_c^P, \widehat{d}^{GX} \circ (g_c^X, eval_{\mathcal{V}(A_c(X))}) \right) && \text{Def. } \widehat{c}^T / \widehat{c}^P \\
&= \left((e_{\widehat{c}}^T \circ \widehat{c}^T) \Big|_{T_{\widehat{c},c}^{\otimes}}, (e_{\widehat{c}}^P \circ \widehat{c}^P) \Big|_{P_{\widehat{c},c}^{\otimes}}, (g_{\widehat{c}}^X, eval_{\mathcal{V}(\dots)}) \circ (\widehat{c}^{SX}, \widehat{c}^t) \right) && \text{Diagramm} \\
&= \left(e_{\widehat{c},c}^T, e_{\widehat{c},c}^P, (g_{\widehat{c}}^X \circ \widehat{c}^{SX}, \mathcal{V}_{\widehat{c}}(eval_{\mathcal{V}(A_{\widehat{c}}(X))}) \circ \widehat{c}^t) \right) && 3.1.4 \\
&= eval_{\widehat{c},c} && T_{OP_c(X)} \text{ ist initial}
\end{aligned}$$

(.5): Dekompositionen von CP/T-Netzen über CP/T-Netzmorphismen (\rightarrow 1.2.3) haben auf die Datentypen des Netzes keinen Einfluß, d.h. sie sind in allen Bestandteilen gleich dem ursprünglichen Datentyp.

$$\mathcal{P}^A(\mathcal{DC}_{eval_{\widehat{c},c}}(co)) = \mathcal{P}^A(N_{\widehat{c},c}^o)$$

Damit gilt die zu zeigende Eigenschaft, wenn sie für die Stellen- und Transitionskomponente von \widehat{d}^N gilt. Wir geben den Beweis für \widehat{c}^T an, der Beweis für \widehat{c}^P erfolgt auf die gleiche Weise.

$$\begin{aligned}
& t_c \in \mathcal{DC}_{eval_c}(co), t_{\widehat{c}} \in N_{\widehat{c},c}^o, \widehat{d}^T(t_c)(t_{\widehat{c}}) > 0 && \text{Annahmen} \\
& \implies \exists s \in \mathcal{S}, t_s \in \mathcal{DC}_{eval_s}(co): \dot{c}_s^T(t_s)(t_c) > 0 && 5.2.23, \text{ Colimes } \mathcal{P}^N \circ \mathcal{D}, \\
& && 6.3.11, \text{ S. 216, 1.2.3} \\
& \implies \exists s \in \mathcal{S}, t_s \in \mathcal{DC}_{eval_s}(co): \widehat{d}^T \circ \dot{c}_s^T(t_s)(t_{\widehat{c}}) > 0 \\
& \implies \exists s \in \mathcal{S}, t_s \in \mathcal{DC}_{eval_s}(co): \widehat{c}^T \circ \dot{c}_s^T(t_s)(t_{\widehat{c}}) > 0 && \text{Def. } \widehat{d}^T \\
& \implies \exists s \in \mathcal{S}, t_s \in \mathcal{DC}_{eval_s}(co): \hookrightarrow_{T_{\widehat{c},s}^{\otimes}, T_{\widehat{c}}^{\otimes}} \circ \overline{d}_s^T(t_s)(t_{\widehat{c}}) > 0 && (k) \\
& \implies t_{\widehat{c}} \in \mathcal{DC}_{eval_{\widehat{c},s}}(co) = \mathcal{DC}_{eval_{\widehat{c},c}}(co) && 2.2.3, 3.1.4, \mathcal{V}_{\widehat{c},s} = \mathcal{V}_{c_s} \circ \mathcal{V}_{\widehat{c}}
\end{aligned}$$

Bis jetzt haben wir gezeigt, daß $(\widehat{d}^G, \widehat{d}^N)$ ein A-Netzmorphismus von AN_c nach $\mathcal{V}_{\widehat{c}}(AN_{\widehat{c}})$ ist. Daraus folgt mit der Definition generalisierten Morphismen (\rightarrow 5.3.2):

$$(\widehat{c}, (\widehat{d}^G, \widehat{d}^N)): AN_c \rightarrow AN_{\widehat{c}} \in \mathcal{GAHL-NETS}.$$

(Kompatibilität): Mit den bisherigen Ergebnissen und der Definition von $d_s^N(c)$ läßt sich für alle Objekte s in \mathcal{S} ableiten:

$$(\widehat{c}, (\widehat{d}^G, \widehat{d}^N)) \circ (c_s, (d_s^G, d_s^N))$$

$$\begin{aligned}
&= (\widehat{c} \circ c_s, \mathcal{V}_{c_s}(\widehat{d}^G, \widehat{d}^N) \circ (d_s^G, d_s^N)) \\
&= \left(\overline{c}_s, \left(\widehat{d}^G \circ d_s^G, \right. \right. && \text{(i), 3.1.4} \\
&\quad \left. \left. \left(\widehat{d}^T \Big|_{T_{\widehat{c},s}^{\otimes}}^{\otimes} \circ d_s^T, \widehat{d}^P \Big|_{P_{\widehat{c},s}^{\otimes}}^{\otimes} \circ d_s^P, \widehat{d}^{GX} \circ d_s^{GX} \right) \right) \right) \\
&= \left(\overline{c}_s, \left(\overline{d}_s^G, \right. \right. && \text{(i), 3.1.4} \\
&\quad \left. \left. \left((\widehat{d}^T \circ \dot{c}_s^T) \Big|_{T_{\overline{c},s}^{\otimes}}^{\otimes}, (\widehat{d}^P \circ \dot{c}_s^P) \Big|_{P_{\overline{c},s}^{\otimes}}^{\otimes}, \overline{d}_s^{GX} \right) \right) \right) && \text{Def. } d_s^T, d_s^P \\
&= \left(\overline{c}_s, \left(\overline{d}_s^G, \right. \right. && T_{\overline{c},s}^{\otimes} \subseteq T_{\overline{c},c}^{\otimes} \\
&\quad \left. \left. \left((\widehat{c}^T \circ \dot{c}_s^T) \Big|_{T_{\overline{c},s}^{\otimes}}^{\otimes}, (\widehat{c}^P \circ \dot{c}_s^P) \Big|_{P_{\overline{c},s}^{\otimes}}^{\otimes}, \overline{d}_s^{GX} \right) \right) \right) && P_{\overline{c},s}^{\otimes} \subseteq P_{\overline{c},c}^{\otimes} \\
&= (\overline{c}_s, (\overline{d}_s^G, (\overline{d}_s^T, \overline{d}_s^P, \overline{d}_s^{GX}))) && \text{(k), 3.2.1} \\
&= (\overline{c}_s, (\overline{d}_s^G, \overline{d}_s^N))
\end{aligned}$$

(Eindeutigkeit): Unter der Annahme, daß $(\tilde{c}, (\tilde{c}^G, \tilde{c}^N)): AN_c \rightarrow AN_{\overline{c}}$ ein beliebiger Morphismus in $\mathcal{GAHL-NETS}$ mit

$$(\tilde{c}, (\tilde{c}^G, \tilde{c}^N)) \circ (c_s, (d_s^G, d_s^N)) = (\overline{c}_s, (\overline{d}_s^G, \overline{d}_s^N))$$

für alle Objekte s in \mathcal{S} ist, zeigen wir:

$$(\tilde{c}, (\tilde{c}^G, \tilde{c}^N)) = (\widehat{c}, (\widehat{d}^G, \widehat{d}^N)).$$

Unsere Annahme impliziert, zusammen mit der Definition der $\mathcal{GAHL-NETS}$ -Morphismen (\rightarrow 5.3.2), der Definition von \mathcal{P}^N (\rightarrow 3.2.1) und $\dot{c}_s^N = \mathcal{P}^N(c_s, (d_s^G, d_s^N))$ für alle Objekte s in \mathcal{S} :

- (m): $\tilde{c} \circ c_s = \overline{c}_s$
- (n): $\tilde{c}^G \circ d_s^G = \overline{d}_s^G$
- (o): $\mathcal{P}^N(\tilde{c}, (\tilde{c}^G, \tilde{c}^N)) \circ \dot{c}_s^N = \mathcal{P}^N(\overline{c}_s, (\overline{d}_s^G, \overline{d}_s^N))$

Aus (m), (n) und (o) folgt aufgrund der Eindeutigkeit von \widehat{c} , \widehat{d}^G , resp. $\widehat{c}\widehat{c}$ direkt:

$$\tilde{c} = \widehat{c}, \quad \tilde{c}^G = \widehat{d}^G, \quad \mathcal{P}^N((\tilde{c}, (\tilde{c}^G, \tilde{c}^N))) = \widehat{c}\widehat{c}$$

und damit

$$\mathcal{P}^N((\tilde{c}, (\tilde{c}^G, \tilde{c}^N))) = \mathcal{P}^N((\widehat{c}, (\widehat{d}^G, \tilde{c}^N))) = \widehat{c}\widehat{c}.$$

Mit den Definitionen der Projektion (\rightarrow 3.2.1), der A-Netzmorphismen (\rightarrow 2.2.3) und von \widehat{d}^N folgt daraus:

$$\widetilde{c}^N = (\widehat{c}c^T \Big|_{T_{\widetilde{c},c}^{\otimes}}, \widehat{c}c^P \Big|_{P_{\widetilde{c},c}^{\otimes}}, \widehat{d}^{G_X}) = \widehat{d}^N.$$

Die Gleichheit aller Komponenten zweier Morphismen in $\mathcal{GAHL-NETS}$ impliziert die Gleichheit der Morphismen selbst, d.h. in unserem Fall:

$$(\widetilde{c}, (\widetilde{c}^G, \widetilde{c}^N)) = (\widehat{c}, (\widehat{d}^G, \widehat{d}^N)).$$

□

□

Beweis (3.2.6 Eigenschaften der erweiterten operationalen Semantik):

(I):

Es muß gezeigt werden:

- für alle generalisierten Morphismen $(h, f): AN_1 \rightarrow AN_2$ ist

$$\mathcal{UF}((h, f)): \mathcal{UF}(AN_1) \rightarrow \mathcal{UF}(AN_2)$$

ein CP/T-Netzmorphismus,

- \mathcal{UF} bewahrt die Identitäten und
- \mathcal{UF} bewahrt die Komposition von Morphismen.

($\mathcal{UF}((h, f)) \in \mathcal{CPTN}$): Wir nehmen für $i = 1, 2$ o.B.d.A. an:

– $ANS_i = (Spec_i, Cond_i, T_{I,i}^{\otimes} \xrightarrow[Post_i]{Pre_i} P_{I,i}^{\otimes} \otimes T_{OP_i(X)})$ ist eine AN-Spezifikation,

– $h = (h^S, h^N): ANS_1 \rightarrow ANS_2$ ist ein AN-Spezifikationsmorphismus,

– $AN_i = (ANS_i, g_i, A_i, N_i^o, eval_i)$ ist ein ANS_i -Modell mit

$$g_i: Spec_i \rightarrow Spec'_i, \quad N_i^o = T_i^{\otimes} \xrightarrow[Post_i]{Pre_i} P_i^{\otimes} \otimes A_i(X) \quad \text{und}$$

$$eval_i = (e_i^T, e_i^P, (g_i^X, eval_{\mathcal{V}(A_i(X))})),$$

– $\mathcal{V}_h(AN_2) = (ANS_1, g_2 \circ h^S, A_2, N_{2,1}^o, eval_{2,1})$ mit

$$N_{2,1}^o = T_{2,1}^{\otimes} \xrightarrow[Post_{2,1}]{Pre_{2,1}} P_{2,1}^{\otimes} \otimes A_2(X) \quad \text{und}$$

– $f: AN_1 \rightarrow \mathcal{V}_h(AN_2)$ ist ein ANS_1 -Modellmorphismus mit

$$f = (f^G, f^N) \quad \text{und} \quad f^N = (f^T, f^P, f^{G_X}).$$

und zeigen zunächst, daß Domäne und Codomäne von $\mathcal{UF}((h, f))$ in \mathcal{CPTN} liegen.

($\mathcal{UF}(AN_i) \in \mathcal{CP}\mathcal{T}\mathcal{N}$): Mit den Definitionen von \mathcal{UF} (\rightarrow 3.2.4) und der CP/T-Netz-Entfaltung (\rightarrow 2.3.3) von A-Netzen gilt:

$$\mathcal{UF}(AN_i) = \mathcal{UF}_{ANS_i}(AN_i) = T_i^{\otimes} \odot Ass_i \xrightarrow[Post_u]{Pre_u} P_i^{\otimes} \otimes A_i(X).$$

$T_i^{\otimes} \odot Ass_i$ und P_i^{\otimes} sind ihrer Definition nach freie abelsche Gruppen (\rightarrow 2.3.2) und $A_i(X)$ ist eine Algebra. $Pre_u, Post_u$ sind auf Basiselementen definiert und liefern Werte in $P_i^{\otimes} \otimes A_i(X)$. Damit sind sie monotone Morphismen von $T_i^{\otimes} \odot Ass_i$ nach $P_i^{\otimes} \otimes A_i(X)$ in den freien abelschen Gruppen. Aus diesen Eigenschaften folgt mit der Definition der CP/T-Netze (\rightarrow 1.1.3) unsere Behauptung.

Mit Definitionen der Transformation von A-Netzmorphismen in CP/T-Netzmorphismen und deren Erweiterung auf generalisierte Modellmorphisamen (\rightarrow 2.3.5, 3.2.4) gilt:

$$\mathcal{UF}\left((h, (f^G, f^N))\right) = \left(\hookrightarrow_{T_{2,1}^{\otimes}, T_2^{\otimes}} \circ f^T \odot f^{G^X}, \hookrightarrow_{P_{2,1}^{\otimes}, P_2^{\otimes}} \circ f^P, f^{G^X}\right).$$

und es muß gezeigt werden:

- (1) $\hookrightarrow_{T_{2,1}^{\otimes}, T_2^{\otimes}} \circ f^T \odot f^{G^X}$ ist ein monotoner Morphismus von $T_1^{\otimes} \odot Ass_1$ nach $T_2^{\otimes} \odot Ass_2$ in \mathcal{AG}_f ,
- (2) $\hookrightarrow_{P_{2,1}^{\otimes}, P_2^{\otimes}} \circ f^P \otimes f^{G^X}$ ist ein monotoner Morphismus von $P_1^{\otimes} \otimes A_1(X)$ nach $P_2^{\otimes} \otimes A_2(X)$ in \mathcal{AG}_f und
- (3) die Kompatibilität von $\mathcal{UF}((h, (f^G, f^N)))$ mit den Markenflußmorphismen.

(.1): Aus der Definition der Vergißfunktoren (\rightarrow 3.1.4) folgt $T_{2,1}^{\otimes} \subseteq T_2^{\otimes}$. Damit bildet die Komposition der Inklusion mit dem \mathcal{AG}_{fm} -Morphismus $f^T: T_1^{\otimes} \rightarrow T_{2,1}^{\otimes}$ einen monotonen Morphismus in den freien abelschen Gruppen mit Ursprung in T_1^{\otimes} und Ziel in T_2^{\otimes} . Da die Verknüpfung eines monotonen Morphismus in \mathcal{AG}_f mit einem generalisierten Algebromorphismus über die Operation \odot wieder zu einem monotonen Morphismus in \mathcal{AG}_f führt (\rightarrow 2.3.2) muß nur noch gezeigt werden, daß das Bild von $\hookrightarrow_{T_{2,1}^{\otimes}, T_2^{\otimes}} \circ f^T \odot f^{G^X}$ in der freien abelschen Gruppe $T_2^{\otimes} \odot Ass_2$ liegt.

Eine Voraussetzung dafür ist für alle Transitionen $t_1 \in T_1, t_2 \in T_2$ die Implikation

$$f^T(t_1)(t_2) > 0 \implies var_2(t_2) = \{x_{fso(s)} \mid x_s \in var_1(t_1)\},$$

deren Gültigkeit wir im folgenden nachweisen.

$$(x_{fso(s)} \in var_2(t_2) \implies x_s \in var_1(t_1)):$$

$$x_{fso(s)} \in var_2(t_2)$$

Annahme

$$\implies \exists p_2 \in P_2, a \in A_2(X):$$

2.1.1

$$x_{fso(s)} \in var(a) \wedge$$

$$(Pre_2(t_2)(p_2, a) > 0 \vee Post_2(t_2)(p_2, a) > 0)$$

$$\begin{aligned}
&\implies \exists p_2 \in P_2, a \in A_2(X): && f^T(t_1)(t_2) > 0 \\
&\quad x_{fso(s)} \in \text{var}(a) \wedge \\
&\quad (Pre_2(f^T(t_1)))(p_2, a) > 0 \vee \\
&\quad Post_2(f^T(t_1))(p_2, a) > 0 \\
&\implies \exists p_2 \in P_2, a \in A_2(X): && 1.1.4 \\
&\quad x_{fso(s)} \in \text{var}(a) \wedge \\
&\quad (f^P \otimes f^{Gx}(Pre_1(t_1)))(p_2, a) > 0 \vee \\
&\quad f^P \otimes f^{Gx}(Post_2(t_1))(p_2, a) > 0 \\
&\implies \exists p_1 \in P_1, a' \in A_1(X): && 1.1.1, 1.1.4 \\
&\quad f^P \otimes f^{Gx}(p_1, a')(p_2, a) > 0 \wedge \\
&\quad (Pre_1(t_1)(p_1, a') > 0 \vee Post_1(t_1)(p_1, a') > 0) \\
&\implies x_s \in \text{var}_1(t_1) && 2.1.1, \\
&&& \text{var}(a') \subseteq \text{var}_1(t_1), \\
&&& f^{Gx}(a') = a, \\
&&& f^{Gx} \text{ bewahrt Variablen} \\
&(x_s \in \text{var}_1(t_1) \implies x_{fso(s)} \in \text{var}_2(t_2)): \\
&x_s \in \text{var}_1(t_1) && \text{Annahme} \\
&\implies \exists p_1 \in P_1, a \in A_1(X): && 2.1.1 \\
&\quad x_s \in \text{var}(a) \wedge \\
&\quad (Pre_1(t_1)(p_1, a) > 0 \vee Post_1(t_1)(p_1, a) > 0) \\
&\implies \exists p_2 \in P_2: && f^T(t_1)(t_2) > 0, \\
&\quad f^P(p_1)(p_2) > 0 \wedge && 1.1.1, 1.1.4 \\
&\quad (Pre_2(t_2)(p_2, f^{Gx}(a)) > 0 \vee \\
&\quad Post_2(t_2)(p_2, f^{Gx}(a)) > 0) \\
&\implies x_{fso(s)} \in \text{var}_2(t_2) && f^{Gx} \text{ bewahrt Variablen}
\end{aligned}$$

Für generierte Transitionen t_1 muß darüber hinaus gezeigt werden:

$$f^{Gx}(\text{var}_1(Cond_1^o(t_1))) = \text{var}_2(Cond_2^o(t_2)).$$

Dazu nehmen wir o.B.d.A an: $t'_1 \in T_{I,1}$, $e_1^T(t'_1)(t_1) > 0$. Mit der Definition der A-Netzmorphismen und der Vergißfunktoren (\rightarrow 2.2.3(IV), 3.1.4) können wir zusätzlich o.B.d.A annehmen:

$$t'_2 \in T_{I,2}, e_2^T(t'_2)(t_2) > 0, h^T(t'_1)(t'_2) > 0.$$

Im Zusammenhang mit der Definition der A-Netze und der AN-Spezifikationsmorphismen (\rightarrow 2.2.1, 3.1.1(III)), folgt daraus:

$$\begin{aligned} \text{Cond}_1^o(t_1) &= g_1^*(\text{Cond}_1(t'_1)) \\ \text{und } \text{Cond}_2^o(t_2) &= g_2^*(\text{Cond}_2(t'_2)) = g_2^*(h^{S^*}(\text{Cond}_1(t'_1))). \end{aligned}$$

Darüber hinaus folgt aus der Definition der A-Netzmorphismen, der Vergißfunktoren und den Funktoreigenschaften von \mathcal{T}^A ($f^{G_X} = \mathcal{T}^A(f^G)$) (\rightarrow 2.2.3(IV), 3.1.4, 6.3.9) $f^{G_X} \circ g_1^x = g_2^x \circ h^{S_x}$. Mit 6.2.3 und 6.3.9 folgt daraus für alle Gleichungsmengen E über der Spec_1 -Signatur mit Variablen aus X :

$$f^{G_X} \circ g_1^*(E) = g_2^* \circ h^{S^*}(E).$$

Damit gilt:

$$f^{G_X}(\text{Cond}_1^o(t_1)) = f^{G_X} \circ g_1^*(\text{Cond}_1(t'_1)) = g_2^* \circ h^{S^*}(\text{Cond}_1(t'_1)) = \text{Cond}_2^o(t_2) \quad .$$

Als direkte Konsequenz daraus folgt unsere Behauptung.

Bis jetzt wissen wir das im Fall $f^T(t_1)(t_2) > 0$ für alle ass aus $Ass_1(t_1)$ gilt:

$$f^{As}(ass) \in [\text{var}_2(t_2) \cup \text{var}_2(\text{Cond}_2^o(t_2)) \rightarrow A_2(X)].$$

Es verbleibt zu zeigen:

$$f^{As}(ass) \in Ass_2(t_2)$$

(\rightarrow 2.3.1). Dazu unterscheiden wir die Fälle:

$$t_2 \in \mathcal{DC}_{eval_2}(co) \quad \text{und} \quad t_2 \in \mathcal{DC}_{eval_2}(im).$$

($t_2 \in \mathcal{DC}_{eval_2}(co)$):

$$\implies \forall t'_2 \in T_{I,2}^{\otimes}: e_2^T(t'_2)(t_2) = 0 \quad 1.2.3$$

$$\implies Ass_2(t_2) = [\text{var}_2(t_2) \rightarrow A_2(X)] \quad 2.3.1$$

$$\implies f^{As}(ass) \in Ass_2(t_2) \quad (\rightarrow \text{oben})$$

($t_2 \in \mathcal{DC}_{eval_2}(im)$): Aus der Annahme folgt: $\exists t'_2 \in T_{I,2}: e_2^T(t'_2)(t_2) > 0$. Wir können daher o.B.d.A. annehmen:

$$t'_2 \in T_{I,2} \wedge e_2^T(t'_2)(t_2) > 0.$$

Daraus folgt mit der Definition der von AN-Spezifikationsmorphismen generierten Vergißfunktoren (\rightarrow 3.1.7) $t_2 \in \mathcal{DC}_{eval_{2,1}}(im)$. Da f^T Komponente eines A-Netzmorphismus (\rightarrow 2.2.3) mit $f^T(t_1)(t_2) > 0$ ist, gilt damit auch $t_1 \in \mathcal{DC}_{eval_1}(im)$, d.h.

$$\text{(a):} \quad \exists t'_1 \in T_{I,1}: e_1^T(t'_1)(t_1) > 0.$$

Für jede Transition t'_1 mit dieser Eigenschaft gilt:

$$\hookrightarrow_{T_{2,1}^{\otimes}, T_2^{\otimes}} \circ f^T \circ e_1^T(t'_1)(t_2) > 0$$

$$\implies e_2^T \circ h^T(t'_1)(t_2) > 0 \quad 3.2.2$$

$$\implies \exists t''_2 \in T_{I,2}: e_2^T(t''_2)(t_2) > 0 \wedge \quad 2.2.1, 3.1.1$$

$$h^S(Cond_1(t'_1)) = Cond_2(t''_2) = Cond_2(t'_2)$$

Mit der Definition der AN-Spezifikationsmorphisamen (\rightarrow 3.1.1) folgt daraus:

$$(b): \quad \forall t'_1 \in T_{I,1}: e_1^T(t'_1)(t_1) > 0 \implies h^S(Cond_1(t'_1)) = Cond_2(t'_2).$$

Sei h^t die h^S -generierte Abbildung (\rightarrow 6.2.3) von $T_{OP_1}(X)$ -Termen auf $T_{OP_2}(X)$ -Terme, dann gilt mit der Definition der von Datentypspezifikationsmorphisamen generierten Vergißfunktoren (\rightarrow 6.1.11) und der Evaluierungsfunktionen (\rightarrow 6.1.9):

$$(c): \quad \forall u \in T_{OP_1}(X): \mathcal{P}^A(h^N)(u) = (\mathcal{T}^S(h^S), eval_{\mathcal{V}(T_{OP_2}(X))})(u) = h^t(u).$$

Aus der komponentweise definierten Komposition der CP/T-Netzmorphisamen und der Definition der Vergißfunktoren folgt das Kommutieren des oberen Quadrats im nachfolgenden Diagramm. Die Definition des Operators \odot (\rightarrow 2.3.2) zusammen mit der Definition der Variablensubstitution (\rightarrow 6.3.2) implizieren das Kommutieren des unteren Quadrats im selben Diagramm, sodaß es insgesamt kommutiert.

(d):

$$\begin{array}{ccc} T_{OP_1}(X) & \xrightarrow{\mathcal{P}^A(h^N)} & T_{OP_2}(X) \\ (g_1^X, eval) \downarrow & \circlearrowleft & \downarrow (g_2^X, eval) \\ A_1(X) & \xrightarrow{f^{G_X}} & A_2(X) \\ \widetilde{ass} \downarrow & \circlearrowleft & \downarrow f^{As}(\widetilde{ass}) \\ A_1(X) & \xrightarrow{f^{G_X}} & A_2(X) \end{array}$$

Damit können wir für alle Transitionen t'_1 aus $T_{I,1}$ mit $e_1^T(t'_1)(t_1) > 0$ zeigen:

$$\forall (Y, l, r) \in Cond_1(t'_1):$$

$$\implies eval_1^{ass} \circ g_1^t(l) = eval_1^{ass} \circ g_1^t(r) \quad 2.3.1$$

$$\implies (g_1^X, eval_1^{ass})(l) = (g_1^X, eval_1^{ass})(r) \quad 6.1.1, 6.1.9, 6.3.4$$

$$\implies \widetilde{ass} \circ (g_1^X, eval_1)(l) = \widetilde{ass} \circ (g_1^X, eval_1)(r) \quad 6.1.9, 6.3.2$$

$$\implies f^{G_X} \circ \widetilde{ass} \circ (g_1^X, eval)(l)$$

$$= f^{G_X} \circ \widetilde{ass} \circ (g_1^X, eval)(r)$$

$$\begin{aligned}
&\implies \widetilde{f^{As}(ass)} \circ (g_2^X, eval_2) \circ \mathcal{P}^A(h^N)(l) && (d) \\
&= f^{As}(ass) \circ (g_2^X, eval_2) \circ \mathcal{P}^A(h^N)(r) \\
&\implies \widetilde{f^{As}(ass)} \circ (g_2^X, eval_2) \circ h^t(l) && (c) \\
&= f^{As}(ass) \circ (g_2^X, eval_2) \circ h^t(r) \\
&\implies (g_2^X, eval_2^{f^{As}(ass)}) \circ h^t(l) && 6.1.9, 6.3.2 \\
&= (g_2^X, eval_2^{f^{As}(ass)}) \circ h^t(r) \\
&\implies eval_2^{f^{As}(ass)} \circ g_2^t(h^t(l)) = eval_2^{f^{As}(ass)} \circ g_2^t(h^t(r)) && 6.1.1, 6.1.9, 6.3.4 \\
&\implies \\
&\forall (Y', l', r') \in Cond_2(t'_2): \\
&\exists (Y, l, r) \in Cond_1(t'_1): \quad l' = h^t(l) \wedge r' = h^t(l) && 6.2.3, (a), (b) \\
&\implies eval_2^{f^{As}(ass)} \circ g_2^t(l') = eval_2^{f^{As}(ass)} \circ g_2^t(r') \\
&\implies f^{As}(ass) \in Cond_2(t_2) && 2.3.1
\end{aligned}$$

Damit ist das Bild eines jeden Paares (t_1, ass) unter $f^T \odot f^{G_X}$ ein Element aus $T_2^\otimes \odot Ass_2$.

(.2): Aus der Definition der Vergißfunktoren (\rightarrow 3.1.4) folgt $P_{2,1}^\otimes \subseteq P_2^\otimes$. Damit bildet die Inklusion $\hookrightarrow_{P_{2,1}^\otimes, P_2^\otimes}$ und die Komposition $\hookrightarrow_{P_{2,1}^\otimes, P_2^\otimes} \circ f^P - f^P$ ist definitionsgemäß ein monotoner Morphismus in \mathcal{AG}_f — einen monotonen Morphismus in den freien abelschen Gruppen mit Ursprung in P_1^\otimes und Ziel in P_2^\otimes . Da die Verknüpfung eines monotonen Morphismus in \mathcal{AG}_f mit einem generalisierten Algebromorphismus über die Operation \oplus wieder zu einem monotonen Morphismus in \mathcal{AG}_f führt (\rightarrow 1.1.1), der die Verknüpfung der Ursprünge in die Verknüpfung der Ziele beider Morphismen abbildet, gilt unsere Behauptung.

(.3): Für alle Paare (t, ass) aus $T_1^\otimes \odot Ass_1$ gilt:

$$\begin{aligned}
&(\hookrightarrow \circ f^P \otimes f^{G_X}) \circ Pre_{1_u}(t, ass) \\
&= (f^P \otimes f^{G_X}) \circ (id \otimes \widetilde{ass}) \circ Pre_1(t) && 2.3.3 \\
&= (f^P \otimes (f^{G_X} \circ \widetilde{ass})) \circ Pre_1(t) && _ \otimes _ \text{ ist Funktor} \\
&= (f^P \otimes \widetilde{f^{As}(ass)}) \circ f^{G_X} \circ Pre_1(t) && (d) \\
&= (id \otimes \widetilde{f^{As}(ass)}) \circ (f^P \otimes f^{G_X}) \circ Pre_1(t) && _ \otimes _ \text{ ist Funktor} \\
&= (id \otimes \widetilde{f^{As}(ass)}) \circ Pre_{2,1}(f^T(t)) && f^N \in \mathcal{CPTN}
\end{aligned}$$

$$= \bigoplus_{t' \in T_{2,1}} f^T(t)(t') * (id \otimes \widetilde{f^{As}(ass)}) \circ Pre_2(t') \quad 3.1.4$$

$$= \bigoplus_{t' \in T_2} \hookrightarrow_{T_{2,1}, T_2^{\otimes}} \circ f^T(t)(t') * Pre_{2u}(t', f^{As}(ass)) \quad 2.3.3$$

$$= Pre_{2u}(\hookrightarrow_{T_{2,1}, T_2^{\otimes}} \circ f^T \odot f^{Gx}(t, ass)) \quad 2.3.2$$

(\mathcal{UF} bewahrt die Identitäten): Identitätsmorphisme in $\mathcal{GAHL-NETS}$ haben die Form $(id_{ANS}, (id_A, id_{No}))$ und die von Identitäten in $\mathcal{AHLN-SPEC}$ generierten Vergißfunktoren sind Identitätsfunktoren (\rightarrow 3.1.8). Damit und mit Definition 2.3.5 gilt:

$$\mathcal{UF}((id, (id, id))) = (id \odot id, id, id)$$

und für jedes Paar (t, ass)

$$id \odot id(t, ass) = \bigoplus_{t' \in T} id(t)(t') * (t', id^{As}(ass)) = (t, ass).$$

Daraus folgt:

$$\mathcal{UF}((id, (id, id))) = (id, id, id)$$

und das ist ein Identitätsmorphismus in \mathcal{CPTN} .

(\mathcal{UF} bewahrt die Komposition von Morphismen): Seien

$$(h_i, (f_i^G, (f_i^T, f_i^P, f_i^{Gx}))) : AN_i \rightarrow AN_{i+1}$$

für $i = 1, 2$ komponierbare Morphisme in $\mathcal{GAHL-NETS}$, dann muß gezeigt werden:

$$\mathcal{UF}((h_2, (f_2^G, f_2^N)) \circ (h_1, (f_1^G, f_1^N))) = \mathcal{UF}((h_2, (f_2^G, f_2^N))) \circ \mathcal{UF}((h_1, (f_1^G, f_1^N))).$$

Mit der Definition der Vergißfunktoren (\rightarrow 3.1.5) und der Annahme (o.B.d.A.)

$$f_i^N = (f_i^T, f_i^P, f_i^{Gx})$$

ergibt sich

$$\mathcal{V}_{h_1}((f_2^G, f_2^N)) = \left(f_2^G, \left(f_2^T \Big|_{T_{2,1}^{\otimes}}^{T_{3,1}^{\otimes}}, f_2^P \Big|_{P_{2,1}^{\otimes}}^{P_{3,1}^{\otimes}}, f_2^{Gx} \right) \right).$$

Daraus folgt:

$$(e): \quad \hookrightarrow_{T_{3,1}, T_3^{\otimes}} \circ f_2^T \Big|_{T_{2,1}^{\otimes}}^{T_{3,1}^{\otimes}} \circ f_1^T = \hookrightarrow_{T_{3,2}, T_3^{\otimes}} \circ f_2^T \circ \hookrightarrow_{T_{2,1}, T_2^{\otimes}} \circ f_1^T \quad \text{und}$$

$$\hookrightarrow_{P_{3,1}, P_3^{\otimes}} \circ f_2^P \Big|_{P_{2,1}^{\otimes}}^{P_{3,1}^{\otimes}} \circ f_1^P = \hookrightarrow_{P_{3,2}, P_3^{\otimes}} \circ f_2^P \circ \hookrightarrow_{P_{2,1}, P_2^{\otimes}} \circ f_1^P.$$

Damit zeigen wir:

$$\begin{aligned}
& \mathcal{UF}\left((h_2, (f_2^G, f_2^P \otimes f_2^{GX})) \circ (h_1, (f_1^G, f_1^P \otimes f_1^{GX}))\right) \\
&= \mathcal{UF}\left((h_2 \circ h_1, \mathcal{V}_{h_1}(f_2^G, f_2^N) \circ (f_1^G, f_1^N))\right) \quad 5.3.2 \\
&= \mathcal{UF}\left(h_2 \circ h_1, \quad 3.1.5 \right. \\
&\quad \left. \left(f_2^G, \left(f_2^T \Big|_{T_{2,1}^\otimes}^{T_{3,1}^\otimes}, f_2^P \Big|_{P_{2,1}^\otimes}^{P_{3,1}^\otimes}, f_2^{GX}\right)\right) \circ (f_1^G, f_1^N)\right) \\
&= \mathcal{UF}\left(h_2 \circ h_1, \right. \\
&\quad \left. \left(f_2^G \circ f_1^G, \left(f_2^T \Big|_{T_{2,1}^\otimes}^{T_{3,1}^\otimes} \circ f_1^T, f_2^P \Big|_{P_{2,1}^\otimes}^{P_{3,1}^\otimes} \circ f_1^P, \right. \right. \\
&\quad \quad \left. \left. f_2^{GX} \circ f_1^{GX}\right)\right)\right) \\
&= \left(\hookrightarrow_{T_{3,2}, T_3^\otimes} \circ f_2^T \circ \hookrightarrow_{T_{2,1}, T_2^\otimes} \circ f_1^T \odot f_2^{GX} \circ f_1^{GX}, \quad 3.2.4, (e) \right. \\
&\quad \left. \left(\hookrightarrow_{P_{3,2}, P_3^\otimes} \circ f_2^P\right) \circ \left(\hookrightarrow_{P_{2,1}, P_2^\otimes} \circ f_1^P\right), f_2^{GX} \circ f_1^{GX}\right)
\end{aligned}$$

Wegen der komponentweisen Komposition der CP/T-Netzmorphismen verbleibt zu zeigen:

$$\begin{aligned}
& \hookrightarrow_{T_{3,2}, T_3^\otimes} \circ f_2^T \circ \hookrightarrow_{T_{2,1}, T_2^\otimes} \circ f_1^T \odot f_2^{GX} \circ f_1^{GX} \\
&= \\
& \left(\hookrightarrow_{T_{3,2}, T_3^\otimes} \circ f_2^T \odot f_2^{GX}\right) \circ \left(\hookrightarrow_{T_{2,1}, T_2^\otimes} \circ f_1^T \odot f_1^{GX}\right) .
\end{aligned}$$

Dazu notieren wir im folgenden abkürzend:

$$ass_2 \stackrel{\text{def}}{=} f_1^{As}(ass), \quad ass_3 \stackrel{\text{def}}{=} (f_2 \circ f_1)^{As}(ass).$$

Mit der Definition von f^{As} (\rightarrow 2.3.2) folgt daraus:

$$(f): \quad ass_3 = f_2^{As}(ass_2).$$

Für jedes Paar (t, ass) aus $T_1 \odot Ass_1$ gilt:

$$\begin{aligned}
& \hookrightarrow_{T_{3,2}, T_3^\otimes} \circ f_2^T \circ \hookrightarrow_{T_{2,1}, T_2^\otimes} \circ f_1^T \odot f_2^{GX} \circ f_1^{GX} (t, ass) \\
&= \bigoplus_{t_3 \in T_3} f_2^T(f_1^T(t))(t_3) * (t_3, (f_2 \circ f_1)^{As}(ass)) \quad 2.3.5 \\
&= \bigoplus_{t_3 \in T_3} f_2^T\left(\bigoplus_{t_2 \in T_2} f_1^T(t)(t_2) * t_2\right)(t_3) * (t_3, ass_3) \\
&= \bigoplus_{t_3 \in T_3} \bigoplus_{t_2 \in T_2} f_1^T(t)(t_2) * f_2^T(t_2)(t_3) * (t_3, ass_3) \quad f_2^T \text{ bewahrt } \oplus
\end{aligned}$$

$$\begin{aligned}
&= \bigoplus_{t_2 \in T_2} f_1^T(t)(t_2) * \bigoplus_{t_3 \in T_3} f_2^T(t_2)(t_3) * (t_3, f_2^{As}(ass_2)) && \text{Kommutativität v. } \bigoplus, \\
& && \text{(f)} \\
&= \bigoplus_{t_2 \in T_2} f_1^T(t)(t_2) * f_2^T \odot f_2^{Gx}(t_2, ass_2) && 2.3.5 \\
&= \hookrightarrow_{T_{3,2}^\otimes, T_3^\otimes} \circ f_2^T \odot f_2^{Gx} \left(\bigoplus_{t_2 \in T_2} f_1^T(t)(t_2) * (t_2, f_1^{As}(ass)) \right) && f_2^T \odot f_2^{Gx} \text{ bewahrt } \bigoplus \\
&= (\hookrightarrow_{T_{3,2}^\otimes, T_3^\otimes} \circ f_2^T \odot f_2^{Gx}) \circ (\hookrightarrow_{T_{2,1}^\otimes, T_2^\otimes} \circ f_1^T \odot f_1^{Gx})(t, ass) && 2.3.5
\end{aligned}$$

Zusammen mit der bisherigen Beweisschritten folgt daraus:

$$\begin{aligned}
&\mathcal{UF}\left((h_2, (f_2^G, f_2^P \otimes f_2^{Gx})) \circ (h_1, (f_1^G, f_1^P \otimes f_1^{Gx}))\right) \\
&= \left((\hookrightarrow_{T_{3,2}^\otimes, T_3^\otimes} \circ f_2^T \odot f_2^{Gx}) \circ (\hookrightarrow_{T_{2,1}^\otimes, T_2^\otimes} \circ f_1^T \odot f_1^{Gx}), \quad \text{s.o.} \right. \\
&\quad \left. (\hookrightarrow_{P_{3,2}^\otimes, P_3^\otimes} \circ f_2^P) \circ (\hookrightarrow_{P_{2,1}^\otimes, P_2^\otimes} \circ f_1^P), f_2^{Gx} \circ f_1^{Gx} \right) \\
&= \mathcal{UF}\left((h_2, (f_2^G, f_2^P \otimes f_2^{Gx}))\right) \circ && \text{Komponentw. Komposi-} \\
&\quad \mathcal{UF}\left((h_1, (f_1^G, f_1^P \otimes f_1^{Gx}))\right) && \text{tion in } \mathcal{CPTN}
\end{aligned}$$

□

(II):

Aus 1.3.11 und 3.2.6(I) folgt, daß es sich bei \mathcal{PC} und \mathcal{UF} jeweils um Funktoren handelt. Die Komposition von Funktoren hat in jedem Fall einen Funktor als Ergebnis, sodaß sich unsere Behauptung direkt aus der Definition von \mathcal{OS} ergibt. □

□

Beweis (3.3.1 Komponentweise Colimites in $\mathcal{AHLN-SPEC}$):

Es muß gezeigt werden:

- $ANS_c = (Spec_c, Cond_c, N_c)$ ist eine AN-Spezifikation,
- für alle Objekte s aus \mathcal{S} ist (c_s^S, c_s^N) ein AN-Spezifikationsmorphismus von ANS_s nach ANS_c ,
- die Verträglichkeit der (c_s^S, c_s^N) mit \mathcal{D} und
- ihre couniverselle Eigenschaft.

Dazu nehmen wir im folgenden o.B.d.A. an:

- $\mathcal{D}(s) = ANS_s = (Spec_s, Cond_s, N_s)$ mit $N_s = T_s^\otimes \xrightarrow[Post_s]{Pre_s} P_s^\otimes \otimes T_{OP_s(X)}$,
- $\mathcal{D}(m) = (h_m^S, h_m^N)$ mit $h_m^N = (h_m^T, h_m^P, (\mathcal{T}^S(h_m^S), h_m^t))$,

- $c_s^N = (c_s^T, c_s^P, (\mathcal{T}^S(c_s^S), c_s^t))$ und
- $N_c = T_c^{\otimes} \begin{array}{c} \xrightarrow{Pre_c} \\ \xrightarrow{Post_c} \end{array} P_c^{\otimes} \otimes \mathcal{P}^A(cod(c_s^N))$.

(ANS_c ist eine AN-Spezifikation):

Im einzelnen muß gezeigt werden:

- (1) $Spec_c$ eine algebraische Spezifikation,
- (2) $Cond_c$ ist Gleichungsmenge mit Variablen in X zur Signatur von $Spec_c$,
- (3) N_c ist ein CP/T-Netz mit $\mathcal{P}^A(N_c) = T_{OP_c(X)}$ und
- (4) für alle Transitionen t aus N_c gilt: $\forall x_{s_1}, y_{s_2} \in var_c(t): s_1 \neq s_2 \implies x \neq y$.

(.1): $Spec_c$ ist Objekt des Colimes zu \mathcal{D}^S und damit eine algebraische Spezifikation.

(.2): $Cond_c$ ist für alle Transitionen aus T_c definiert und hat jeweils eine Gleichungsmenge als Ergebnis. Diese Gleichungsmenge entspricht der Vereinigung von Bildern der Gleichungsmengen zu Transitionen aus N_s unter $c_s^{S^*}$ für Objekte s aus \mathcal{S} . Da $c_s^{S^*}$ lediglich die Sortenzuordnung der Variablen an die Signatur der Codomäne von c_s^S anpasst und die Variablen der Gleichungsmengen $Cond_s(t)$ mit t aus N_s nur Variablen aus X enthalten, sind auch die Variablen der Gleichungen aus $Cond_c$ in X .

(.3): Nach Voraussetzung gilt

$$N_c = cod(c_s^N) \quad \text{und} \quad \mathcal{P}^A(c_s^N) = (\mathcal{T}^S(c_s^S), c_s^t): T_{OP_s(X)} \rightarrow T_{OP_c(X)}$$

und damit

$$\mathcal{P}^A(N_c) = \mathcal{P}^A(cod(c_s^N)) = cod((\mathcal{T}^S(c_s^S), c_s^t)) = T_{OP_c(X)}.$$

(.4):

$$\forall t \in T_c, x_{s_1}, y_{s_2} \in var_c(t):$$

$$\exists t_s \in T_s: c_s^T(t_s)(t) > 0$$

1.2.2, 5.2.23,
Colimes \mathcal{D}^N , 6.3.11

$$\begin{aligned}
&\implies \forall (p, a) \in P_c \times T_{OP_c(X)}: && c_s^N \in \mathcal{CPTN}, \\
&\quad Pre_c(t)(p, a) > 0 && \text{Kompat. mit } Pre_c, Post_c \\
&\quad \implies c_s^P \otimes (\mathcal{T}^S(c_s^S), c_s^t)(Pre_s(t_s))(p, a) > 0 \\
&\quad \wedge \\
&\quad Post_c(t)(p, a) > 0 \\
&\quad \implies c_s^P \otimes (\mathcal{T}^S(c_s^S), c_s^t)(Post_s(t_s))(p, a) > 0 \\
&\implies \exists x_{s'_1}, y_{s'_2} \in var_s(t_s): && c_s^t \text{ bewahrt Var.} \\
&\quad (\mathcal{T}^S(c_s^S), c_s^t)(x_{s'_1}) = x_{s_1} \wedge (\mathcal{T}^S(c_s^S), c_s^t)(y_{s'_1}) = y_{s_1} \\
&\implies s_1 \neq s_2 \implies s'_1 \neq s'_2 \implies x \neq y && ANS_s \in \mathcal{AHLN-SPEC}, \\
& && 2.1.3
\end{aligned}$$

□

$((c_s^S, c_s^N): ANS_s \rightarrow ANS_c \in \mathcal{AHLN-SPEC})$:

Im einzelnen muß gezeigt werden:

- (1) c_s^S ist ein Spezifikationsmorphismus von $Spec_s$ nach $Spec_c$,
- (2) c_s^N ist ein CP/T-Netzmorphismus von N_s nach N_c mit

$$\mathcal{P}^A(c_s^N) = (\mathcal{T}^S(c_s^S, c_s^t)): T_{OP_s(X)} \rightarrow T_{OP_c(X)}$$

und

- (3) $\forall t_s \in T_s, t_c \in T_c: c_s^T(t_s)(t_c) > 0 \implies c_s^{S*}(Cond_s(t_s)) = Cond_c(t_c)$.

(.1): c_s^S ist ein Morphismus im Colimes zu \mathcal{D}^S und damit ein Spezifikationsmorphismus von $Spec_s$ nach $Spec_c$.

(.2): c_s^N ist ein Morphismus im Colimes zu \mathcal{D}^N und damit ein CP/T-Netzmorphismus von N_s nach N_c . $\mathcal{P}^A(c_s^N) = (\mathcal{T}^S(c_s^S, c_s^t)): T_{OP_s(X)} \rightarrow T_{OP_c(X)}$ gilt nach Voraussetzung.

(.3): Mit der Definition von $Cond_c$ gilt die Behauptung genau dann, wenn

$$\begin{aligned}
&\forall s, s' \in \mathcal{S}, t_s \in T_s, t'_{s'} \in T_{s'}, t_c \in T_c: \\
&\quad c_s^T(t_s)(t_c) > 0 \wedge c_{s'}^T(t'_{s'})(t_c) \implies c_s^{S*}(Cond_s(t_s)) = c_{s'}^{S*}(Cond_{s'}(t'_{s'}))
\end{aligned}$$

gilt.

Wir zeigen zunächst:

- (a):** $\forall s, s' \in Obj(\mathcal{S}), t_s \in T_s, t'_{s'} \in T_{s'}:$
 $t_s \Rightarrow t'_{s'} \implies c_s^{S*}(Cond_s(t_s)) = c_{s'}^{S*}(Cond_{s'}(t'_{s'}))$

$\forall s, s' \in \text{Obj}(\mathcal{S}), t_s \in T_s, t'_{s'} \in T_{s'}:$

$t_s \rightarrow t'_{s'}$

$\implies \exists m: s \rightarrow s' \in \mathcal{S}: h_m^T(t_s)(t'_{s'}) > 0$ 5.2.23

$\implies \exists m: s \rightarrow s' \in \mathcal{S}: \quad \quad \quad 3.1.1, \text{ Colimes } \mathcal{D}^S$

$h_m^{S^*}(\text{Cond}_s(t_s)) = \text{Cond}_{s'}(t'_{s'}) \wedge$

$c_s^{S^*}(\text{Cond}_s(t_s)) = c_{s'}^{S^*} \circ h_m^{S^*}(\text{Cond}_s(t_s))$

$\implies c_s^{S^*}(\text{Cond}_s(t_s)) = c_{s'}^{S^*}(\text{Cond}_{s'}(t'_{s'}))$ 3.1.1

mit

$t_s \rightleftharpoons t'_{s'} \implies \exists s_1, \dots, s_n \in \text{Obj}(\mathcal{S}), t_1 \in T_{s_1}, \dots, t_n \in T_{s_n}:$

$(t_s \rightarrow t_1 \vee t_1 \rightarrow t_s) \wedge$

$(t_i \rightarrow t_{i+1} \vee t_{i+1} \rightarrow t_i), i = 1, \dots, n-1 \wedge$

$(t_n \rightarrow t'_{s'} \vee t'_{s'} \rightarrow t_n)$

folgt daraus:

$\exists s_1, \dots, s_n \in \text{Obj}(\mathcal{S}), t_1 \in T_{s_1}, \dots, t_n \in T_{s_n}:$

$c_s^{S^*}(\text{Cond}_s(t_s)) = c_{s_1}^{S^*}(\text{Cond}_{s_1}(t_{s_1}))$

\vdots

$= c_{s_n}^{S^*}(\text{Cond}_{s_n}(t_{s_n}))$

$= c_{s'}^{S^*}(\text{Cond}_{s'}(t'_{s'})).$

Damit können wir zeigen:

$\forall s, s' \in \mathcal{S}, t_s \in T_s, t'_{s'} \in T_{s'}, t_c \in T_c:$

$c_s^T(t_s)(t_c) > 0 \wedge c_{s'}^T(t'_{s'})(t_c) > 0$

$\implies t_s \rightleftharpoons t'_{s'} \quad \quad \quad 5.2.23, \text{ Colimes } \mathcal{D}^N$

6.3.11

$\implies c_s^{S^*}(\text{Cond}_s(t_s)) = c_{s'}^{S^*}(\text{Cond}_{s'}(t'_{s'}))$ (a)

□

(Vertäglichkeit):

Folgt direkt aus der komponentweisen Komposition der Morphismen in $\mathcal{AHLN-SPEC}$ und der Vertäglichkeit von c_s^S und c_s^N mit den Funktoren $\mathcal{D}^S, \mathcal{D}^N$, die mit der Spezifikations- resp. der Netzkomponente von \mathcal{D} übereinstimmen. □

(Couniverselle Eigenschaft):

Folgt direkt aus der komponentweisen Komposition der Morphismen in \mathcal{AHLN} - \mathcal{SPEC} und der couniversellen Eigenschaft der Colimites für \mathcal{D}^S , \mathcal{D}^N , die mit der Spezifikations- resp. der Netzkomponente von \mathcal{D} übereinstimmen. \square

□

Beweis (3.3.4 Kompositionsdiagramme):

Aufgrund der komponentweise definierten Komposition und den aus den Identitäten der Komponenten zusammengesetzten Identitäten in \mathcal{AHLN} - \mathcal{SPEC} und \mathcal{CPTN} existieren folgende Funktoren

- $\mathcal{D}^S: \mathcal{S}_{cp} \rightarrow \mathcal{SPEC}$,
- $\mathcal{D}^A: \mathcal{S}_{cp} \rightarrow \mathcal{GSIG-ALG}$,
- $\mathcal{D}^T: \mathcal{S}_{cp} \rightarrow \mathcal{AG}_{fm}$,
- $\mathcal{D}^P: \mathcal{S}_{cp} \rightarrow \mathcal{AG}_{fm}$ und
- $\mathcal{D}^N: \mathcal{S}_{cp} \rightarrow \mathcal{CPTN}$

mit

$$\begin{aligned} \mathcal{CP}(m) &= (\mathcal{D}^S(m), \mathcal{D}^N(m)) \quad \text{und} \\ \mathcal{D}^N(m) &= (\mathcal{D}^T(m), \mathcal{D}^P(m), \mathcal{D}^A(m)) \end{aligned}$$

für alle Morphismen m in \mathcal{S}_{cp} .

Wir zeigen im folgenden, daß \mathcal{D}^S , \mathcal{D}^N , \mathcal{D}^A , \mathcal{D}^T und \mathcal{D}^P Colimites haben.

(\mathcal{D}^S hat einen Colimes): Folgt mit 6.2.7 aus der Endlichkeit von \mathcal{S}_{cp} .

(\mathcal{D}^A hat einen Colimes): Aus der Eigenschaft des Funktors \mathcal{T}^S (\rightarrow 6.3.4) die Colimites endlicher Diagramme zu bewahren (\rightarrow 6.3.5), der Existenz eines Colimites für \mathcal{D}^S und der Endlichkeit von \mathcal{S}_{cp} , folgt die Existenz eines Colimites für $\mathcal{T}^S \circ \mathcal{D}^S$.

Aufgrund der Eigenschaften von AN-Spezifikationsmorphismen (\rightarrow 3.1.1) (\mathcal{D}^A ist Komponente von \mathcal{CP}) und der Initialität der Termalgebren $T_{OP_s(X)}$ für Objekte s in \mathcal{S}_{cp} , gilt für alle Morphismen $m: s \rightarrow s'$ in \mathcal{S}_{cp} :

$$\mathcal{D}^A(m) = (\mathcal{T}^S(\mathcal{D}^S(m)), eval): T_{OP_s(X)} \rightarrow T_{OP_{s'}(X)}.$$

Da außerdem die Kategorie der Signaturen algebraischer Spezifikationen (\mathcal{SIG}) zusammen mit den Algebrakategorien und der Vergißfunktorkonstruktion einen Spezifikationsrahmen mit generalisierter Modellkategorie $\mathcal{GSIG-ALG}$ bilden (\rightarrow 6.1.12), sind alle Voraussetzung für die Anwendung von Lemma 5.3.3 gegeben und der Colimes von \mathcal{D}^A läßt sich aus dem Colimes von $\mathcal{T}^S \circ \mathcal{D}^S$ konstruieren.

(\mathcal{D}^T hat einen Colimes): Aufgrund der Annahmen in 3.3.4 sind zwei Fälle zu unterscheiden:

- für alle Morphismen m in \mathcal{S}_{cp} ist $\mathcal{D}^T(m)$ von einer Funktion generiert oder
- es existiert höchstens ein Morphismus m in \mathcal{S}_{cp} der nicht von einer Funktion generiert ist und alle anderen Morphismen m' in \mathcal{S}_{cp} sind jeweils von einer injektiven Funktion generiert.

Im ersten Fall folgt die Existenz eines Colimes zu \mathcal{D}^T aus Korollar 5.2.11. Im zweiten Fall entspricht die zu \mathcal{S}_{cp} duale Kategorie \mathcal{S}_{cp}^{op} der Kategorie \mathcal{S} in Lemma 5.2.21 und der zu \mathcal{D}^T duale Funktor $\mathcal{D}^{T^{op}}$ erfüllt die an den Funktor \mathcal{D} in 5.2.21 gestellten Bedingungen. Damit existiert ein Limes zu $\mathcal{D}^{T^{op}}$ und als Konsequenz daraus ein Colimes zu \mathcal{D}^T (\rightarrow 5.2.9, 5.1.17).

(\mathcal{D}^P hat einen Colimes): Die Existenz eines Colimes zu \mathcal{D}^P folgt aus den selben Gründen wie die zuvor gezeigte Existenz eines Colimes zu \mathcal{D}^T .

(\mathcal{D}^N hat einen Colimes): Aus den Colimites von \mathcal{D}^T , \mathcal{D}^P und \mathcal{D}^A folgt mit Lemma 1.2.2 die Existenz eines Colimes für \mathcal{D}^N .

Sei $((c_s^S)_{s \in \text{Obj}(\mathcal{S}_{cp})}, \text{Spec}_c)$ der Colimes für \mathcal{D}^S , dann bildet

$$((\mathcal{T}^S(c_s^S))_{s \in \text{Obj}(\mathcal{S}_{cp})}, \mathcal{T}^S(\text{Spec}_c))$$

einen Colimes zu $\mathcal{T}^S \circ \mathcal{D}^S$ (\rightarrow 6.3.5) und

$$((\mathcal{T}^S(c_s^S), \text{eval})_{s \in \text{Obj}(\mathcal{S}_{cp})}, T_{OP_c(X)})$$

bildet einen Colimes zu \mathcal{D}^A (\rightarrow (\mathcal{D}^A hat einen Colimes), 5.3.3).

Da sich der Colimes $((c_s^N)_{s \in \text{Obj}(\mathcal{S}_{cp})}, N_c)$ zu \mathcal{D}^N komponentweise aus den Colimites zu \mathcal{D}^T , \mathcal{D}^P und \mathcal{D}^A konstruieren läßt (\rightarrow 1.2.2) gilt mit der Definition der Projektion \mathcal{P}^A (\rightarrow 1.2.1):

$$\mathcal{P}^A(c_s^N) = (\mathcal{T}^S(c_s^S), \text{eval}): T_{OP_s(X)} \rightarrow T_{OP_c(X)}.$$

Damit sind alle Voraussetzungen für Lemma 3.3.1 erfüllt und der Colimes zu \mathcal{CP} kann in der dort beschriebenen Weise konstruiert werden.

□

Beweis (3.3.6 Kompositionsmuster und Colimites):

Aufgrund der komponentweisen definierten Komposition und den aus den Identitäten der Komponenten zusammengesetzten Identitäten in $\mathcal{GAHL-NETS}$, $\mathcal{AHLN-SPEC}$ und \mathcal{CPTN} existieren folgende Funktoren

- $\mathcal{D}^S: \mathcal{S}_{cp} \rightarrow \mathcal{SPEC}$,

- $\mathcal{D}^N: \mathcal{S}_{cp} \rightarrow \mathcal{CPTN}$
- $\mathcal{D}^{AS}: \mathcal{S}_{cp} \rightarrow \mathcal{AHLN-SPEC}$,
- $\mathcal{D}^A: \mathcal{S}_{cp} \rightarrow \mathcal{GSP\mathcal{E}C-ALG}$,
- $\mathcal{D}^T: \mathcal{S}_{cp} \rightarrow \mathcal{AG}_{fm}$ und
- $\mathcal{D}^P: \mathcal{S}_{cp} \rightarrow \mathcal{AG}_{fm}$

mit

$$\begin{aligned} \mathcal{D}^{AS}(m) &= (\mathcal{D}^S(m), \mathcal{D}^N(m)), \\ \mathcal{D}(m) &= (\mathcal{D}^{AS}(m), (\mathcal{D}^A(m), f_m^N)) \quad \text{und} \\ \mathcal{P}^N(\mathcal{D}(m)) &= (\mathcal{D}^T(m), \mathcal{D}^P(m), \mathcal{T}^A(\mathcal{D}^A(m))) \quad {}^{10} \end{aligned}$$

für alle Morphismen m in \mathcal{S}_{cp} . Wir zeigen im folgenden, daß \mathcal{D}^S , \mathcal{D}^N , \mathcal{D}^{AS} , \mathcal{D}^A , \mathcal{D}^T , \mathcal{D}^P und $\mathcal{P}^N \circ \mathcal{D}$ Colimites haben.

(\mathcal{D}^S , \mathcal{D}^N und \mathcal{D}^{AS} haben Colimites): Die Existenz eines, aus den Colimites der Komponenten (\mathcal{D}^S , \mathcal{D}^N) zusammengesetzten, Colimites für das Kompositionsdiagramm \mathcal{D}^{AS} folgt aus Satz 3.3.4 und der im zugehörigen Beweis verwendeten Konstruktion.

(\mathcal{D}^A hat einen Colimes): Folgt mit der Endlichkeit von \mathcal{S}_{cp} aus Satz 6.2.23.

(\mathcal{D}^T hat einen Colimes): Aufgrund der Annahmen in 3.3.6 und der Definition der Projektion \mathcal{P}^N (\rightarrow 3.2.1), sind zwei Fälle zu unterscheiden:

- für alle Morphismen m in \mathcal{S}_{cp} ist $\mathcal{D}^T(m)$ von einer Funktion generiert oder
- es existiert höchstens ein Morphismus in \mathcal{S}_{cp} , der nicht von einer Funktion generiert ist und alle anderen Morphismen in \mathcal{S}_{cp} sind jeweils von einer injektiven Funktion generiert.

Im ersten Fall folgt die Existenz eines Colimes zu \mathcal{D}^T aus Korollar 5.2.11. Im zweiten Fall entspricht die zu \mathcal{S}_{cp} duale Kategorie \mathcal{S}_{cp}^{op} der Kategorie \mathcal{S} in Lemma 5.2.21 und der zu \mathcal{D}^T duale Funktor $\mathcal{D}^{T^{op}}$ erfüllt die an den Funktor \mathcal{D} in 5.2.21 gestellten Bedingungen. Damit existiert ein Limes zu $\mathcal{D}^{T^{op}}$ und als Konsequenz daraus ein Colimes zu \mathcal{D}^T (\rightarrow 5.2.9, 5.1.17)

(\mathcal{D}^P hat einen Colimes): Die Existenz eines Colimes zu \mathcal{D}^P folgt aus den selben Gründen wie die zuvor gezeigte Existenz eines Colimes zu \mathcal{D}^T .

($\mathcal{P}^N \circ \mathcal{D}$ hat einen Colimes): Aus dem Colimes für \mathcal{D}^A und der Eigenschaft des Funktors \mathcal{T}^A Colimites endlicher Diagramme zu bewahren (\rightarrow 6.3.10), folgt die Existenz eines Colimites für $\mathcal{T}^A \circ \mathcal{D}^A$.

¹⁰ $\mathcal{P}^A(f_m^N) = \mathcal{T}^A(\mathcal{D}^A(m))$ (\rightarrow 2.2.3)

Da auch für \mathcal{D}^T und \mathcal{D}^P Colimites existieren, folgt aus der Definition der Projektion \mathcal{P}^N (\rightarrow 3.2.1),

$$\mathcal{P}^A \circ \mathcal{P}^N \circ \mathcal{D} = \mathcal{P}^A \circ \mathcal{D} = \mathcal{T}^A \circ \mathcal{D}^A$$

(\rightarrow 2.2.3, 1.2.1) und Lemma 1.2.2, die Existenz eines Colimes für $\mathcal{P}^N \circ \mathcal{D}$.

Sei $((\dot{c}_s^N)_{s \in \mathcal{S}_{cp}}, N_c^o)$, der aus den Komponentocolimites zu \mathcal{D}^T , \mathcal{D}^P und $\mathcal{T}^A \circ \mathcal{D}^A$ konstruierte Colimes für $\mathcal{P}^N \circ \mathcal{D}$, dann folgt mit der Definition der Projektion \mathcal{P}^A

$$\mathcal{P}^A(\dot{c}_s^N) = \mathcal{T}^A(\mathcal{D}^A(s))$$

für alle Objekte s in \mathcal{S}_{cp} . Damit sind alle Voraussetzungen von Lemma 3.2.3 erfüllt und der Colimes zu \mathcal{D} , kann in der dort beschriebenen Weise konstruiert werden.

□

Beweis (3.3.7 Verträglichkeit von Komposition und Modellsemantik):

Sei \mathcal{D} ein Funktor von einem Kompositonsmuster \mathcal{S}_{cp} nach $\mathcal{GAHL-NETS}$, dessen Spezifikationskomponente \mathcal{CP} ein Kompositionsdiagramm mit dem Wert

$$((c_s)_{s \in \text{Obj}(\mathcal{S}_{cp})}, ANS_c)$$

bildet und

$$\mathcal{D}(m) = (\mathcal{CP}(m), id)$$

für alle Morphismen m in \mathcal{S}_{cp} .

Wir nehmen im folgenden o.B.d.A. für alle Objekte s in \mathcal{S} an:

- $s_0 \in \mathcal{S}_{cp}$ mit $\forall s \in \text{Obj}(\mathcal{S}_{cp}): \exists m \in \text{Mor}(\mathcal{S}_{cp})[s_0, s]$,
- $\mathcal{D}(s) = (\mathcal{CP}(s), g_s, A_s, N_s^o, eval_s)$ mit $N_s^o = T_s^{\otimes} \xrightarrow[Post_s]{Pre_s} P_s^{\otimes} \otimes A_s(X)$,
- $eval_s = (e_s^T, e_s^P, e_s^G)$ und
- $c_s = (c_s^S, c_s^N)$ mit $c_s^N = (c_s^T, c_s^P, c_s^G)$.

Aus Satz 3.3.4 und dessen Beweis, der sich auf Lemma 3.3.1 stützt folgt, daß sich der Wert (Colimes) von \mathcal{CP} (\rightarrow 3.3.3) komponentweise aus den Colimites von dessen Spezifikations- und Netzkomponente zusammensetzen läßt, die unter den gegebenen Voraussetzungen existieren (\rightarrow Beweis von 3.3.4). Aufgrund der Eindeutigkeit (bis auf Isomorphie) von Colimites und der komponentweisen Komposition der Morphismen in $\mathcal{AHLN-SPEC}$ folgt daraus, daß die Spezifikations- und Netzanteile im Colimes für \mathcal{CP} jeweils Colimites für den Spezifikations- bzw. Netzanteil von \mathcal{CP} bilden, d.h. für alle Funktoren

$$\mathcal{D}^S: \mathcal{S}_{cp} \rightarrow \mathcal{SPEC}, \quad \mathcal{D}^N: \mathcal{S}_{cp} \rightarrow \mathcal{CPTN} \quad \text{mit} \quad \mathcal{CP}(m) = (\mathcal{D}^S(m), \mathcal{D}^N(m))$$

gilt:

$$(a): \quad ((c_s^S)_{s \in \text{Obj}(\mathcal{S}_{cp}), \text{Spec}_c) \text{ ist Colimes für } \mathcal{D}^S \quad \text{und} \\ ((c_s^N)_{s \in \text{Obj}(\mathcal{S}_{cp}), N_c) \text{ ist Colimes für } \mathcal{D}^N.$$

Aus $\mathcal{D}(m) = (\mathcal{CP}(m), id)$ für alle Morphismen m in \mathcal{S}_{cp} , der Vergißfunktordefinition (\rightarrow 3.1.4) und der Struktur von \mathcal{S}_{cp} ($\forall s \in \text{Obj}(\mathcal{S}_{cp}): \exists m \in \text{Mor}(\mathcal{S}_{cp}) [s_0, s]$) folgt:

$$A_s = A_{s_0} \quad \text{und} \quad \mathcal{P}^N(\mathcal{D}(m)) = (\hookrightarrow_{T_s^\otimes, T_{s'}^\otimes}, \hookrightarrow_{P_s^\otimes, P_{s'}^\otimes}, \mathcal{T}^A(id_{A_s}))$$

für alle Objekte s und Morphismen $m: s \rightarrow s'$ in \mathcal{S}_{cp} . Wegen der Gleichheit aller A_s lassen wir deren Index weg und bezeichnen sie einheitlich mit A .

Projiziert man den Funktor $\mathcal{P}^N \circ \mathcal{D}$ auf seine Komponenten, ergeben sich die Funktoren:

- $\mathcal{D}^T: \mathcal{S}_{cp} \rightarrow \mathcal{AG}_{fm}$ mit $\mathcal{D}^T(m) = \hookrightarrow_{T_s^\otimes, T_{s'}^\otimes}$,
- $\mathcal{D}^P: \mathcal{S}_{cp} \rightarrow \mathcal{AG}_{fm}$ mit $\mathcal{D}^P(m) = \hookrightarrow_{P_s^\otimes, P_{s'}^\otimes}$,
- $\mathcal{D}^A: \mathcal{S}_{cp} \rightarrow \mathcal{GSP\mathcal{E}C-ALG}$ mit $\mathcal{D}^A(m) = id_A$

für alle Morphismen $m: s \rightarrow s'$ in \mathcal{S}_{cp} und es gilt:

$$\mathcal{P}^N(\mathcal{D}(m)) = (\mathcal{D}^T(m), \mathcal{D}^P(m), \mathcal{T}^A(\mathcal{D}^A(m))).$$

\mathcal{D}^T und \mathcal{D}^P sind für alle Morphismen m in \mathcal{S}_{cp} von den jeweiligen Inklusionen zwischen den Basismengen ihrer Domäne und Codomäne generiert. Für Diagramme in der Kategorie der Mengen \mathcal{SET} deren Morphismen Inklusionen sind, ergeben sich die Colimites aus der Vereinigung der Diagrammobjekte mit den jeweiligen Inklusionen. Da die freie Konstruktion von den Mengen in die freien abelschen Gruppen (\rightarrow 5.2.10), als Linksadjunktion Colimites bewahrt (\rightarrow [HS73] 27.5, 27.7), ergeben sich die Colimites von \mathcal{D}^T resp. von \mathcal{D}^P zu:

$$(b): \quad ((\hookrightarrow_{T_s^\otimes, T_c^\otimes})_{s \in \text{Obj}(\mathcal{S}_{cp}), T_c^\otimes) \quad \text{und} \quad ((\hookrightarrow_{P_s^\otimes, P_c^\otimes})_{s \in \text{Obj}(\mathcal{S}_{cp}), P_c^\otimes);$$

wobei T_c und P_c jeweils die Vereinigung der T_s resp. der P_s bezeichnen. Der Colimes des trivialen Diagramms \mathcal{D}^A ergibt sich zu:

$$((id_A)_{s \in \text{Obj}(\mathcal{S}_{cp}), A).$$

Da der Funktor \mathcal{T}^A (\rightarrow 6.3.9) Colimites endlicher Diagramme bewahrt (\rightarrow 6.3.10), bildet die Senke:

$$((id_{A(X)})_{s \in \text{Obj}(\mathcal{S}_{cp}), A(X)).$$

einen Colimes für $\mathcal{T}^A \circ \mathcal{D}^A$.

Aus diesen Colimites der $\mathcal{P}^N \circ \mathcal{D}$ -Komponenten ergibt sich nach Lemma 1.2.2 ein Colimes für $\mathcal{P}^N \circ \mathcal{D}$ zu:

$$\left(\left((\hookrightarrow_{T_s^\otimes, T_c^\otimes}, \hookrightarrow_{P_s^\otimes, P_c^\otimes}, id_{A(X)}) \right)_{s \in Obj(\mathcal{S}_{cp})}, N_c^o \right)$$

mit
$$N_c^o = T_c^\otimes \begin{array}{c} \xrightarrow{Pre_c} \\ \xrightarrow{Post_c} \end{array} P_c^\otimes \otimes A(X) \quad \text{und}$$

$$\mathcal{T}^A(id_A) = \mathcal{P}^A((\hookrightarrow_{T_s^\otimes, T_c^\otimes}, \hookrightarrow_{P_s^\otimes, P_c^\otimes}, id_{A(X)}))$$

für alle Objekte s in \mathcal{S}_{cp} .

(3.3.7 (I)):

Die Funktoren \mathcal{D}^S , \mathcal{D}^N , \mathcal{CP} , \mathcal{D}^A und $\mathcal{P}^N \circ \mathcal{D}$ erfüllen zusammen mit ihren Colimites alle Voraussetzungen für Lemma 3.2.3, sodaß die Senke

(c):
$$\left((c_s, (id_A, d_s^N)) \right)_{s \in Obj(\mathcal{S})}, AN_c$$

mit
$$AN_c = (ANS_c, g_c, A_c, N_c^o, eval_c) \quad \text{und}$$

$$\mathcal{P}^N(c_s, (id_A, d_s^N)) = (\hookrightarrow_{T_s^\otimes, T_c^\otimes}, \hookrightarrow_{P_s^\otimes, P_c^\otimes}, id_{A(X)}) \quad {}^{11}$$

einen Colimes für \mathcal{D} bildet. Die AN_c -Komponenten g_c und $eval_c$ ergeben sich eindeutig aus den Eigenschaften:

(d):
$$g_c \circ c_s^S = g_s \quad \text{und} \quad eval_c \circ c_s^N = \mathcal{P}^N(c_s, (id_A, d_s^N)) \circ eval_s$$

(→ 3.2.3) für alle Objekte s in \mathcal{S}_{cp} und wir nehmen o.B.d.A an: $eval_c = (e_c^T, e_c^P, e_c^G)$.

Sei für alle Objekte s aus \mathcal{S}_{cp}

$$\mathcal{V}_{c_s}(AN_c) = (ANS_s, g_c \circ c_s^S, A_c, N_{c,s}^o, eval_{c,s}),$$

dann muß gezeigt werden:

- $g_c \circ c_s^S = g_s$,
- $A_c = A_s$,
- $N_{c,s}^o = N_s^o$ und
- $eval_{c,s} = eval_s$.

¹¹ $\mathcal{P}^N(c_s, id_{AN_s}) \circ d_s^N = \mathcal{P}^N(c_s, id_{AN_s}) \circ \mathcal{P}^N(id_{ANS_s}, d_s^N) = \mathcal{P}^N(c_s, (id_A, d_s^N))$ (→ 3.2.1)

$(g_c \circ c_s^S = g_s)$: $(\rightarrow (d))$

$(A_c = A_s)$: Folgt direkt aus (c).

$(N_{c,s}^o = N_s^o)$: Mit der Definition der Vergißfunktoren (\rightarrow 3.1.4) und (c) folgt:

$$\begin{aligned} N_{c,s}^o &= T_{c,s}^{\otimes} \begin{array}{c} \xrightarrow{Pre_{c,s}} \\ \xrightarrow{Post_{c,s}} \end{array} P_{c,s}^{\otimes} \otimes A(X), \\ eval_{c,s} &= (e_c^T \circ c_s^T, e_c^T \circ c_s^T, (\mathcal{T}^S(g_c \circ c_s^S), eval)) \quad \text{und} \\ d_s^N &= (\hookrightarrow_{T_s^{\otimes}, T_{c,s}^{\otimes}}, \hookrightarrow_{P_s^{\otimes}, P_{c,s}^{\otimes}}, id_{A(X)}). \end{aligned}$$

Damit muß gezeigt werden:

- (1) $T_{c,s}^{\otimes} = T_s^{\otimes}$, $P_{c,s}^{\otimes} = P_s^{\otimes}$ und
- (2) $Pre_{c,s} = Pre_s$, $Post_{c,s} = Post_s$.

(.1): Aus (b) und der Definition der Vergißfunktoren (\rightarrow 3.1.4) folgt:

$$(e): \quad T_s^{\otimes} \subseteq T_{c,s}^{\otimes} \subseteq T_c^{\otimes} \quad \text{und} \quad P_s^{\otimes} \subseteq P_{c,s}^{\otimes} \subseteq P_c^{\otimes}$$

für alle Objekte s in \mathcal{S} ; verbleibt zu Zeigen: $T_{c,s}^{\otimes} \subseteq T_s^{\otimes}$ und $P_{c,s}^{\otimes} \subseteq P_s^{\otimes}$. Wir führen den Beweis separat für die, aus der Zerlegungen von $N_{c,s}^o$ über $eval_{c,s}$ entstehenden Teilnetze $\mathcal{DC}_{eval_{c,s}}(im)$ und $\mathcal{DC}_{eval_{c,s}}(co)$, die zusammen jede Stelle und jede Transition aus $N_{c,s}^o$ enthalten.

$t \in \mathcal{DC}_{eval_{c,s}}(im)$	Annahme
$\implies \exists t' \in N_s: e_c^T \circ c_s^T(t')(t) > 0$	1.2.3, 3.1.4
$\implies \exists t' \in N_s: \hookrightarrow_{T_s^{\otimes}, T_c^{\otimes}} \circ e_s^T(t')(t) > 0$	(d)
$\implies t \in T_s^{\otimes}$	
$t \in \mathcal{DC}_{eval_{c,s}}(co)$	Annahme
$\implies t \in \mathcal{DC}_{eval_c}(co)$	3.1.4
$\implies \forall s' \in \mathcal{S}: t \in \mathcal{DC}_{eval_{c,s'}}(co)$	3.1.4
$\implies \exists s' \in \mathcal{S}: t \in \mathcal{DC}_{eval_{s'}}(co)$	(b), 5.2.23(I), (c), 2.2.3
$\implies \exists s' \in \mathcal{S}: t \in \mathcal{DC}_{eval_{s'}, s_0}(co)$	$\exists m: s_0 \rightarrow s' \in \mathcal{S}$, 3.1.4
$\implies t \in \mathcal{DC}_{eval_{s_0}}(co)$	$\mathcal{V}_{\mathcal{CP}(m)}(AN_{s'}) = AN_{s_0}$
$\implies t \in \mathcal{DC}_{eval_{s,s_0}}(co) \subseteq T_s$	$\exists m: s_0 \rightarrow s \in \mathcal{S}$, $\mathcal{V}_{\mathcal{CP}(m)}(AN_s) = AN_{s_0}$ 2.2.3, 3.1.4

Der Beweis für die $P_{c,s} \subseteq P_s$ verläuft analog. Damit gilt für alle Objekte s in \mathcal{S} :

$$(f): \quad d_s^N = (id_{T_s^\otimes}, id_{P_s^\otimes}, id_{A(X)}) = id_{N_{c,s}^o}.$$

(.2): Da es sich bei den $d_s^N: N_s^o \rightarrow N_{c,s}^o$ um CP/T-Netzmorphismen handelt gilt mit (f) für alle Objekte s in \mathcal{S}_{cp} :

$$\begin{aligned} Pre_{c,s} &= Pre_{c,s} \circ id_{T_s^\otimes} = id_{P_s^\otimes} \circ Pre_s = Pre_s \quad \text{und} \\ Post_{c,s} &= Post_{c,s} \circ id_{T_s^\otimes} = id_{P_s^\otimes} \circ Post_s = Post_s. \end{aligned}$$

($eval_{c,s} = eval_s$): Die Abbildung $(id_A, d_s^N): AN_s \rightarrow \mathcal{V}_{c_s}(AN_c)$ ist für alle s aus \mathcal{S} ein A-Netzmorphismus. Daraus folgt mit deren Definition (\rightarrow 2.2.3) und (f)

$$eval_{c,s} = d_s^N \circ eval_s = id_{N_s^o} \circ eval_s = eval_s.$$

□

(3.3.7 (II)): Wir zeigen, daß für ein beliebiges ANS_c -Modell

$$AN_{\bar{c}} = (ANS_c, g_{\bar{c}}, A_{\bar{c}}, N_{\bar{c}}^o, eval_{\bar{c}}) \quad \text{mit} \quad \mathcal{V}_{c_s}(AN_{\bar{c}}) = \mathcal{D}(s)$$

für alle s in \mathcal{S} , $AN_{\bar{c}} = AN_c$ gilt, d.h.

- $g_{\bar{c}} = g_c$,
- $A_{\bar{c}} = A_c$,
- $N_{\bar{c}}^o = N_c^o$ und
- $eval_{\bar{c}} = eval_c$.

Mit der Vergißfunktordefinition (\rightarrow 3.1.4) folgt aus unseren Annahmen o.B.d.A. für alle s in \mathcal{S} :

$$(g): \quad (c_s, id_{\mathcal{V}_{c_s}(AN_{\bar{c}})}): \mathcal{D}(s) \rightarrow AN_{\bar{c}} \in \mathcal{GAHL-NETS} \quad \text{und}$$

$$(h): \quad \mathcal{P}^N((c_s, id_{\mathcal{V}_{c_s}(AN_{\bar{c}})})) = (\hookrightarrow_{T_s^\otimes, T_{\bar{c}}^\otimes}, \hookrightarrow_{P_s^\otimes, P_{\bar{c}}^\otimes}, id): N_s^o \rightarrow N_{\bar{c}}^o \in \mathcal{CPTN}.$$

$$(g_{\bar{c}} = g_c):$$

Aus $\mathcal{V}_{c_s}(AN_{\bar{c}}) = \mathcal{D}(s)$ für alle Objekte s aus \mathcal{S}_{cp} folgt mit der Definition der Vergißfunktoren (\rightarrow 3.1.4)

$$\forall s \in \mathcal{S}: \mathcal{T}^S(g_s) = \mathcal{T}^S(g_{\bar{c}} \circ c_s^S) \quad 3.1.4$$

$$\begin{aligned} \implies \forall s \in \mathcal{S}: g_s &= g_{\bar{c}} \circ c_s^{\mathcal{S}} && \mathcal{T}^{\mathcal{S}} \text{ faithful} \\ \implies g_{\bar{c}} &= g_c && \text{(d)} \end{aligned}$$

□

$(A_{\bar{c}} = A_c)$:

Folgt mit $A_c = A_s = A$ (\rightarrow (c)) und $\mathcal{V}_{c_s}(AN_{\bar{c}}) = \mathcal{D}(s)$ aus der Definition der Vergißfunktoren (\rightarrow 3.1.4). □

$(N_{\bar{c}}^o = N_c^o)$:

Aus den bisherigen Ergebnissen folgt: $A_{\bar{c}}(X) = A(X)$ und wir können o.B.d.A annehmen:

$$N_{\bar{c}}^o = T_{\bar{c}}^{\otimes} \begin{array}{c} \xrightarrow{\text{Pre}_{\bar{c}}} \\ \xrightarrow{\text{Post}_{\bar{c}}} \end{array} P_{\bar{c}}^{\otimes} \otimes A(X) \quad \text{und} \quad \text{eval}_{\bar{c}} = (e_{\bar{c}}^T, e_{\bar{c}}^P, e_{\bar{c}}^G).$$

Damit muß gezeigt werden:

- (1) $T_{\bar{c}} = T_c, \quad P_{\bar{c}} = P_c$ und
- (2) $\text{Pre}_{\bar{c}} = \text{Pre}_c, \quad \text{Post}_{\bar{c}} = \text{Post}_c.$

(.1): Wir zeigen zunächst: $T_c \subseteq T_{\bar{c}}$

$$\begin{aligned} \forall t \in T_c, \exists s \in \mathcal{S}: t \in T_s &&& 5.2.23(\text{I}), (\text{b}) \\ \implies \forall t \in T_c: t \in T_{\bar{c}} &&& \text{(h), 3.2.1} \end{aligned}$$

Um $T_{\bar{c}} \subseteq T_c$ zu zeigen, nutzen wir den Zusammenhang

$$\text{eval}_{\bar{c}} \circ c_s^N = \mathcal{P}^N((c_s, \text{id}_{\mathcal{V}_{c_s}(AN_{\bar{c}})})) \circ \text{eval}_s$$

(\rightarrow 3.2.2) für alle s in \mathcal{S} . Zusammen mit (g) ergibt sich daraus:

$$\text{(i):} \quad \forall s \in \mathcal{S}: e_{\bar{c}}^T \circ c_s^T = \hookrightarrow_{T_s^{\otimes}, T_{\bar{c}}^{\otimes}} \circ e_s^T.$$

Für den Beweis unterscheiden wir für alle t aus $T_{\bar{c}}$ die Fälle:

$$t \in \mathcal{DC}_{\text{eval}_{\bar{c}}}(im) \quad \text{oder} \quad t \in \mathcal{DC}_{\text{eval}_{\bar{c}}}(co).$$

$$\begin{aligned} t \in \mathcal{DC}_{\text{eval}_{\bar{c}}}(im) &&& \text{Annahme} \\ \implies \exists t_c \in N_c: e_{\bar{c}}^T(t_c)(t) > 0 &&& 1.2.3 \\ \implies \exists s \in \mathcal{S}, t_s \in N_s: e_{\bar{c}}^T \circ c_s^T(t_s)(t) > 0 &&& \text{(a), 6.3.11, 5.2.23(I)} \\ \implies \exists s \in \mathcal{S}, t_s \in N_s: \hookrightarrow_{T_s^{\otimes}, T_{\bar{c}}^{\otimes}} \circ e_s^T(t_s)(t) > 0 &&& \text{(i)} \\ \implies \exists s \in \mathcal{S}: t \in T_s &&& \\ \implies t \in T_c &&& \text{(b)} \end{aligned}$$

$$\begin{array}{ll}
t \in \mathcal{DC}_{eval_{\bar{c}}}(co) & \text{Annahme} \\
\implies \forall s \in \mathcal{S}: t \in \mathcal{DC}_{eval_s}(co) \subseteq T_s & \mathcal{V}_{c_s}(AN_{\bar{c}}) = \mathcal{D}(s), 3.1.4 \\
\implies \forall s \in \mathcal{S}: t \in T_c & (b)
\end{array}$$

Der Beweis für $P_{\bar{c}}^{\otimes} = P_c^{\otimes}$ verläuft analog.

(.2):

$$\forall t \in T_{\bar{c}}, \exists s \in \mathcal{S}: t \in T_s \quad 5.2.23, (b), T_{\bar{c}} = T_c$$

\implies

$$\forall t \in T_{\bar{c}}, \exists s \in \mathcal{S}:$$

$$\begin{array}{ll}
Pre_{\bar{c}}(t) = Pre_{\bar{c}}(\hookrightarrow_{T_s^{\otimes}, T_c^{\otimes}}(t)) & T_{\bar{c}} = T_c \\
= \hookrightarrow_{P_s^{\otimes}, P_c^{\otimes}} \otimes id \circ Pre_s(t) & (h), P_{\bar{c}} = P_c \\
= Pre_c(\hookrightarrow_{T_s^{\otimes}, T_c^{\otimes}}(t)) & (c), \mathcal{P}^N(c_s, id) \in \mathcal{CPTN} \\
= Pre_c(t) &
\end{array}$$

Der Beweis von $Post_{\bar{c}} = Post_c$ verläuft analog. \square

($eval_{\bar{c}} = eval_c$):

Mit (h), (c), den bisher gezeigten Ergebnissen und der Definition der Projektion \mathcal{P}^N (\rightarrow 3.2.1) folgt:

$$\mathcal{P}^N((c_s, id_{\mathcal{V}_{c_s}(AN_{\bar{c}})})) = \mathcal{P}^N((c_s, (id_A, d_s^N)))$$

für alle Objekte s in \mathcal{S}_{cp} . Zusammen mit Lemma 3.2.2 ergibt sich daraus:

$$eval_{\bar{c}} \circ c_s^N = \mathcal{P}^N((c_s, id_{\mathcal{V}_{c_s}(AN_{\bar{c}})})) \circ eval_s = \mathcal{P}^N((c_s, (id_A, d_s^N))) \circ eval_s$$

für alle Objekte s in \mathcal{S}_{cp} und damit $eval_{\bar{c}} = eval_c$, da $eval_c$ bzgl. dieser Eigenschaft eindeutig ist (\rightarrow (d)). \square

\square

Beweis (3.4.4 Freie Konstruktion):

Wir nehmen o.B.d.A an:

- $PAR = (Spec_p, Cond_p, T_p^{\otimes} \xrightleftharpoons[Post_p]{Pre_p} P_p^{\otimes} \otimes T_{OP_p(X)})$,
- $ANS = (Spec, Cond, T^{\otimes} \xrightleftharpoons[Post]{Pre} P^{\otimes} \otimes T_{OP(X)})$,
- $I_p = (PAR, id, T_{Spec_p}, N_p^o, eval_p)$ ist das initiale PAR -Modell,

- $I = (ANS, id, T_{Spec}, N^o, eval)$ ist das initiale ANS -Modell und
- $AN = (PAR, g, A, N_a^o, eval_a)$ mit

$$g: Spec_p \rightarrow Spec_a, \quad N_a^o = T_a^\otimes \xrightarrow[Post_a]{Pre_a} P_a^\otimes \otimes A(X) \quad \text{und}$$

$$eval_a = (e_a^T, e_a^P, \mathcal{T}^A(g, eval_{\mathcal{V}(A)})).$$

(I):

Sei $(\hookrightarrow, eval_{\mathcal{V}(I)}): I_P \rightarrow I$ der generalisierte Modellmorphismus vom initialen PAR -Modell I_P in das initiale ANS -Modell I , dann gilt aufgrund von Lemma 3.2.2, der Definition der Projektion \mathcal{P}^N (\rightarrow 3.2.1) und der Definition initialer AN-Spezifikations-Modelle (\rightarrow 2.2.6)

$$eval_{\mathcal{V}(I)} = (\hookrightarrow, \hookrightarrow, (\hookrightarrow, eval)).$$

Damit erfüllt das Diagramm in 3.4.4 alle Voraussetzungen von Satz 3.3.6 und der Pushout

$$\begin{array}{ccc} I_P & \xrightarrow{(\hookrightarrow, eval)} & I \\ (id, eval) \downarrow & (PO) & \downarrow (id, eval) \\ AN & \xrightarrow{(\hookrightarrow, u_{AN})} & \mathcal{F}(AN) \end{array}$$

existiert. Aus Lemma 5.3.4 folgt in diesem Fall, daß $\mathcal{F}(AN)$ frei über AN konstruiert ist mit universeller Abbildung u_{AN} . □

(II):

Aus den Anfangsannahmen folgen mit der Definition der freien Konstruktion als Pushout (\rightarrow 3.4.3), der Inklusionsbeziehung zwischen PAR und ANS (\rightarrow 3.4.1) und der Definition der initialen Modelle (\rightarrow 2.2.6) die folgenden Pushouts:

$$\begin{array}{ccc} PAR & \xrightarrow{\hookrightarrow} & ANS \\ \parallel & (PO) & \parallel \\ PAR & \xrightarrow{\hookrightarrow} & ANS \end{array} \qquad \begin{array}{ccc} Spec_p & \xrightarrow{\hookrightarrow} & Spec \\ g \downarrow & (PO) & \downarrow g_c \\ Spec_a & \xrightarrow{\hookrightarrow} & Spec_c \end{array}$$

$$\begin{array}{ccc} T_p^\otimes & \xrightarrow{\hookrightarrow} & T^\otimes \\ e_a^T \downarrow & (PO) & \downarrow e_c^T \\ T_a^\otimes & \xrightarrow{\hookrightarrow} & T_c^\otimes \end{array} \qquad \begin{array}{ccc} P_p^\otimes & \xrightarrow{\hookrightarrow} & P^\otimes \\ e_a^P \downarrow & (PO) & \downarrow e_c^P \\ P_a^\otimes & \xrightarrow{\hookrightarrow} & P_c^\otimes \end{array}$$

Die Voraussetzung

$$\forall s \in SO_p, op \in OP_s, eq \in EQ_s: op \in OP_{p,s} \wedge eq \in EQ_{p,s}$$

der Inklusion von $Spec_p$ in $Spec$, übertägt sich auf die Inklusion von $Spec_a$ in $Spec_c$ und auf deren Erweiterung, die Inklusion von $Spec_a(X)$ in $Spec_c(X)$. Das ist im ersten Fall die Eigenschaft des Pushouts als dessen Colimesobjekt wir $Spec_c$ erhalten und im zweiten Fall die Konsequenz aus der für alle Sorten gleichen Variablenmenge X_{fix} . Damit sind beide Inklusionen „strongly liberal“ (\rightarrow [EBO91]) und es gilt:

$$\mathcal{V}_{\hookrightarrow} \circ \mathcal{F}_{\hookrightarrow} = \mathcal{ID}.$$

Nach Lemma 5.3.3 bildet das obere der beiden Rechtecke im nachfolgenden Diagramm, bei dessen Objekten es sich um die Quotiententermalgebren zu den jeweiligen Spezifikationen handelt, einen Pushout in der Kategorie der generalisierten Algebren $\mathcal{GSP\mathcal{E}C-ALG}$. Das darunterliegend Rechteck, mit der über A frei bzgl. der Inklusion von $Spec_a$ in $Spec_c$ konstuierten Algebra $\mathcal{F}(A)$, bildet nach Lemma 5.3.4 ebenfalls einen Pushout in $\mathcal{GSP\mathcal{E}C-ALG}$ und damit auch das äußere umfassende Rechteck.

$$\begin{array}{ccc} T_{Spec_p} & \xrightarrow{(\hookrightarrow, eval)} & T_{Spec} \\ (g, eval) \downarrow & \text{(PO)} & \downarrow (g_c, eval) \\ T_{Spec_a} & \xrightarrow{(\hookrightarrow, eval)} & T_{Spec_c} \\ eval \downarrow & \text{(PO)} & \downarrow eval \\ A & \xrightarrow{(\hookrightarrow, id)} & \mathcal{F}(A) \end{array}$$

In Zusammenhang mit der Eigenschaft der Transformation \mathcal{T}^A (\rightarrow 6.3.9) endliche Colimites zu bewahren (\rightarrow 6.3.10), ergibt sich damit aus dem Vorangehenden, der nachfolgende Pushout in $\mathcal{GSP\mathcal{E}C-ALG}$.

$$\begin{array}{ccc} T_{Spec_p}(X) & \xrightarrow{\mathcal{T}^A(\hookrightarrow, eval)} & T_{Spec}(X) \\ \mathcal{T}^A(g, eval) \downarrow & \text{(PO)} & \downarrow \mathcal{T}^A(g_c, eval) \\ A(X) & \xrightarrow{\mathcal{T}^A(\hookrightarrow, id)} & \mathcal{F}(A)(X) \end{array}$$

Mit der Definition der Projektion \mathcal{P}^N (\rightarrow 3.2.1) folgt:

$$\begin{aligned} \mathcal{P}^N((\hookrightarrow, eval): I_P \rightarrow I) &= (\hookrightarrow_{T_p^{\otimes}, T^{\otimes}}, \hookrightarrow_{P_p^{\otimes}, P^{\otimes}}, \mathcal{T}^A(\hookrightarrow, eval_{\mathcal{V}(T_{Spec})})) \\ \mathcal{P}^N((id, eval): I_P \rightarrow AN) &= (e_a^T, e_a^P, \mathcal{T}^A(g, eval_{\mathcal{V}(A)})). \end{aligned}$$

Daraus ergibt sich, wegen der komponentweisen Colimesbildung in $\mathcal{CP}\mathcal{TN}$, der folgende Pushout in $\mathcal{CP}\mathcal{TN}$.

$$\begin{array}{ccc}
 N_p^o & \xrightarrow{\mathcal{P}^N(\hookrightarrow, eval)} & N^o \\
 \mathcal{P}^N(id, eval) \downarrow & \text{(PO)} & \downarrow (e_c^T, e_c^P, \mathcal{T}^A(g_c, eval)) \\
 N_a^o & \xrightarrow{(\hookrightarrow, \hookrightarrow, \mathcal{T}^A(\hookrightarrow, id))} & N_c^o
 \end{array}$$

Aufgrund der komponentweisen Colimiteskonstruktion in $\mathcal{GAHL}\text{-}\mathcal{NETS}$ (\rightarrow 3.3.1) und $\mathcal{AHLN}\text{-}\mathcal{SPEC}$ (\rightarrow 3.3.4) in Kombination mit den bisherigen Ergebnissen, erhalten wir die folgende, zur ursprünglichen Definition der freien Konstruktion (\rightarrow 3.4.3) isomorphe, Konstruktion:

$$\begin{array}{ccc}
 I_p & \xrightarrow{(\hookrightarrow, eval)} & I \\
 eval \downarrow & \text{(PO)} & \downarrow eval \\
 AN & \xrightarrow{(\hookrightarrow, u_{AN})} & \mathcal{F}(AN)
 \end{array}$$

mit

$$\begin{aligned}
 \mathcal{F}(AN) &= (ANS, g_c, \mathcal{F}(A), N_c^o, eval_c), \\
 \mathcal{V}_{\hookrightarrow}(\mathcal{F}(AN)) &= (PAR, g_c \circ \hookrightarrow, \mathcal{F}(A), N_{\mathcal{V}}^o, eval_{\mathcal{V}}) \quad \text{und} \\
 u_{AN} &= \left((\hookrightarrow, id), (\hookrightarrow_{T_a^{\otimes}, T_{\mathcal{V}}^{\otimes}}, \hookrightarrow_{T_a^{\otimes}, T_{\mathcal{V}}^{\otimes}}, \mathcal{T}^A(\hookrightarrow, id)) \right).
 \end{aligned}$$

Es verbleibt zu zeigen:

- (1) $T_{\mathcal{V}}^{\otimes} = T_a^{\otimes}$,
- (2) $P_{\mathcal{V}}^{\otimes} = P_a^{\otimes}$ und
- (3) $\mathcal{T}^A(\hookrightarrow_{Spec_a, Spec_c}, id_A) = (\hookrightarrow_{Spec_a(X), Spec_c(X)}, id_{A(X)})$.

(.1): Aufgrund der Transitionskomponente des CP/T-Netzmorphismus zwischen N_a^o und $N_{\mathcal{V}}^o$, wissen wir: $T_a \subseteq T_{\mathcal{V}}$, $T_{\mathcal{V}} \subseteq T_a$ muß noch gezeigt werden. Jede Transition aus $T_{\mathcal{V}}$ liegt oder liegt nicht im Bild von $eval_{\mathcal{V}}$, dem entsprechend gliedert sich der Beweis in zwei Teile.

$t \in \mathcal{DC}_{eval_{\mathcal{V}}}(im)$	Annahme
$\implies \exists t' \in T_p: e_c^T \circ \hookrightarrow (t')(t) > 0$	3.1.4
$\implies \exists t' \in T_p: \hookrightarrow \circ e_p^T (t')(t) > 0$	(\rightarrow Pushout)
$\implies t \in T_a$	

$t \in \mathcal{DC}_{eval_y}(co)$	Annahme
$\implies t \in T_c \wedge t \in \mathcal{DC}_{eval_c}$	3.1.4
$\implies t \in T_a \vee \exists t' \in T: e_c^T(t')(t) > 0$	5.2.23
$\implies t \in T_a$	$t \in \mathcal{DC}_{eval_c}$, 1.2.3

(.2): Läßt sich mit der gleichen Argumentation nachweisen wie die entsprechende Eigenschaft für Transitionen.

(.3): Im folgenden führen wir notationelle Abkürzungen für eine Reihe von Inklusionen in der Kategorie der Datentypspezifikationen ein. Die ersten drei der dabei aufgeführten Inklusionen sind „strongly liberal“. In den ersten beiden Fällen haben wir die Begründung bereits gegeben. Die Inklusion der um A erweiterten Spezifikation mit Variablen $Spec_m(X, A)$ (\rightarrow 6.2.15) in die um $\mathcal{F}(A)$ erweiterte Spezifikation $Spec_c(X, \mathcal{F}(A))$ ist „strongly liberal“, weil diese Eigenschaft bereits für die Spezifikationen ohne Erweiterung gilt (\hookrightarrow_2) und wegen $\mathcal{V}_{\hookrightarrow_1} \circ \mathcal{F}_{\hookrightarrow_1} = \mathcal{ID}$ als Konsequenz dieser Eigenschaft in Bezug auf \hookrightarrow_1 .

$\hookrightarrow_1: Spec_a$	$\longrightarrow Spec_c$	strongly liberal
$\hookrightarrow_2: Spec_a(X)$	$\longrightarrow Spec_c(X)$	strongly liberal
$\hookrightarrow_3: Spec_a(X, A)$	$\longrightarrow Spec_c(X, \mathcal{F}(A))$	strongly liberal
$\hookrightarrow_4: Spec_a$	$\longrightarrow Spec_a(X)$	
$\hookrightarrow_5: Spec_a(X)$	$\longrightarrow Spec_a(X, A)$	

Aus Lemma 5.3.4 folgt zusammen mit Lemma 5.3.3 die Bewahrung von initialen Objekten durch freie Konstruktionen. Damit und mit \mathcal{V}_i als Abkürzung für $\mathcal{V}_{\hookrightarrow_i}$ resp. \mathcal{F}_i als Abkürzung für $\mathcal{F}_{\hookrightarrow_i}$ gilt:

$\mathcal{F}_3(T_{Spec_a}(X, A)) = T_{Spec_c}(X, \mathcal{F}_1(A))$	$T_{Spec\dots}(\dots)$ ist initial
$\implies \mathcal{V}_3(T_{Spec_c}(X, \mathcal{F}_1(A))) = \mathcal{V}_3 \circ \mathcal{F}_3(T_{Spec_a}(X, A))$	\hookrightarrow_3 strongly liberal
$= T_{Spec_a}(X, A)$	
$\implies eval: T_{Spec_a}(X, A) \rightarrow \mathcal{V}_3(T_{Spec_c}(X, \mathcal{F}_1(A)))$	Initialität
$= id_{T_{Spec_a}(X, A)}$	
$\implies \mathcal{V}_5(eval: T_{Spec_a}(X, A) \rightarrow \mathcal{V}_3(T_{Spec_c}(X, \mathcal{F}_1(A))))$	6.2.17, 6.2.15
$= id_{A(X)}$	

Daraus folgt mit $\mathcal{V}_1 \circ \mathcal{F}_1 = \mathcal{ID}$ (\hookrightarrow_1 strongly liberal) und 6.2.15

$$\begin{aligned} \hookrightarrow^{X_V} &= \mathcal{V}_5 (eval: T_{Spec_a}(X, \mathcal{V}_{\hookrightarrow} \circ \mathcal{F}_1(A)) \rightarrow \mathcal{V}_3(T_{Spec_c}(X, \mathcal{F}_1(A)))) \\ &= \mathcal{V}_5(eval: T_{Spec_a}(X, A) \rightarrow \mathcal{V}_3(T_{Spec_c}(X, \mathcal{F}_1(A)))) \\ &= id_{A(X)} \end{aligned}$$

und damit

$$\begin{aligned} &\mathcal{T}^A(\hookrightarrow_1, id_A) \\ &= (\hookrightarrow_2, \hookrightarrow^{X_V} \circ \mathcal{F}_4(id_A)) && 6.3.7 \\ &= (\hookrightarrow_2, id_{A(X)} \circ id_{A(X)}) && (\rightarrow \text{oben}) \\ &= (\hookrightarrow_2, id_{A(X)}) \end{aligned}$$

□

□

Beweis (3.4.5 Operationale Semantik der Parameterisierung):

(I):

Die Behauptung folgt aus der Eigenschaft von Funktoren die Komposition von Morphismen zu bewahren, der Tatsache das \mathcal{OS} ein Funktor ist (\rightarrow 3.2.6) und der Definition der freien Konstruktion (\rightarrow 3.4.3). □

(II):

Aus den Annahmen: $ANS(PAR)$ ist eine parameterisierte Spezifikation und $\mathcal{F}_{\hookrightarrow}(AN)$ das über AN frei konstruierte ANS -Modell, folgt mit Satz 3.4.4 die Existenz des universellen PAR -Modellmorphimus

$$u_{AN} = ((\hookrightarrow, id), (id, id, (\hookrightarrow, id))): AN \rightarrow \mathcal{V} \circ \mathcal{F}(AN) .$$

Daraus ergibt sich unmittelbar die Gleichheit der Transitions- und Stellenmenge der Netzstruktur in AN , im weiteren mit N^o bezeichnet, und der Netzstruktur in $\mathcal{V} \circ \mathcal{F}(AN)$, im weiteren mit $N_{\mathcal{V}\mathcal{F}}^o$ bezeichnet.

Der durch den CP/T-Netzmorphismus $(id, id, (\hookrightarrow, id))$ gegebene Zusammenhang beider Netzstrukturen impliziert darüber hinaus die Gleichheit der Bilder jeder Transition unter den Markenflußmorphimus beider Netze, d.h.

$$(a): \quad Pre = Pre_{\mathcal{V}\mathcal{F}} \Big|_{P^{\otimes} \otimes A(X)} \quad \text{und} \quad Post = Post_{\mathcal{V}\mathcal{F}} \Big|_{P^{\otimes} \otimes A(X)}$$

und die Gleichheit der Variablenmengen zu den Transitionen, d.h.

$$\forall t \in T: var(t) = var_{\mathcal{V}\mathcal{F}}(t).$$

Letzteres führt mit $(\hookrightarrow, id): A(X) \rightarrow \mathcal{F}(A)(X)$ ¹², der Voraussetzung

$$\forall s \in SO_p, op \in OP_s, eq \in EQ_s: op \in OP_{p,s} \wedge eq \in EQ_{p,s} \quad ^{13}$$

und der Definition der gültigen Variablenbelegungen (\rightarrow 2.3.1), zur Gleichheit der Variablenbelegungen für jede Transition, d.h.

$$\forall t \in T: Ass(t) = Ass_{\mathcal{V}\mathcal{F}}(t).$$

Unter Berücksichtigung der bisherigen Ergebnisse folgt aus der Definition der Netzentfaltung (\rightarrow 2.3.3):

$$\begin{aligned} \mathcal{UF}(AN) &= T^{\otimes} \odot Ass \begin{array}{c} \xrightarrow{Pre} \\ \xrightarrow{Post} \end{array} P^{\otimes} \otimes A(X) \\ \mathcal{UF}(\mathcal{V} \circ \mathcal{F}(AN)) &= T^{\otimes} \odot Ass \begin{array}{c} \xrightarrow{Pre_{\mathcal{V}\mathcal{F}}} \\ \xrightarrow{Post_{\mathcal{V}\mathcal{F}}} \end{array} P^{\otimes} \otimes \mathcal{F}(A)(X). \end{aligned}$$

Zusammen mit der Definition von \mathcal{OS} (\rightarrow 3.2.5) (a) als Komposition aus Entfaltung und Prozeßsemantik ($\mathcal{OS} = \mathcal{PC} \circ \mathcal{UF}$) und der Definition von \mathcal{PC} (\rightarrow 1.3.1, 1.3.11) ergibt sich daraus unsere Behauptung. \square

\square

Beweise zu Kapitel 5

Beweis (5.1.19 Limites generiert aus Produkten und Equalizern):

Aufgrund der Konstruktion von $\widehat{\ell}_o$ und $\widehat{\ell}_p$ gilt:

- (a): $\forall m \in Mor(\mathcal{S}): \ell_m \circ \widehat{\ell}_o = \ell_{cod(m)}$ und
 (b): $\forall m \in Mor(\mathcal{S}): \ell_m \circ \widehat{\ell}_p = \mathcal{D}(m) \circ \ell_{dom(m)}$.

(Verträglichkeit): Für alle Morphismen $m: s \rightarrow s'$ in \mathcal{S} gilt:

$$\begin{aligned} \ell_{s'}^e &= \ell_{cod(m)} \circ \ell && \text{Def.} \\ &= \ell_m \circ \widehat{\ell}_o \circ \ell && \text{(a)} \\ &= \ell_m \circ \widehat{\ell}_p \circ \ell && \text{Equalizer} \end{aligned}$$

¹² $\mathcal{F}(A)(X)$ bezeichnet die Algebra der Netzstruktur in $\mathcal{F}(AN)$ und der Netzstruktur $\mathcal{V} \circ \mathcal{F}(AN)$ (\rightarrow 3.4.3, 3.1.4)

¹³ SO_p , OP_p und EQ_p bezeichnen die Sorten-, Operations- und Gleichungsmenge in der Datentypspezifikation des formalen Parameters und SO , OP und EQ bezeichnen die entsprechenden Mengen in der Datentypspezifikation des Spezifikationsrumpfes.

$$= \mathcal{D}(m) \circ \ell_{\text{dom}(m)} \circ \ell \quad (\text{b})$$

$$= \mathcal{D}(m) \circ \ell_s^e \quad \text{Def.}$$

(Universelle Eigenschaft): Wir nehmen an: $(\mathcal{Q}, (\bar{\ell}_s)_{s \in \mathcal{S}})$ ist eine Quelle zu \mathcal{D} und konstruieren einen eindeutigen \mathcal{C} -Morphismus

$$\widehat{\ell}: \mathcal{Q} \rightarrow \mathcal{E} \quad \text{mit} \quad \forall s \in \text{Obj}(\mathcal{S}): \ell_s^e \circ \widehat{\ell} = \bar{\ell}_s .$$

Jede Quelle zu \mathcal{D} ist gleichzeitig eine Quelle zu dem im Diagramm links stehenden Produkt. Das impliziert, aufgrund der universellen Eigenschaft der Produkte, die Existenz und Eindeutigkeit eines \mathcal{C} -Morphismus

$$(c): \quad \widehat{\ell}_q: \mathcal{Q} \rightarrow \prod_{s \in \text{Obj}(\mathcal{S})} \mathcal{D}(s) \quad \text{mit der Eigenschaft} \quad \forall s \in \text{Obj}(\mathcal{S}): \ell_s \circ \widehat{\ell}_q = \bar{\ell}_s .$$

Darüber hinaus erhalten wir mit $(\mathcal{Q}, (\ell_m \circ \widehat{\ell}_p \circ \widehat{\ell}_q)_{m \in \text{Mor}(\mathcal{S})})$ eine Quelle für das im Diagramm rechts stehende Produkt und damit den eindeutigen Morphismus

$$\widehat{\ell}_r: \mathcal{Q} \rightarrow \prod_{m \in \text{Mor}(\mathcal{S})} \mathcal{D}(m) \quad \text{mit} \quad \forall m \in \text{Mor}(\mathcal{S}): \ell_m \circ \widehat{\ell}_r = \ell_m \circ \widehat{\ell}_p \circ \widehat{\ell}_q .$$

Indem wir

$$\widehat{\ell}_o \circ \widehat{\ell}_q \stackrel{?}{=} \widehat{\ell}_r \stackrel{?}{=} \widehat{\ell}_p \circ \widehat{\ell}_q$$

nachweisen, erhalten wir wegen der Eindeutigkeit des Equilizers (\mathcal{E}, ℓ) mit dieser Eigenschaft, den Morphismus $\widehat{\ell}: \mathcal{Q} \rightarrow \mathcal{E}$.

Die rechte der beiden Gleichungen folgt mit der Eindeutigkeit von $\widehat{\ell}_r$ direkt aus der Definition der Morphismenfamilie, die linke Gleichung folgt für alle Morphismen m in \mathcal{S} aus:

$$\begin{aligned} & \ell_m \circ \widehat{\ell}_o \circ \widehat{\ell}_q \\ &= \ell_{\text{cod}(m)} \circ \widehat{\ell}_q \quad (\text{a}) \end{aligned}$$

$$= \bar{\ell}_{\text{cod}(m)} \quad (\text{c})$$

$$= \mathcal{D}(m) \circ \bar{\ell}_{\text{dom}(m)} \quad \text{Quelle}$$

$$= \mathcal{D}(m) \circ \ell_{\text{dom}(m)} \circ \widehat{\ell}_q \quad (\text{c})$$

$$= \ell_m \circ \widehat{\ell}_p \circ \widehat{\ell}_q \quad (\text{b})$$

$$\implies \widehat{\ell}_o \circ \widehat{\ell}_q = \widehat{\ell}_r \quad \text{Eindeutigkeit v. } \widehat{\ell}_r$$

Damit existiert $\widehat{\ell}: \mathcal{Q} \rightarrow \mathcal{E}$ und ist bzgl. der nachfolgenden Eigenschaft eindeutig:

$$(d): \quad \ell \circ \widehat{\ell} = \widehat{\ell}_q .$$

Daraus folgt mit (c) und (d) für alle Objekte s in \mathcal{S} :

$$\ell_s^e \circ \widehat{\ell} = \ell_s \circ \ell \circ \widehat{\ell} = \ell_s \circ \widehat{\ell}_q = \bar{\ell}_s .$$

Für jeden Morphismus $\tilde{c}: \mathcal{Q} \rightarrow \mathcal{E}$ mit der Eigenschaft:

$$\forall s \in \text{Obj}(\mathcal{S}): \ell_s^e \circ \tilde{c} = \bar{\ell}_s$$

folgt mit (c) und (d): $\ell_s^e \circ \tilde{c} = \ell_s \circ \ell \circ \tilde{c} = \bar{\ell}_s \implies \ell \circ \tilde{c} = \widehat{\ell}_q \implies \tilde{c} = \widehat{\ell}$.

□

Beweis (5.2.9 Eigenschaften von $(-)^{op}$):

(I):

$(-)^{op}$ liefert als Bilder Morphismen in \mathcal{AG}_f . Da jeder Morphismus von \mathcal{AG}_f gleichzeitig Morphismus in \mathcal{AG}_f^{op} ist (\rightarrow 5.1.7), kann $(-)^{op}$ auch als Abbildung nach \mathcal{AG}_f^{op} betrachtet werden. Um nachzuweisen das $(-)^{op}$ ein Funktor ist müssen darüber hinaus die folgenden Eigenschaften gezeigt werden:

- $(-)^{op}$ bewahrt die Identitäten und
- $(-)^{op}$ bewahrt die Komposition von Morphismen.

$((-)^{op}$ bewahrt die Identitäten): Für jede freie abelsche Gruppe A^\otimes gilt:

$$\forall a, a' \in A: id_{A^\otimes}(a)(a) = 1 \wedge a' \neq a \implies id_{A^\otimes}(a)(a') = 0$$

$$\implies \forall a \in A: id_{A^\otimes}^{op}(a) = \bigoplus_{a' \in A} id_{A^\otimes}(a)(a') * a = a \quad 5.2.7$$

$((-)^{op}$ bewahrt die Komposition von Morphismen): Seien

$$f: A^\otimes \rightarrow B^\otimes, \quad g: B^\otimes \rightarrow C^\otimes$$

zwei Morphismen in den freien abelschen Gruppen, dann gilt für jedes c in C :

$$(g \circ f)^{op}(c)$$

$$= \bigoplus_{a \in A} g \circ f(a)(c) * a \quad 5.2.7$$

$$= \bigoplus_{a \in A} g \left(\bigoplus_{b \in B} f(a)(b) * b \right) (c) * a$$

$$= \bigoplus_{a \in A} \bigoplus_{b \in B} f(a)(b) * g(b)(c) * a \quad 5.2.3, g \text{ bewahrt } \bigoplus$$

$$= \bigoplus_{b \in B} \bigoplus_{a \in A} g(b)(c) * f(a)(b) * a \quad 5.2.1 \text{ (VI)}$$

$$= \bigoplus_{b \in B} (g(b)(c) * \bigoplus_{a \in A} f(a)(b) * a) \quad 5.2.1 \text{ (VI)}$$

$$= \bigoplus_{b \in B} (g(b)(c) * f^{op}(b)) \quad 5.2.7$$

$$= f^{op}(\bigoplus_{b \in B} g(b)(c) * b)$$

$$= f^{op}(g^{op}(c)) = f^{op} \circ g^{op}(c)$$

□

(II und III):

Die Äquivalenzen gelten, wenn $(-)^{op}$ bijektiv ist und die Monotonie bewahrt. Mit (IV) folgt aus

$$\forall f \in \text{Mor}(\mathcal{AG}_f^{op}): f^{op} \in \mathcal{AG}_f \wedge (f^{op})^{op} = f$$

die Surjektivität von $(-)^{op}$.

Für alle Morphismen f in \mathcal{AG}_f , alle a in $\text{Base}(\text{dom}(f))$ und alle b in $\text{Base}(\text{cod}(f))$ gilt:

$$(a): \quad f^{op}(b)(a) = \left(\bigoplus_{a' \in A} f(a')(b) * a' \right)(a) = f(a)(b).$$

Damit zeigen wir für alle A^\otimes, B^\otimes in \mathcal{AG}_f und f, g in $\text{Mor}(\mathcal{AG}_f)[A^\otimes, B^\otimes]$:

$$f \neq g$$

$$\implies \exists a \in A, b \in B: f(a)(b) \neq g(a)(b)$$

$$\implies \exists a \in A, b \in B: \quad (a)$$

$$f^{op}(b)(a) = f(a)(b) \neq g(a)(b) = g^{op}(b)(a)$$

$$\implies f^{op} \neq g^{op}$$

und damit die Injektivität von $(-)^{op}$.

Die Summenbildung im Zuge der Konstruktion dualer Morphismen erhält jeweils die Vorzeichen des zugrundeliegenden Morphismus, d.h. $f(a)(b) > 0 \implies f^{op}(b)(a) > 0$. Damit bewahrt $(-)^{op}$ die Monotonie. □

(IV):

Mit (a) gilt für alle Morphismen f in \mathcal{AG}_f , alle a in $\text{Base}(\text{dom}(f))$ und alle b in $\text{Base}(\text{cod}(f))$:

$$(f^{op})^{op}(a) = \bigoplus_{b \in B} f^{op}(b)(a) * b = \bigoplus_{b \in B} f(a)(b) * b = f(a).$$

Daraus folgt unsere Behauptung. □

□

Beweis (5.2.10 Freie-Konstruktion):

(I):

Wir definieren zu jeder Menge A die Abbildung u_A durch:

$$u_A: A \rightarrow \mathcal{V}(A^\otimes) \stackrel{\text{def}}{=} \hookrightarrow_{A, A^\otimes}$$

und beweisen, daß das Paar (u_A, A^\otimes) eine \mathcal{V} -universelle Abbildung zu A (\rightarrow [HS73] Def. 26.1, S. 178) bildet.

$((u_A, A^\otimes)$ ist eine \mathcal{V} -universelle Abbildung zu A): Sei $f: A \rightarrow \mathcal{V}(B^\otimes)$ eine Funktion, dann definieren wir $f': A^\otimes \rightarrow B^\otimes$ als den \mathcal{AG} -Morphismus für den

$$\forall a \in A: f'(a) = f(a)$$

gilt und zeigen daß er das nachfolgende Diagramm eindeutig kommutativ ergänzt.

$$\begin{array}{ccc} A & \xrightarrow{u_A} & \mathcal{V}(A^\otimes) & & A^\otimes \\ & \searrow f & \downarrow \mathcal{V}(f') & & \downarrow f' \\ & & \mathcal{V}(B^\otimes) & & B^\otimes \end{array}$$

Für alle a aus A gilt:

$$\mathcal{V}(f') \circ u_A(a) = \mathcal{V}(f')(a) = f(a)$$

und für alle Morphismen $g: A^\otimes \rightarrow B^\otimes$ mit $\mathcal{V}(g) \circ u_A = f$ gilt:

$$\forall a \in A: \mathcal{V}(g)(a) = f(a) = \mathcal{V}(f')(a) \implies g = f' \quad .$$

□

(II):

Zunächst zeigen wir die Funktoreigenschaften von $(-)^{\otimes}$, d.h.

- $(-)^{\otimes}$ bewahrt die Identitäten und
- $(-)^{\otimes}$ bewahrt die Komposition.

$(-)^{\otimes}$ bewahrt die Identitäten): Für alle Mengen A und alle Elemente v der freien abelschen Gruppe A^\otimes gilt:

$$id_{A^\otimes}(v)$$

$$= id_{A^\otimes} \left(\bigoplus_{a \in A} v(a) * a \right)$$

$$= \bigoplus_{a \in A} v(a) * id_A(a)$$

5.2.5

$$= \bigoplus_{a \in A} v(a) * a$$

$$= v$$

$(-)^{\otimes}$ bewahrt die Komposition): Seien f, g zwei komponierbare Funktionen und v ein Element aus $\text{dom}(g)^{\otimes}$, dann gilt:

$$(f \circ g)^{\otimes}(v)$$

$$= (f \circ g)^{\otimes} \left(\bigoplus_{a \in \text{dom}(g)} v(a) * a \right)$$

$$= \bigoplus_{a \in \text{dom}(g)} v(a) * f(g(a)) \tag{5.2.5}$$

$$= f^{\otimes} \left(\bigoplus_{a \in \text{dom}(g)} v(a) * g(a) \right) \tag{5.2.5}$$

$$= f^{\otimes} (g^{\otimes} \left(\bigoplus_{a \in \text{dom}(g)} v(a) * a \right)) \tag{5.2.5}$$

$$= f^{\otimes} \circ g^{\otimes}(v) \tag{5.2.5}$$

Wir zeigen, daß

$$u \stackrel{\text{def}}{=} (u_A)_{A \in \text{Obj}(\mathcal{SET})}: \mathcal{ID}_{\mathcal{SET}} \rightarrow \mathcal{V} \circ (-)^{\otimes}$$

eine Funktortransformation bildet. Da im Fall, daß es zu jeder Menge A eine \mathcal{V} -universelle Abbildung (u_A, A^{\otimes}) gibt, ein bzgl. \mathcal{V} freier Funktor \mathcal{F} mit

$$\forall A \in \text{Obj}(\mathcal{SET}): \mathcal{F}(A) = A^{\otimes} \quad \text{und} \quad u: \mathcal{ID}_{\mathcal{SET}} \rightarrow \mathcal{V} \circ \mathcal{F}$$

eindeutig existiert (\rightarrow [HS73] Theor. 26.11, S. 191), folgt daraus $\mathcal{F} = (-)^{\otimes}$ und damit unsere Behauptung $(-)^{\otimes}$ sei ein freier Funktor.

$(u$ ist eine Funktortransformation): Sei $f: A \rightarrow \mathcal{V}(B^{\otimes})$ eine Funktion in \mathcal{SET} . Wir zeigen, daß f^{\otimes} das nachfolgende Diagramm kommutativ ergänzt.

$$\begin{array}{ccc}
 A & \xrightarrow{u_A} & \mathcal{V}(A^{\otimes}) & & A^{\otimes} \\
 \downarrow f & & \downarrow \mathcal{V}(f^{\otimes}) & \circlearrowleft & \downarrow f^{\otimes} \\
 B & \xrightarrow{u_B} & \mathcal{V}(B^{\otimes}) & & B^{\otimes}
 \end{array}$$

Für alle a aus A gilt: $\mathcal{V}(f^{\otimes}) \circ u_A(a) = \mathcal{V}(f^{\otimes})(a) = f(a) = u_B \circ f(a)$ □

□

Beweis (5.2.11 Covollständigkeit):

Für alle endlichen Kategorien \mathcal{S} und Funktoren \mathcal{D} von \mathcal{S} in \mathcal{AG}_{fg} existiert ein Funktor

$$\mathcal{G}: \mathcal{S} \rightarrow \mathcal{SET}_f \text{ mit der Eigenschaft: } \mathcal{D} = (-)^\otimes \circ \mathcal{G}$$

(\mathcal{SET}_f bezeichnet die Kategorie der endlichen Mengen) (\rightarrow 5.2.1, 5.2.5). Da $(-)^\otimes$ ein freier Funktor ist (\rightarrow 5.2.10) bewahrt er Colimites (\rightarrow [HS73] 26.11, S. 191) und die existieren, für jede endliche Kategorie \mathcal{S} , in der Kategorie der endlichen Mengen (\rightarrow [HS73] 26.9 (1), S. 160). Damit existiert der Colimes von \mathcal{D} und, da wir keine Einschränkungen für \mathcal{D} verlangen, ist \mathcal{AG}_{fg} endlich covollständig.

□

Beweis (5.2.14 Funktor): **$(-)^\mathcal{M}$ ist ein Funktor):**

Um nachzuweisen, daß die Abbildung $(-)^\mathcal{M}$ ein Funktor ist, muß gezeigt werden:

- $(-)^\mathcal{M}$ bewahrt die Identitäten und
- $(-)^\mathcal{M}$ bewahrt die Kompositon.

$(-)^\mathcal{M}$ bewahrt Identitäten): Sei A^\otimes eine freie abelsche Gruppe, $|A| = n$ und $(id_{A^\otimes})^\mathcal{M} = (m_{i,j})_{i,j=1,\dots,n}$, dann gilt:

$$\forall i, j \in \{1, \dots, n\} : id_{A^\otimes}(a_i) = a_i$$

$$\implies m_{i,i} = 1 \wedge m_{j,k} = 0 \text{ für } j \neq i \vee k \neq i \quad 5.2.13$$

$$\implies (id_{A^\otimes})^\mathcal{M} = E_n$$

Die Einheitsmatrizen E_n bilden bzgl. der Multiplikation die neutralen Elemente innerhalb der Matrizen.

$(-)^\mathcal{M}$ bewahrt die Kompositon): Zwei $m \times n$ -Matrizen M, K sind genau dann gleich, wenn jede ihrer Spalten gleich ist. Das heißt wenn

$$M * e_i = K * e_i, \quad i = 1, \dots, n$$

gilt, wobei e_i den i -ten Spaltenvektor der $n \times n$ -Einheitsmatrix bezeichnet. Übertragen auf, von Morphismen in den freien abelschen Gruppen, generierte Matrizen folgt daraus für alle Morphismen f, g mit $dom(f) = dom(g)$ und $cod(f) = cod(g)$

$$(f)^\mathcal{M} = (g)^\mathcal{M} \iff \forall a \in Base(dom(f)): (f)^\mathcal{M} * \vec{a} = (g)^\mathcal{M} * \vec{a} .$$

Aus der Definition von $(-)^\mathcal{M}$ (\rightarrow 5.2.13) folgt für jeden \mathcal{AG}_f -Morphismus f und jedes Element v aus $dom(f)$:

$$(f)^\mathcal{M} * \vec{v} = f(\vec{v}) .$$

Damit läßt sich für alle komponierbaren Morphismen f, g und alle a aus $\text{dom}(g)$ zeigen:

$$(f \circ g)^{\mathcal{M}} * \vec{a} = f(g(\vec{a})) = (f)^{\mathcal{M}} * g(\vec{a}) = (f)^{\mathcal{M}} * (g)^{\mathcal{M}} * \vec{a}$$

und damit $((f \circ g))^{\mathcal{M}} = (f)^{\mathcal{M}} * (g)^{\mathcal{M}}$.

□

$(-)^{\mathcal{M}}$ ist faithful):

Alle Morphismen zwischen zwei Gruppen, unterscheiden sich mindestes im Bild eines Basiselements. Die daraus resultierenden Matrizen unterscheiden sich dann genau in dem diesem Basiselement entsprechenden Spaltenvektor.

□

$(-)^{\mathcal{M}}$ ist full):

Für jede $m \times n$ -Matrix M gilt:

$$M = \left(Mp \left(M, \{a_1, \dots, a_n\}^{\otimes}, \{b_1, \dots, b_m\}^{\otimes} \right) \right)^{\mathcal{M}}$$

(\rightarrow 5.2.15), d.h. sie liegt im Bild von $(-)^{\mathcal{M}}$.

□

□

Beweis (5.2.17 Produkte):

Die disjunkte Vereinigung von Mengen ist eine Menge und sie ist endlich, wenn alle beteiligten Mengen endliche sind. Damit bildet das Limesobjekt

$$\left(\biguplus_{s \in \mathcal{S}} \text{Base}(\mathcal{D}(s)) \right)^{\otimes}$$

nach Definition der freien abelschen Gruppen (\rightarrow 5.2.1) ein Objekt in \mathcal{AG}_f resp. in \mathcal{AG}_{fm} .

Basiselemente werden von den universellen Abbildungen ℓ_s auf das jeweilige Basiselement der freien abelschen Gruppe im Bild von $\mathcal{D}(s)$ abgebildet, wenn sie dieser Basis entstammen und sonst auf das neutrale Element. Ihre Verträglichkeit mit \oplus und dem neutralen Element folgt direkt aus der Definition. Sie bilden damit monotone Morphismen von der freien abelschen Gruppe über der disjunkten Vereinigung der Basismengen, in die Gruppen über denen das Produkt gebildet wurde.

Damit verbleibt noch zu Zeigen:

- die Verträglichkeit mit den Morphismen aus \mathcal{S}
- und die universellen Eigenschaften.

(Verträglichkeit): In \mathcal{S} sind alle Morphismen Identitäten, damit folgt aus

$$\ell_s \circ id_{\mathcal{D}(s)} = \ell_s$$

die Verträglichkeit der universellen Abbildungen mit den Morphismen aus \mathcal{S} .

(universelle Eigenschaft): Wir nehmen an, daß $(\bar{L}^\otimes, (\bar{\ell}_s)_{s \in \mathcal{S}})$ eine weitere Senke für \mathcal{D} ist und definieren den \mathcal{AG}_f -Morphismus

$$\widehat{\ell}: \bar{L}^\otimes \rightarrow \left(\biguplus_{s \in \mathcal{S}} Base(\mathcal{D}(s)) \right)^\otimes \text{ durch: } \forall q \in \bar{L}: \widehat{\ell}(q) = \bigoplus_{s \in \mathcal{S}} \bigoplus_{a \in Base(\mathcal{D}(s))} \bar{\ell}_s(q)(a) * a_s.$$

$\widehat{\ell}$ bewahrt \oplus und das neutrale Element, da alle $\bar{\ell}_s$ als \mathcal{AG}_f -Morphismen diese Eigenschaften haben und $\widehat{\ell}$ ist monoton, wenn alle Morphismen der Senke monoton sind. Die Verträglichkeit von $\widehat{\ell}$ mit den Morphismen der Senke folgt aus:

$$\forall q \in \bar{L}, s \in \mathcal{S}:$$

$$\begin{aligned} & \ell_s \circ \widehat{\ell}(q) \\ &= \ell_s \left(\bigoplus_{s \in \mathcal{S}} \bigoplus_{a \in Base(\mathcal{D}(s))} \bar{\ell}_s(q)(a) * a_s \right) && \text{Def. } \widehat{\ell} \\ &= \bigoplus_{a \in Base(\mathcal{D}(s))} \bar{\ell}_s(q)(a) * a && \text{Def. } \ell_s \\ &= \bar{\ell}_s(q) && 5.2.1 \end{aligned}$$

Um die Eindeutigkeit von $\widehat{\ell}$ zu zeigen nehmen an,

$$\widetilde{\ell}: \bar{L}^\otimes \rightarrow \left(\biguplus_{s \in \mathcal{S}} Base(\mathcal{D}(s)) \right)^\otimes$$

ist ein weiterer, mit den Morphismen der Senke verträglicher Morphismus in den freien abelschen Gruppen, d.h. es gilt $\ell_s \circ \widetilde{\ell} = \bar{\ell}_s$ für alle s in \mathcal{S} . Daraus folgt:

$$\forall q \in \bar{L}, s \in \mathcal{S}:$$

$$\begin{aligned} & \ell_s \circ \widetilde{\ell}(q) \\ &= \bigoplus_{a \in Base(\mathcal{D}(s))} \widetilde{\ell}(q)(a_s) * a && \text{Def. } \ell_s \\ &= \bigoplus_{a \in Base(\mathcal{D}(s))} \bar{\ell}_s(q)(a) * a && \text{Def. } \ell_s, \ell_s \circ \widetilde{\ell} = \bar{\ell}_s \\ &= \bigoplus_{a \in Base(\mathcal{D}(s))} \widehat{\ell}(q)(a_s) * a \\ &\implies \widetilde{\ell} = \widehat{\ell} \end{aligned}$$

□

Beweis (5.2.19 Existenz von Equalizern):

(Equalizer in \mathcal{AG}_f):

Betrachtet man

$$DF = (\mathcal{D}(m))^{\mathcal{M}} - (\mathcal{D}(m'))^{\mathcal{M}}$$

als Matrix über dem Ring $(\mathbb{Q}, +, *)$, dann ergibt sich die Dimension des Lösungsraums zum linearen Gleichungssystem

$$DF * x = 0$$

(\rightarrow [KM95], Folgerung 3.4.8 e) zu $n - \text{Rang}(DF) = r$.

Für die Spaltenvektoren $\vec{l}_1, \dots, \vec{l}_r$ der Lösungsmatrix L folgt aus $DF * L = 0$:

$$DF * \vec{l}_i = 0, \quad i = 1, \dots, r.$$

Da sie darüber hinaus aufgrund der Eigenschaft der Folge $1 \leq j_1 \leq \dots \leq j_r \leq n$ linearunabhängig sind, bilden sie eine Basis von $\text{Kern}(DF)$. Damit existiert zu jedem Vektor \vec{v} aus $\text{Kern}(DF)$ genau ein Vektor \vec{u}_v mit der Eigenschaft:

$$L * \vec{u}_v = \vec{v}.$$

Diese Vektoren sind von der Form:

$$\vec{u}_v = \begin{pmatrix} v_{j_1} \\ \vdots \\ v_{j_r} \end{pmatrix},$$

da für jedes $i = 1, \dots, r$ nur \vec{l}_i an der Stelle j_i einen Wert ungleich Null, nämlich eins, enthält.

Aus der Form der \vec{u}_v folgt unmittelbar, daß L auch dann eine Basis von $\text{Kern}(DF)$ bildet, wenn DF als Matrix über dem Ring \mathbb{Z} aufgefasst wird. Darüber hinaus liegt der Vektor \vec{u}_v in \mathbb{N}^n , wenn \vec{v} in \mathbb{N}^n liegt. Damit läßt sich jeder $n \times q$ -Matrix M über \mathbb{Z} (\mathbb{N}) mit Spaltenvektoren $\vec{m}_1, \dots, \vec{m}_q$ in $\text{Kern}(DF)$ eindeutig eine $r \times q$ -Matrix U über \mathbb{Z} (\mathbb{N}) mit Vektoren $u_{\vec{m}_1}, \dots, u_{\vec{m}_r}$ und der Eigenschaft

$$L * U = M$$

zuordnen.

Auf dieser Grundlage können wir nun unsere Behauptung nachweisen. Dazu ist im einzelnen zu Zeigen:

- (1) $(E^\otimes, Mp(L, E^\otimes, A^\otimes))$ bildet eine Quelle für \mathcal{D} in \mathcal{AG}_f und
- (2) zu jedem anderen potentiellen Equalizer $(\bar{E}^\otimes, \bar{\ell})$ existiert eindeutig ein \mathcal{AG}_f -Morphismus $\hat{\ell}: \bar{E}^\otimes \rightarrow E^\otimes$ mit

$$Mp(L, E^\otimes, A^\otimes) \circ \hat{\ell} = \bar{\ell}.$$

(.1): E^\otimes ist als Bild des Funktors $(-)^{\otimes}$ (\rightarrow 5.2.14) eine freie abelsche Gruppe und $Mp(L, E^\otimes, A^\otimes)$ (\rightarrow 5.2.15) ein Morphismus in \mathcal{AG}_f . Aus

$$\left((\mathcal{D}(m))^{\mathcal{M}} - (\mathcal{D}(m'))^{\mathcal{M}} \right) * L = 0$$

folgt:

$$\begin{aligned} (\mathcal{D}(m))^{\mathcal{M}} * L - (\mathcal{D}(m'))^{\mathcal{M}} * L &= 0 && \text{[KM95] 3.1.23} \\ \implies (\mathcal{D}(m))^{\mathcal{M}} * L &= (\mathcal{D}(m'))^{\mathcal{M}} * L \\ \implies \mathcal{D}(m) \circ Mp(L, E^\otimes, A^\otimes) &= \mathcal{D}(m') \circ Mp(L, E^\otimes, A^\otimes) && (-)^{\mathcal{M}} \text{ ist faithful, 5.2.16} \end{aligned}$$

(.2): Sei $(\bar{E}^\otimes, \bar{\ell})$ ein potentieller Equalizer für \mathcal{D} in \mathcal{AG}_f , d.h. es gilt:

$$\mathcal{D}(m) \circ \bar{\ell} = \mathcal{D}(m') \circ \bar{\ell},$$

dann folgt:

$$\begin{aligned} (\mathcal{D}(m))^{\mathcal{M}} * (\bar{\ell})^{\mathcal{M}} &= (\mathcal{D}(m'))^{\mathcal{M}} * (\bar{\ell})^{\mathcal{M}} && (-)^{\mathcal{M}} \text{ ist faithful, 5.2.16} \\ \implies \left((\mathcal{D}(m))^{\mathcal{M}} - (\mathcal{D}(m'))^{\mathcal{M}} \right) * (\bar{\ell})^{\mathcal{M}} &= 0 && \text{[KM95] 3.1.23} \end{aligned}$$

d.h. die Spaltenvektoren von $(\bar{\ell})^{\mathcal{M}}$ liegen in $\text{Kern}(DF)$. Damit existiert eindeutig eine $r \times q$ -Matrix U über \mathbb{Z} und es gilt:

$$\begin{aligned} L * U &= (\bar{\ell})^{\mathcal{M}} \\ \implies Mp(L, E^\otimes, A^\otimes) \circ Mp(U, \bar{E}^\otimes, E^\otimes) &= \bar{\ell} && (-)^{\mathcal{M}} \text{ ist faithful, 5.2.16} \\ \implies \text{mit } \hat{\ell} = Mp(U, \bar{E}^\otimes, E^\otimes) : Mp(L, E^\otimes, A^\otimes) \circ \hat{\ell} &= \bar{\ell} && \text{5.2.16} \end{aligned}$$

□

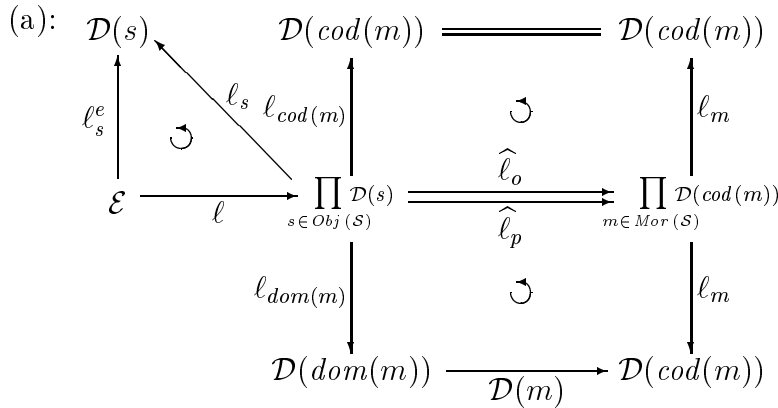
(Equalizer in \mathcal{AG}_{fm}):

Bildet L eine Matrix über \mathbb{N} , dann ist $Mp(L, E^\otimes, A^\otimes)$ monoton (\rightarrow 5.2.15). Da es sich bei den, von Matrizen über \mathbb{N} mit Spaltenvektoren in $\text{Kern}(DF)$, generierten Matrizen U ebenfalls um Matrizen über \mathbb{N} handelt, sind auch die daraus gebildeten Morphismen $\hat{\ell} = Mp(U, \bar{E}^\otimes, E^\otimes)$ monoton. Im Falle $\text{cod}(\mathcal{D}) \subset \mathcal{AG}_{fm}$ folgt daraus zusammen mit den bisher gezeigten Ergebnissen: $(E^\otimes, Mp(L, E^\otimes, A^\otimes))$ ist ein Equalizer für \mathcal{D} in den freien abelschen Gruppen mit monotonen Morphismen. □

□

Beweis (5.2.21 Limites spezieller Diagramme):

Nach Satz 5.1.19 läßt sich der Limes eines jeden endlichen Diagramms mit Hilfe des Equalizers $(\mathcal{E}, \ell)^{14}$ der Morphismen $\widehat{\ell}_o, \widehat{\ell}_p$ im nachfolgenden Diagramm (a) ermitteln:



d.h. unsere Behauptung ist equivalent zu:

$$\left(E^\otimes, Mp \left(L, E^\otimes, \prod_{s \in \text{Obj}(\mathcal{S})} \mathcal{D}(s) \right) \right) \text{ ist Equalizer von } \prod_{s \in \text{Obj}(\mathcal{S})} \mathcal{D}(s) \xrightleftharpoons[\widehat{\ell}_p]{\widehat{\ell}_o} \prod_{m \in \text{Mor}(\mathcal{S})} \text{cod}(m)$$

in \mathcal{AG}_{fm} . Der Beweis dazu erfolgt mit Hilfe von Lemma 5.2.19. Zu diesem Zweck zeigen wir zunächst:

$$(b) : \left((\widehat{\ell}_p)^{\mathcal{M}} - (\widehat{\ell}_o)^{\mathcal{M}} \right) * L = 0$$

und im Anschluß daran die übrigen Voraussetzungen von Lemma 5.2.19.

(Gleichung (b) gilt):

Die in Diagramm (a) enthaltenen Produkte¹⁵ (\rightarrow 5.2.17) entsprechen bis auf Isomorphie den freien abelschen Gruppen

$$(B \uplus A_1 \uplus \dots \uplus A_m)^\otimes \quad \text{bzw.} \quad (B^{m_1} \uplus \dots \uplus B^{m_{n+1}} \uplus B \uplus A_1 \uplus \dots \uplus A_m)^\otimes$$

mit $B^{m_i} \stackrel{\text{def}}{=} \{ b_j^{m_i} \mid b_j \in B \}$ für alle Morphismen m_i in \mathcal{S} .

Ihre Basen enthalten beide die disjunkte Vereinigung aller abelschen Gruppen im Diagramm, die Basis des rechten Produkts darüber hinaus $n + 1$ Instanzen der Menge B , entsprechend den $n + 1$ Morphismen mit Codomäne B^\otimes . Das Enthaltensein des Linken im rechten Produkt ist eine Folge des Identitätsmorphismus zu jeder Gruppe im Diagramm. Er bewirkt über seine Codomäne die Inklusion der jeweiligen Gruppe im rechten Produkt.

¹⁴sofern ein derartiger Equalizer existiert

¹⁵Genaugenommen müßte man in diesem Zusammenhang von den Produktobjekten, im Gegensatz zu den universellen Morphismen des Produkts, reden.

Die Anzahl der Elemente in den Basismengen und damit die Dimension der Matrizen zu den Morphismen in (b), ergibt sich mit Hilfe der Bijektivität der

$$g_i: A_i \setminus \text{Kern}(f_i) \rightarrow B, \quad 1 \leq i \leq n$$

und der Definition von r (\rightarrow 5.2.19) aus

$$\begin{aligned} & \left(\sum_{i=1}^m |A_i| \right) + |B| \\ &= \sum_{i=1}^n |A_i \setminus \text{Kern}(f_i)| + \sum_{i=1}^n |A_i \cap \text{Kern}(f_i)| + \sum_{i=n+1}^m |A_i| \\ & \quad + |B| \\ &= \left(\sum_{i=1}^n |\text{dom}(g_i)| \right) + r + |B| && \text{Def. } r \\ &= (n+1) * |B| + r && \text{Bijektivität v. } g_i \end{aligned}$$

für das linke bzw.

$$(n+1) * |B| + r + (n+1) * |B| = 2 * (n+1) * |B| + r \quad n+1 \text{ Mor. mit Cod. } B$$

für das rechte Produkt.

Die Morphismen

$$\ell_s: \prod_{s \in \text{Obj}(\mathcal{S})} \mathcal{D}(s) \rightarrow \mathcal{D}(s)$$

(\rightarrow 5.2.17) für alle Objekte s in \mathcal{S} des linken Produkts bzw.

$$\ell_m: \prod_{m \in \text{Mor}(\mathcal{S})} \mathcal{D}(\text{cod}(m)) \rightarrow \mathcal{D}(\text{cod}(m))$$

für alle Morphismen m in \mathcal{S} des rechten Produkts ergeben sich aus der vorangegangenen Definition der Produktobjekte zu:

$$(c): \quad \ell_s(x) = \begin{cases} a_j^i & ; x = a_j^i \wedge s = s_i \\ b_j & ; x = b_j \wedge s = s' \\ \ell_s(y) \oplus \ell_s(z) & ; x = y \oplus z \\ 0 & ; \text{sonst} \end{cases}$$

respektive zu:

$$(d): \quad \ell_m(x) = \begin{cases} a_j^i & ; x = a_j^i \wedge m = id_{s_i} \\ b_j & ; x = b_j^m \vee (x = b_j \wedge m = id_{s'}) \\ \ell_m(y) \oplus \ell_m(z) & ; x = y \oplus z \\ 0 & ; \text{sonst} \end{cases}$$

Wir zeigen im folgenden, daß unter Annahme der folgenden Ordnung¹⁶ auf den Basiselementen des rechten Produkts

$$(e) : b_1^{m_1} < \dots < b_{|B|}^{m_1} < \dots < b_1^{m_{n+1}} < \dots < b_{|B|}^{m_{n+1}} < b_1 < \dots < a_{|A_n|}^n$$

(Elemente zwischen b_1 und $a_{|A_n|}^n$ wie in 5.2.21) und mit Hilfe der in Lemma 5.2.21 vorgegebenen Ordnung auf der Basis des linken Produkts gilt:

$$\begin{aligned} (\widehat{\ell}_o)^{\mathcal{M}} &= \left(\begin{array}{c|c} \begin{array}{c} E_{|B|} \\ \vdots \\ E_{|B|} \end{array} & 0 \\ \hline E_{(n+1)*|B|+r} \end{array} \right) \\ (\widehat{\ell}_p)^{\mathcal{M}} &= \left(\begin{array}{c|c|c|c} 0 & E_{n*|B|} & 0 & 0 \\ \hline & 0 & (f_{n+1})^{\mathcal{M}} & \\ \hline E_{(n+1)*|B|+r} \end{array} \right) \end{aligned}$$

und damit (f):

$$(\widehat{\ell}_p)^{\mathcal{M}} - (\widehat{\ell}_o)^{\mathcal{M}} = \left(\begin{array}{c|c|c|c} -E_{|B|} & E_{n*|B|} & 0 & 0 \\ \hline \vdots & & & \\ \hline -E_{|B|} & 0 & (f_{n+1})^{\mathcal{M}} & \end{array} \right)$$

Die $(n + 1) * |B| + r$ nur mit Nullen gefüllten Zeilen am Ende der Matrix wurden weggelassen, da sie auf die Lösung der Matrixgleichung keinen Einfluß haben.

(Die Matrix zu $\widehat{\ell}_o$ hat die angegebene Form): Mit der Definition der Produkte (\rightarrow 5.2.17) gilt:

$$\forall i \in \{1, \dots, n + 1\}, b_j \in B: \quad \text{(a), 5.2.17, (c),}$$

$$\ell_{cod(m_i)}(b_j) = b_j = \ell_{m_i} \circ \widehat{\ell}_o(b_j) \quad cod(m_i) = s'$$

$$\implies \forall b_j \in B: \widehat{\ell}_o(b_j) = \bigoplus_{i=1}^{n+1} b_j^{m_i} \oplus b_j \quad \text{(d), } cod(id_{s'}) = s',$$

Eindeutigkeit v. $\widehat{\ell}_o$

¹⁶Die Lösung von Matrixgleichungen ist gegenüber Spalten- und Reihenvertauschung invariant. Daher hat die Wahl der Ordnungen keinen Einfluß auf die Lösung.

\implies

$$\forall i \in \{1, \dots, n+1\} : \left(\widehat{\ell}_o \Big|_{B^{\otimes}}^{B^{m_i \otimes}} \right)^{\mathcal{M}} = E_{|B|} \quad (\text{e}), (\text{c})$$

und

$$\left(\widehat{\ell}_o \Big|_{B^{\otimes}}^{B^{\otimes}} \right)^{\mathcal{M}} = E_{|B|} \quad (\text{e}), (\text{c})$$

Für die übrigen Elemente des Produkts über alle Bilder der Objekte in \mathcal{S} gilt, da sie nur im Bild des jeweiligen Identitätsmorphisms liegen:

$$\forall i, x \in \{1, \dots, m\}, a_j^i \in A_i, a_y^x \in A_x:$$

$$\widehat{\ell}_o(a_y^x)(a_j^i) = \ell_{id_i} \circ \widehat{\ell}_o(a_y^x)(a_j^i) = \ell_{cod(id_i)}(a_y^x)(a_j^i) \quad (\text{a}), (\text{c})$$

$$\implies \widehat{\ell}_o(a_y^x)(a_j^i) = 1 \iff x = i \wedge y = j \quad (\text{c})$$

$$\widehat{\ell}_o(a_y^x)(a_j^i) = 0 \iff x \neq i \vee y \neq j$$

Daraus folgt zusammen mit dem Bild von B^{\otimes} unter $\widehat{\ell}_o$ (\rightarrow s.o.) und der übereinstimmenden Ordnung auf den gemeinsamen Elementen der Basen beider Produkte:

$$\left(\widehat{\ell}_o \Big|_{(B \uplus A_1 \uplus \dots \uplus A_m)^{\otimes}}^{(B \uplus A_1 \uplus \dots \uplus A_m)^{\otimes}} \right)^{\mathcal{M}} = E_{(n+1) * |B| + r} \quad \text{und} \quad \left(\widehat{\ell}_o \Big|_{(A_1 \uplus \dots \uplus A_m)^{\otimes}}^{(B^{m_1 \uplus \dots \uplus B^{m_{n+1}}})^{\otimes}} \right)^{\mathcal{M}} = 0.$$

Zusammen mit den vorher gezeigten Ergebnissen für die anderen Submatrizen folgt daraus, daß die Matrix zu $\widehat{\ell}_o$ die angegebene Form hat.

(Die Matrix zu $\widehat{\ell}_p$ hat die angegebene Form): $\left(\widehat{\ell}_p \right)^{\mathcal{M}}$ läßt sich in fünf Untermatrizen separieren (vier Spalten und die untere Reihe), für die wir die jeweilige Teilbehauptung zeigen.

Dazu benötigen wir den folgenden Zusammenhang:

$$(g) : \forall x \in \prod_{s \in Obj(\mathcal{S})} \mathcal{D}(s), m \in Mor(\mathcal{S}), b^m \in \prod_{m \in Mor(\mathcal{S})} \mathcal{D}(cod(m)), z \in \mathbb{N} \setminus \{0\}:$$

$$\widehat{\ell}_p(x)(b^m) = z \iff x \in \mathcal{D}(dom(m)) \wedge \mathcal{D}(m)(x)(b) = z$$

$$\forall x \in \prod_{s \in Obj(\mathcal{S})} \mathcal{D}(s), m \in Mor(\mathcal{S}), b^m \in \prod_{m \in Mor(\mathcal{S})} \mathcal{D}(cod(m)), z \in \mathbb{N} \setminus \{0\}:$$

$$\widehat{\ell}_p(x)(b^m) = z$$

$$\iff \ell_m \circ \widehat{\ell}_p(x)(b) = z \quad 5.2.17, (\text{d})$$

$$\iff \mathcal{D}(m) \circ \ell_{dom(m)}(x)(b) = z \quad (\text{a})$$

$$\iff x \in \mathcal{D}(\text{dom}(m)) \wedge \mathcal{D}(m)(x)(b) = z \quad 5.2.17, (c)$$

(Obere Zeile, erste Spalte):

$$\forall m \in \text{Mor}(\mathcal{S}), b_j \in B, b_y^m \in \prod_{m \in \text{Mor}(\mathcal{S})} \mathcal{D}(\text{cod}(m)) :$$

$$\implies \widehat{\ell}_p(b_j)(b_y^m) = 0 \quad \begin{array}{l} (g), b_j \in \mathcal{D}(\text{dom}(m)) \\ \iff m = \text{id}_s \end{array}$$

$$\implies \left(\widehat{\ell}_p \Big|_{B^\otimes} \left\{ b_1^{m_1}, \dots, b_{|B|}^{m_{n+1}} \right\}^\otimes \right)^\mathcal{M} = 0 \quad \begin{array}{l} b_y^d \notin \prod_{m \in \text{Mor}(\mathcal{S})} \mathcal{D}(\text{cod}(m)) , \\ (e) \end{array}$$

(Obere Zeile, zweite Spalte): Sei

$$\left(\widehat{\ell}_p \Big|_{\left\{ a_1^1, \dots, a_{|B|}^n \right\}^\otimes} \left\{ b_1^{m_1}, \dots, b_{|B|}^{m_{n+1}} \right\}^\otimes \right)^\mathcal{M} = (k_{u,v})$$

dann gilt aufgrund der Ordnung auf den Basiselementen (\rightarrow (e), 5.2.21) der Produkte:

$$\widehat{\ell}_p(a_j^i)(b_y^{m_x}) = k_{(x-1)*|B|+y, (i-1)*|B|+j} .$$

Daraus folgt:

$$\forall i, x \in \{1, \dots, n\}, j, y \in \{1, \dots, |B|\} :$$

$$\widehat{\ell}_p(a_j^i)(b_y^{m_x}) \neq 0 \implies i = x \quad (g), a_j^i \in \mathcal{D}(\text{dom}(m_i))$$

und

$$\widehat{\ell}_p(a_j^i)(b_y^{m_i}) = f_i(a_j^i)(b_y) = g_i^\otimes(a_j^i)(b_y) \quad \begin{array}{l} (g), a_j^i \in A_i \setminus \text{Kern}(f_i) \\ 5.2.21 \end{array}$$

$$\implies \widehat{\ell}_p(a_j^i)(b_y^{m_x}) = 1 \iff i = x \wedge j = y \quad g_i \text{ ist eine inj. Fkt.}$$

$$\widehat{\ell}_p(a_j^i)(b_y^{m_x}) = 0 \iff i \neq x \vee j \neq y \quad g_i(a_j^i) = b_j$$

$$\implies k_{u,v} = 1 \iff u = v \wedge k_{u,v} = 0 \iff u \neq v$$

$$\implies \left(\widehat{\ell}_p \Big|_{\left\{ a_1^1, \dots, a_{|B|}^n \right\}^\otimes} \left\{ b_1^{m_1}, \dots, b_{|B|}^{m_{n+1}} \right\}^\otimes \right)^\mathcal{M} = \begin{pmatrix} E_{n*|B|} \\ 0 \end{pmatrix} \quad \begin{array}{l} u \leq (n+1) * |B|, \\ v \leq n * |B| \end{array}$$

(Obere Zeile, dritte Spalte):

$$\forall i \in \{1, \dots, n+1\}, j, y \in \{1, \dots, |B|\} :$$

$$\widehat{\ell}_p(a_j^{n+1})(b_y^{m_x}) \neq 0 \implies x = n+1 \quad (g), a_j^i \in \mathcal{D}(\text{dom}(m_i))$$

und

$$\widehat{\ell}_p(a_j^{n+1})(b_y^{m_{n+1}}) = f_{n+1}(a_j^{n+1})(b_y) \quad (\text{g})$$

$$\implies \left(\widehat{\ell}_p \Big|_{A_{n+1}^{\otimes}} \left\{ b_1^{m_1}, \dots, b_{|B|}^{m_{n+1}} \right\}^{\otimes} \right)^{\mathcal{M}} = \left(\frac{0}{(f_{n+1})^{\mathcal{M}}} \right) \quad (\text{e})$$

(Obere Zeile, vierte Spalte):

$\forall i, x \in \{1, \dots, n+1\}, j, y \in \{|B|+1, \dots, |A_i|\}$:

$$\widehat{\ell}_p(a_j^i)(b_x^{m_x}) = 0, \quad i \neq x \quad (\text{g}), a_j^i \in \mathcal{D}(\text{dom}(m_i))$$

und

$$\widehat{\ell}_p(a_j^i)(b_x^{m_x}) = f_i(a_j^i)(b_x) = 0 \quad (\text{g}), (\text{e}), a_j^i \in \text{Kern}(f_i)$$

und

$\forall i \in \{n+2, \dots, m\}, j \in \{1, \dots, |A_i|\},$
 $y \in \{1, \dots, |B|\}, m \in \text{Mor}(\mathcal{S})$:

$$\widehat{\ell}_p(a_j^i)(b_y^m) = 0 \quad (\text{g}), a_j^i \in \mathcal{D}(\text{dom}(m))$$

$$\iff m = \text{id}_{s_i^{\otimes}}$$

$$\implies \left(\widehat{\ell}_p \Big|_{\left\{ a_{|B|+1}^1, \dots, a_{|A_m|}^m \right\}^{\otimes}} \left\{ b_1^{m_1}, \dots, b_{|B|}^{m_{n+1}} \right\}^{\otimes} \right)^{\mathcal{M}} = 0$$

$$b_y^{\text{id}} \notin \prod_{m \in \text{Mor}(\mathcal{S})} \mathcal{D}(\text{cod}(m))$$

(Untere Zeile): Sei

$$\left(\widehat{\ell}_p \Big|_{(B \uplus A_1 \uplus \dots \uplus A_m)^{\otimes}} \right)^{\mathcal{M}} = (k_{u,v}),$$

dann gilt:

$\forall x, y \in B \uplus A_1 \uplus \dots \uplus A_m$:

$$\widehat{\ell}_p(x)(y) = \ell_{\text{id}} \circ \widehat{\ell}_p(x)(y) = \text{id} \circ \ell_{\text{dom}(\text{id})}(x)(y) \quad (\text{d}), \ell_m(z)(y) > 0$$

$$\implies z = y \wedge m = \text{id}$$

$$\implies \widehat{\ell}_p(x)(y) = 1 \iff x = y \wedge \widehat{\ell}_p(x)(y) = 0 \iff x \neq y$$

$$\implies k_{u,v} = 1 \iff u = v \wedge k_{u,v} = 0 \iff u \neq v \quad (\text{e})$$

$$\implies \left(\widehat{\ell}_p \Big|_{(B \uplus A_1 \uplus \dots \uplus A_m)^{\otimes}} \right)^{\mathcal{M}} = E_{(n+1)*|B|+r}$$

Damit haben alle Submatrizen die behaupteten Eigenschaften, d.h. die Matrix zu $\widehat{\ell}_p$ hat die angegebene Form.

(Die Matrixgleichung gilt): Mit Hilfe der speziellen Form der Differenzmatrix

$$\left(\widehat{\ell}_p \right)^{\mathcal{M}} - \left(\widehat{\ell}_o \right)^{\mathcal{M}}$$

kann die Gültigkeit der Matrixgleichung, d.h.

$$\left(\begin{array}{c|c|c|c} \hline -E_{|B|} & & & \\ \hline \vdots & & & \\ \hline -E_{|B|} & & & \\ \hline \end{array} \middle| \begin{array}{c|c} \hline E_{n*|B|} & \\ \hline 0 & \\ \hline \end{array} \middle| \begin{array}{c|c} \hline 0 & \\ \hline (f_{n+1})^{\mathcal{M}} & \\ \hline \end{array} \middle| 0 \right) * \begin{pmatrix} \hline (f_{n+1})^{\mathcal{M}} & | & 0 \\ \hline \vdots & & \\ \hline (f_{n+1})^{\mathcal{M}} & | & 0 \\ \hline E_r \end{pmatrix} = 0$$

leicht nachgewiesen werden. Dazu zerlegen wir die Differenzmatrix vertikal in $n + 1$ Matrizen der Dimension $|B| \times ((n + 1) * |B| + r)$ und L vertikal in $n + 1$ identische Matrizen der Dimension $|B| \times r$ ergänzt um die $r \times r$ -Einheitsmatrix

$$\left(\widehat{\ell}_p \right)^{\mathcal{M}} - \left(\widehat{\ell}_o \right)^{\mathcal{M}} \stackrel{\text{def}}{=} \begin{pmatrix} \hline M_1 \\ \hline \vdots \\ \hline M_{n+1} \\ \hline \end{pmatrix}, \quad L \stackrel{\text{def}}{=} \begin{pmatrix} \hline L_1 \\ \hline \vdots \\ \hline L_{n+1} \\ \hline E_r \end{pmatrix} \quad \text{mit } L_i = \left((f_{n+1})^{\mathcal{M}} \mid 0 \right).$$

Dann gilt für die Multiplikation der M_i mit L ,

– im Fall $i = 1$:

$$M_1 * L = -E_{|B|} * L_1 + E_{|B|} * L_2 + 0 * L_3 + \cdots + 0 * E_r = 0$$

– im Fall $i = 2, \dots, n - 1$:

$$M_i * L = -E_{|B|} * L_1 + 0 * L_2 + \cdots + E_{|B|} * L_i + 0 * L_{i+1} + \cdots + 0 * E_r = 0$$

– im Fall $i = n$:

$$M_n * L = -E_{|B|} * L_1 + 0 * L_2 + \cdots + E_{|B|} * L_n + 0 * E_r = 0$$

– im Fall $i = n + 1$:

$$M_{n+1} * L = -E_{|B|} * L_1 + 0 * L_2 + \cdots + (f_{n+1})^{\mathcal{M}} * E_{|A_{n+1}|} + 0 * E_{r-|A_{n+1}|} = 0$$

und damit

$$\left(\left(\widehat{\ell}_p \right)^{\mathcal{M}} - \left(\widehat{\ell}_o \right)^{\mathcal{M}} \right) * L = \begin{pmatrix} \hline M_1 * L \\ \hline \vdots \\ \hline M_{n+1} * L \\ \hline \end{pmatrix} = 0$$

□

(Die übrigen Voraussetzungen von Lemma 5.2.19 sind erfüllt):

Die Mächtigkeit der Basis des linken Produkts in Diagramm (a) ist gleich $(n+1)*|B|+r$ und der Rang der Differenzmatrix entspricht ihrer Zeihlenanzahl¹⁷, also $(n+1)*|B|$. Daraus ergibt sich die Dimension des Lösungsraums für den Kern der Differenzmatrix zu:

$$((n+1)*|B|+r) - ((n+1)*|B|) = r$$

d.h. die $((n+1)*|B|+r) \times r$ -Matrix L hat die korrekten Dimensionen.

Für die Folge

$$1 \leq j_1 = (n+1)*|B|+1 \leq \dots \leq j_r = (n+1)*|B|+r$$

und $L = (l_{u,v})$ gilt: $\forall i, p \in \{1, \dots, r\} : l_{j_i, i} = 1$ und $l_{j_i, p} = 0$ für alle $p \neq i$.

Aus der Definition der Produkte (\rightarrow 5.2.17), $\text{cod}(\mathcal{D}) = \mathcal{AG}_{fm}$ und Diagramm (a) folgt: $\widehat{\ell}_p$ und $\widehat{\ell}_o$ sind monotone Morphismen in den freien abelschen Gruppen.

Die Matrixeinträge von L sind entweder von dem monotonen Morphismus f_{n+1} erzeugt, gleich Null oder stammen aus der Einheitsmatrix, d.h. sie sind in jedem Fall aus \mathbb{N} . \square

Damit sind alle Voraussetzungen für die Anwendung von Lemma 5.2.19 gegeben, d.h.

$$\left(E^\otimes, Mp \left(L, E^\otimes, \prod_{s \in \text{Obj}(\mathcal{S})} \mathcal{D}(s) \right) \right)$$

ist ein Equalizer zu $\widehat{\ell}_p$ und $\widehat{\ell}_o$ in \mathcal{AG}_{fm} . Mit Satz 5.1.19 folgt daraus unsere Behauptung.

\square

Beweis (5.2.23 Colimeseigenschaften):

(I):

Wir zeigen, daß die Annahme:

$$\exists c \in C, \forall s \in \text{Mor}(\mathcal{S}), a_s \in A_s : c_s(a_s)(c) = 0$$

zum Widerspruch führt. Dazu definieren wir die freie abelsche Gruppe \overline{C}^\otimes durch: $\overline{C} = C \cup \{\overline{c}\}$ und die Morphismenfamilie

$$(\overline{c}_s : A_s^\otimes \rightarrow \overline{C}^\otimes)_{s \in \text{Obj}(\mathcal{S})} \text{ durch: } \overline{c}_s \Big|_{C^\otimes} = c_s \text{ für alle } s \in \text{Mor}(\mathcal{S}).$$

Aufgrund unserer Eingangsannahme ist es leicht einzusehen, daß

$$((\overline{c}_s)_{s \in \text{Obj}(\mathcal{S})}, \overline{C}^\otimes)$$

¹⁷ohne die nur mit Nullen gefüllten Zeilen am Ende

eine Senke für \mathcal{D} in \mathcal{AG}_{fm} bildet. In diesem Fall darf es nur einen Morphismus

$$\hat{c}: C^{\otimes} \rightarrow \overline{C}^{\otimes} \text{ mit der Eigenschaft } \hat{c} \circ c_s = \overline{c}_s \text{ für alle } s \in \text{Obj}(\mathcal{S})$$

in \mathcal{AG}_{fm} geben.

Da das Bild von c unter \hat{c} durch diese Bedingung nicht festgelegt ist, existiert kein c_s in dessen Bild c liegt und es gibt mit dem neutralen Element und \overline{c} in \overline{C}^{\otimes} mindestens zwei Möglichkeiten c abzubilden ohne die Bedingung zu verletzen. Die Existenz mehrerer (unterschiedlicher) Morphismen mit der genannten Eigenschaft steht im Widerspruch zu der Annahme, daß $((c_s)_{s \in \text{Obj}(\mathcal{S})}, C^{\otimes})$ einen Colimes zu \mathcal{D} in \mathcal{AG}_{fm} bildet. \square

(II):

Wir zeigen, daß die Annahme:

$$\begin{aligned} \exists s, s' \in \text{Mor}(\mathcal{S}), a_s \in A_s, a_{s'} \in A_{s'}, c \in C: \\ c_s(a_s)(c) > 0 \wedge c_{s'}(a_{s'})(c) > 0 \wedge a_s \neq a_{s'} \end{aligned}$$

zum Widerspruch führt. Dazu definieren wir die freie abelsche Gruppe \overline{C}^{\otimes} durch: $\overline{C} = C \cup \{\overline{c}, \overline{c}'\}$ und die Morphismenfamilie

$$(\overline{c}_s: A_s^{\otimes} \rightarrow C^{\otimes})_{s \in \text{Obj}(\mathcal{S})}$$

für alle Objekte r aus \mathcal{S} , x aus A_r und y aus \overline{C}^{\otimes} durch:

$$\overline{c}_r(x)(y) = \begin{cases} 0 & ; (x \rightleftharpoons a_s \wedge y = c) \vee (x \not\rightleftharpoons a_s \wedge y = \overline{c}) \\ c_r(x)(c) & ; (x \rightleftharpoons a_s \wedge y = \overline{c}) \vee (x \not\rightleftharpoons a_s \wedge y = c) \\ c_r(x)(c) & ; y = \overline{c}' \\ \overline{c}_r(u)(y) + \overline{c}_r(v)(y) & ; x = u \oplus v \\ c_r(x)(y) & ; \text{sonst} \end{cases}$$

und zeigen, daß $((\overline{c}_s)_{s \in \text{Obj}(\mathcal{S})}, \overline{C}^{\otimes})$ eine Senke zu \mathcal{D} in \mathcal{AG}_{fm} bildet. Im zweiten Schritt leiten wir daraus einen Widerspruch ab.

$((\overline{c}_s)_{s \in \text{Obj}(\mathcal{S})}, \overline{C})$ ist eine Senke): Das die \overline{c}_s monotone Morphismen in den freien abelschen Gruppen sind, folgt direkt aus ihrer Definition. Verbleibt noch der Nachweis ihrer Verträglichkeit mit den Diagrammmorphismen, d.h. für alle $m: r \rightarrow r'$

$$\overline{c}_{r'} \circ \mathcal{D}(m) = \overline{c}_r .$$

Wir zeigen diese Eigenschaft für alle Basiselemente b_r aus A_r und alle Basiselemente d aus \overline{C} . Für alle anderen Elemente ergibt sie sich dann aus der Verträglichkeit der \overline{c}_s und $\mathcal{D}(m)$ mit der Gruppenoperation \oplus .

$$\overline{c}_{r'} \circ \mathcal{D}(m)(b_r)(d)$$

$$= \bar{c}_{r'} \left(\bigoplus_{b'_{r'} \in A_{r'}} \mathcal{D}(m)(b_r)(b'_{r'}) * b'_{r'} \right)(d)$$

$$\text{cod}(\mathcal{D}(m)) = \mathcal{AG}_{fm}$$

$$= \bigoplus_{b'_{r'} \in A_{r'}} \mathcal{D}(m)(b_r)(b'_{r'}) * \bar{c}_{r'}(b'_{r'})(d)$$

$$\bar{c}_{r'} \in \mathcal{AG}_{fm}$$

- Fallunterscheidung:
- $b_r \rightleftharpoons a_s \wedge d = c$
 - $b_r \rightleftharpoons a_s \wedge d = \bar{c}$
 - $d = \bar{c}'$
 - $d \neq c \wedge d \neq \bar{c} \wedge d \neq \bar{c}'$
 - $b_r \not\rightleftharpoons a_s \wedge d = c$
 - $b_r \not\rightleftharpoons a_s \wedge d = \bar{c}$

($b_r \rightleftharpoons a_s \wedge d = c$):

$$= \bigoplus_{b'_{r'} \in A_{r'}} \mathcal{D}(m)(b_r)(b'_{r'}) * 0 = 0 = \bar{c}_r(b_r)(d)$$

$$b'_{r'} \rightleftharpoons b_r \rightleftharpoons a_s \\ \implies b'_{r'} \rightleftharpoons a_s, d = c$$

($b_r \rightleftharpoons a_s \wedge d = \bar{c}$):

$$= \bigoplus_{b'_{r'} \in A_{r'}} \mathcal{D}(m)(b_r)(b'_{r'}) * c_{r'}(b'_{r'})(c)$$

$$b'_{r'} \rightleftharpoons b_r \rightleftharpoons a_s \\ \implies b'_{r'} \rightleftharpoons a_s, d = \bar{c}$$

$$= c_{r'} \circ \mathcal{D}(m)(b_r)(c) = c_r(b_r)(c) = \bar{c}_r(b_r)(d)$$

Senke zu \mathcal{D}

$$b_r \rightleftharpoons a_s \wedge d = \bar{c}$$

($d = \bar{c}'$):

$$= \bigoplus_{b'_{r'} \in A_{r'}} \mathcal{D}(m)(b_r)(b'_{r'}) * c_{r'}(b'_{r'})(c)$$

Def. $\bar{c}_{r'}$

$$= c_{r'} \circ \mathcal{D}(m)(b_r)(c) = c_r(b_r)(c) = \bar{c}_r(b_r)(d)$$

Senke zu \mathcal{D} , Def. \bar{c}_r

($d \neq c \wedge d \neq \bar{c}$):

$$= \bigoplus_{b'_{r'} \in A_{r'}} \mathcal{D}(m)(b_r)(b'_{r'}) * c_{r'}(b'_{r'})(d)$$

$$d \neq c \wedge d \neq \bar{c}$$

$$= c_{r'} \circ \mathcal{D}(m)(b_r)(d) = c_r(b_r)(d) = \bar{c}_r(b_r)(d)$$

$$d \neq c \wedge d \neq \bar{c}$$

($b_r \not\rightleftharpoons a_s \wedge d = c$):

$$= \bigoplus_{b'_{r'} \in A_{r'}} \mathcal{D}(m)(b_r)(b'_{r'}) * c_{r'}(b'_{r'})(d)$$

$$b'_{r'} \rightleftharpoons b_r \wedge b_r \not\rightleftharpoons a_s \\ \implies b'_{r'} \not\rightleftharpoons a_s, d = c$$

$$= c_{r'} \circ \mathcal{D}(m)(b_r)(c) = c_r(b_r)(c) = \bar{c}_r(b_r)(d)$$

Senke zu \mathcal{D}

($b_r \not\rightleftharpoons a_s \wedge d = \bar{c}$):

$$= \bigoplus_{b'_{r'} \in A_{r'}} \mathcal{D}(m)(b_r)(b'_{r'}) * 0 = 0 = \bar{c}_r(b_r)(d)$$

$$b'_{r'} \rightleftharpoons b_r \wedge b_r \not\rightleftharpoons a_s \\ \implies b'_{r'} \not\rightleftharpoons a_s, d = \bar{c}$$

(Senke \implies Widerspruch): Aus der neuen Senke folgt die Existenz und Eindeutigkeit eines Morphismus

$$\widehat{c}: C^{\otimes} \rightarrow \overline{C}^{\otimes} \text{ mit der Eigenschaft } \widehat{c} \circ c_s = \overline{c}_s \text{ f\u00fcr alle } s \text{ aus } \mathcal{S}.$$

Zusammen mit der Definition der \overline{c}_s l\u00e4\u00dft sich daraus ableiten:

$$\begin{aligned} \overline{c}_s(a_s)(c) = 0 &\implies \widehat{c} \circ c_s(a_s)(c) = 0 \implies \widehat{c}(c)(c) = 0 \\ \overline{c}_{s'}(a'_{s'}) (\overline{c}) = 0 &\implies \widehat{c} \circ c_{s'}(a'_{s'}) (\overline{c}) = 0 \implies \widehat{c}(c)(\overline{c}) = 0 \quad . \end{aligned}$$

Wir definieren im folgenden mit f und g zwei weitere von einander verschiedene \mathcal{AG}_{fm} -Morphismen und zeigen, da\u00df sie die gleiche Eigenschaft wie \widehat{c} haben. Wegen der Eindeutigkeit von \widehat{c} ergibt sich daraus der Widerspruch.

(($f \circ c_s = \overline{c}_s$)): Sei f ein Morphismus in \mathcal{AG}_{fm} der auf den Elementen aus C und \overline{C} folgenderma\u00dfen definiert ist:

$$f(x)(y) = \begin{cases} 0 & ; x = c \wedge y = \overline{c}' \\ \widehat{c}(x)(c) + \widehat{c}(x)(\overline{c}) & ; x \neq c \wedge y = \overline{c}' \\ \widehat{c}(x)(y) & ; y \neq \overline{c}' \end{cases}$$

Wir zeigen die Eigenschaft zun\u00e4chst f\u00fcr alle Objekte s aus \mathcal{S} , Elemente x aus der Basis von $\mathcal{D}(s)$ bzgl. des Elements \overline{c}' .

$$\begin{aligned} &f \circ c_s(x)(\overline{c}') \\ &= \left(\sum_{c' \in C \setminus \{c\}} c_s(x)(c') * f(c')(\overline{c}') \right) + c_s(x)(c) * f(c)(\overline{c}') \\ &= \sum_{c' \in C \setminus \{c\}} c_s(x)(c') * (\widehat{c}(x)(c) + \widehat{c}(x)(\overline{c})) && \text{Def. f, } f(c)(\overline{c}') = 0 \\ &= \sum_{c' \in C} c_s(x)(c') * (\widehat{c}(x)(c) + \widehat{c}(x)(\overline{c})) && \widehat{c}(c)(c) = 0, \widehat{c}(c)(\overline{c}) = 0 \\ &= \widehat{c} \circ c_s(x)(c) + \widehat{c} \circ c_s(x)(\overline{c}) \\ &= \overline{c}_s(x)(c) + \overline{c}_s(x)(\overline{c}) && \text{Eigensch. } \widehat{c} \\ &= c_s(x)(c) && \text{Def. } \overline{c}_s \\ &= \overline{c}_s(x)(\overline{c}') && \text{Def. } \overline{c}_s \end{aligned}$$

F\u00fcr alle \u00fcrbrigen Elemente y aus \overline{C} , Objekte s aus \mathcal{S} und x aus der Basis von $\mathcal{D}(s)$ folgt aus der Definition von f : $f \circ c_s(x)(y) = \widehat{c} \circ c_s(x)(y) = \overline{c}_s(x)(y)$.

(($g \circ c_s = \overline{c}_s$)): Sei g ein Morphismus in \mathcal{AG}_{fm} der auf den Elementen aus C und \overline{C} folgenderma\u00dfen definiert ist:

$$g(x)(y) = \begin{cases} 0 & ; x \neq c \wedge y = \overline{c}' \\ 1 & ; x = c \wedge y = \overline{c}' \\ \widehat{c}(x)(y) & ; y \neq \overline{c}' \end{cases}$$

$\forall s \in \text{Obj}(\mathcal{S}), x \in \text{Base}(\mathcal{D}(s)):$ Def g

$$g \circ c_s(x)(\bar{c}') = c_s(x)(c) = \bar{c}_s(x)(\bar{c}')$$

und

$\forall s \in \text{Obj}(\mathcal{S}), x \in \text{Base}(\mathcal{D}(s)), y \in \bar{C}:$ Def g

$$g \circ c_s(x)(\bar{c}') = \hat{c} \circ c_s(x)(y) = \bar{c}_s(x)(y)$$

□

□

Beweis (5.3.3 Colimites für initiale Modelle):

Im folgenden bezeichnen wir aus Gründen der Übersichtlichkeit den von einem \mathcal{C} -Morphismus h generierten Funktor $\mathcal{M}(h)$ mit \mathcal{V}_h und für alle $s, m: s \rightarrow s'$ aus \mathcal{S} , den jeweils vom initialen Objekt $I_{\mathcal{D}(s)}$ aus eindeutig existierenden $\mathcal{M}(\mathcal{D}(s))$ -Morphismus mit

$$\text{eval}_s: I_{\mathcal{D}(s)} \rightarrow \mathcal{V}_{c_s}(I_C) \quad \text{resp. mit} \quad \text{eval}_m: I_{\mathcal{D}(s)} \rightarrow \mathcal{V}_{\mathcal{D}(m)}(I_{\mathcal{D}(s')}).$$

(Kommutativität): Für alle \mathcal{S} -Morphismen $m: s \rightarrow s'$ gilt:

$$\mathcal{V}_{\mathcal{D}(m)}(\text{eval}_{s'}: I_{\mathcal{D}(s')} \rightarrow \mathcal{V}_{c_{s'}}(I_C)) = \mathcal{V}_{\mathcal{D}(m)}(\text{eval}_{s'}): \mathcal{V}_{\mathcal{D}(m)}(I_{\mathcal{D}(s')}) \rightarrow \mathcal{V}_{c_s}(I_C) \quad {}^{18}$$

mit der Initialität von $I_{\mathcal{D}(s)}$ in $\mathcal{D}(s)$ folgt daraus:

$$(a): \quad \mathcal{V}_{\mathcal{D}(m)}(\text{eval}_{s'}) \circ \text{eval}_m = \text{eval}_s \quad .$$

Damit können wir zeigen:

$$\begin{aligned} (c_{s'}, \text{eval}_{s'}) \circ \mathcal{G}(m) &= (c_{s'}, \text{eval}_{s'}) \circ (\mathcal{D}(m), \text{eval}_m) \\ &= (c_{s'} \circ \mathcal{D}(m), \mathcal{V}_{\mathcal{D}(m)}(\text{eval}_{s'}) \circ \text{eval}_m) \\ &= (c_s, \text{eval}_s) \end{aligned} \quad \text{Eigens. Colim. v. } \mathcal{D}, (a)$$

(Couniverselle Eigenschaft): Sei

$$((\bar{c}_s, \bar{c}_s^M)_{s \in \mathcal{S}}, M) \text{ mit } M \in \mathcal{M}(\bar{C})$$

eine weitere Senke für \mathcal{G} in $\mathcal{G}\mathcal{S}\mathcal{F}$. Dann bildet

$$((\bar{c}_s)_{s \in \mathcal{S}}, \bar{C})$$

¹⁸ $\mathcal{V}_{\mathcal{D}(m)} \circ \mathcal{V}_{c_{s'}} = \mathcal{V}_{\mathcal{D}(m) \circ c_{s'}} = \mathcal{V}_{c_s}$; die Indices sind Morphismen in \mathcal{C}^{op} !

eine Senke zu \mathcal{D} in \mathcal{SF} und es existiert eindeutig ein \mathcal{C} -Morphismus

$$\widehat{c}: C \rightarrow \overline{C} \text{ mit der Eigenschaft } \forall s \in \text{Obj}(\mathcal{S}): \widehat{c} \circ c_s = \overline{c}_s \quad .$$

Daraus konstruieren wir den \mathcal{GSF} -Morphismus

$$(\widehat{c}, \text{eval}_C): I_C \rightarrow M$$

für dessen zweite Komponente bzgl. aller \mathcal{S} -Objekte s gilt:

$$\mathcal{V}_{c_s}(\text{eval}_C: I_C \rightarrow \mathcal{V}_{\widehat{c}}(M)) = \mathcal{V}_{c_s}(\text{eval}_C): \mathcal{V}_{c_s}(I_C) \rightarrow \mathcal{V}_{\overline{c}_s}(M) \quad ^{19}.$$

Mit der Initialität von $I_{\mathcal{D}(s)}$ in $\mathcal{D}(s)$ folgt daraus:

$$(b): \mathcal{V}_{c_s}(\text{eval}_C) \circ \text{eval}_s = \overline{c}_s^M: I_{\mathcal{D}(s)} \rightarrow \mathcal{V}_{\overline{c}_s}(M)$$

und damit:

$$\begin{aligned} & (\widehat{c}, \text{eval}_C) \circ (c_s, \text{eval}_s) \\ &= (\widehat{c} \circ c_s, \mathcal{V}_{c_s}(\text{eval}_C) \circ \text{eval}_s) \\ &= (\overline{c}_s, \overline{c}_s^M) \qquad \text{Eigens. von } \widehat{c}, (b) \end{aligned}$$

(Eindeutigkeit): \widehat{c} ist eindeutig mit der Eigenschaft: „Verträglichkeit mit allen \overline{c}_s “ und eval_C ist aufgrund der Initialität von I_C eindeutig. Damit ist auch $(\widehat{c}, \text{eval}_C)$ eindeutig.

□

¹⁹ $\mathcal{V}_{c_s} \circ \mathcal{V}_{\widehat{c}} = \mathcal{V}_{c_s \circ \widehat{c}} = \mathcal{V}_{\overline{c}_s}$; die Indices sind Morphismen in \mathcal{C}^{op} !

Beweis (5.3.4 Pushouts und freie Konstruktionen):

(Freie Konstruktion \implies Pushout):

Wir nehmen an, daß (u_A, C) eine \mathcal{M}_h -universelle Abbildung zu A (\rightarrow 5.1.10) ist, d.h. zu jedem $\mathcal{M}(E)$ -Morphismus f existiert eindeutig ein $\mathcal{M}(D)$ -Morphismus f' der das nachfolgende Diagramm kommutativ ergänzt

$$\begin{array}{ccc}
 A & \xrightarrow{u_A} & \mathcal{M}_h(C) & & C \\
 & \searrow f & \downarrow \mathcal{M}_h(f') & & \downarrow f' \\
 & & \mathcal{M}_h(B) & & B
 \end{array}$$

und zeigen, daß in diesem Fall das Diagramm

$$\begin{array}{ccc}
 I_E & \xrightarrow{(h, h^m)} & I_D \\
 (id_E, eval_A) \downarrow & \curvearrowright & \downarrow (id_D, eval_C) \\
 A & \xrightarrow{(h, u_A)} & C
 \end{array}$$

einen Pushout in \mathcal{GSF} bildet.

(Kommutativität): Unsere Annahme, I_E ist das initiale Objekt in $\mathcal{M}(E)$, impliziert für alle Morphismen k, k' von I_E in ein $\mathcal{M}(E)$ -Objekt C : $k = k'$. Damit können wir zeigen:

$$\begin{aligned}
 & (id_D, eval_C) \circ (h, h^m) \\
 &= (h, \mathcal{M}_h(eval_C) \circ h^m) \\
 &= (h, u_A \circ eval_A) && I_E \text{ ist initial} \\
 &= (h, u_A) \circ (id_E, eval_A)
 \end{aligned}$$

(couniverselle Eigenschaft): Sei

$$((\bar{c}_s, \bar{c}_s^M)_{s \in \mathcal{S}}, \bar{C})$$

eine weitere Senke mit $\bar{C} \in \mathcal{M}(G)$ und $G \in \mathcal{SF}$. Wir zeigen, daß in diesem Fall genau ein \mathcal{GSF} -Morphismus

$$\hat{c}: C \rightarrow \bar{C}$$

existiert, mit der Eigenschaft

$$\hat{c} \circ (h, u_A) = (\bar{c}_1, \bar{c}_1^M) \quad \text{und} \quad \hat{c} \circ (id_D, eval_C) = (\bar{c}_2, \bar{c}_2^M).$$

Aus der neuen Senke folgt

$$(\bar{c}_1, \bar{c}_1^M) \circ (id_E, eval_A) = (\bar{c}_2, \bar{c}_2^M) \circ (h, h^m) \implies \bar{c}_1 = \bar{c}_2 \circ h$$

und damit:

$$\mathcal{M}_{\bar{c}_1}(\bar{C}) = \mathcal{M}_{\bar{c}_2 \circ h}(\bar{C}) = \mathcal{M}_h(\mathcal{M}_{\bar{c}_2}(\bar{C})) \implies \bar{c}_1^M: A \rightarrow \mathcal{M}_h(\mathcal{M}_{\bar{c}_2}(\bar{C}))^{20} .$$

Wegen der universellen Eigenschaft von (u_A, C) existiert in diesem Fall ein $\mathcal{M}(D)$ -Morphismus

$$\hat{c}^m: C \rightarrow \mathcal{M}_{\bar{c}_2}(\bar{C})$$

der das Diagramm

$$(a): \begin{array}{ccc} A & \xrightarrow{u_A} & \mathcal{M}_h(C) & & C \\ & \searrow \bar{c}_1^M & \downarrow \mathcal{M}_h(\hat{c}^m) & & \downarrow \hat{c}^m \\ & & \mathcal{M}_h(\mathcal{M}_{\bar{c}_2}(\bar{C})) & & \mathcal{M}_{\bar{c}_2}(\bar{C}) \end{array}$$

kommutativ ergänzt. Daraus läßt sich der $\mathcal{G}\mathcal{S}\mathcal{F}$ -Morphismus

$$\hat{c} = (\bar{c}_2, \hat{c}^m): C \rightarrow \bar{C}$$

konstruieren, mit der Eigenschaft:

$$(\bar{c}_2, \hat{c}^m) \circ (h, u_A) = (\bar{c}_2 \circ h, \mathcal{M}_h(\hat{c}^m) \circ u_A) = (\bar{c}_1, \bar{c}_1^M) \quad (a)$$

und

$$(\bar{c}_2, \hat{c}^m) \circ (id_D, eval_C) = (\bar{c}_2, \hat{c}^m \circ eval_C) = (\bar{c}_2, \bar{c}_2^M) \quad I_D \text{ ist initial}$$

(Eindeutigkeit): Die Eindeutigkeit von \hat{c} in $\mathcal{G}\mathcal{S}\mathcal{F}$ folgt aus $\bar{c}_1 = \bar{c}_2 \circ h$ und der Eindeutigkeit von \hat{c}^m in (a). □

(Pushout \implies freie Konstruktion):

Es muß zu jedem $\mathcal{M}(E)$ -Morphismus f die Existenz und Eindeutigkeit eines $\mathcal{M}(D)$ -Morphismus f' nachgewiesen werden, der das Diagramm

$$\begin{array}{ccc} A & \xrightarrow{u_A} & \mathcal{M}_h(C) & & C \\ & \searrow f & \downarrow \mathcal{M}_h(f') & & \downarrow f' \\ & & \mathcal{M}_h(\bar{C}) & & \bar{C} \end{array}$$

²⁰ $\bar{c}_2 \circ h = h \bullet \bar{c}_2$, wobei \bullet die Komposition in C^{op} bezeichnet (\rightarrow 5.1.7)

kommutativ ergänzt.

Sei $f: A \rightarrow \mathcal{M}_h(\overline{C})$ gegeben. Wir konstruieren daraus den generalisierten Morphismus

$$(h, f): A \rightarrow \overline{C} \quad .$$

Aufgrund der Initialität von I_E ergänzt f das nachfolgende Diagramm kommutativ.

$$\begin{array}{ccc} I_E & \xrightarrow{(h, h^m)} & I_D \\ (id_E, eval_A) \downarrow & \circlearrowright & \downarrow (id_D, eval_{\overline{C}}) \\ A & \xrightarrow{(h, f)} & \overline{C} \end{array}$$

Damit bildet $((h, f), (id_D, eval_{\overline{C}}), \overline{C})$ eine weitere Senke zum Pushout. Wegen dessen couniversellen Eigenschaft existiert in diesem Fall genau ein \mathcal{GSF} -Morphismus

$$(id_D, f'): C \rightarrow \overline{C} \text{ mit } (h, f) = (id_D, f') \circ (h, u_A).$$

Daraus ergibt sich der gesuchte $\mathcal{M}(D)$ -Morphismus

$$f': C \rightarrow \overline{C} \text{ mit } f = \mathcal{M}_h(f') \circ u_A.$$

(Eindeutigkeit): Die Eindeutigkeit von f' ergibt sich aus der Eindeutigkeit von (id_D, f') mit der Eigenschaft $(h, f) = (id_D, f') \circ (h, u_A)$. \square

\square

Beweis (5.3.5 Konstruktion von Coequalizern):

(Verträglichkeit):

$$\begin{aligned} & (c, \mathcal{M}_c(d) \circ u_{A'}) \circ (h_1, f_1) \\ &= (c \circ h_1, \mathcal{M}_{h_1}(\mathcal{M}_c(d) \circ u_{A'}) \circ f_1) \\ &= (k, \mathcal{M}_k(d) \circ \mathcal{M}_{h_1}(u_{A'}) \circ f_1) && k = c \circ h_1 \\ &= (k, \mathcal{M}_k(d) \circ \mathcal{M}_k(g_1) \circ u_A) && 5.3.5 \\ &= (k, \mathcal{M}_k(d \circ g_1) \circ u_A) \\ &= (k, \mathcal{M}_k(d \circ g_2) \circ u_A) && \text{Coequalizer } d \\ &= (k, \mathcal{M}_k(d) \circ \mathcal{M}_k(g_2) \circ u_A) \\ &= (k, \mathcal{M}_k(d) \circ \mathcal{M}_{h_2}(u_{A'}) \circ f_2) && 5.3.5 \\ &= (c \circ h_2, \mathcal{M}_{h_2}(\mathcal{M}_c(d) \circ u_{A'}) \circ f_2) && k = c \circ h_2 \end{aligned}$$

$$= (c, \mathcal{M}_c(d) \circ u_{A'}) \circ (h_2, f_2)$$

(Couniverselle Eigenschaft):

Sei $(\bar{c}, \bar{d}): A' \rightarrow \bar{D}$ mit $(\bar{c}, \bar{d}) \circ (h_1, f_1) = (\bar{c}, \bar{d}) \circ (h_2, f_2)$ ein generalisierter Morphismus in \mathcal{GSF} . Wir zeigen, daß unter diesen Umständen ein generalisierter Morphismus

$$(\hat{c}, \hat{d}): D \rightarrow \bar{D}$$

existiert, der mit der Eigenschaft $(\hat{c}, \hat{d}) \circ (c, d) = (\bar{c}, \bar{d})$ bis auf Isomorphie eindeutig ist.

Aus der Eigenschaft von (\bar{c}, \bar{d}) folgt: $\bar{c} \circ h_1 = \bar{c} \circ h_2$ und damit Aufgrund der co-universellen Eigenschaft des Equalizers (c, \mathcal{C}) die Existenz und Eindeutigkeit eines Morphismus

$$\hat{c}: C \rightarrow \bar{C} \text{ mit } \hat{c} \circ c = \bar{c}.$$

Aus dem Modellmorphismus \bar{d} und den universellen Abbildungen, ergeben sich die folgenden kommutativen Diagramme:

$$(a): \begin{array}{ccc} A & \xrightarrow{u_A} & \mathcal{M}_k(B) \\ & \searrow \mathcal{M}_{h_i}(\bar{d}) \circ f_i & \downarrow \mathcal{M}_k(q) \\ & & \mathcal{M}_k(\mathcal{M}_{\hat{c}}(\bar{D})) \end{array} \quad \begin{array}{ccc} B & & \\ & \downarrow q & \\ & \mathcal{M}_{\hat{c}}(\bar{D}) & \end{array} \quad i = 1, 2$$

$$(b): \begin{array}{ccc} A' & \xrightarrow{u_{A'}} & \mathcal{M}_c(B') \\ & \searrow \bar{d} & \downarrow \mathcal{M}_c(r) \\ & & \mathcal{M}_c(\mathcal{M}_{\hat{c}}(\bar{D})) \end{array} \quad \begin{array}{ccc} B' & & \\ & \downarrow r & \\ & \mathcal{M}_{\hat{c}}(\bar{D}) & \end{array}$$

Daraus folgt:

$$\begin{aligned} & \mathcal{M}_k(r \circ g_i) \circ u_A \\ &= \mathcal{M}_k(r) \circ \mathcal{M}_k(g_i) \circ u_A \\ &= \mathcal{M}_k(r) \circ \mathcal{M}_{h_i}(u_{A'}) \circ f_i && 5.3.5 \\ &= \mathcal{M}_{h_i}(\mathcal{M}_c(r) \circ u_{A'}) \circ f_i && k = c \circ h_i \\ &= \mathcal{M}_{h_i}(\bar{d}) \circ f_i && (b) \\ \implies & r \circ g_1 = q = r \circ g_2 && (a) \end{aligned}$$

Mit der Eindeutigkeit von c in Bezug auf die Eigenschaft $c \circ g_1 = c \circ g_2$, folgt daraus die Existenz und Eindeutigkeit eines Morphismus

$$\hat{d}: C \rightarrow \mathcal{M}_{\hat{c}}(\bar{D}) \text{ mit } \hat{d} \circ d = r.$$

Kombiniert mit \hat{c} ergibt sich daraus der generalisierte Morphismus

$$(\hat{c}, \hat{d}): D \rightarrow \bar{D}.$$

für den wir zeigen: $(\widehat{c} \circ \widehat{d}) \circ (c, \mathcal{M}_c(d) \circ u_{A'}) = (\bar{c}, \bar{d})$.

$$\begin{aligned}
& (\widehat{c}, \widehat{d}) \circ (c, \mathcal{M}_c(d) \circ u_{A'}) \\
&= (\widehat{c} \circ \widehat{d}, \mathcal{M}_c(\widehat{d}) \circ \mathcal{M}_c(d) \circ u_{A'}) \\
&= (\bar{c}, \mathcal{M}_c(\widehat{d} \circ d) \circ u_{A'}) && \text{Eigensch. v. } \widehat{c} \\
&= (\bar{c}, \mathcal{M}_c(r) \circ u_{A'}) && \text{Eigensch. v. } \widehat{d} \\
&= (\bar{c}, \bar{d}) && \text{(b)}
\end{aligned}$$

Die Eindeutigkeit von \widehat{c} folgt direkt aus unserer Annahme, daß c ein Equalizer für g_1, g_2 bildet. Angenommen es gäbe einen Modellmorphismus

$$\tilde{d}: D \rightarrow \mathcal{M}_{\widehat{c}}(\bar{D}) \quad \text{mit} \quad (\widehat{c} \circ \tilde{d}) \circ (c, \mathcal{M}_c(d) \circ u_{A'}) = (\bar{c}, \bar{d})$$

dann folgt daraus:

$$\begin{aligned}
& \implies \mathcal{M}_c(\tilde{d} \circ d) \circ u_{A'} = \bar{d} \\
& \implies \tilde{d} \circ d = r && \text{Eindeutigk. v. } r \\
& \implies \widehat{d} = \tilde{d} && \text{Eindeutigk. v. } \widehat{d}
\end{aligned}$$

□

Beweise zu Kapitel 6

Beweis (6.1.3 Coprodukte in $SI\mathcal{G}$):

Wir notieren im weiteren das Coprodukt $\coprod_{s \in \text{Obj}(\mathcal{S})} SO_s$ durch SO_c und das Paar (SO_c, OP_c) durch Sig_c .

Es muß gezeigt werden:

- $((c_s^{SO}, c_s^{OP})_{s \in \text{Obj}(\mathcal{S})}, Sig_c)$ ist eine Senke und
- diese Senke ist Eindeutig.

$((c_s^{SO}, c_s^{OP})_{s \in \text{Obj}(\mathcal{S})}, Sig_c)$ ist eine Senke): (SO_c, OP_c) ist eine Signatur. c_s^{SO} und c_s^{OP} sind für alle Objekte s aus \mathcal{S} Abbildungen von der jeweiligen Komponente des Diagrammobjekts in die entsprechende Komponente des Colimesobjektes. c_s^{OP} ist darüber hinaus per Definition mit den Sorten der Operationen verträglich. Damit bilden die Paare (c_s^{SO}, c_s^{OP}) Signaturmorphisamen von den Diagrammobjekten in das Colimesobjekt. Da es sich bei der Domäne des Diagrammfunktors um eine diskrete Kategorie handelt, folgt daraus unsere Behauptung.

(Eindeutigkeit): Sei $((\bar{c}_s^{SO}, \bar{c}_s^{OP})_{s \in \text{Obj}(\mathcal{S})}, \text{Sig}_{\bar{c}})$ eine weitere Senke für \mathcal{D} . Wir zeigen, daß in diesem Fall ein eindeutiger Signaturmorphismus

$$(\hat{c}^{SO}, \hat{c}^{OP}): \text{Sig}_c \rightarrow \text{Sig}_{\bar{c}} \quad \text{mit} \quad (\hat{c}^{SO}, \hat{c}^{OP}) \circ (c_s^{SO}, c_s^{OP}) = (\bar{c}_s^{SO}, \bar{c}_s^{OP})$$

existiert.

Unter der Annahme $\text{Sig}_{\bar{c}} = (SO_{\bar{c}}, OP_{\bar{c}})$ folgt aus der Eindeutigkeit des Coprodukts der Sorten die Existenz und Eindeutigkeit einer Abbildung

$$\hat{c}^{SO}: SO_c \rightarrow SO_{\bar{c}} \quad \text{mit} \quad \hat{c}^{SO} \circ c_s^{SO} = \bar{c}_s^{SO}$$

für alle Objekte s in \mathcal{S} .

Da zu jeder Operation op in OP_c eindeutig ein Objekt s aus \mathcal{S} existiert mit $op \in OP_s$ und $c_s^{OP}(op) = op$, läßt sich die Operationsabbildung $\hat{c}^{OP}: OP_c \rightarrow OP_{\bar{c}}$ wie folgt konstruieren

$$\hat{c}^{OP}(op) \stackrel{\text{def}}{=} \bar{c}_s^{OP}(op): \bar{c}_s^{SO}(so_1), \dots, \bar{c}_s^{SO}(so_n) \rightarrow \bar{c}_s^{SO}(so)$$

für alle Operationen $op = c_s^{OP}(op): c_s^{SO}(so_1), \dots, c_s^{SO}(so_n) \rightarrow c_s^{SO}(so)$ aus OP_c . Aus

$$\begin{aligned} \hat{c}^{OP}(op) &= \bar{c}_s^{OP}(op): \bar{c}_s^{SO}(so_1), \dots, \bar{c}_s^{SO}(so_n) \rightarrow \bar{c}_s^{SO}(so) \\ &= \bar{c}_s^{OP}(op): \hat{c}^{SO}(c_s^{SO}(so_1)), \dots, \hat{c}^{SO}(c_s^{SO}(so_n)) \rightarrow \hat{c}^{SO}(c_s^{SO}(so)) \end{aligned}$$

für alle Operationen op in OP_c folgt die Verträglichkeit von \hat{c}^{OP} mit den Sorten der Operationen. Damit ist $(\hat{c}^{SO}, \hat{c}^{OP})$ ein Signaturmorphismus von Sig_c nach $\text{Sig}_{\bar{c}}$.

Für alle Objekte s in \mathcal{S} und alle Operationen op in OP_s gilt:

$$\hat{c}^{OP}(c_s^{OP}(op)) = \hat{c}^{OP}(op) = \bar{c}_s^{OP}(op)$$

und damit $(\hat{c}^{SO}, \hat{c}^{OP}) \circ (c_s^{SO}, c_s^{OP}) = (\hat{c}^{SO} \circ c_s^{SO}, \hat{c}^{OP} \circ c_s^{OP}) = (\bar{c}_s^{SO}, \bar{c}_s^{OP})$.

Sei $(\tilde{c}^{SO}, \tilde{c}^{OP}): \text{Sig}_c \rightarrow \text{Sig}_{\bar{c}}$ ein weiterer Signaturmorphimus mit der Eigenschaft

$$(\tilde{c}^{SO}, \tilde{c}^{OP}) \circ (c_s^{SO}, c_s^{OP}) = (\bar{c}_s^{SO}, \bar{c}_s^{OP})$$

für alle Objekte s in \mathcal{S} . Wir zeigen, daß in diesem Fall $(\tilde{c}^{SO}, \tilde{c}^{OP}) = (\hat{c}^{SO}, \hat{c}^{OP})$ gilt.

Aus der komponentweisen Komposition der Signaturmorphismen und der Eindeutigkeit von \hat{c}^{SO} als Morphismus vom Colimesobjekt SO_s nach SO_c folgt $\tilde{c}^{SO} = \hat{c}^{SO}$.

Darüber hinaus folgt aus der Kompatibilität von $(\tilde{c}^{SO}, \tilde{c}^{OP})$ mit den Diagrammmorphismen, der Konstruktion von OP_c und der Definition der Abbildungen c_s^{OP} für alle Operationen op aus OP_c :

$$\exists s \in \text{Obj}(\mathcal{S}): \tilde{c}^{OP}(op) = \tilde{c}^{OP}(c_s^{OP}(op)) = \bar{c}_s^{OP}(op) = \hat{c}^{OP}(op)$$

Damit gilt $\tilde{c}^{OP} = \hat{c}^{OP}$ und zusammen mit der zuvor gezeigten Gleichheit der Sortenabbildungen gilt $(\tilde{c}^{SO}, \tilde{c}^{OP}) = (\hat{c}^{SO}, \hat{c}^{OP})$.

□

Beweis (6.1.4 Coequalizer in STG):

Es muß gezeigt werden:

- $((c, d), (SO, OP))$ ist eine Senke und
- diese Senke ist eindeutig.

$((c, d), (SO, OP))$ ist eine Senke: (SO, OP) ist eine Signatur. c und d sind Abbildungen von $SO_{s'}$ bzw. $OP_{s'}$ nach SO bzw. OP . d ist darüber hinaus per Definition mit den Sorten der Operationen verträglich. Damit bildet (c, d) einen Signaturmorphismus von $\mathcal{D}(s')$ nach (SO, OP) . Die Verträglichkeit von (c, d) mit den Diagrammorphismen folgt aus der komponentweisen Komposition der Signaturmorphismen und den Eigenschaften der beiden Coequalizer.

(Eindeutigkeit): Sei $((\bar{c}^{SO}, \bar{c}^{OP}), Sig_{\bar{c}})$ eine weitere Senke für \mathcal{D} . Wir zeigen, daß in diesem Fall ein eindeutiger Signaturmorphismus

$$(\hat{c}^{SO}, \hat{c}^{OP}): Sig_c \rightarrow Sig_{\bar{c}} \quad \text{mit} \quad (\hat{c}^{SO}, \hat{c}^{OP}) \circ (c, d) = (\bar{c}^{SO}, \bar{c}^{OP})$$

existiert.

Unter der Annahme $Sig_{\bar{c}} = (SO_{\bar{c}}, OP_{\bar{c}})$ folgt aus der Eindeutigkeit der Coequalizer die Existenz und Eindeutigkeit einer Abbildung

$$\hat{c}^{SO}: SO \rightarrow SO_{\bar{c}} \quad \text{mit} \quad \hat{c}^{SO} \circ c = \bar{c}^{SO}$$

und einer Abbildung

$$\alpha: C \rightarrow OP_{\bar{c}} \quad \text{mit} \quad \alpha \circ d = \bar{c}^{OP}.$$

Da die Menge C und die Operationsfamilie OP von ihren Elementen her gleich sind (d ist als Morphis in einem Coequalizer der Kategorie der Mengen \mathcal{SET} surjektiv) läßt sich die Operationsabbildung $\hat{c}^{OP}: OP \rightarrow OP_{\bar{c}}$ wie folgt konstruieren

$$\hat{c}^{OP}(op) \stackrel{\text{def}}{=} \alpha(op): \bar{c}_s^{OP}(so_1), \dots, \bar{c}_s^{OP}(so_n) \rightarrow \bar{c}_s^{OP}(so)$$

für alle Operationen $op = d(op'): c(so_1), \dots, c(so_n) \rightarrow c(so_n)$ aus OP . Aus

$$\begin{aligned} \hat{c}^{OP}(op) &= \alpha(op): \bar{c}_s^{SO}(so_1), \dots, \bar{c}_s^{SO}(so_n) \rightarrow \bar{c}_s^{SO}(so) \\ &= \alpha(op): \hat{c}_s^{SO}(c_s^{SO}(so_1)), \dots, \hat{c}_s^{SO}(c_s^{SO}(so_n)) \rightarrow \hat{c}_s^{SO}(c_s^{SO}(so)) \end{aligned}$$

für alle Operationen op in OP folgt die Verträglichkeit von \hat{c}^{OP} mit den Sorten der Operationen. Damit ist $(\hat{c}^{SO}, \hat{c}^{OP})$ ein Signaturmorphismus von (SO, OP) nach $Sig_{\bar{c}}$.

Für alle Objekte s in \mathcal{S} und alle Operationen op in $OP_{s'}$ gilt:

$$\hat{c}^{OP}(d(op)) = \alpha(d(op)) = \bar{c}^{OP}(op)$$

und damit $(\hat{c}^{SO}, \hat{c}^{OP}) \circ (c, d) = (\hat{c}^{SO} \circ c, \hat{c}^{OP} \circ d) = (\bar{c}^{SO}, \bar{c}^{OP})$.

Die Eindeutigkeit $(\widehat{c}^{SO}, \widehat{c}^{OP})$ folgt aus der Eindeutigkeit der Morphismen \widehat{c}^{SO} und α .

□

Beweis (6.2.6 Colimites in $\mathcal{SP}\mathcal{E}\mathcal{C}$):

Es muß gezeigt werden:

- $((c_s)_{s \in \text{Obj}(\mathcal{S})}, (Sig, EQ))$ ist eine Senke und
- diese Senke ist eindeutig.

$((c_s)_{s \in \text{Obj}(\mathcal{S})}, (Sig, EQ))$ ist eine Senke: (Sig, EQ) ist eine Spezifikation. c_s ist für alle Objekte s aus \mathcal{S} ein Signaturmorphismus von dem jeweiligen Diagrammobjekt in das Colimesobjekt. Das c_s mit den Gleichungen verträglich ist folgt direkt aus der Konstruktion der Gleichungsmenge EQ . Damit bilden die c_s Spezifikationsmorphismen von den Diagrammobjekten in das Colimesobjekt. Die Verträglichkeit dieser Morphismen mit den Diagrammmorphismen folgt aus ihrer Verträglichkeit mit den Morphismen des Diagramms \mathcal{G} , die definitionsgemäß mit der Einschränkung der \mathcal{D} -Morphismen auf die jeweiligen Signaturen übereinstimmen und aus der Definition der Spezifikationsmorphismen als gleichungsbewahrende Signaturmorphismen.

(Eindeutigkeit): Sei $((\bar{c}_s)_{s \in \text{Obj}(\mathcal{S})}, Spec_{\bar{c}})$ mit $Spec_{\bar{c}} = (Sig_{\bar{c}}, EQ_{\bar{c}})$ eine weitere Senke für \mathcal{D} . Wir zeigen, daß in diesem Fall ein eindeutiger Spezifikationsmorphismus

$$\widehat{c}: (Sig, EQ) \rightarrow Spec_{\bar{c}} \quad \text{mit} \quad \widehat{c} \circ c_s = \bar{c}_s$$

für alle \mathcal{S} -Objekte s existiert.

Als Senke für \mathcal{D} bildet $((\bar{c}_s)_{s \in \text{Obj}(\mathcal{S})}, Spec_{\bar{c}})$ auch eine Senke für \mathcal{G} . Aufgrund der Eindeutigkeit des Colimes der Signaturen folgt daraus die Existenz und Eindeutigkeit eines Signaturmorphismus

$$\widehat{c}: Sig \rightarrow Sig_{\bar{c}} \quad \text{mit} \quad \widehat{c} \circ c_s = \bar{c}_s$$

für alle Objekte s in \mathcal{S} . In dem wir zeigen, daß \widehat{c} die Gleichungen respektiert, ergibt sich daraus der gesuchte eindeutige Spezifikationsmorphismus.

Für alle Gleichungen

$$(X^{c_s}, c_s^t(l), c_s^t(r)), \quad s \in \text{Obj}(\mathcal{S}), \quad (X, l, r) \in EQ_s$$

in EQ gilt:

$$(X^{\widehat{c}(c_s)}, \widehat{c}(c_s)^t(l), \widehat{c}(c_s)^t(r)) = (X^{\bar{c}_s}, \bar{c}_s^t(l), \bar{c}_s^t(r))$$

und damit

$$EQ_{\bar{c}} \models (X^{\widehat{c}(c_s)}, \widehat{c}(c_s)^t(l), \widehat{c}(c_s)^t(r))$$

da \bar{c}_s als Spezifikationsmorphismus mit den Gleichungen verträglich ist.

□

Beweis (6.2.21 Coprodukte für Algebren):

Im weiteren bezeichnen wir die Spezifikation (SO_c, OP_c, EQ_c) mit $Spec_c$ und die Algebra

$$((A_{so}^c)_{so \in SO_c}, (op_{A_c}: A_{so_1}^c \times \cdots \times A_{so_n}^c \rightarrow A_{so}^c)_{op \in OP_c^{so_1 \cdots so_n, so}})$$

mit A_c .

Es muß gezeigt werden:

- $((c_s, id)_{s \in Obj(\mathcal{S})}, A_c)$ ist eine Senke und
- diese Senke ist eindeutig.

$((c_s, id)_{s \in Obj(\mathcal{S})}, A_c)$ ist eine Senke): Gemäß unseren Eingangsannahmen bildet $((c_s)_{s \in Obj(\mathcal{S})}, Spec_c)$ ein Coprodukt für \mathcal{G} in der Kategorie der Spezifikationen \mathcal{SPEC} (\rightarrow 6.2.6, 6.1.3), daher existiert zu jeder Sorte so und jeder Operation op in A_c genau ein Objekt s aus \mathcal{S} und eine Sorte so' bzw. eine Operation op' aus $\mathcal{G}(s)$ mit $c_s^{SO}(so') = so$ bzw. $c_s^{OP}(op') = op$. Damit sind die Trägermengen und Operationen in A_c wohldefiniert und A_c ist eine Algebra in $\mathcal{ALG}(Spec_c)$.

Darüber hinaus folgt aus der Definition von A_c und der Definition der Vergißfunktoren (\rightarrow 6.1.11) $\mathcal{D}(s) = \mathcal{V}_{c_s}(A_c)$ für alle Objekte s aus \mathcal{S} , sodaß (c_s, id) für alle Objekte s aus \mathcal{S} einen generalisierten Algebromorphismus von $\mathcal{D}(s)$ nach A_c bildet. Da es sich bei \mathcal{S} um eine diskrete Kategorie handelt, folgt daraus unsere Behauptung.

(Eindeutigkeit): Sei $((\bar{c}_s, \bar{c}_s^A)_{s \in Obj(\mathcal{S})}, A_{\bar{c}})$ mit $Spec_{\bar{c}} = cod(\bar{c}_s)$ eine weitere Senke für \mathcal{D} . Mit der Eindeutigkeit der Coprodukte folgt daraus die Existenz und Eindeutigkeit eines Spezifikationsmorphismus

$$\hat{c}: Spec_c \rightarrow Spec_{\bar{c}} \quad \text{mit} \quad \hat{c} \circ c_s = \bar{c}_s$$

für alle Objekte s aus \mathcal{S} .

Wir definieren den Algebromorphismus $\hat{c}^A: A_c \rightarrow \mathcal{V}_{\hat{c}}(A_{\bar{c}})$ durch

$$\hat{c}_{c_s^{SO}(so)}^A(a) = \bar{c}_{sso}^A(a)$$

für alle Objekte s aus \mathcal{S} , Sorten so aus SO_s und Elemente a aus $A_{c_s^{SO}(so)}^c = A_{so}^s$. Aus

$$\begin{aligned} & \hat{c}_{c_s^{SO}(so)}^A(c_s^{OP}(op)_{A_c}(a_1, \dots, a_n)) \\ &= \bar{c}_{sso}^A(op_{A_s}(a_1, \dots, a_n)) && \text{Def. } \hat{c}^A, 6.2.21 \\ &= op_{\mathcal{V}_{\bar{c}}(A_{\bar{c}})}(\bar{c}_{sso_1}^A(a_1), \dots, \bar{c}_{sso_n}^A(a_n)) && \bar{c}_s^A \in \mathcal{ALG}(\mathcal{G}(s)) \end{aligned}$$

$$= c_s^{OP}(op)\mathcal{V}_{\widehat{c}(A_{\overline{c}})}(\widehat{c}_{c_s^{SO}(so_1)}^A(a_1), \dots, \widehat{c}_{c_s^{SO}(so_n)}^A(a_n)) \quad \text{Def. } \widehat{c}^A, 6.1.11, \overline{c}_s = \widehat{c} \circ c_s$$

$$= op\mathcal{V}_{\widehat{c}(A_{\overline{c}})}(\widehat{c}_{c_s^{SO}(so_1)}^A(a_1), \dots, \widehat{c}_{c_s^{SO}(so_n)}^A(a_n)) \quad \text{Def. } \widehat{c}^A, 6.2.21$$

für alle Objekte s aus \mathcal{S} und Operationssymbole $op: so_1, \dots, so_n \rightarrow so$ aus OP_s , folgt die Verträglichkeit von \widehat{c}^A mit den Operationen.

Mit der Definition der Vergißfunktoren (\rightarrow 6.1.11) und $\widehat{c} \circ c_s = \overline{c}_s$ folgt

$$\mathcal{V}_{c_s}(\widehat{c}^A) = \overline{c}_s^A$$

für alle Objekte s aus \mathcal{S} und wir erhalten den generalisierten Algebromorphismus $(\widehat{c}, \widehat{c}^A): A_c \rightarrow A_{\overline{c}}$ mit

$$(\widehat{c}, \widehat{c}^A) \circ (c_s, id) = (\widehat{c} \circ c_s, \mathcal{V}_{c_s}(\widehat{c}^A) \circ id) = (\overline{c}_s, \overline{c}_s^A)$$

für alle Objekte s aus \mathcal{S} .

Für alle anderen generalisierten Algebromorphismen

$$(\tilde{c}, \tilde{c}^A): A_c \rightarrow A_{\overline{c}} \quad \text{mit} \quad (\tilde{c}, \tilde{c}^A) \circ (c_s, id) = (\overline{c}_s, \overline{c}_s^A)$$

für alle Objekte s aus \mathcal{S} , folgt

$$\forall s \in \text{Obj}(\mathcal{S}): \tilde{c} \circ c_s = \overline{c}_s$$

und damit aufgrund der Eindeutigkeit von \widehat{c} mit dieser Eigenschaft $\tilde{c} = \widehat{c}$. Außerdem gilt für alle Objekte s aus \mathcal{S}

$$\mathcal{V}_{c_s}(\tilde{c}^A) = \overline{c}_s^A = \mathcal{V}_{c_s}(\widehat{c}^A)$$

und damit, da für alle Spec_c -Sorten so ein Objekt s aus \mathcal{S} und eine Sorte so' aus SO_s existiert mit $c_s^{SO}(so') = so$ und $\widehat{c}_{so'}^A = \mathcal{V}_{c_s}(\widehat{c}^A)_{so'}$ (\rightarrow 6.1.3, 6.2.6, 6.1.11), $\tilde{c}^A = \widehat{c}^A$.

□

Beweis (6.2.22 Coequalizer für Algebren):

Im weiteren bezeichnen wir die Algebra $((C_{so})_{so \in SO}, (op_c)_{op \in OP})$ mit C .

Es muß gezeigt werden:

- (c, C) ist eine Senke und
- diese Senke ist eindeutig.

$((c, C)$ ist eine Senke): C ist eine Spec -Algebra und c eine Familie von Abbildungen zwischen den Trägermengen von B und C . Aus der Definition der Operationen in C folgt darüber hinaus deren Verträglichkeit mit c . Damit bildet c einen Algebromorphismus von B nach C . Die Verträglichkeit von c mit den Diagrammorphismen folgt aus der Definition der Trägermengen in C .

(Eindeutigkeit): Sei (\bar{c}, D) eine weitere Senke für \mathcal{D} . Wir zeigen, daß in diesem Fall ein eindeutiger Algebromorphismus

$$\hat{c}: C \rightarrow D \quad \text{mit} \quad \hat{c} \circ c = \bar{c}$$

existiert.

Die Abbildungsfamilie $\hat{c} = (\hat{c}_{so})_{so \in SO}$ wird für alle *Spec*-Sorten so und alle Elemente $[b]$ aus C_{so} durch:

$$\hat{c}_{so}([b]) = \bar{c}_{so}(b)$$

definiert. Zunächst muß nachgewiesen werden, daß \hat{c} wohldefiniert ist, d.h. daß

$$\forall so \in SO, [b] \in C_{so}, b, b' \in [b]: \bar{c}_{so}(b) = \bar{c}_{so}(b')$$

gilt. Der Beweis erfolgt durch strukturelle Induktion.

(Induktions Basis): b, b' sind Konstanten oder nicht generierte Elemente.

Im Fall von Konstanten, folgt aus der Verträglichkeit der Algebromorphismen f, g mit den Operationen und der Definition der Kongruenzrelation: $b, b' \in [b] \implies b = b'$. Damit gilt die Behauptung für Konstanten. Für nicht generierte Elemente folgt sie aus der Definition der Kongruenzrelation in Zusammenhang mit der Verträglichkeit des Algebromorphismus \bar{c} mit den Diagrammorphismen, d.h. aus $\bar{c} \circ f = \bar{c} \circ g$.

(Induktions Schritt): Die Behauptung gilt für $b_i, b'_i, i = 1, \dots, n, b = op_B(b_1, \dots, b_n)$ und $b' = op_B(b'_1, \dots, b'_n)$.

$$\begin{aligned} & \bar{c}_{so}(op_B(b_1, \dots, b_n)) \\ &= op_D(\bar{c}_{so_1}(b_1), \dots, \bar{c}_{so_n}(b_n)) && \bar{c} \in \mathcal{ALG}(spec) \\ &= op_D(\bar{c}_{so_1}(b'_1), \dots, \bar{c}_{so_n}(b'_n)) && \text{Ind. Annahme} \\ &= \bar{c}_{so}(op_B(b'_1, \dots, b'_n)) && \bar{c} \in \mathcal{ALG}(spec) \end{aligned}$$

Für alle Sorten so aus SO und Elemente b aus B_{so} gilt:

$$\hat{c}_{so} \circ c_{so}(b) = \hat{c}_{so}([b]) = \bar{c}_{so}(b)$$

und damit $\hat{c} \circ c = \bar{c}$. Mit Hilfe dieser Eigenschaft können wir die Verträglichkeit von \hat{c} mit den Operationen zeigen.

Für alle Sortentupel so_1, \dots, so_n aus SO^ω , Sorten so aus SO , Operationssymbole op aus $OP^{so_1, \dots, so_n, so}$ und Elemente $[b_i]$ aus $C_{so_i}, i = 1, \dots, n$ gilt:

$$\begin{aligned} & \hat{c}_{so}(op_C([b_1], \dots, [b_n])) \\ &= \hat{c}_{so}(op_C(c_{so_1}(b_1), \dots, c_{so_n}(b_n))) && \text{Def. } c \\ &= \hat{c}_{so}(c_{so}(op_B(b_1, \dots, b_n))) && 6.2.22 \end{aligned}$$

$$\begin{aligned}
&= \bar{c}_{so}(op_B(b_1, \dots, b_n)) && \hat{c} \circ c = \bar{c} \\
&= op_D(\bar{c}_{so_1}(b_1), \dots, \bar{c}_{so_n}(b_n)) && \bar{c} \in \mathcal{ALG}(Spec) \\
&= op_D(\hat{c}_{so_1}([b_1]), \dots, \hat{c}_{so_n}([b_n])) && \text{Def. } \hat{c}
\end{aligned}$$

Damit bildet \hat{c} einen, mit den Morphismen der neuen Senke verträglichen, Algebromorphismus von C nach D .

Für jeden anderen Algebromorphismus

$$\tilde{c}: C \rightarrow D \quad \text{mit} \quad \tilde{c} \circ c = \bar{c}$$

gilt für alle Sorten so aus SO und Elemente $[b]$ aus C_{so} :

$$\tilde{c}_{so}([b]) = \tilde{c}_{so} \circ c_{so}(b) = \bar{c}_{so}(b) = \hat{c}_{so}([b])$$

und damit $\tilde{c} = \hat{c}$.

□

Beweis (6.3.5 Transformation von Colimites):

(Funktoreigenschaften von \mathcal{T}^S):

Es muß gezeigt werden:

- \mathcal{T}^S bewahrt die Identitäten und
- \mathcal{T}^S bewahrt die Komposition.

(\mathcal{T}^S bewahrt Identitäten): Sei $Spec = (SO, OP, EQ)$ eine Spezifikation und $\mathcal{T}^S(Spec) = Spec(X) = (SO, OP(X), EQ)$ ihre Erweiterung um die Variablenfamilie X als zusätzliche Konstanten, dann gilt:

$$\begin{aligned}
&\mathcal{T}^S(id_{Spec}) \\
&= \mathcal{T}^S((id_{SO}, id_{OP})) && 6.2.4
\end{aligned}$$

$$= (id_{SO}, (id_{OP})^x) \quad 6.3.4$$

und für alle $Spec(X)$ -Operationen op

$$(id_{OP})^x(op) = \begin{cases} id_{OP}(op), & op \in OP; \\ x: \rightarrow id_{SO}(s), & op = x: \rightarrow s \wedge x \in X. \end{cases} \quad 6.3.4$$

$$= id_{OP(X)}(op)$$

$$\implies \mathcal{T}^S(id_{Spec}) = (id_{SO}, id_{OP(X)}) = id_{Spec(X)} \quad 6.2.4$$

(\mathcal{T}^S bewahrt die Composition): Seien

- $Spec_i = (SO_i, OP_i, EQ_i)$, $i = 1, 2, 3$, Spezifikationen,
- $\mathcal{T}^S(Spec_i) = Spec_i(X) = (SO_i, OP_i(X), EQ_i)$, $i = 1, 2, 3$, ihre Erweiterungen um die Variablenfamilie X als zusätzliche Konstanten,
- $h_i = (h_i^{SO}, h_i^{OP}): Spec_i \rightarrow Spec_{i+1}$, $i = 1, 2$, Spezifikationsmorphisme und
- $\mathcal{T}^S(h_i) = (h_i^{SO}, h_i^{OP_x})$, $i = 1, 2$,

dann gilt:

$$\begin{aligned} & \mathcal{T}^S(h_2 \circ h_1) \\ &= \mathcal{T}^S(h_2^{SO} \circ h_1^{SO}, h_2^{OP} \circ h_1^{OP}) \end{aligned} \quad 6.2.4$$

$$= (h_2^{SO} \circ h_1^{SO}, (h_2^{OP} \circ h_1^{OP})^x) \quad 6.3.4$$

und für alle $Spec_1$ -Operationen op

$$\begin{aligned} & (h_2^{OP} \circ h_1^{OP})^x(op) \\ &= h_2^{OP} \circ h_1^{OP}(op) \end{aligned} \quad 6.3.4$$

$$= h_2^{OP_x} \circ h_1^{OP_x}(op) \quad h_1^{OP}(op) \in OP_2, 6.3.4$$

und für alle $Spec_1(X)$ -Sorten s und Variablen x aus X_s

$$\begin{aligned} & (h_2^{OP} \circ h_1^{OP})^x(x: \rightarrow s) \\ &= x: \rightarrow h_2^{SO} \circ h_1^{SO}(s) \end{aligned} \quad 6.3.4$$

$$= h_2^{OP_x}(x: \rightarrow h_1^{SO}(s)) \quad 6.3.4, x \in X_{h_1^{SO}(s)},$$

$$x: \rightarrow h_1^{SO}(s) \in Spec_2(X)$$

$$= h_2^{OP_x} \circ h_1^{OP_x}(x: \rightarrow s) \quad 6.3.4, x \in X_s,$$

$$x: \rightarrow s \in Spec_2(X)$$

\implies

$$\begin{aligned} & \mathcal{T}^S(h_2 \circ h_1) \\ &= (h_2^{SO} \circ h_1^{SO}, h_2^{OP_x} \circ h_1^{OP_x}) \\ &= \mathcal{T}^S(h_2) \circ \mathcal{T}^S(h_1) \end{aligned} \quad 6.3.4$$

□

(\mathcal{T}^S ist treu):

Folgt direkt aus der Definition von \mathcal{T}^S .

□

(\mathcal{T}^S bewahrt endliche Colimites):

Sei \mathcal{S} eine endliche Kategorie. Wir zeigen für jeden Funktor $\mathcal{D}: \mathcal{S} \rightarrow \mathcal{SPEC}$ mit Colimites

$$\left((c_s)_{s \in \text{Obj}(\mathcal{S})}, \text{Spec}_c \right) \quad \text{ist} \quad \left((\mathcal{T}^S(c_s))_{s \in \text{Obj}(\mathcal{S})}, \text{Spec}_c(X) \right)^{21}$$

ein Colimes für $\mathcal{T}^S \circ \mathcal{D}$. Dazu nehmen wir o.B.d.A an:

- $\text{Spec}_c = (SO_c, OP_c, EQ_c)$,
- $\text{Spec}_c(X) = (SO_c, OP_c(X), EQ_c)$,
- $\{ \mathcal{D}(m) \mid m: s \rightarrow s' \in \text{Mor}(\mathcal{S}) \} = \{ h_m: \text{Spec}_s \rightarrow \text{Spec}_{s'} \mid m: s \rightarrow s' \in \text{Mor}(\mathcal{S}) \}$,
- $c_s: \text{Spec}_s \rightarrow \text{Spec}_c$ für alle \mathcal{S} -Objekte s

und notieren die Erweiterung $\mathcal{T}^S(h)$ eines Spezifikationsmorphisms h mit h^x .

Das es sich beim Bild des Colimes für \mathcal{D} , unter \mathcal{T}^S , um eine Senke für $\mathcal{T}^S \circ \mathcal{D}$ handelt, folgt aus den Funktoreigenschaften von \mathcal{T}^S . Es verbleibt der Nachweis der couniversellen Eigenschaft dieser Senke.

(Couniverselle Eigenschaft): Sei $((\bar{c}_s)_{s \in \text{Obj}(\mathcal{S})}, \text{Spec}_{\bar{c}})$ eine Senke für $\mathcal{T}^S \circ \mathcal{D}$, d.h. es gilt:

$$(a): \quad \bar{c}_s = \bar{c}_{s'} \circ h_m^x$$

für alle \mathcal{S} -Morphismen $m: s \rightarrow s'$, dann konstruieren wir einen Spezifikationsmorphimus $\hat{c}^x: \text{Spec}_c(X) \rightarrow \text{Spec}_{\bar{c}}$, der bzgl. der Eigenschaft $\hat{c}^x \circ c_s^x = \bar{c}_s$ für alle \mathcal{S} -Objekte s , eindeutig ist.

Mit der Definition von \mathcal{T}^S (\rightarrow 6.3.4) gilt für alle Spezifikationsmorphismen h

$$(b): \quad \mathcal{T}^S(h) \Big|_{\text{dom}(h)} = h.$$

Wir definieren für jede Sorte o aus SO_c eine Menge

$$\langle o \rangle \stackrel{\text{def}}{=} \{ (o', s) \mid s \in \text{Obj}(\mathcal{S}), o' \in SO_s, c_s^{SO}(o') = o \}$$

und zeigen, daß für jede Spec_c -Sorte o und jedes Paar $(o', s), (o'', s')$ aus $\langle o \rangle$ gilt:

$$(c): \quad \langle o \rangle \neq \emptyset \quad \text{und} \quad \bar{c}_s(o') = \bar{c}_{s'}(o'').$$

Colimites in der Kategorie der Spezifikationen \mathcal{SPEC} werden aus den entsprechenden Colimites der unterliegenden Signaturen gebildet (\rightarrow 6.2.6). Deren Colimites werden wiederum aus den Colimites ihrer Sorten- und Operationskomponenten zusammengesetzt (\rightarrow 6.1.3, 6.1.4), d.h. unsere, nur die Sorten betreffende Behauptung, kann auf dem Colimes der Sortenkomponenten nachgewiesen werden. Dabei handelt es sich

²¹ $\mathcal{T}^S(\text{Spec}_c) = \text{Spec}_c(X)$

um einen Colimes in der Kategorie der Mengen \mathcal{SET} . Da nach Satz 5.1.19 in Zusammenhang mit Folgerung 5.1.17 jeder Colimes unter ausschließlicher Verwendung von Coprodukten und Coequalizern konstruiert werden kann, genügt es unsere Behauptung für diese beiden Colimeskonstruktionen in \mathcal{SET} nachzuweisen.

Im Fall von Coprodukten, die in \mathcal{SET} durch die disjunkte Vereinigung der Diagrammobjekte und mit entsprechenden Inklusionen gebildet werden, enthält die Menge $\langle o \rangle$ für jede Sorte o des Coproduktobjekts SO_c genau ein Element, sodaß (c) offensichtlich erfüllt ist.

Das Objekt eines Coequalizers zweier Funktionen f, g in \mathcal{SET} wird aus der Codomäne der Funktionen durch Faktorisierung nach der kleinsten Äquivalenzrelation gewonnen, die alle Paare $(f(a), g(a))$ für die Elemente a der Funktionendomäne enthält (\rightarrow Differenzcokerne [ML70] S. 69).

Damit existiert, im Falle einer von dem Graphen $s \rightrightarrows s'$ generierten Kategorie \mathcal{S} , d.h. immer dann wenn der Colimes zu \mathcal{D} ein Coequalizer ist, für alle Paare (o_1, s') , (o_n, s') aus $\langle o \rangle$ mit $o_1 \neq o_n$ eine Folge o_1, \dots, o_n von Sorten aus $SO_{s'}$ mit der Eigenschaft, daß zu jedem Paar von aufeinanderfolgenden Sorten o_i, o_{i+1} eine Sorte o' aus SO_s existiert, sodaß für die \mathcal{S} -Morphismen m, m' von s nach s' gilt:

$$h_m^{SO}(o') = o_i \quad \text{und} \quad h_{m'}^{SO}(o') = o_{i+1}.$$

Mit (a) folgt daraus:

$$\bar{c}_{s'}^{SO}(o_i) = \bar{c}_{s'}^{SO} \circ h_m^{SO}(o') = \bar{c}_{s'}^{SO} \circ h_{m'}^{SO}(o') = \bar{c}_{s'}^{SO}(o_{i+1})$$

für $i = 1, \dots, n-1$ und damit $\bar{c}_{s'}^{SO}(o_1) = \bar{c}_{s'}^{SO}(o_n)$.

Für alle Paare $(o', s), (o'', s')$ aus $\langle o \rangle$ gilt: $(h_m^{SO}(o'), s) \in \langle o \rangle$. Zusammen mit dem im vorangehenden Absatz gezeigten Ergebnis folgt daraus $\bar{c}_{s'}^{SO}(o'') = \bar{c}_{s'}^{SO} \circ h_m^{SO}(o')$ und mit (a)

$$\bar{c}_s^{SO}(o') = \bar{c}_{s'}^{SO} \circ h_m^{SO}(o') = \bar{c}_{s'}^{SO}(o'').$$

Für alle Paare $(o', s), (o'', s)$ aus $\langle o \rangle$ gilt:

$$\begin{aligned} c_s(o') &= o = c_s(o'') && \text{Def. } \langle o \rangle \\ \implies c_{s'} \circ h_m^{SO}(o') &= o = c_{s'} \circ h_m^{SO}(o'') && \text{Colimes} \\ \implies (h_m^{SO}(o'), s'), (h_m^{SO}(o''), s') &\in \langle o \rangle && \text{Def. } \langle o \rangle \\ \implies \bar{c}_{s'}^{SO} \circ h_m^{SO}(o') &= \bar{c}_{s'}^{SO} \circ h_m^{SO}(o'') && \text{s.o.} \\ \implies \bar{c}_s^{SO}(o') &= \bar{c}_s^{SO}(o'') && \text{(a)} \end{aligned}$$

Damit haben unsere Behauptung für alle Sorten o aus SO_c und alle Paarkombinationen aus $\langle o \rangle$ gezeigt.

Aus (a) und (b) folgt für alle \mathcal{S} -Morphismen $m: s \rightarrow s'$

$$\bar{c}_s \Big|_{Spec_s} = \bar{c}_{s'} \Big|_{Spec_{s'}} \circ h_m$$

und damit, wegen des Colimites zu \mathcal{D} mit dieser Eigenschaft, die Existenz und Eindeutigkeit eines Spezifikationsmorphisms

$$(d): \quad \hat{c} = (\hat{c}^{SO}, \hat{c}^{OP}): Spec_c \rightarrow Spec_{\bar{c}} \quad \text{mit} \quad \hat{c} \circ c_s = \bar{c}_s \Big|_{Spec_s}.$$

Mit Hilfe dieses Morphismus und den Mengen $\langle o \rangle$ zu den $Spec_c(X)$ -Sorten definieren wir den Spezifikationsmorphisms $\hat{c}^x: Spec_c(X) \rightarrow Spec_{\bar{c}}$ durch:

$$\hat{c}^x = (\hat{c}^{SO}, \hat{c}^{OP_x}) \quad \text{mit} \quad \hat{c}^{OP_x}(p) = \begin{cases} \hat{c}^{OP}(p), & p \in OP_c; \\ \bar{c}_s(x: \rightarrow o'), & p = x: \rightarrow o \wedge (o', s) \in \langle o \rangle; \end{cases}$$

für alle $Spec_c(X)$ -Operationen p . Die Definition ist wegen (c) vollständig und deterministisch. Da die Sortenabbildung von \hat{c}^x derjenigen von \hat{c} entspricht gilt wegen (d):

$$\hat{c}^{SO_x} \circ c_s^{SO_x} = \bar{c}_s^{SO}.$$

Für die Operationsabbildung zeigen wir

$$\hat{c}^{OP_x} \circ c_s^{OP_x} = \bar{c}_s^{OP}$$

für alle \mathcal{S} -Objekte s , zunächst für alle Operationen p aus $OP_c \subset OP_c(X)$ und dann für die neu hinzu gekommenen Konstanten $x: \rightarrow o$.

$$\begin{aligned} p \in OP_s & && \text{Annahme} \\ \implies & && \\ \hat{c}^{OP_x} \circ c_s^{OP_x}(p) & && \\ = \hat{c}^{OP} \circ c_s^{OP}(p) & && \text{Def. } \hat{c}^{OP_x}, (b) \\ = \bar{c}_s^{OP}(p) & && (d) \end{aligned}$$

$$\begin{aligned} p = x: \rightarrow o & && \text{Annahme} \\ \implies & && \\ \hat{c}^{OP_x} \circ c_s^{OP_x}(x: \rightarrow o) & && \\ = \hat{c}^{OP_x}(x: \rightarrow c_s^{SO}(o)) & && 6.3.4 \\ = \bar{c}_s(x: \rightarrow o) & && \text{Def. } \hat{c}^{OP_x}, \\ & && (o, s) \in \langle c_s^{SO}(o) \rangle \end{aligned}$$

Die Verträglichkeit der Sorten- und der Operationsabbildung mit den Spezifikationsmorphisms \bar{c}_s impliziert

$$\hat{c}^x \circ c_s^x = \bar{c}_s$$

für alle \mathcal{S} -Objekte s .

Für jeden anderen Spezifikationsmorphismus $\tilde{c}: \text{Spec}_c(X) \rightarrow \text{Spec}_{\tilde{c}}$ mit:

$$\tilde{c} \circ c_s^x = \tilde{c}_s$$

für alle \mathcal{S} -Objekte s , gilt wegen

$$\tilde{c} \Big|_{\text{Spec}_c} \circ c_s = \tilde{c}_s \Big|_{\text{Spec}_s}$$

für alle \mathcal{S} -Objekte s und damit aufgrund der Eindeutigkeit von \hat{c} mit dieser Eigenschaft: $\tilde{c} \Big|_{\text{Spec}_c} = \hat{c}$.

Verbleibt noch für Sorten o und alle Variablen x zur Sorte o zu Zeigen:

$$\tilde{c}^{OP}(x: \rightarrow o) = \hat{c}^{OP_x}(x: \rightarrow o).$$

Aufgrund von (c) und der Definition von \mathcal{T}^S (\rightarrow 6.3.4), existiert für alle Operationen $x: \rightarrow o$ aus $OP_c(X) \setminus OP_c$, ein \mathcal{S} -Objekt s , eine $\text{Spec}_s(X)$ -Sorte o' und eine Operation $x: \rightarrow o'$ mit $c_s^{OP}(x: \rightarrow o') = x: \rightarrow o$. Damit können wir für jede Operation $x: \rightarrow o$ zeigen:

$$\begin{aligned} & \tilde{c}^{OP}(x: \rightarrow o) \\ &= \tilde{c}^{OP} \circ c_s^{OP}(x: \rightarrow o') && \text{s.o.} \\ &= \tilde{c}_s^{OP}(x: \rightarrow o') \\ &= \hat{c}^{OP_x} \circ c_s^{OP}(x: \rightarrow o') \\ &= \hat{c}^{OP_x}(x: \rightarrow o) && \text{s.o.} \end{aligned}$$

□

□

Beweis (6.3.8 Natürliche Transformation):

Wir nehmen o.B.d.A an

- $h: \text{Spec}_1 \rightarrow \text{Spec}_2$ ist ein Spezifikationsmorphismus,
- $f: A \rightarrow B$ ist ein Spec_2 -Algebromorphismus,
- für $i = 1, 2$ ist \mathcal{F}_i der freie Funktor zur Inklusion von Spec_i in $\text{Spec}_i(X)$

und zeigen

$$\mathcal{V}_{\mathcal{T}^S(h)}(\mathcal{F}_2(f)) \circ h_A^\mathcal{V} = h_B^\mathcal{V} \circ \mathcal{F}_1(\mathcal{V}_h(f)).$$

Der Beweis erfolgt mit Hilfe struktureller Induktion über den Elementen der Termalgebra $T_{OP_1(X, \mathcal{V}_h(A))}$ zur Spezifikation $\text{Spec}_1(X, \mathcal{V}_h(A))$ (\rightarrow 6.2.15). Faktorisiert nach

den Spezifikationsgleichungen ergeben sich daraus die Elemente (Kongruenzklassen) in $\mathcal{V}_h(A)(X)$.

Für den Beweis benötigen wir die folgenden Zusammenhänge, die sich aus 6.1.11, 6.2.17, 6.2.18, 6.3.7 und 6.3.7 ergeben

$$(a): \forall h \in \text{Mor}(\mathcal{SPEC}), f \in \text{Mor}(\mathcal{ALG}(\text{cod}(h))), a \in \text{dom}(\mathcal{V}_h(f)): \mathcal{V}_h(f)(a) = f(a)$$

$$(b): \forall h \in \text{Mor}(\mathcal{SPEC}), f \in \text{Mor}(\mathcal{ALG}(\text{dom}(h))), a \in \text{dom}(f): \mathcal{F}_h(f)([a]) = [f(a)]$$

$$(c): \forall h \in \text{Mor}(\mathcal{SPEC}), A \in \text{Obj}(\mathcal{ALG}(\text{cod}(h))), a \in \mathcal{V}_h(A): h_A^\mathcal{V}([a]) = [a]$$

Für alle Sorten s aus $\text{Spec}_1(X, \mathcal{V}_h(A))$ und Terme t aus $T_{OP_1(X, \mathcal{V}_h(A))_s}$:

Induktionsbasis: $t \in OP_1^{\lambda, s}$.

Es gilt: $t \in \mathcal{V}_h(A)$ oder $t \notin \mathcal{V}_h(A)$

$$t \in \mathcal{V}_h(A)$$

Annahme

\implies

$$\mathcal{V}_{\mathcal{T}^s(h)}(\mathcal{F}_2(f)) \circ h_A^\mathcal{V}([t])$$

$$= \mathcal{V}_{\mathcal{T}^s(h)}(\mathcal{F}_2(f))([t]) = [f(t)] = \mathcal{F}_1(\mathcal{V}_h(f))([t]) \quad (a), (b), (c)$$

$$= h_B^\mathcal{V} \circ \mathcal{F}_1(\mathcal{V}_h(f))([t]) \quad (c)$$

$$t \notin \mathcal{V}_h(A)$$

Annahme

\implies

$$\mathcal{V}_{\mathcal{T}^s(h)}(\mathcal{F}_2(f)) \circ h_A^\mathcal{V}([t])$$

$$= \mathcal{V}_{\mathcal{T}^s(h)}(\mathcal{F}_2(f)) \circ h_A^\mathcal{V}(t_{\mathcal{V}_h(A)(X)}) \quad \text{Quotiententermalg.}$$

$$= \mathcal{V}_{\mathcal{T}^s(h)}(\mathcal{F}_2(f))(t_{\mathcal{V}_{\mathcal{T}^s(h)}(A(X))}) \quad h_A^\mathcal{V} \in \mathcal{ALG}(\text{Spec}_1(X))$$

$$= t_{\mathcal{V}_{\mathcal{T}^s(h)}(B(X))} \quad \mathcal{V}_{\mathcal{T}^s(h)}(\mathcal{F}_2(f)) \in \mathcal{ALG}(\text{Spec}_1(X))$$

$$= h_B^\mathcal{V}(t_{\mathcal{V}_h(B)(X)}) \quad h_B^\mathcal{V} \in \mathcal{ALG}(\text{Spec}_1(X))$$

$$= h_B^\mathcal{V} \circ \mathcal{F}_1(\mathcal{V}_h(f))(t_{\mathcal{V}_h(A)(X)}) \quad \mathcal{F}_1(\mathcal{V}_h(f)) \in \mathcal{ALG}(\text{Spec}_1(X))$$

$$= h_B^\mathcal{V} \circ \mathcal{F}_1(\mathcal{V}_h(f))([t]) \quad \text{Quotiententermalg.}$$

Induktionsschritt: $t = op(t_1, \dots, t_n)$ und für $i = 1, \dots, n$

$$\mathcal{V}_{\mathcal{T}^s(h)}(\mathcal{F}_2(f)) \circ h_A^\mathcal{V}([t_i]) = h_B^\mathcal{V} \circ \mathcal{F}_1(\mathcal{V}_h(f))([t_i])$$

$$\begin{aligned}
& \mathcal{V}_{\mathcal{T}^S(h)}(\mathcal{F}_2(f)) \circ h_A^\mathcal{V}([t]) \\
&= \mathcal{V}_{\mathcal{T}^S(h)}(\mathcal{F}_2(f)) \circ h_A^\mathcal{V}(op_{\mathcal{V}_h(A)(X)}([t_1], \dots, [t_n])) && \text{Quotiententermalg.} \\
&= \mathcal{V}_{\mathcal{T}^S(h)}(\mathcal{F}_2(f))(op_{\mathcal{V}_{\mathcal{T}^S(h)}(A)(X)}(h_A^\mathcal{V}([t_1]), \dots, h_A^\mathcal{V}([t_n]))) && h_A^\mathcal{V} \in \mathcal{ALG}(Spec_1(X)) \\
&= op_{\mathcal{V}(B(X))}(\mathcal{V}(\mathcal{F}(f)) \circ h_A^\mathcal{V}([t_1]), \dots, \mathcal{V}(\mathcal{F}(f)) \circ h_A^\mathcal{V}([t_n])) && \mathcal{V}_{\mathcal{T}^S(h)}(\mathcal{F}_2(f)) \in \\
& && \mathcal{ALG}(Spec_1(X)) \\
&= op_{\mathcal{V}(B(X))}(h_B^\mathcal{V} \circ \mathcal{F}_1(\mathcal{V}_h(f))([t_1]), \dots, h_B^\mathcal{V} \circ \mathcal{F}_1(\mathcal{V}_h(f))([t_n])) && \text{Ind. Annahme} \\
&= h_B^\mathcal{V}(op_{\mathcal{V}_h(B)(X)}(\mathcal{F}_1(\mathcal{V}_h(f))([t_1]), \dots, \mathcal{F}_1(\mathcal{V}_h(f))([t_n]))) && \mathcal{F}_1(\mathcal{V}_h(f)) \in \\
& && \mathcal{ALG}(Spec_1(X)) \\
&= h_B^\mathcal{V} \circ \mathcal{F}_1(\mathcal{V}_h(f))(op_{\mathcal{V}_h(A)(X)}([t_1], \dots, [t_n])) && h_B^\mathcal{V} \in \mathcal{ALG}(Spec_1(X)) \\
&= h_B^\mathcal{V} \circ \mathcal{F}_1(\mathcal{V}_h(f))([t]) && \text{Quotiententermalg.}
\end{aligned}$$

□

Beweis (6.3.10 Eigenschaften von \mathcal{T}^A):

Im weiteren bezeichnet:

- \mathcal{F} (mit leerem Index) den freien Funktor von einer Spezifikation $Spec$ in ihre Erweiterung $Spec(X)$ und
- \mathcal{V} (mit leerem Index) den Vergißfunktor zur Inklusion einer Spezifikation mit Variablen $Spec(X)$ in die, durch Konstanten zu den Elementen einer $Spec$ -Algebra A , erweiterte Spezifikation $Spec(X, A)$.

(Funktoreigenschaften von \mathcal{T}^A):

Es muß gezeigt werden:

- \mathcal{T}^A bewahrt die Identitäten und
- \mathcal{T}^A bewahrt die Komposition.

(\mathcal{T}^A bewahrt Identitäten): Für alle Spezifikationen $Spec$ und alle $Spec$ -Algebren A gilt:

$$\begin{aligned}
id_A^\mathcal{V} &= \mathcal{V}(eval: T_{Spec}(X, \mathcal{V}_{id}(A)) \rightarrow \mathcal{V}_{id}(T_{Spec}(X, A))) && 6.3.7, 6.3.5 \\
&= \mathcal{V}(eval: T_{Spec}(X, A) \rightarrow T_{Spec}(X, A)) && 6.1.12 \\
&= \mathcal{V}(id_{T_{Spec}(X, A)}) && T_{Spec}(X, A) \text{ ist initial} \\
&= id_{A(X)} && \text{Fkt.Eigens. } \mathcal{V}_{\hookrightarrow}
\end{aligned}$$

Daraus folgt:

$$\mathcal{T}^A(id_A) = \mathcal{T}^A((id_{Spec}, id_A))$$

$$\begin{aligned}
&= (\mathcal{T}^S(id_{Spec}), id_A^\mathcal{V} \circ \mathcal{F}(id_A)) && 6.3.7 \\
&= (id_{Spec(X)}, id_A^\mathcal{V} \circ id_{A(X)}) && \text{Fkt.Eigens. } \mathcal{T}^S, \mathcal{F} \\
&= (id_{Spec(X)}, id_{A(X)} \circ id_{A(X)}) && \text{s.o.} \\
&= id_{A(X)}
\end{aligned}$$

(\mathcal{T}^A bewahrt die Composition): Wir zeigen, daß für alle Paare generalisierter Morphismen $(h, f), (k, g)$ mit $cod((h, f)) = dom((k, g))$ gilt:

$$\mathcal{T}^A((k, g) \circ (h, f)) = \mathcal{T}^A((k, g)) \circ \mathcal{T}^A((h, f)).$$

Dazu nehmen wir o.B.d.A an:

- $h: Spec_1 \rightarrow Spec_2,$
- $k: Spec_2 \rightarrow Spec_3,$
- $(h, f): A \rightarrow B,$
- $(k, g): B \rightarrow C$

und definieren $d: Spec_1 \rightarrow Spec_3 = k \circ h.$

Der Beweis nutzt den Zusammenhang

$$(a): \quad d_C^\mathcal{V} = \mathcal{V}_h(k_C^\mathcal{V}) \circ h_{\mathcal{V}_k(C)}^\mathcal{V}$$

dessen Gültigkeit wir zunächst nachweisen.

$$(d_C^\mathcal{V} = \mathcal{V}_h(k_C^\mathcal{V}) \circ h_{\mathcal{V}_k(C)}^\mathcal{V}):$$

Seien

- $h^e: Spec_1(X, \mathcal{V}_h(\mathcal{V}_k(C))) \rightarrow Spec_2(X, \mathcal{V}_k(C)),$
- $k^e: Spec_2(X, \mathcal{V}_k(C)) \rightarrow Spec_3(X, C)$ und
- $d^e: Spec_1(X, \mathcal{V}_d(C)) \rightarrow Spec_3(X, C)$

die jeweiligen Erweiterungen (\rightarrow 6.3.7) der Spezifikationsmorphismen h, k, d dann gilt wegen $d = k \circ h$ und $\mathcal{V}_d(C) = \mathcal{V}_{k \circ h}(C) = \mathcal{V}_h(\mathcal{V}_k(C))$:

$$d^e = k^e \circ h^e.$$

In Zusammenhang mit der Initialität von $T_{Spec_1(X, \mathcal{V}_d(C))}$ in $Spec_1(X, \mathcal{V}_d(C))$ folgt da-

raus das Kommutieren des nachfolgenden Diagramms

$$(b): \quad \begin{array}{ccccc} T_{Spec_1(X, \mathcal{V}_d(C))} & \xrightarrow{eval} & \mathcal{V}_{h^e}(T_{Spec_2(X, \mathcal{V}_k(C))}) & & T_{Spec_2(X, \mathcal{V}_k(C))} \\ & \searrow eval & \downarrow \mathcal{V}_{h^e}(eval) & & \downarrow eval \\ & & \mathcal{V}_{d^e}(T_{Spec_3(X, C)}) & & \mathcal{V}_{k^e}(T_{Spec_3(X, C)}) \end{array}$$

und aus den Definitionen der h -Erweiterungen $\mathcal{T}^S(h)$ (\rightarrow 6.3.4) und h^e (\rightarrow 6.3.7) folgt

$$(c): \quad \begin{array}{ccccc} Spec_1 & & Spec_1(X) & \xrightarrow{\hookrightarrow} & Spec_1(X, A) \\ \downarrow h & & \downarrow \mathcal{T}^S(h) & & \downarrow h^e \\ Spec_2 & & Spec_2(X) & \xrightarrow{\hookrightarrow} & Spec_2(X, A) \end{array}$$

Aus (c) ergibt sich zusammen mit der Definition der Algebraerweiterungen (\rightarrow 6.3.7)

$$\mathcal{V}_d(C)(X) = \mathcal{V}_{\hookrightarrow}(T_{Spec_1(X, \mathcal{V}_d(C))}) \quad 6.3.7$$

und

$$\begin{aligned} & \mathcal{V}_{\mathcal{T}^S(h)}(\mathcal{V}_k(A)(X)) \\ &= \mathcal{V}_{\mathcal{T}^S(h)}(\mathcal{V}_{\hookrightarrow}(T_{Spec_2(X, \mathcal{V}_k(C))})) \quad 6.3.7 \\ &= \mathcal{V}_{\hookrightarrow \circ \mathcal{T}^S(h)}(T_{Spec_2(X, \mathcal{V}_k(C))}) \\ &= \mathcal{V}_{h^e \circ \hookrightarrow}(T_{Spec_2(X, \mathcal{V}_k(C))}) \quad (c) \end{aligned}$$

$$= \mathcal{V}_{\hookrightarrow}(\mathcal{V}_{h^e}(T_{Spec_2(X, \mathcal{V}_k(C))}))$$

und

$$\begin{aligned} & \mathcal{V}_{\mathcal{T}^S(d)}(C(X)) \\ &= \mathcal{V}_{\hookrightarrow \circ \mathcal{T}^S(d)}(T_{Spec_3(X, C)}) \quad 6.3.7 \\ &= \mathcal{V}_{d^e \circ \hookrightarrow}(T_{Spec_3(X, C)}) \quad (c) \end{aligned}$$

$$= \mathcal{V}_{\hookrightarrow}(\mathcal{V}_{d^e}(T_{Spec_3(X, C)}))$$

und

$$h_{\mathcal{V}_k(C)}^{\mathcal{V}}$$

$$= \mathcal{V}_{\hookrightarrow}(\text{eval}: T_1(X, \mathcal{V}_d(C)) \rightarrow \mathcal{V}_{h^e}(T_2(X, \mathcal{V}_k(C)))) \quad T_i = T_{\text{Spec}_i}, 6.3.7$$

und

$$\mathcal{V}_{\mathcal{T}^S(h)}(k_C^\mathcal{V})$$

$$= \mathcal{V}_{\mathcal{T}^S(h)}(\mathcal{V}_{\hookrightarrow}(\text{eval}: T_2(X, \mathcal{V}_k(C)) \rightarrow \mathcal{V}_{k^e}(T_3(X, C)))) \quad T_i = T_{\text{Spec}_i}, 6.3.7$$

$$= \mathcal{V}_{\hookrightarrow \circ \mathcal{T}^S(h)}(\text{eval})$$

$$= \mathcal{V}_{h^e \circ \hookrightarrow}(\text{eval}) \quad (\text{c})$$

$$= \mathcal{V}_{\hookrightarrow}(\mathcal{V}_{h^e}(\text{eval}))$$

und

$$d_C^\mathcal{V}$$

$$= \mathcal{V}_{\hookrightarrow}(\text{eval}: T_{\text{Spec}_1}(X, \mathcal{V}_d(C)) \rightarrow \mathcal{V}_{d^e}(T_{\text{Spec}_2}(X, C))) \quad 6.3.7$$

Damit entspricht unsere Behauptung dem Bild von Diagramm (b) unter dem Vergißfunktorkomplex $\mathcal{V}_{\hookrightarrow}$ zur Inklusion von $\text{Spec}_1(X)$ in $\text{Spec}_1(X, \mathcal{V}_d(C))$. Ihre Gültigkeit folgt mit dem Kommutieren von (b) aus der Bewahrung von Kompositionen durch $\mathcal{V}_{\hookrightarrow}$.

Mit Hilfe von (a) und der h -generierten natürlichen Transformation

$$h^\mathcal{V}: \mathcal{F} \circ \mathcal{V}_h \rightarrow \mathcal{V}_{\mathcal{T}^S(h)} \circ \mathcal{F}$$

(\rightarrow 6.3.8), die das Kommutieren des nachfolgenden Diagramms impliziert

$$(\text{d}): \quad \begin{array}{ccc} \mathcal{V}_h(B)(X) & \xrightarrow{h_B^\mathcal{V}} & \mathcal{V}_{\mathcal{T}^S(h)}(B(X)) \\ \mathcal{F}(\mathcal{V}_h(g)) \downarrow & \wr & \downarrow \mathcal{V}_{\mathcal{T}^S(h)}(\mathcal{F}(g)) \\ \mathcal{V}_h(\mathcal{V}_k(C))(X) & \xrightarrow{h_{\mathcal{V}_k(C)}^\mathcal{V}} & \mathcal{V}_{\mathcal{T}^S(h)}(\mathcal{V}_k(C)(X)) \end{array}$$

folgt:

$$\mathcal{T}^A((k, g) \circ (h, f))$$

$$= \mathcal{T}^A((k \circ h, \mathcal{V}_h(g) \circ f))$$

$$= (\mathcal{T}^S(k \circ h), (k \circ h)_C^\mathcal{V} \circ \mathcal{F}(\mathcal{V}_h(g)) \circ \mathcal{F}(f)) \quad \text{Fkt.Eigens. } \mathcal{F}, 6.3.7$$

$$= (\mathcal{T}^S(k \circ h), \mathcal{V}_{\mathcal{T}^S(h)}(k_C^\mathcal{V}) \circ h_{\mathcal{V}_k(C)}^\mathcal{V} \circ \mathcal{F}(\mathcal{V}_h(g)) \circ \mathcal{F}(f)) \quad (\text{a})$$

$$= (\mathcal{T}^S(k \circ h), \mathcal{V}_{\mathcal{T}^S(h)}(k_C^\mathcal{V}) \circ \mathcal{V}_{\mathcal{T}^S(h)}(\mathcal{F}(g)) \circ h_B^\mathcal{V} \circ \mathcal{F}(f)) \quad (\text{d})$$

$$= (\mathcal{T}^S(k) \circ \mathcal{T}^S(h), \mathcal{V}_{\mathcal{T}^S(h)}(k_C^\mathcal{V} \circ \mathcal{F}(g)) \circ h_B^\mathcal{V} \circ \mathcal{F}(f)) \quad (\text{d}), \text{Fkt.Eigens. } \mathcal{V}_{\mathcal{T}^S(h)}$$

$$= (\mathcal{T}^S(k), k_C^\mathcal{V} \circ \mathcal{F}(g)) \circ (\mathcal{T}^S(h), h_B^\mathcal{V} \circ \mathcal{F}(f))$$

$$= \mathcal{T}^A((k, g)) \circ \mathcal{T}^A((h, f)) \quad 6.3.7$$

□

(\mathcal{T}^A bewahrt endliche Colimites):

Sei \mathcal{S} eine endliche Kategorie. Wir zeigen für jeden Funktor $\mathcal{D}: \mathcal{S} \rightarrow \mathcal{GSP\mathcal{E}C-AL\mathcal{G}}$ mit Colimes

$$\left(((c_s, d_s))_{s \in \text{Obj}(\mathcal{S})}, G \right) \quad \text{ist} \quad \left((\mathcal{T}^A((c_s, d_s)))_{s \in \text{Obj}(\mathcal{S})}, \mathcal{T}^A(G) \right)$$

ein Colimes für $\mathcal{T}^A \circ \mathcal{D}$. Dazu nehmen wir o.B.d.A an:

- $\{ \mathcal{D}(m) \mid m: s \rightarrow s' \in \text{Mor}(\mathcal{S}) \} = \{ (h_m, f_m): A_s \rightarrow A_{s'} \},$
- $h_m: \text{Spec}_s \rightarrow \text{Spec}_{s'},$
- $c_s: \text{Spec}_s \rightarrow \text{Spec}_c$ für alle \mathcal{S} -Objekte s

und notieren die Erweiterung $\mathcal{T}^S(h)$ eines Spezifikationsmorphisms h mit h^x .

Die Verträglichkeit des transformierten Colimes mit den Morphismen des transformierten Diagramms $\mathcal{T}^A \circ \mathcal{D}$ folgt aus den Funktoreigenschaften von \mathcal{T}^A . Es verbleibt der Nachweis der couniversellen Eigenschaft.

(Couniverselle Eigenschaft): Mit Definition 6.3.9 gilt:

$$\left((\mathcal{T}^A((c, d)))_{s \in \text{Obj}(\mathcal{S})}, \mathcal{T}^A(G) \right) = ((c_s^x, c_s^y(G) \circ \mathcal{F}(d_s)), G(X)).^{22}$$

Wir zeigen, daß zu jeder Senke

$$\left(((\bar{c}_s, q_s))_{s \in \text{Obj}(\mathcal{S})}, Q \right)$$

eindeutig ein Morphismus

$$(\hat{c}, \hat{d}): G(X) \rightarrow Q \quad \text{mit} \quad (\hat{c}, \hat{d}) \circ (c_s^x, c_s^y(G) \circ \mathcal{F}(d_s)) = (\bar{c}_s, q_s)$$

für alle \mathcal{S} -Objekte s existiert.

Für den Beweis benötigen wir die folgenden Zusammenhänge, die sich aus 6.2.17, 6.2.18 und 6.3.7 ergeben

$$\text{(e): } \forall h \in \text{Mor}(\mathcal{SP\mathcal{E}C}), f \in \text{Mor}(\mathcal{AL\mathcal{G}}(\text{dom}(h))), a \in \text{dom}(f): \mathcal{F}_h(f)([a]) = [f(a)]$$

$$\text{(f): } \forall h \in \text{Mor}(\mathcal{SP\mathcal{E}C}), A \in \text{Obj}(\mathcal{AL\mathcal{G}}(\text{cod}(h))), a \in \mathcal{V}_h(A): h_A^y([a]) = [a].$$

²² $c_s^y(G) = c_{s_G}^y$ (\rightarrow 6.3.8)

Da die erste Komponente von Colimites generalisierter Morphismen in $\mathcal{G}\mathcal{S}\mathcal{P}\mathcal{E}\mathcal{C}\text{-}\mathcal{A}\mathcal{L}\mathcal{G}$ einen Colimes in $\mathcal{S}\mathcal{P}\mathcal{E}\mathcal{C}$ bildet (\rightarrow 5.3.5, 6.2.18) impliziert jede Senke, wegen $\bar{c}_{s'} \circ h_m = \bar{c}_s$ für alle \mathcal{S} -Objekte s , die Existenz und Eindeutigkeit eines Morphismus

$$(g): \quad \widehat{c}: \text{Spec}_c \rightarrow \text{cod}(\bar{c}) \quad \text{mit} \quad \widehat{c} \circ c_s = \bar{c}_s$$

für alle \mathcal{S} -Objekte s . Dieser Morphismus bildet die erste Komponente des generalisierten Morphismus von $G(X)$ nach Q . Zur Konstruktion der zweiten Komponente definieren wir zunächst für jedes \mathcal{S} -Objekt s einen Spec_s -Algebromorphismus

$$q'_s: G \rightarrow \mathcal{V}_{\bar{c}_s|_{\text{Spec}_s}}(Q) \quad \text{mit} \quad q'_{s_o}(a) = q_{s_o}([a])$$

für alle Spec_s -Sorten o und Elemente a der o -Trägermenge von A_s . Zusammen mit der Einschränkung von \bar{c}_s auf Spec_s für alle \mathcal{S} -Objekte s ergibt sich daraus die Familie generalisierter Morphismen

$$\left(\left(\bar{c}_s|_{\text{Spec}_s}, q'_s \right) : G \rightarrow Q \right)_{s \in \text{Obj}(\mathcal{S})}$$

deren Verträglichkeit mit den \mathcal{D} -Morphismen wir im folgenden nachweisen. Dazu ist für alle \mathcal{S} -Morphismen $m: s \rightarrow s'$, die Gleichung

$$\mathcal{V}_{h_m}(q'_{s'}) \circ f_m = q_s$$

als korrekt zu beweisen.

$$\forall o \in \mathcal{S}O_s, a \in A_o^s:$$

$$\begin{aligned} & \mathcal{V}_{h_m}(q'_{s'}) \circ f_m(a) \\ &= \mathcal{V}_{h_m^x}(q'_{s'})([f_m(a)]) && \text{Def. } q'_{s'} \\ &= \mathcal{V}_{h_m^x}(q'_{s'}) \circ h_{m_B}^y \circ \mathcal{F}(f_m)([a]) && (e), (f) \\ &= q_s([a]) && \text{Senkeneigens. } (\bar{c}_{s'}, q'_{s'}) \\ &= q'_s(a) && \text{Def. } q'_s \end{aligned}$$

Zusammen mit der aus Definition 6.3.4 folgenden Gleichung

$$h_m^x|_{\text{Spec}_s} = h_m$$

für alle \mathcal{S} -Morphismen $m: s \rightarrow s'$ und der Annahme das $((\bar{c}_s, q_s)_{s \in \text{Obj}(\mathcal{S})}, Q)$ eine Senke für $\mathcal{T}^A \circ \mathcal{D}$ bildet, gilt für alle \mathcal{S} -Morphismen $m: s \rightarrow s'$:

$$\left(\bar{c}_{s'}|_{\text{Spec}_{s'}}, q'_{s'} \right) \circ (h_m, f_m) = \left(\bar{c}_{s'}|_{\text{Spec}_{s'}} \circ h_m, \mathcal{V}_{h_m}(q'_{s'}) \circ f_m \right) = \left(\bar{c}_s|_{\text{Spec}_s}, q'_s \right) \quad .$$

Aus der couniversellen Eigenschaft des Colimes $((c_s, d_s)_{s \in \text{Obj}(\mathcal{S})}, G)$ folgt die Existenz und Eindeutigkeit eines generalisierten Morphismus

$$(h): \quad \left(\widehat{c}|_{\text{Spec}_c}, r \right) : G \rightarrow Q \quad \text{mit} \quad \left(\widehat{c}|_{\text{Spec}_c}, r \right) \circ (c_s, d_s) = \left(\bar{c}_s|_{\text{Spec}_s}, q'_s \right)$$

für alle \mathcal{S} -Objekte s . Mit Hilfe von r definieren wir eine Menge von Konstanten

$$Const = \{ g_R = r(g): \rightarrow \mathcal{V}_{\widehat{c}}(Q)_o \mid o \in SO_c, g \in G_o \},$$

eine $Spec_c(X, G)$ -Algebra

$$R = ((\mathcal{V}_{\widehat{c}}(Q)_o)_{o \in SO_c}, OP_{\mathcal{V}_{\widehat{c}}(Q)} \cup Const)$$

mit der Eigenschaft $\mathcal{V}_{\hookrightarrow}(R) = \mathcal{V}_{\widehat{c}}(Q)$ (\hookrightarrow bezeichnet die Inklusion von $Spec_c(X)$ in $Spec_c(X, G)$) und die zweite Komponente des generalisierten Morphismus von $G(X)$ nach Q

$$\widehat{d}: G(X) \rightarrow \mathcal{V}_{\widehat{c}}(Q) = \mathcal{V}_{\hookrightarrow}(eval: T_{Spec_c(X, G)} \rightarrow R).$$

Das Paar $(\widehat{c}, \widehat{d})$ bildet einen generalisierten Morphismus von $G(X)$ nach Q . Um für alle \mathcal{S} -Objekte s dessen Eigenschaft

$$(\widehat{c}, \widehat{d}) \circ (c_s^x, c_s^y(G) \circ \mathcal{F}(d_s)) = (\overline{c}_s, q_s)$$

nachzuweisen, muß

$$\mathcal{V}_{c_s^x}(\widehat{d}) \circ c_s^y(G) \circ \mathcal{F}(d_s) = q_s$$

gezeigt werden. Der Beweis dazu erfolgt für alle \mathcal{S} -Objekte s mit Hilfe struktureller Induktion über den Elementen der Termalgebra zur Spezifikation $Spec_s(X, A_s)$ (\rightarrow 6.2.15). Faktorisiert nach den Spezifikationsgleichungen ergeben sich daraus die Elemente (Kongruenzklassen) in $A_s(X)$, der Domäne von q_s .

Für alle Sorten o aus $Spec_s(X, A_s)$ und Terme t aus $T_{OP_s(X, A_s)_o}$:

Induktionsbasis: $t \in OP_s^{\lambda, o}$. Es gilt: $t \in A_s$ oder $t \notin A_s$

$t \in A_s$	Annahme
\implies	
$\mathcal{V}_{c_s^x}(\widehat{d}) \circ c_{s_G}^y \circ \mathcal{F}(d_s)([t])$	
$= \mathcal{V}_{c_s^x}(\widehat{d})([d_s(t)])$	(e), (f)
$= \mathcal{V}_{c_s^x}(\widehat{d})(d_s(t)_{T_{Spec_c(X, G)}})$	$d_s(t) \in G_{c^{so}(o)}$
$= d_s(t)_R$	Def. \widehat{d}
$= r(d_s(t))$	Def. $Const$
$= \mathcal{V}_{c_s}(r) \circ d_s(t)$	$d_s(t) \in \mathcal{V}_{c_s}(G)_o$
$= q'_s(t)$	(h)
$= q_s([t])$	Def. q'_s
$t \notin A_s$	Annahme

\implies

$$\begin{aligned}
& \mathcal{V}_{c_s^x}(\widehat{d}) \circ c_{s_G}^{\mathcal{V}} \circ \mathcal{F}(d_s)([t]) \\
&= \mathcal{V}_{c_s^x}(\widehat{d}) \circ c_{s_G}^{\mathcal{V}} \circ \mathcal{F}(d_s)(t_{A_s(X)}) && t \notin A_s, \text{ Q.-T.-Alg.} \\
&= t_{\mathcal{V}_{\bar{c}_s}(Q)} && \text{Alg.-Mor.-Eigens., (g)} \\
&= q_s(t_{A_s(X)}) && \text{Alg.-Mor.-Eigens.} \\
&= q_s([t]) && \text{Quotiententermalg.}
\end{aligned}$$

Induktionsschritt: $t = op(t_1, \dots, t_n)$ und für $i = 1, \dots, n$

$$\mathcal{V}_{c_s^x}(\widehat{d}) \circ c_s^{\mathcal{V}}(G) \circ \mathcal{F}(d_s)([t_i]) = q_s([t_i])$$

$$\begin{aligned}
& \mathcal{V}_{c_s^x}(\widehat{d}) \circ c_s^{\mathcal{V}}(G) \circ \mathcal{F}(d_s)([t]) \\
&= \mathcal{V}_{c_s^x}(\widehat{d}) \circ c_s^{\mathcal{V}}(G) \circ \mathcal{F}(d_s)(op_{A_s(X)}([t_1], \dots, [t_n])) && \text{Quotiententermalg.} \\
&= op_{\mathcal{V}_{\bar{c}_s}(Q)}(\mathcal{V}_{c_s^x}(\widehat{d}) \circ c_s^{\mathcal{V}}(G) \circ \mathcal{F}(d_s)([t_1]), \dots, && \text{Alg.-Mor.-Eigens.} \\
&\quad \mathcal{V}_{c_s^x}(\widehat{d}) \circ c_s^{\mathcal{V}}(G) \circ \mathcal{F}(d_s)([t_n])) \\
&= op_{\mathcal{V}_{\bar{c}_s}(Q)}(q_s([t_1]), \dots, q_s([t_n])) && \text{Ind. Annahme, (g)} \\
&= q_s(op_{A_s(X)}([t_1], \dots, [t_n])) && \text{Alg.-Mor.-Eigens.} \\
&= q_s([t]) && \text{Quotiententermalg.}
\end{aligned}$$

Zusammen mit (g) folgt daraus:

$$(\widehat{c}, \widehat{d}) \circ (c_s^x, c_s^{\mathcal{V}}(G) \circ \mathcal{F}(d_s)) = (\widehat{c} \circ c, \mathcal{V}_{c_s^x}(\widehat{d}) \circ c_s^{\mathcal{V}}(G) \circ \mathcal{F}(d_s)) = (\bar{c}_s, q_s).$$

Für jeden anderen generalisierten Morphismus

$$(i): \quad (\widetilde{c}, \widetilde{d}): G(X) \rightarrow Q \quad \text{mit} \quad (\widetilde{c}, \widetilde{d}) \circ (c_s^x, c_s^{\mathcal{V}}(G) \circ \mathcal{F}(d_s)) = (\bar{c}, q)$$

gilt, wegen $\widetilde{c} \circ c_s^x = \bar{c}$ für alle \mathcal{S} -Objekte s und der Eindeutigkeit von \widehat{c} , $\widetilde{c} = \widehat{c}$.

Um $\widetilde{d} = \widehat{d}$ zu zeigen genügt es nachzuweisen, daß \widetilde{d} die Konstanten $g_{T_{Spec_c(X,G)}}$ bewahrt. In diesem Fall wäre \widetilde{d} ein $Spec_c(X, G)$ -Algebromorphismus von $T_{Spec_c(X,G)}$ nach R und damit bis auf Umbenennung gleich $eval_R$. Da \widehat{d} als Vergißbild von $eval_R$ unter $\mathcal{V}_{\hookrightarrow}$ definiert ist und $\mathcal{V}_{\hookrightarrow}$ die Morphismen unverändert läßt,²³ folgt daraus $\widetilde{d} = \widehat{d}$.

²³der Vergißfunktork zur Inklusion von $Spec_c(X)$ in $Spec_s(X, G)$ vergißt lediglich die Konstanten zu den Elementen aus G

Wir definieren den $Spec_c$ -Algebromorphismus

$$\tilde{d}': G \rightarrow \mathcal{V}_{\mathcal{C}}(Q) \quad \text{mit} \quad \tilde{d}'_o(g) = \tilde{d}'_o([g])$$

für alle $Spec_c$ -Sorten o und Elemente g aus der o -Trägermenge von G und zeigen für alle \mathcal{S} -Objekte $s: \mathcal{V}_{c_s}(\tilde{d}') \circ d_s = q'_s$.

$\forall o \in SO_s, b \in B_o$:

$$\begin{aligned} & \mathcal{V}_{c_s}(\tilde{d}') \circ d_s(b) \\ &= \mathcal{V}_{c_s^x}(\tilde{d}')([d_s(b)]) && \text{Def. } \tilde{d}' \\ &= \mathcal{V}_{c_s^x}(\tilde{d}') \circ c_s^y(G) \circ \mathcal{F}(d_s)([b]) && \text{(e), (f)} \\ &= q_s([b]) && \text{(i)} \\ &= q'_s(b) && \text{Def. } q'_s \end{aligned}$$

Mit der Eindeutigkeit von r bzgl. dieser Eigenschaft folgt daraus $\tilde{d}' = r$ und zusammen mit den Definitionen von \tilde{d}' und $Const$ gilt:

$$\forall g \in G: \tilde{d}'(g_{T_{Spec_c}(X,G)}) = \tilde{d}'([g]) = \tilde{d}'(g) = r(g) = g_R,$$

d.h. \tilde{d}' ist mit den Konstanten zu den Elementen aus G verträglich. □

□

Literaturverzeichnis

- [BCML91] E. Battiston, F. De Cindio, G. Mauri, and Rapanotti L. Morphisms and Minimal Models for OBJSA Nets. Technical Report 4/26, CONSIGLIO NAZIONALE DELLE RICERCHE, 1991.
- [BDCM88] E. Battiston, F. De Cindio, and G. Manri. OBJSA Nets a class of high-level nets having objects as domains. *Lecture Notes in Computer Science*, 340, 1988.
- [BF86] E. Best and C. Fernandez. Notations and Terminology on Petri Net Theory. Arbeitspapiere der GMD 195, 1986.
- [BG91] D. Buchs and N. Guelfi. CO-OPN: A CONCURRENT OBJECT ORIENTED PETRI NET APPROACH. *Lecture Notes in Computer Science*, 1991.
- [BHR84] SD. Brooks, C. Hoare, and AW. Roscoe. A theory of communicating sequential processes. *ACM*, 31, 1984.
- [BW90] M. Barr and C. Wells. *Category Theory for Computing Science*. Prentice Hall, International Series in Computer Science, 1990.
- [Chm85] S. Chmielnicki. *Flexible Fertigungssysteme: Simulation der Prozesse als Hilfsmittel zur Planung und zum Test von Steuerungsprogrammen*. Springer Verlag, Berlin Heidelberg New York, 1985.
- [DGK⁺92] Jörg Desel, Dominik Gomm, Ekkart Kindler, Barbara Paech, and Rolf Walter. Bausteine eines kompositionalen Beweiskalküls für netzmodellierete Systeme. SFB-Bericht 342/16/92 A, Technische Universität München, Juli 1992.
- [DHP90] C. Dimitrovici, U. Hummert, and L. Petrucci. The properties of algebraic nets in some semantics. In *11th International Conference on Applications and Theory of Petri nets*, Paris, 1990.
- [DHP91] C. Dimitrovici, U. Hummert, and L. Petrucci. Semantics, Composition and Net Properties of Algebraic High-Level Nets. *Lecture Notes in Computer Science*, 483, 1991.

- [DMM89] P. Degano, J. Meseguer, and U. Montanari. Axiomatizing Net Computations and Processes. *IEEE Symposium on Logic in Computer Science*, pages 175–185, 1989.
- [EBO91] H. Ehrig, M. Baldamus, and F. Orejas. New concepts for amalgamation and extension in the framework of specification logics. Technical Report 91/05, Technische Universität Berlin, 1991.
- [EGR91] H. Ehrig and M. Grosse-Rode. Structural Theory of Algebraic Specifications in a Specification Logic; Part 1. Technical Report 91/23, Technische Universität Berlin, 1991.
- [EGR94] H. Ehrig and M. Grosse-Rode. Functorial theory of parameterized specifications in a general specification framework. *Theoretical Computer Science*, 135:221–266, 1994.
- [EGRH92] H. Ehrig, M. Grosse-Rode, and A. Heise. Specification Techniques for Concurrent and Distributed Systems. Technical Report 92/5, Technische Universität Berlin, 1992.
- [Ehr89] H. Ehrig. Algebraic specification of modules and modular software systems within the framework of specification logics. Technical Report 89/17, Technische Universität Berlin, 1989.
- [EM85] H. Ehrig and B. Mahr. *Fundamentals of Algebraic Specifications 1*. Springer Verlag, Berlin Heidelberg New York, 1985.
- [EM90] H. Ehrig and B. Mahr. *Fundamentals of Algebraic Specifications 2*. Springer Verlag, Berlin Heidelberg New York, 1990.
- [EW91] H. Eertink and D. Wolz. Symbolic Execution of Lotos. Technical Report 2304, ESPRIT, 1991.
- [Gro81] CIP Language Group. Report on a Wide Spectrum Language for Program Specification and Development. *Lecture Notes in Computer Science*, 183, 1981.
- [GTW78] J.V. Goguen, J.W. Thatcher, and E.G. Wagner. *An Initial Algebra Approach to the Specification, Correctness and Implementation of Abstract Data Types*, pages 80 – 149. Prentice Hall, 1978.
- [Han88] H. Hansen. The act-system - experiences and future enhancements. *Lecture Notes in Computer Science*, 332, 1988.
- [HD90] A. Heise and C. Dimitrovici. Transformation und Komposition von P/T-Netzen unter Erhaltung wesentlicher Eigenschaften. Technical Report 342-29-90 A, Sonderforschungsbereich 342: Methoden und Werkzeuge für die Nutzung paralleler Rechnerarchitekturen, München, 1990.

- [Hei88] A. Heise. High Level Nets with Capacities. Studienarbeit an der TU-Berlin, 1988.
- [HJS90] P. Huber, K. Jensen, and R. M. Shapiro. Hierachies in Coloured Petri Nets. *Lecture Notes in Computer Science*, 483:313–341, 1990.
- [Hoa72] C. Hoare. Proof of Correctness of data representations. *Acta Informatica*, 1:271 – 281, 1972.
- [HS73] H. Herrlich and G. E. Strecker. *Category Theroy*. Allyn and Bacon Inc., Boston, 1973.
- [Hum89] U. Hummert. Algebraische High-Level-Netze. Dissertation, Technische Universität Berlin, 1989.
- [Int87] International Standard, ISO. *Information Processing Systems-Open System Interconnection-LOTOS-A formal description technique based on temporal ordering behavior*, 1987.
- [ISO83] ISO/TC97/SC16/WG1N157. *Information Processing System - Open Systems Interconnection - The definition of the specification language LOTOS*, 1983.
- [JCHH91] K. Jensen, S. Christensen, M. Holla, and P. Huber. *Design/CPN. A Reference*. 125 Cambridge Park Drive, Cambridge MA 02140, USA, 1991.
- [Jen81] K. Jensen. Coloured Petri Nets and the Invariant-Method. *Theoretical Computer Science*, 14, 1981.
- [Jen90] K. Jensen. Coloured Petri Nets: A High Level Language for System Design and Analysis. *Lecture Notes in Computer Science*, 483:342–416, 1990.
- [Jen92] K. Jensen. Basic Concepts, Analysis Methods and Practical Use. Volume 1. ETACS Monographs on Theoretical Computer Science, 1992.
- [Jen94] K. Jensen. Basic Concepts, Analysis Methods and Practical Use. Volume 2. ETACS Monographs on Theoretical Computer Science, 1994.
- [KM95] H.-J. Kowalsky and G. O. Michler. *Lineare Algebra*. de Gruyter, Berlin, New York, 10. edition, 1995.
- [Kra87] B. Kraemer. SEGRAS a formal language combining Petri nets and Abstract Data Types for specifying distributed systems. In *Proc. 9th Int. Conf. on Software Engineering*, pages 116–125, 1987.
- [Lil91] J. Lilius. On the Compositionality and Analisys of Algebraic High-Level Nets. Technical Report 16/ISBN 951-22-0828-8, Helsinki University of Technology, Digital Systems Laboratory, October 1991.

- [Lil95] J. Lilius. ON THE STRUCTURE OF HIGH-LEVEL NETS. Technical Report 33/ISBN 951-22-2522-0, Helsinki University of Technology, Digital Systems Laboratory, March 1995.
- [Mil80] R. Milner. A calculus for communication systems. *Lecture Notes in Computer Science*, 92, 1980.
- [Mil95] R. Milner. *Action Structures and the Pi-Calculus*. Springer Verlag, Berlin Heidelberg New York, 1995.
- [ML70] S. Mac Lane. *Categories for the working mathematician*. Springer Verlag, Berlin Heidelberg New York, 1970.
- [MM90] J. Meseguer and U. Montanari. Petri Nets Are Monoids. *Theoretical Computer Science*, 88, 1990.
- [Pad96] J. Padberg. Abstract Petri Nets: A Uniform Approach and Rule-Based Refinement. Dissertation, Technische Universität, Berlin, 1996.
- [Pet62] C.A. Petri. Kommunikation mit Automaten. Dissertation, Universität Bonn, 1962.
- [Rei85] W. Reisig. *Petri Nets*. Springer Verlag, Berlin Heidelberg New York, 1985.
- [Rei91] W. Reisig. Petri Nets and Algebraic Specifications. *Theoretical Computer Science*, 80:1–34, Mai 1991.
- [Rei92] W. Reisig. Elements of a Temporal Logic Coping with Concurrency. SFB-Bericht 342/23/92 A, Technische Universität München, 1992.
- [Sch88] H.W. Schmidt. Specification and Correct Implementation of Non-Sequential Systems Combining Abstract Data Types and Petri Nets. Dissertation, Universität Bremen, November 1988.
- [Sta90] P. Starke. *Analyse von Petri-Netz-Modellen*. Teubner Verlag, Stuttgart, 1990.
- [Val81] R. Valk. Generalizations of Petri Nets. *LNCS*, 458:140–155, 1981.
- [Vau86] J. Vautherin. Parallel Systems Specifications with Colored Petri Nets and Algebraic Abstract Data Types. In *7th European Workshop on Applications and Theory of Petri Nets*, 1986.
- [Vie85] B. Viehweger. *Planung von Fertigungssystemen mit automatisiertem Werkzeugfluß*, volume 50 of *Produktionstechnik-Berlin*. Carl Hanser Verlag, 1985.
- [Win87] G. Winskel. Petri Nets, Algebras, Morphisms, and Compositionality. *Theoretical Computer Science*, 72, 1987.