Technische Universität Dresden

Fakultät Informatik

Diplomarbeit

zum Thema

Lineare Logik und Quantoren erster Ordnung

von
Hendrik Tews
geboren am 22.8.69 in Leipzig

Eingereicht am 8. November 1995

Verantwortlicher Hochschullehrer und Betreuer: Prof. Dr. rer. nat. habil. H. Reichel

Diese Diplomarbeit wurde mit dem Textsatzsystem LaTeX erstellt. Die kategorientheoretischen Diagramme und Beweisbäume wurden mit Hilfe von Makropaketen von Paul Taylor (Commutative diagrams, Version 3.82) und Makoto Tatsuta (proof.sty, Version 1.0) erstellt.

Inhaltsverzeichnis

Abbildungsverzeichnis

1. Einführung

Funktionale Programmiersprachen konnten durch die Einführung statischer Typsysteme wesentlich weiterentwickelt werden. Eines der bemerkenswerten Ergebnisse in diesem Zusammenhang ist der Curry–Howard–Isomorphismus [GLT88], der es gestattet, den Typ eines Programms als Formel einer intuitionistischen Logik[1] und das Programm selbst als einen Beweis aufzufassen. Wesentlich für diesen Zusammenhang sind kartesisch abgeschlossene Kategorien, die sowohl ein Modell des einfach getypten λ–Kalküls [Bar92] als auch eines für intuitionistische Logik sind.

Auf dem Gebiet der parallelen Sprachen sind die Fortschritte bescheidener geblieben. Es existieren eine Vielzahl verschiedener Kalküle zur Modellierung von Parallelität. Aber kein Kalkül besitzt eine ähnlich unangefochtene Stellung wie der λ–Kalkül innerhalb der funktionalen Sprachen.

Eine bisher nicht verfolgte Möglichkeit, Fortschritte bei der Modellierung paralleler Sprachen zu erzielen, besteht darin, in Analogie zu der Situation bei funktionalen Sprachen nach einem Curry–Howard–Isomorphismus für parallele Sprachen zu suchen. Mit klassischer Linearer Logik kann der Verbrauch von Ressourcen modelliert werden. Durch Robert A. Seely wurde in [See89] ein kategorientheoretisches Modell für Lineare Logik vorgestellt: um einen Comonad bereicherte $*$–autonome Kategorien.

In einer Reihe von Publikationen [AGN93, Abr94] entwickelt Samson Abramsky diese Idee und faßt Prozeßspezifikationen, Lineare Logik und $*$–autonome Kategorien zu einem einheitlichen System, den Interaktionskategorien zusammen. Prozesse werden durch Morphismen der Kategorie modelliert, die Komposition von Morphismen ist Synchronisation, bzw. in Begriffen von Robin Milners CCS [Mil89], parallele Komposition in Verbindung mit Restriktion.

Eine allgemeine Definition von Interaktionskategorien gibt es noch nicht. In den betreffenden Publikationen werden lediglich Beispiele genannt. Am intensivsten untersucht ist die Interaktionskategorie der synchronen Prozesse, $\mathcal{SProc}$. In Anlehnung an den Curry–Howard–Isomorphismus ist $\mathcal{SProc}$ ein Modell für propositionale Lineare Logik, das heißt für klassische Lineare Logik ohne Quantoren.

In der Untersuchung der Semantik von formalen Systemen gibt es mindestens zwei Traditionen. Denotatielle Semantik ordnet Formeln Werte aus einem gewählten Interpretationsbereich, für gewöhnlich den Wahrheitswerten, zu. Der Prozeß des Berechnens oder Beweisens wird nicht modelliert.

Der Gegensatz dazu ist eine Semantik in der Tradition von Heyting. Hier stehen Beweise im Mittelpunkt. Zum Beispiel besteht das Modell einer Konjunktion $A \wedge B$ aus einem Paar zweier Beweise, jeweils einen für A und B. Eine Implikation $A \Longrightarrow B$ wird durch eine Funktion interpretiert, die jedem Beweis von A einen Beweis von B zuordnet.

Heyting'sche Semantik ist eng verknüpft mit intuitionistischer Logik und dem Curry–

[1] Siehe zum Beispiel [FKKK94] für eine zusammenfassende Darstellung.

Howard–Isomorphismus.

Betrachtet man Logiken mit Quantoren, können solche Semantiken mit Fibrationen oder indizierten Kategorien generiert werden.

Schwerpunkt der Aufgabenstellung dieser Diplomarbeit war die Frage, ob sich eine Semantik für Lineare Logik mit Quantoren finden läßt, indem das Modell der Interaktionskategorie $\mathcal{SP}roc$ ausgebaut oder bereichert wird. Dazu war eine Erarbeitung des Begriffes der Fibration notwendig. Die Motivation für diese Arbeit bildeten die im Großen Beleg [Tew95] untersuchten Schwächen von $\mathcal{SP}roc$ bei der Modellierung von zusammengesetzten Systemen.

Im Verlauf der Bearbeitung des Themas wurde klar, daß eine Darstellung Linearer Logik, so wie sie für die Zwecke dieser Arbeit benötigt wurde, noch nicht veröffentlicht ist. Eine umfassende Untersuchung einer „prädikatenlogischen Version" Linearer Logik hätte jedoch den Rahmen dieser Arbeit gesprengt. Deshalb wird vom formalen Apparat der Linearen Logik nur soviel entwickelt, wie für den Rest der Arbeit absolut notwendig ist.

Zur Erweiterung der Interaktionskategorie $\mathcal{SP}roc$ wurden zwei Ansätze untersucht. Zum einen die Bildung einer Potenzmengenfibration getypter Prozesse $\textsf{\textbf{ProcSet}} \longrightarrow \mathcal{SP}roc$. Ein großer Teil der Struktur der Kategorie $\mathcal{SP}roc$ kann in diese Prozeßmengenfibration übertragen werden. In **ProcSet** können auch Quantoren und die Regeln zur Einführung von Quantoren in Formeln interpretiert werden. Allerdings ist **ProcSet** kein Modell für Lineare Logik, es fehlt die Struktur für die lineare Negation und die Modalitäten. Im letzten Fall konnte das auf Eigenschaften von $\mathcal{SP}roc$ zurückgeführt werden.

Der zweite Ansatz, eine Teilobjektfibration über $\mathcal{SP}roc$ aufzubauen, lieferte keine Ergebnisse und wird deshalb in der vorliegenden Arbeit nicht beschrieben. Es gelang nicht, eine geeignete Charakterisierung für die Teilobjekte in $\mathcal{SP}roc$ zu finden.

Die Arbeit ist wie folgt gegliedert: Kapitel 2 enthält eine Reihe von Definitionen, die für das Verständnis der Arbeit notwendig sind. Kapitel 3 beschreibt die Variante der Linearen Logik, für die später Modelle entwickelt werden sollen. Im Kapitel 4 wird die Struktur, die ein kategorientheoretisches Modell für die im Kapitel 3 vorgestellte Logik haben muß, allgemein definiert. Im darauffolgenden Kapitel 5 wird ein konkretes Beispiel vorgestellt: Die Interaktionskategorie $\mathcal{SP}roc$. Kapitel 6 untersucht die Struktur der Potenzmengenfibration getypter Prozesse über $\mathcal{SP}roc$. Im letzten Kapitel werden die erzielten Ergebnisse bewertet.

Ich möchte allen aus der „Familie" danken, die an dem Wochenende, an dem Dresden sich gegen die Zukunft entschieden hat, nichts wichtigeres zu tun hatten, als Kommafehler in dieser Arbeit zu suchen und Essen für mich zu kochen.

2. Grundlagen

Aufgrund des Umfangs war es nicht möglich, alle in der vorliegenden Arbeit verwendeten Konzepte und Werkzeuge in einem Grundlagenkapitel vorzustellen. Deshalb wird folgender Kompromiß gewählt. Ich gehe davon aus, daß der Leser Grundkenntnisse in Kategorientheorie besitzt, die etwa den ersten beiden Kapiteln von [Pie91] oder dem Umfang von [Wal92] entsprechen. Darüber hinausgehende Begriffe werden kurz definiert, auch wenn sie in Standardwerken des Gebietes diskutiert werden.

Da alle für diese Arbeit ausgewerteten Publikationen in Englisch abgefaßt sind, ergab sich für mich das Problem, für bestimmte Begriffe eine deutsche Übersetzung zu finden. Falls sich im Deutschen noch keine Übersetzung durchgesetzt hat, übertrage ich Fachbegriffe so wörtlich wie möglich. In einigen Fällen, zum Beispiel bei „cloven fibration“ oder „split fibration“, hätte das aber keinen Sinn ergeben. Aus diesem Grund werden bestimmte Begriffe nicht übersetzt.

Im folgenden Abschnitt 2.1 werden Begriffe und Operationen definiert, die in dieser Arbeit häufig verwendet werden, aber in der Standardliteratur so nicht vorkommen. Zugleich werden in den Fällen, in denen sich in der Fachliteratur noch keine Konvention durchgesetzt hat, die in dieser Arbeit verwendeten Bezeichnungen vorgestellt.

Im Abschnitt 2.2 werden Comonads und Comonoide definiert. Abschnitt 2.3 kann als kurze Einführung in die Theorie der Fibrationen betrachtet werden.

2.1 Bezeichnungskonventionen

> In our notation we use brackets no more than neccessary (locigally or psychologically).
> S. Eilenberg and G. M. Kelly

Seien Ω und Σ Alphabete und $f : \Omega \longrightarrow \Sigma$ eine Funktion. Mit $f^* : \Omega^* \longrightarrow \Sigma^*$ soll die zwischen den freien (Wort–)Monoiden Ω^* und Σ^* durch f induzierte Funktion bezeichnet werden.

Sei zum Beispiel $\pi_1 : \Omega \times \Sigma \longrightarrow \Omega$ die erste Projektion des kartesischen Produktes zweier Mengen, die jedes Paar (o, s) auf die erste Komponente o abbildet. Dann ist $\pi_1^* : (\Omega \times \Sigma)^* \longrightarrow \Omega^*$ eine Funktion, die aus einem Wort $\rho \in (\Omega \times \Sigma)^*$ ein Wort $\pi_1^*(\rho)$ extrahiert, indem quasi π_1 auf jedes Zeichen in ρ angewandt wird.

Ein **markiertes Transitionssystem** ist ein Tripel $(X, L, \longrightarrow)$, wobei X eine Menge von Zuständen, L eine Menge von Markierungen oder Aktionen und $\longrightarrow \subseteq X \times L \times X$ die Transitionsrelation ist. In dieser Arbeit wird anstelle von $(x, a, y) \in \longrightarrow$ einfach $x \xrightarrow{a} y$ geschrieben. Markierte Transitionssysteme können als markierte gerichtete Graphen dargestellt werden. Dabei entsprechen die Zustände den Knoten. Die Transitionsrelation wird durch die Menge der Kanten dargestellt.

Unter einem **Prozeß** wird in dieser Arbeit eine abstrakte Einheit verstanden, die verschiedene Zustände einnehmen kann und eine Reihe von Kanälen besitzt. Mittels von außen an den Kanälen beobachtbaren Aktionen kann ein Prozeß zwischen zwei Zuständen wechseln. Für jeden Kanal existiert eine Menge zulässiger Aktionen.

Es sollen nur synchrone Prozesse betrachtet werden. Synchron bedeutet hierbei, daß an allen Kanälen, gleichsam von zentraler Stelle getaktet, zu jeder Zeiteinheit genau eine Aktion sichtbar ist. Asynchrone Prozesse, bei denen an einigen Kanälen beliebig lange keine Aktion sichtbar werden kann, werden nicht in Betracht gezogen. In dieser Arbeit werden Prozesse mittels Transitionssystemen bzw. Synchronisationsbäumen beschrieben.

Die Menge der **Synchronisationsbäume** über einem Alphabet Σ ist die größte Lösung der Gleichung

$$\mathsf{ST}_\Sigma \quad = \quad \mathbb{P}(\Sigma \times \mathsf{ST}_\Sigma)$$

Mit Hilfe Peter Aczels Theorie nicht wohlfundierter Mengen [Acz88] läßt sich für diese Gleichung eine Lösung bestimmen. Und zwar wird ST_Σ durch die finale Coalgebra zum Funktor

$$X \longmapsto \mathbb{P}(\Sigma \times X)$$

charakterisiert.

Sei p_0 ein Prozeß über dem Alphabet L. Die Menge der Pfade $\mathsf{traces}(p_0)$, die p_0 realisieren kann, wird wie folgt definiert:

$$\mathsf{traces}(p_0) \quad \stackrel{\text{def}}{=} \quad \{l_1 \dots l_n \mid n \in \mathbb{N} \text{ und } \forall i < n \,.\, \exists p_{i+1} \,.\, p_i \xrightarrow{l_{i+1}} p_{i+1}\}$$

Man beachte, daß die Definition so gewählt ist, daß der leere Pfad $\langle\rangle$ immer in $\mathsf{traces}(p_0)$ enthalten ist.

Eine weitere wesentliche Eigenschaft der Menge $\mathsf{traces}(p_0)$ ist ihre Abgeschlossenheit gegenüber Präfixbildung, das heißt, wenn ω ein Wort aus L^* und $\omega l \in \mathsf{traces}(p_0)$ ist, dann gilt auch $\omega \in \mathsf{traces}(p_0)$.

Abschließend einige Bemerkungen zur hier verwendeten kategorientheoretischen Sprache: Wenn $Y \xrightarrow{f} Z$ und $X \xrightarrow{g} Y$ zwei Morphismen sind, dann wird die Komposition mit $- \circ -$ in algebraischer Ordnung $X \xrightarrow{f \circ g} Z$ dargestellt.

Die disjunkte Vereinigung $\uplus$ bestimmt zu je zwei Morphismen $f : X \longrightarrow Z$ und $g : Y \longrightarrow Z$ den Morphismus $[f, g]$. Die Injektionen sind inl und inr.

Die Projektionen des kartesischen Produktes sind π_1 bzw. π_2. Die Paarung zweier Morphismen $Z \xrightarrow{f} X$ und $Z \xrightarrow{g} Y$ ist $Z \xrightarrow{\langle f,g\rangle} X \times Y$.

Im Kontext Linearer Logik werden die Konnektoren der Linearen Logik verwendet. Dort ist $X \,\&\, Y$ das Produkt, $X \oplus Y$ die Summe, $\mathbf{0}$ das initiale und $\top$ das terminale Objekt.

Sei $F \dashv G : \mathbb{C} \longrightarrow \mathbb{D}$ eine Adjunktion und $F(X) \xrightarrow{f} Y$ ein Morphismus in $\mathbb{C}$. Der zu f gehörende transponierte Morphismus in $\mathbb{D}$ wird mit $f^\vee$ bezeichnet. Umgekehrt wird durch $X \xrightarrow{g} G(Y)$ eineindeutig der Morphismus $g^\wedge$ in $\mathbb{C}$ bestimmt.

Die Kategorie (kleiner) Kategorien heißt **Cat**, die Kategorie aller Mengen **Set**. Ist $\mathbb{C}$ eine Kategorie, dann ist $|\mathbb{C}|$ die Menge der Objekte und $\mathbb{C}(X,Y)$ die Menge der Morphismen zwischen den Objekten X und Y aus $\mathbb{C}$.

Ich hoffe mich in der Arbeit weitgehend an folgende Bezeichnungskonvention gehalten zu haben.

Lateinische Großbuchstaben	$A, B, \ldots$	logische Formeln
	F, G	Funktoren
	I, J, K	Objekte einer Basiskategorie
	M, N, Q	Mengen von Prozessen
	T	Monads
	$X, Y, \ldots$	Objekte einer Kategorie
Lateinische Kleinbuchstaben	$a, b, c, \ldots$	Aktionen
	$f, g, \ldots$	Morphismen
	p	Fibrationen
	$p, q, r, \ldots$	Prozesse
	r, s, t	Terme
	u, v	Variablen einer Typtheorie
	u, v, w	Morphismen einer Basiskategorie
Kalligraphische Buchstaben	$\mathcal{T}, \mathcal{F}$	Mengen von Typen oder Operationssymbolen
Fette Großbuchstaben	$\mathbb{C}, \mathbb{E}, \mathbb{B}$	Kategorien
	$\mathbb{N}$	Menge der natürlichen Zahlen
Große griechische Buchstaben	$\Gamma, \Delta, \ldots$	Listen von logischen Formeln
	Ψ	indizierte Kategorien
	Σ	Alphabete oder Signaturen
	Υ, Θ, Ξ	Kontexte
kleine griechische Buchstaben	$\epsilon, \eta, \mu, \ldots$	natürliche Transformationen
	κ, ψ	ausgezeichnete Morphismen

2.2 Comonads und Comonoide

> Hence we pursue the strategy again of refering to the literature (namly MacLane's book [ML71]).
>
> Axel Poigné

In den Kapiteln 4 und 5 werden die Begriffe des Comonads und des Comonoids zur kategorientheoretischen Interpretation der modalen Operatoren von Linearer Logik benötigt. Zur besseren Übersichtlichkeit werden die Definitionen der beiden Begriffe in diesem Abschnitt zusammengefaßt. Die Darstellung ist allerdings auf die für diese Arbeit wesentlichen Aspekte reduziert. Der interessierte Leser wird auf [ML71] oder [Poi92] verwiesen.

Comonads

Monads und Comonads sind kategorientheoretische Strukturen, die eng mit Adjunktionen verknüpft sind. Jede Adjunktion definiert je einen Monad und einen Comonad. Umgekehrt kann aus einem Monad bzw. einem Comonad auf einer Kategorie $\mathbb{C}$ eine zweite Kategorie[2] konstruiert werden, so daß zwischen beiden Kategorien eine Adjunktion besteht, die genau den gegebenen Monad bzw. Comonad definiert. In einigen Publikationen werden Monads auch als Tripel und Comonads als Cotripel bezeichnet. In dieser Arbeit bezeichnet jedoch Tripel immer ein 3–Tupel.

Definition 2.1 (Monad) *Ein **Monad** über $\mathbb{C}$ ist ein Tripel (T, μ, η), bestehend aus einem Endofunktor $\mathbb{C} \xrightarrow{T} \mathbb{C}$ und zwei natürlichen Transformationen $\mu : T^2 \longrightarrow T$ und $\eta : \mathsf{Id} \longrightarrow T$, so daß die folgenden Diagramme kommutieren:*

$$\begin{tikzcd}
T^3 \arrow[r, "T\mu"] \arrow[d, "\mu T"'] & T^2 \arrow[d, "\mu"] \\
T^2 \arrow[r, "\mu"'] & T
\end{tikzcd}
\qquad
\begin{tikzcd}
T \arrow[r, "T\eta"] \arrow[dr, equal] & T^2 \arrow[d, "\mu"] & T \arrow[l, "\eta T"'] \arrow[dl, equal] \\
& T &
\end{tikzcd}$$

Beide Diagramme der Definition 2.1 liegen in der Funktorkategorie $\mathbb{C}^{\mathbb{C}}$ und legen die Eigenschaften der Verkettung von Morphismen bestimmter natürlicher Transformationen fest. $T\mu$ ist die natürliche Transformation, die man erhält, wenn T auf jeden Morphismus von μ angewendet wird. Die natürliche Transformation μT dagegen besteht aus den Morphismen $\{\mu_{T(X)} \mid X \in |\mathbb{C}|\}$.

Das linke Diagramm in der Definition 2.1 besagt damit, daß für alle $X \in |\mathbb{C}|$ in der Kategory $\mathbb{C}$ die Gleichung $\mu_X \circ T(\mu_X) = \mu_X \circ \mu_{T(X)}$ gilt.

[2] Zum Beispiel Kleisli's Kategorie [BW84]

Jede Adjunktion $F \dashv G : \mathbb{D} \longrightarrow \mathbb{C}$ definiert mit $(G \circ F\,,\, G\epsilon F\,,\, \eta)$ einen Monad. Ein Comonad ist ein dualisierter Monad.

Definition 2.2 (Comonads) *Eine* ***Comonad*** *über $\mathbb{C}$ ist ein Tripel (T, δ, ϵ), bestehend aus einem Endofunktor $\mathbb{C} \xrightarrow{T} \mathbb{C}$ und zwei natürlichen Transformationen $\delta : T \longrightarrow T^2$ und $\epsilon : T \longrightarrow \mathsf{Id}$, so daß die folgenden Diagramme kommutieren:*

$$\begin{tikzcd}
T \arrow[r, "\delta"] \arrow[d, "\delta"'] & T^2 \arrow[d, "T\delta"] \\
T^2 \arrow[r, "\delta T"'] & T^3
\end{tikzcd}
\qquad
\begin{tikzcd}
 & T \arrow[dl, equal] \arrow[d, "\delta"] \arrow[dr, equal] & \\
T & T^2 \arrow[l, "T\epsilon"] \arrow[r, "\epsilon T"'] & T
\end{tikzcd}$$

Sowohl Definition 2.1 als auch Definition 2.2 führen den zu definierenden Begriff auf eine ganze Menge von zu testenden Bedingungen zurück. Ernest G. Manes entwickelte eine weniger komplexe, alternative Definition von Monads. Diese alternative Definition wird hier nur für die in dieser Arbeit wichtigen Comonads angegeben.

Satz 2.3 *Sei $(T, \epsilon, (-)^!)$ ein Tripel, so daß*

1. *T eine Funktion $|\mathbb{C}| \longrightarrow |\mathbb{C}|$ und*

2. *ϵ eine Familie von Morphismen $\{T(X) \xrightarrow{\epsilon_X} X \mid X \in |\mathbb{C}|\}$ ist und*

3. *$(-)^!$ eine Operation $\mathbb{C}(T(X), Y) \longrightarrow \mathbb{C}(T(X), T(Y))$, die jedem Morphismus $T(X) \xrightarrow{f} Y$ einen Morphismus $T(X) \xrightarrow{f^!} T(Y)$ zuordnet.*

Das Tripel $(T, \epsilon, (-)^!)$ charakterisiert genau dann einen Comonad auf der Kategorie $\mathbb{C}$, wenn für beliebige Objekte $X, Y, Z \in |\mathbb{C}|$ und Morphismen $Y \xrightarrow{g} Z$ und $X \xrightarrow{f} Y$

$$\begin{aligned} \epsilon_Y \circ f^! &= f \\ {\epsilon_X}^! &= \mathsf{id}_{T(X)} \\ g^! \circ f^! &= (g \circ f^!)^! \end{aligned} \tag{2.4}$$

gilt.

Die Gleichungen 2.4 spezifizieren genau die auf der rechten Seite einer Adjunktion auftretende Struktur.

Beweisidee Ist ein Tripel nach Ernest G. Manes gegeben, läßt sich aufgrund der Gleichungen 2.4 die Funktion T zu einem Funktor

$$T(f) \quad = \quad (f \circ \epsilon)^!$$

erweitern. Ebenfalls aufgrund der Gleichungen 2.4 ist ϵ eine natürliche Transformation. Die zweite natürliche Transformation erhält man als

$$\delta_X \quad = \quad (\mathsf{id}_{T(X)})^!$$

Ist hingegen ein Comonad nach Definition 2.2 gegeben, dann erfüllt das Tripel

$$(T\,,\,\epsilon\,,\,f \longmapsto T(f) \circ \delta)$$

die Gleichungen 2.4. □

Comonoide

Im Gegensatz zum Monad, der eine Eigenschaft für die gesamte Kategorie definiert, spezifiziert die Definition des Comonoiden nur Eigenschaften einzelner Objekte der betreffenden Kategorie. Auch für Comonoide gilt, daß sie dualisierte Monoide sind, die hier aber nicht definiert werden sollen.

Definition 2.5 *Ein Objekt A aus $\mathbb{C}$ ist ein* ***kommutativer Comonoid****, falls es zwei Morphismen $\eta_A : A \longrightarrow I$ und $\mu_A : A \longrightarrow A \otimes A$ gibt, so daß die Diagramme*

$$\begin{tikzcd}
A \otimes A \arrow[d, "\mathsf{id}_A \otimes \mu_A"'] & A \arrow[l, "\mu_A"'] \arrow[r, "\mu_A"] & A \otimes A \arrow[d, "\mu_A \otimes \mathsf{id}_A"] \\
A \otimes (A \otimes A) \arrow[rr, "\mathsf{assoc}_{A,A,A}"'] & & (A \otimes A) \otimes A
\end{tikzcd}$$

$$\begin{tikzcd}
 & A \arrow[dl, "\mathsf{unitl}_A^{-1}"'] \arrow[d, "\mu_A"] \arrow[dr, "\mathsf{unitr}_A^{-1}"] & \\
I \otimes A & A \otimes A \arrow[l, "\eta_A \otimes \mathsf{id}_A"] \arrow[r, "\mathsf{id}_A \otimes \eta_A"'] & A \otimes I
\end{tikzcd}
\qquad
\begin{tikzcd}
 & A \arrow[dl, "\mu_A"'] \arrow[dr, "\mu_A"] & \\
A \otimes A \arrow[rr, "\mathsf{symm}_{A,A}"'] & & A \otimes A
\end{tikzcd}$$

kommutieren.

2.3 Indizierte Kategorien und Fibrationen

> if Q's depend on P's, then Q's are fibred over P's
>
> Bart Jacobs

Im Rahmen dieser Arbeit spielen die Anwendungen indizierter Familien in der Semantik für Prädikatenlogiken bzw. für bestimmte λ–Kalkuli eine Rolle. Dafür ist es notwendig, den Begriff einer indizierten Familie kategorientheoretisch zu erfassen. In diesem Abschnitt werden zwei Möglichkeiten diskutiert: Indizierte Kategorien und Fibrationen. Es wird sich herausstellen, daß beide Ansätze eng miteinander verbunden sind.

Bei der Auswahl der hier vorgestellten Definitionen und Sätze wurden nur die Erfordernisse der vorliegenden Arbeit berücksichtigt. Es wird keinerlei Anspruch auf Vollständigkeit erhoben. Aus Platzgründen können nicht alle Aussagen dieses Abschittes bewiesen werden. Der interessierte Leser sei auf [Jac91] verwiesen. Zum besseren Verständnis werden die Definitionen durch einige Beispiele erläutert.

Indizierte Kategorien

Indizierten Kategorien liegt die Idee zugrunde, daß man eine durch eine Menge I indizierte Familie $(X_i)_{i\in I}$ als eine Funktion $I \longrightarrow \mathbb{X}$ modellieren kann. Verschiedene indizierte Familien über der gleichen Indexmenge I können durch verschiedene Funktionen dargestellt werden. In kategorientheoretische Begriffe übertragen, bedeutet das: die mit Hilfe einer Indexmenge I indizierten Kategorien werden durch die Funktorkategorie $\mathbb{X}^I$ modelliert. Durch weitere Verallgemeinerung auf beliebige Indexmengen gelangt man zu der folgenden Definition.

Definition 2.6 (Indizierte Kategorie) *Eine **indizierte Kategorie** Ψ ist ein Pseudofunktor $\Psi : \mathbb{B}^{\mathsf{op}} \longrightarrow \mathbf{Cat}$. Der zu einem Morphismus $I \xrightarrow{u} J$ gehörende Funktor $\Psi(J) \xrightarrow{\Psi(u)} \Psi(I)$ wird mit u^* bezeichnet. Die zum Pseudofunktor gehörenden natürlichen Isomorphismen $\eta_I : \mathsf{Id} \longrightarrow \mathsf{id}_I^*$ und $\mu_{u,v} : u^* \circ v^* \longrightarrow (v \circ u)^*$ müssen folgende Kohärenzbedingungen erfüllen:*

$$
\begin{tikzcd}
 & u^* \arrow[dl, "\eta_I u^*"'] \arrow[d, equal] \arrow[dr, "u^*\eta_J"] & \\
id_I^* u^* \arrow[r, "\mu_{\mathsf{id}_I,u}"'] & u^* & u^* id_J^* \arrow[l, "\mu_{u,\mathsf{id}_J}"]
\end{tikzcd}
\tag{2.7}
$$

$$
\begin{tikzcd}
u^*v^*w^* \arrow[r, "u^*\mu_{v,w}"] \arrow[d, "\mu_{u,v}w^*"] & u^*(w \circ v)^* \arrow[d, "\mu_{u,w\circ v}"'] \\
(v \circ u)^* w^* \arrow[r, "\mu_{v\circ u,w}"'] & (w \circ v \circ u)^*
\end{tikzcd}
\tag{2.8}
$$

Genau wie in Definition 2.1 ist $u^*\eta_I$ die natürliche Transformation, die die Morphismen von η_I nach Anwendung von u^* enthält und $\eta_I u^*$ besteht aus den Morphismen, deren Index im Wertebereich des Funktors u^* liegt.

Das linke Dreieck des Diagramms 2.7 besagt, daß die Komposition $(\mu_{\mathsf{id}_I,u})_X \circ (\eta_I)_{u^*(X)}$ gleich $\mathsf{id}_{u^*(X)}$ in $\Psi(I)$ sein muß. Damit wird festgelegt, wie der Pseudofunktor das Identitätsgesetz $u \circ \mathsf{id}_I = u$ respektiert. In analoger Weise beschreibt die rechte Seite von Diagramm 2.7 das andere Identitätsgesetz und Diagramm 2.8 das Verhältnis von Ψ zur Assoziativität der Verkettung.

Der zu einem Morphismus u der Basiskategorie $\mathbb{B}$ gehörende Funktor u^* wird gelegentlich auch als **Pullbackfunktor** bezeichnet.

In vielen Fällen ergibt sich, daß eine indizierte Kategorie die Axiome eines Funktors erfüllt. Das heißt insbesondere, daß für zwei Morphismen $J \xrightarrow{v} K$ und $I \xrightarrow{u} J$ in $\mathbb{B}$ gilt, daß $u^* \circ v^* = (v \circ u)^*$. In diesem Fall wird die Kategorie in der englischsprachigen Literatur als „split indexed category“ bezeichnet. In Ermangelung einer geeigneten Übersetzung werde ich eine solche indizierte Kategorie als split bezeichnen.

Beispiel 2.9 (Mengenindizierte Familien) In diesem Beispiel wird ausgenutzt, daß Mengen sich als diskrete Kategorien modellieren lassen.

Die durch Mengen indizierten Familien von Objekten einer Kategorie $\mathbb{C}$ werden durch einen Funktor[3] $\Psi : \mathsf{Set}^{\mathsf{op}} \longrightarrow \mathbf{Cat}$ modelliert. Der Funktor Ψ bildet Mengen I auf die Funktorkategorie $\mathbb{C}^I$ ab. Jede Funktion $f : I \longrightarrow J$ induziert einen Funktor $f^* = \Psi(f) = \mathbb{C}^f : \mathbb{C}^J \longrightarrow \mathbb{C}^I$. Der Funktor f^* läßt sich durch Präkomposition von f definieren. Allerdings muß man dazu f als Funktor zwischen den diskreten Kategorien I und J betrachten: Einem Funktor $F : J \longrightarrow C$ wird der Funktor

$$f^*(F) \quad = \quad \Psi(f)(F) \quad : \quad I \longrightarrow C \quad : \quad i \longmapsto F(f(i))$$

zugeordnet. Da J eine Menge ist und somit als Morphismen nur Identitäten enthält, erfordert die Definition der Morphismenkomponente von $f^*(F)$ keine weitere Beachtung.

Eine natürliche Transformation $\eta : F \longrightarrow G : J \longrightarrow C$ wird durch f^* auf die natürliche Transformation $\{f^*(\eta)_i = \eta_{f(i)}\}$ abgebildet.

Fibrationen

Eine zweite Möglichkeit, indizierte Familien zu modellieren, besteht darin, sämtliche indizierte Familien in einer Kategorie mit besonderen Eigenschaften zusammenzufassen. Ein Funktor ordnet jedem Objekt die zugehörige Indexmenge zu.

Definition 2.10 (Kartesische Morphismen) *Sei $\mathbb{E} \xrightarrow{p} \mathbb{B}$ ein Funktor. Ein Morphismus $X \xrightarrow{f} Y$ in $\mathbb{E}$ heißt* ***kartesisch****, wenn für alle Morphismen $Z \xrightarrow{g} Y$ und jede Faktorisierung $p(g) = p(f) \circ w$ in $\mathbb{B}$ genau einen Morphismus $\psi : Z \longrightarrow X$ gibt, so daß $p(\psi) = w$ und $f \circ \psi = g$ gilt (siehe Abbildung 1).*

[3] In diesem Beispiel sind sowohl η_I als auch $\mu_{u,v}$ gleich den Identitäten der jeweiligen Objekte.

$$\text{in } \mathbb{E}: \quad \begin{tikzcd} Z \arrow[dr, dashed, "\psi"'] \arrow[drr, "g"] & & \\ & X \arrow[r, "f"'] & Y \end{tikzcd}$$

$$\text{in } \mathbb{B}: \quad \begin{tikzcd} pZ \arrow[dr, "w"'] \arrow[drr, "pg"] & & \\ & pX \arrow[r, "pf"'] & pY \end{tikzcd}$$

Abbildung 1: Zur Definition kartesischer Morphismen

Definition 2.11 (Fibration) *Ein Funktor $\mathbb{E} \xrightarrow{p} \mathbb{B}$ ist eine* ***Fibration****, wenn es für alle $Y \in |\mathbb{E}|$ und alle Morphismen $I \xrightarrow{u} pY$ in B mindestens einen kartesischen Pfeil f über u, das heißt mit $pf = u$ gibt.*

$$\begin{tikzcd} \textit{in } \mathbb{E}: \arrow[d, "p"] & ? \arrow[r, "\exists f"] & Y \\ \textit{in } \mathbb{B}: & I \arrow[r, "u"] & pY \end{tikzcd}$$

Die Kategorie $\mathbb{B}$ wird ***Basiskategorie*** *und $\mathbb{E}$* ***totale Kategorie*** *genannt.*

Beispiel 2.12 (Codomänfibration) Sei $\mathbb{B}$ eine Kategorie mit Pullbacks. Der Codomänfunktor $\mathsf{cod} : \mathbb{B}^{\rightarrow} \longrightarrow \mathbb{B}$ bildet Objekte $X \xrightarrow{f} Y$ auf Y und Morphismen aus $\mathbb{B}^{\rightarrow}$, das heißt kommutierende Quadrate $(u, v) : f \longrightarrow g$ auf v ab. Um zu zeigen, daß cod eine Fibration ist, überzeugt man sich zunächst mit Hilfe von Abbildung 2, daß die Pullbackquadrate in $\mathbb{B}$ genau die kartesischen Morphismen von $\mathbb{B}^{\rightarrow}$ sind: Für einen Morphismus $(a, b) : h \longrightarrow f$ in $\mathbb{B}^{\rightarrow}$ und eine Faktorisierung $b = u \circ w$ in $\mathbb{B}$ gibt es genau einen Morphismus $\psi : K \longrightarrow U$, da $(u, v) : g \longrightarrow f$ ein Pullback ist. Folglich gibt es in $\mathbb{B}^{\rightarrow}$ genau einen Morphismus $(\psi, w) : h \longrightarrow g$ mit den geforderten Eigenschaften.

Da in $\mathbb{B}$ zu jedem $X \xrightarrow{f} Y$ und $V \xrightarrow{u} Y$ ein Pullbackquadrat existiert, ist $\mathbb{B}^{\rightarrow} \xrightarrow{\mathsf{cod}} \mathbb{B}$ eine Fibration.

Begriffe Einige im folgenden verwendete Begriffe sollen hier kurz eingeführt werden. Sei im folgenden $\mathbb{E} \xrightarrow{p} \mathbb{B}$ eine Fibration. Ein Morphismus $f : X \longrightarrow Y$ aus $\mathbb{E}$ ist

$$\begin{tikzcd}
K \arrow[dr, dashed, "\nexists"'] \arrow[drr, "a"] \arrow[ddd, "h"'] & & \\
& U \arrow[r, "v"] \arrow[d, "g"'] \arrow[dr, phantom, "\lrcorner", very near start] & X \arrow[d, "f"] \\
& V \arrow[r, "u"] & Y \\
L \arrow[ur, "w"] \arrow[urr, "b"'] & &
\end{tikzcd}$$

Abbildung 2: Kartesische Morphismen in der Codomänfibration

vertikal, wenn er durch p auf die Identität abgebildet wird, das heißt $p(f) = \mathsf{id}_{pX}$.

Zu einem beliebigen Objekt I aus $\mathbb{B}$ bilden die durch p auf I abgebildeten Objekte zusammen mit vertikalen Morphismen eine Teilkategorie von $\mathbb{E}$. Eine solche Teilkategorie wird (in diesem Falle die zu I gehörende) **Faser** genannt und mit $\mathbb{E}_I$ bezeichnet.

Die Definition 2.11 verlangt nur die Existenz von kartesischen Morphismen für geeignete Morphismen der Basiskategorie. Existiert darüber hinaus eine Zuordnung kartesischer Morphismen zu Morphismen in der Basiskategorie, so spricht man in der englischen Literatur von einer „cloven fibration“. Hier wird eine solche Zuordnung **Cleavage** genannt. Falls $u : I \longrightarrow p(Y)$ ein Morphismus aus $\mathbb{B}$ ist, so wird der durch ein gegebenes Cleavage u zugeordnete kartesische Morphismus mit $\overline{u}(Y) : u^*(Y) \longrightarrow Y$ bezeichnet.

Konstruktion einer indizierten Kategorie aus einer Fibration mit Cleavage

Sei $\mathbb{E} \xrightarrow{p} \mathbb{B}$ eine Fibration mit Cleavage. Jeder Morphismus $I \xrightarrow{u} J$ in der Basiskategorie induziert kontravariant einen Funktor $u^* : \mathbb{E}_J \longrightarrow \mathbb{E}_I$ zwischen den entsprechenden Fasern. Einem Objekt Y aus der Faser $\mathbb{E}_J$ wird das durch das Cleavage bestimmte Objekt $u^*(Y)$ in $\mathbb{E}_I$ zugeordnet. Jedem Morphismus $X \xrightarrow{f} Y$ der Faser $\mathbb{E}_J$ kann genau ein vertikaler Morphismus $u^*(f) : u^*(X) \longrightarrow u^*(Y)$ zugeordnet werden, da $\overline{u}(Y)$ kartesisch ist und $p(f \circ \overline{u}(x)) = p(f) \circ p(\overline{u}(x)) = \mathsf{id}_J \circ u = u \circ \mathsf{id}_I$.

$$\text{in } \mathbb{E}: \quad \begin{tikzcd} u^*(X) \arrow[r, "\overline{u}(X)"] \arrow[d, dashed, "u^* f"] & X \arrow[d, "f"] \\ u^*(Y) \arrow[r, "\overline{u}(Y)"] & Y \end{tikzcd}$$

$$\text{in } \mathbb{B}: \quad I \xrightarrow{\quad u \quad} J = pX$$

Mit Hilfe dieser Beobachtung kann aus einer Fibration mit Cleavage eine induzierte Kategorie gewonnen werden. Wenn $\mathbb{E} \xrightarrow{p} \mathbb{B}$ eine Fibration mit Cleavage ist, dann läßt sich $\Psi : \mathbb{B}^{\mathsf{op}} \longrightarrow \textsf{\textbf{Cat}}$ durch $I \longmapsto \mathbb{E}_I$ und $(I \xrightarrow{u} J) \longmapsto (\mathbb{E}_J \xrightarrow{u^*} \mathbb{E}_I)$ definieren.

Eine Konstruktion der natürlichen Isomorphismen η_I und $\mu_{u,v}$ und der Nachweis, daß die Diagramme 2.7 und 2.8 kommutieren, führt hier zu weit. Ich empfehle Interessierten nochmals das ausgezeichnet Buch [Jac].

Beispiel 2.13 (Codomänfibration) Die eben skizzierte Konstruktion soll anhand der Codomänfibration aus Beispiel 2.12 illustriert werden. Sei deshalb $\mathbb{B}$ eine Kategorie mit ausgezeichneten Pullbacks.

Betrachten wir zunächst die Faser über einem beliebigen Objekt J. Objekte dieser Faser sind Morphismen $X \longrightarrow J$ aus $\mathbb{B}$. Morphismen $f \xrightarrow{h} g$ der Faser sind kommutierende Dreiecke $g \circ h = f$, wobei die Codomäne von f als auch von g gleich J ist. Komposition in der Faser wird durch die Komposition kommutierender Dreicke in $\mathbb{B}$ realisiert. Eine so konstruierte Kategorie wird in der englischsprachigen Literatur als „slice category“ $\mathbb{B}/J$ bezeichnet.

Die indizierte Kategorie Ψ wird deshalb für Objekte aus $\mathbb{B}$ durch $\Psi(J) = \mathbb{B}/J$ definiert. Abbildung 3 illustriert die Konstruktion des Morphismenteils des Pseudofunktors Ψ. Für Objekte g aus $\mathbb{B}^{\rightarrow}_{J}$, das heißt aus $\mathbb{B}/J$, wird das Bild $u^*(g) = \Psi(u)(g)$ mit Hilfe des Pullbacks entlang $I \xrightarrow{u} J$ definiert. Desweiteren bestimmt das Pullback von g entlang u für jeden Morphismus $X \xrightarrow{h} Y$ mit $g \circ h = f$ genau einen Morphismus $u^*(X) \longrightarrow u^*(Y)$, der als das Bild von h definiert wird.

Im Gegensatz zum Beispiel der mengenindizierten Familien ist die eben vorgestellte indizierte Kategorie im allgemeinen kein Funktor, sondern lediglich ein Pseudofunktor. Die fehlenden natürlichen Isomorphismen lassen sich für ein gegebenes Cleavage der Codomänfibration leicht konstruieren. Sie bestehen aus eindeutig bestimmten Morphismen zwischen Pullbackquadraten.

$$\begin{tikzcd}
u^*(X) \arrow[rr] \arrow[dr, dashed, "u^*(h)"] \arrow[dd, "u^*(f)"'] & & X \arrow[dr, "h"] \arrow[dd, "f"] & \\
& u^*(Y) \arrow[rr] \arrow[dl, "u^*(g)"] & & Y \arrow[dl, "g"] \\
I \arrow[rr, "u"] & & J &
\end{tikzcd}$$

Abbildung 3: Zur Konstruktion der Pullbackfunktoren in der Codomänfibration

Die Grothendieckkonstruktion

Bereits im vorhergehenden Teilabschnitt wurde eine Konstruktion skizziert, mit deren Hilfe aus einer Fibration mit Cleavage eine indizierte Kategorie gewonnen werden kann. Mit Hilfe der Grothendieckkonstruktion wird die Transformation in die andere Richtung möglich. Beide Konstruktionen sind zueinander invers. Fibrationen und indizierte Kategorien erscheinen als zwei verschiedenen Sichten auf ein und dieselbe Struktur.

Definition 2.14 (Grothendieckkonstruktion) *Sei* $\Psi : \mathbb{B}^{\mathsf{op}} \longrightarrow \mathbf{Cat}$ *eine indizierte Kategorie mit natürlichen Isomorphismen* η_I *und* $\mu_{u,v}$ *wie in Definition 2.6 beschrieben. Die Kategorie* $\int \Psi$ *besteht aus Objekten* (I, X) *mit* $I \in |\mathbb{B}|$ *und* $X \in |\Psi(I)|$*. Ein Morphismus in* $\int \Psi$ *ist ein Paar* $(u, f) : (I, X) \longrightarrow (J, Y)$*, wobei* $I \xrightarrow{u} J$ *ein Morphismus aus* $\mathbb{B}$ *und* $X \xrightarrow{f} u^*(Y)$ *ein Morphismus aus* $\mathbb{E}_I$ *ist. Für die Identitäten in* $\int \Psi$ *gilt* $\mathsf{id}_{(I,X)} = (\mathsf{id}_I, \eta_I)$.

Die Komposition zweier Morphismen $(J, Y) \xrightarrow{(v,g)} (K, Z)$ *und* $(I, X) \xrightarrow{(u,f)} (J, Y)$ *wird durch* $(v, g) \circ (u, f) = (v \circ u, \mu_{u,v} \circ u^*(g) \circ f) : (I, X) \longrightarrow (K, Z)$ *definiert.*

Satz 2.15 *Für eine indizierte Kategorie* $\Psi : \mathbb{B}^{\mathsf{op}} \longrightarrow \mathbf{Cat}$ *gilt: Die erste Projektion* $\int \Psi \longrightarrow \mathbb{B}$ *ist eine Fibration mit Cleavage.*

Der Beweis wird zur Illustration aus der Standardliteratur übernommen.

Beweis Zuerst muß gezeigt werden, daß $\int \Psi$ eine Kategorie ist:

$$
\begin{tikzcd}
X \arrow[r, "f"] \arrow[dddrr] & u^*Y \arrow[ddr, "u^*g"] & & & Y \arrow[ddr, "g"] & & & \\
& & \arrow[rr, phantom, "\overset{u^*}{\Longleftarrow}"] & & & \arrow[rr, phantom, "\overset{v^*}{\Longleftarrow}"] & & \\
& & u^*v^*Z \arrow[d, "\mu_{u,v}"] & & & v^*Z & & Z \\
& & (v \circ u)^*Z & & & & &
\end{tikzcd}
$$

$$
\begin{tikzcd}
\Psi(I) \arrow[rr, "u"] & & \Psi(J) \arrow[r, "v"] & \Psi(K)
\end{tikzcd}
$$

Abbildung 4: Komposition für die Grothendieckkonstruktion

- Zum Nachweis des Identitätsgesetzes ist der entsprechende Teil des Diagrammes 2.7 auszunutzen:

$$
\begin{aligned}
(u, f) \circ \mathsf{id}_{I,X} &= (u \circ \mathsf{id}_I, \mu_{\mathsf{id}_I,u} \circ \mathsf{id}_I^*(f) \circ \eta_I \\
&= (u, \mu_{\mathsf{id}_I,u} \circ \eta_I \circ f) \\
&\overset{2.7}{=} (u, f)
\end{aligned}
$$

- Auf ähnliche Weise folgt aus Diagramm 2.8, daß die Verkettung assoziativ ist.

Um zu zeigen, daß $\int \Psi$ eine Fibration ist, muß für jeden Morphismus $J \xrightarrow{v} K$ und jedes Objekt (K, Z) aus $\int \Psi$ ein kartesischer Morphismus in $\int \Psi$ bestimmt werden.

- Der zu v gehörende kartesische Morphismus ist $(v, \mathsf{id}_{v^*(Z)})$.
- Sei $(u, f) : (I, X) \longrightarrow (K, Z)$ ein beliebiger Morphismus in $\int \Psi$, so daß $u = v \circ w$ in $\mathbb{B}$ gilt. Dann schließt der Morphismus $(w, \mu_{w,v}^{-1} \circ f)$ das folgende Diagramm:

$$
\begin{tikzcd}
(I, X) \arrow[d, "{(w, \mu_{w,v}^{-1} \circ f)}"'] \arrow[dr, "{(u, f)}"] & \\
(J, v^*(Z)) \arrow[r, "{(v, \mathsf{id}_{v^*(Z)})}"'] & (K, Z)
\end{tikzcd}
$$

Denn es gilt

$$\begin{aligned}\Big(v\ ,\ \mathsf{id}_{v^*(Z)}\Big) \circ \Big(w\ ,\ \mu_{w,v}^{-1} \circ f\Big) &= \Big(v \circ w\ ,\ \mu_{w,v} \circ w^*(\mathsf{id}_{v^*(Z)}) \circ \mu_{w,v}^{-1} \circ f\Big) \\ &= \Big(u\ ,\ \mu_{w,v} \circ \mathsf{id}_{w^*v^*(Z)} \circ \mu_{w,v}^{-1} \circ f\Big) \\ &= (u\ ,\ f)\end{aligned}$$

Da $\mu_{w,v}$ ein natürlicher Isomorphismus ist, gibt es zu jedem anderen Morphismus $(w, f') \neq (w, \mu_{w,v}^{-1} \circ f)$ einen Morphismus $f' \circ \mu_{w,v} \neq f$, und $(w, \mu_{w,v}^{-1} \circ f)$ ist der einzige Morphismus, der obiges Diagramm schließt. □

Beispiel 2.16 (Mengenindizierte Familien) Hier soll die Grothendieckkonstruktion auf die im Beispiel 2.9 vorgestellte indizierte Kategorie mengenindizierter Familien angewandt werden. Die resultierende Fibration wird in der englischsprachigen Literatur „family fibration“ genannt und mit $\mathsf{Fam}(\mathbb{C})$ bezeichnet.

Nach Definition 2.14 sind die Objekte von $\mathsf{Fam}(\mathbb{C})$ Paare (I, X), wobei I eine Menge und X ein Objekt aus $\mathbb{C}^I$ ist. Da I eine Menge, das heißt eine diskrete Kategorie ist, besteht X nur aus einer Funktion $I \longrightarrow |\mathbb{C}|$. Somit läßt sich das Paar (I, X) auch als eine mit I indizierte Familie $(X_i)_{i \in I}$ auffassen.

Ein Morphismus $(I, X) \xrightarrow{(u,f)} (J, Y)$ besteht laut Definition 2.14 aus einer Funktion u zwischen I und J und aus einem Morphismus (der in diesem Fall eine natürliche Transformation ist) $f : X \longrightarrow u^*(Y)$ in $\mathbb{C}^I$. Dabei ist $u^*(Y)$ der durch $i \longmapsto Y(u(i))$ definierte Funktor. Auch hier kann durch die Tatsache, daß I eine Menge ist, die Charakterisierung von f vereinfacht werden: f ist eine Familie $(f_i)_{i \in I}$ von Morphismen, so daß $f_i : X_i \longrightarrow Y_{u(i)}$ gilt.

Die erste Projektion $\mathsf{Fam}(\mathbb{C}) \longrightarrow \mathbf{Set}$ ist eine Fibration mit Cleavage. Man überzeugt sich leicht davon, daß ein Morphismus $(u, (f_i))$ in $\mathsf{Fam}(\mathbb{C})$ genau dann kartesisch ist, wenn alle f_i Isomorphismen sind. Das Cleavage ordnet einer Funktion $I \xrightarrow{u} J$ den Morphismus $(u, (\mathsf{id}_i))$ zu.

Das Charakteristische einer Fibration ist, daß sich jeder Morphismus der totalen Kategorie in eineindeutiger Weise in eine horizontale und eine vertikale Komponente faktorisieren läßt. Der folgende Satz präzisiert diese Beobachtung.

Satz 2.17 *Sei $\mathbb{E} \xrightarrow{p} \mathbb{B}$ eine Fibration mit Cleavage, X und Y Objekte aus $\mathbb{E}$. Dann gilt*

$$\mathbb{E}(X, Y) \qquad \cong \qquad \biguplus_{u \in \mathbb{B}(pX, pY)} \mathbb{E}_{pX}(X, u^*(Y))$$

Beweis Der Beweis soll hier zu illustrativen Zwecken übernommen werden und besteht darin, eine eineindeutige Funktion zu konstruieren, die beide Morphismenmengen aufeinander abbildet. Die Situation ist in Abbildung 5 dargestellt.

in $\mathbb{E}$:

$$\begin{tikzcd}
X \arrow[d, "\overline{f}"] \arrow[dr, "f"] & \\
u^*(Y) \arrow[r, "\overline{u}"] & Y
\end{tikzcd}$$

in $\mathbb{B}$:

$$\begin{tikzcd}
p(X) \arrow[d, equal] \arrow[dr, "u = p(f)"] & \\
p(X) \arrow[r, "u"] & p(Y)
\end{tikzcd}$$

Abbildung 5: Zum Beweis von Satz 2.17

Sei mit $f \in \mathbb{E}(X, Y)$ ein Element der linken Seite gegeben. Mit dem durch das Cleavage zu $u = p(f)$ bestimmten kartesischen Morphismus erhält man genau eine Faktorisierung $f = \overline{u} \circ \overline{f}$.

Ist $pX \xrightarrow{u} pY$ und $X \xrightarrow{\overline{f}} u^*(Y)$ gegeben, dann läßt sich f als Komposition $\overline{u} \circ \overline{f}$ bestimmen. Da $\overline{f}$ ein Morphismus über $p(X)$ ist, ergibt sich $p(f) = u$. Der Morphismus f ist zugleich der einzige, der das Diagramm aus der Abbildung kommutativ schließt, weil $\overline{u}$ kartesisch ist. □

Änderung der Basiskategorie

Unter bestimmten Bedingungen läßt sich mittels einer gegebenen Fibration eine neue Fibration erzeugen.

Satz 2.18 (Basisänderung) *Sei* $\mathbb{E} \xrightarrow{p} \mathbb{B}$ *eine Fibration und* $\mathbb{A} \xrightarrow{K} \mathbb{B}$ *ein Funktor. Wenn das Diagramm*

$$\begin{tikzcd}
\mathbb{A} \times_{\mathbb{B}} \mathbb{E} \arrow[r] \arrow[d, "K^*(p)"'] \arrow[dr, phantom, "\lrcorner", very near start] & \mathbb{E} \arrow[d, "p"] \\
\mathbb{A} \arrow[r, "K"'] & \mathbb{B}
\end{tikzcd}$$

ein Pullbackdiagramm in **Cat** *ist, dann ist* $\mathbb{A} \times_{\mathbb{B}} \mathbb{E} \xrightarrow{K^*(p)} \mathbb{A}$ *eine Fibration.*

Als Pullback enthält die Kategorie $\mathbb{A} \times_{\mathbb{B}} \mathbb{E}$ Objekte der Form (I, X), wobei $I \in |\mathbb{A}|$ und $X \in |\mathbb{E}|$ ist, so daß $K(I) = p(X)$ gilt. Das Gleiche gilt im übertragenen Sinne für Morphismen.

Beweis Auch hier wird der Beweis zu illustrativen Zwecken aus [Jac] übernommen.

Sei (J, Y) ein Objekt aus $\mathbb{A} \times_{\mathbb{B}} \mathbb{E}$ und $I \xrightarrow{u} J$ ein Morphismus aus $\mathbb{A}$. Da $K(J) = p(Y)$ gibt es einen kartesischen Morphismus $X \xrightarrow{f} Y$ über $K(I) \xrightarrow{K(u)} K(J)$ in $\mathbb{E}$ und $(I, X) \xrightarrow{(u,f)} (J, Y)$ ist kartesisch in $\mathbb{A} \times_{\mathbb{B}} \mathbb{E}$. $\square$

Beispiel 2.19 (Endlich indizierte Familien)
Im Beispiel 2.16 wurde die Fibration $\mathsf{Fam}(\mathbb{C})$ vorgestellt. Bezeichne **FinSet** die Kategorie der endlichen Mengen und **FinSet** $\hookrightarrow$ **Set** die Einbettung in die Kategorie der Mengen, dann definiert das Diagramm

$$\begin{tikzcd}
\mathbf{FinFam}(\mathbb{C}) \arrow[r] \arrow[d] \arrow[dr, phantom, "\lrcorner", very near start] & \mathsf{Fam}(\mathbb{C}) \arrow[d] \\
\mathbf{FinSet} \arrow[r, hook] & \mathbf{Set}
\end{tikzcd}$$

die Fibration endlich indizierter Familien $\mathbf{FinFam}(\mathbb{C}) \longrightarrow \mathbf{FinSet}$ von Objekten aus $\mathbb{C}$.

3. Lineare Logik

Linear logic is a logic behind logic.
Jean–Yves Girard

Lineare Logik wurde zuerst in [Gir87] vorgestellt. Das besondere an dieser Logik ist, daß die aus der Klassischen Logik bekannten Regeln zur Hinzunahme weiterer Annahmen („weakening“) und zum Zusammenfassen gleicher Formeln („contraction“) völlig fehlen (siehe Abbildung 6). Dadurch zerfallen zum einen die bekannten Konnektoren $\wedge$ („und“) und $\vee$ („oder“) in jeweils zwei Versionen, die sich unterschiedlich bezüglich des vorhandenen Kontexts verhalten. Zum anderen erhalten Beweise in einem bestimmten Sinn die verwendeten Ressourcen. Jede Annahme muß genau einmal in einem Beweis verwendet werden.

In den meisten Veröffentlichungen zu Linearer Logik (zum Beispiel [Gir87, GLT88, See89, FKKK94]) wird diese im Stil einer Aussagenlogik dargestellt, das heißt atomare Formeln sind dort propositionale Konstanten. Die einzige mir bekannte Ausnahme ist [Amb91]. Da der Schwerpunkt in der vorliegenden Arbeit auf Quantoren erster Ordnung liegt, soll im folgenden eine „prädikatenlogische Version“ der Linearen Logik entwickelt werden. Der Ansatz von [Amb91] kann hier jedoch nicht verwendet werden, weil Simon Ambler erstens intuitionistische Lineare Logik betrachtet und zweitens die Idee der Linearität auch auf die Typebene überträgt. In dieser Arbeit sollen weniger Einschränkungen bezüglich der betrachteten Logik gemacht werden.

Nach einem Beispiel, das die Ressourcen erhaltenden Eigenschaften der Linearen Logik verdeutlichen soll, werden im nächsten Abschnitt 3.1 Signaturen und atomare Formeln nach dem Vorbild von [Jac] definiert. Die Definitionen sind im wesentlichen einer von Bart Jacobs freundlicherweise zur Verfügung gestellten Vorabversion seines Buches [Jac] entnommen. Im Abschnitt 3.2 werden Terme und Formeln über einer Signatur definiert. Die klassifizierende Kategorie, die die Typtheorie über einer Signatur modelliert, ist damit eng verbunden. Schließlich werden Kontexte und Sequenzen vorgestellt und der Begriff der mehrsortigen Linearen Logik definiert. Die Darstellung Linearer Logik schließt mit Bemerkungen zur Cut–Elimination und einer Einschätzung der in diesem Kapitel zusammengetragenen Ergebnisse.

$$\frac{\Gamma \vdash \Delta}{\Gamma \vdash \Delta, A}\ \text{Weakening L} \qquad \frac{\Gamma, A, A \vdash \Delta}{\Gamma, A \vdash \Delta}\ \text{Contraction L}$$

$$\frac{\Gamma \vdash \Delta}{\Gamma, A \vdash \Delta}\ \text{Weakening R} \qquad \frac{\Gamma \vdash \Delta, A, A}{\Gamma \vdash \Delta, A}\ \text{Contraction R}$$

Abbildung 6: Regeln der Klassischen Logik, die in Linearer Logik *nicht* gelten

Ein Beispiel linearen Schließens

Lineare Logik wird häufig mit einer Version des folgenden Beispiels illustriert: Angenommen, es soll der Kauf von Schokolade mittels Logik modelliert werden. Zwei Axiome der Form

$$\begin{array}{rcl} 1\mathrm{DM} & \vdash & \mathrm{RitterSport\ Trauben{-}Nuß} \\ 1\mathrm{DM} & \vdash & \mathrm{RitterSport\ Pfefferminz} \end{array} \tag{3.1}$$

modellieren die Tatsache, daß mit einer Mark eine entsprechende Tafel Schokolade gekauft werden kann. In Klassischer Logik kann daraus die Sequenz

$$1\mathrm{DM} \quad \vdash_{\mathrm{KL}} \quad \mathrm{RitterSport\ Trauben{-}Nuß} \ \wedge \ \mathrm{RitterSport\ Pfefferminz}$$

abgeleitet werden, die im allgemeinen der Wirklichkeit nicht mehr entspricht.

Formalisiert man den gleichen Diskursbereich mit Linearer Logik, dann gibt es keine gültige Ableitung obiger Sequenz:

$$1\mathrm{DM} \quad \nvdash_{\mathrm{LL}} \quad \mathrm{RitterSport\ Trauben{-}Nuß} \ \otimes \ \mathrm{RitterSport\ Pfefferminz}$$

Der Konnektor $\otimes$ bezeichnet hierbei das multiplikative „und“, siehe Abschnitt 3.3.

Dagegen kann man mit Hilfe der beiden Axiome aus 3.1 die Sequenz

$$1\mathrm{DM} \ \otimes \ 1\mathrm{DM} \quad \vdash_{\mathrm{LL}} \quad \mathrm{RitterSport\ Trauben{-}Nuß} \ \otimes \ \mathrm{RitterSport\ Pfefferminz}$$

beweisen, oder mehr nach meinem Geschmack (mit einem entsprechenden Axiom):

$$1\mathrm{DM} \ \otimes \ 1\mathrm{DM} \quad \vdash_{\mathrm{LL}} \quad \mathrm{RitterSport\ Alpenvollmilch} \ \otimes \ \mathrm{RitterSport\ Alpenvollmilch}$$

Es gibt auch gültige Sequenzen der Form:

$$1\mathrm{DM} \quad \vdash_{\mathrm{LL}} \quad \mathrm{RitterSport\ Alpenvollmilch} \ \& \ \mathrm{RitterSport\ Pfefferminz}$$

Allerdings hat eine solche Sequenz die folgende Bedeutung: Der Käufer kann *entweder* eine Tafel Rittersport Pfefferminz *oder* eine Tafel Rittersport Alpenvollmilch für eine Mark kaufen, jedoch nicht beide zugleich.

3.1 Signaturen

Wie schon in der Einleitung zu diesem Kapitel bemerkt, ist das folgende Material im wesentlichen [Jac] entnommen. Gleichartige Definitionen für Signaturen sind auch in [Rei95] zu finden.

Unter einer mehrsortigen Signatur wird eine Zusammenfassung von einfachen bzw. Grundtypen[4] und getypten Operationssymbolen verstanden. Eine mehrsortige Signatur Σ ist

[4] in der englischen Literatur mit „ground types“ bezeichnet

deshalb ein Paar $(\mathcal{T}, \mathcal{F})$, wobei $\mathcal{T}$ eine Menge von (Grund–)Typen und $\mathcal{F} : \mathcal{T}^* \times \mathcal{T} \longrightarrow \textsf{Set}$ eine Abbildung ist, die jeder Kombination von Typen eine Menge von Operationssymbolen zuordnet. Falls $F \in \mathcal{F}(\langle\sigma_1, \ldots, \sigma_n\rangle, \sigma)$ gilt, wird vereinfachend $F : \sigma_1, \ldots, \sigma_n \longrightarrow \sigma$ geschrieben. Die Menge der Typen $\mathcal{T}$ einer Signatur Σ wird auch mit $|\Sigma|$ bezeichnet.

Ein Morphismus zwischen zwei Signaturen $\Sigma = (\mathcal{T}, \mathcal{F})$ und $\Delta = (\mathcal{U}, \mathcal{G})$ ist ein Paar $(u, (f_\alpha)_{\alpha \in \mathcal{T}^* \times \mathcal{T}})$, wobei $u : \mathcal{T} \longrightarrow \mathcal{U}$ eine Abbildung zwischen den Typen ist. Die Funktionen f_α bilden die Mengen von Operationssymbolen aufeinander ab, so daß die Typen der Operationssymbole respektiert werden:

$$F : \sigma_1, \ldots, \sigma_n \longrightarrow \sigma \quad \Longrightarrow \quad f_\alpha(F) : u(\sigma_1), \ldots, u(\sigma_n) \longrightarrow u(\sigma)$$

Hier steht α abkürzend für $(\langle\sigma_1, \ldots, \sigma_n\rangle, \sigma)$. Diese eher informale Beschreibung wird mit folgender Definition präzisiert:

Definition 3.2 (Mehrsortige Signaturen) *Die Kategorie der mehrsortigen Signaturen* **Sig** *wird als das Resultat der Basisänderung*

$$\begin{tikzcd}
\textsf{Sig} \arrow[r] \arrow[d] \arrow[dr, phantom, very near start, "\lrcorner"] & \textsf{Fam}(\textsf{Set}) \arrow[d] \\
\textsf{Set} \arrow[r, "T \longmapsto T^* \times T"'] & \textsf{Set}
\end{tikzcd}$$

definiert.

In einer mehrsortigen Signatur mit Prädikaten sind zusätzlich getypte Relationssymbole enthalten. Informal besteht eine mehrsortige Signatur mit Prädikaten aus einem Paar (Σ, Π), wobei Σ eine mehrsortige Signatur ist und $\Pi : |\Sigma|^* \longrightarrow \textsf{Set}$ jeder Sequenz von Typen eine Menge von Relationssymbolen zuordnet. Auch hier soll für $R \in \Pi(\sigma_1, \ldots, \sigma_n)$ abkürzend $R : \sigma_1, \ldots, \sigma_n$ geschrieben werden.

Ein Morphismus zwischen zwei Signaturen mit Prädikaten ist ein Tripel (u, f_α, g_α), wobei u die Typen, f_α die Mengen der Operationssymbole und g_α die Mengen der Relationssymbole aufeinander abbildet, so daß gilt:

$$\begin{aligned}
F : \sigma_1, \ldots, \sigma_n \longrightarrow \sigma \quad &\Longrightarrow \quad f_\alpha(F) : u(\sigma_1), \ldots, u(\sigma_n) \longrightarrow u(\sigma) \\
R : \sigma_1, \ldots, \sigma_n \quad &\Longrightarrow \quad g_\beta(R) : u(\sigma_1), \ldots, u(\sigma_n)
\end{aligned}$$

wobei α und β jeweils abkürzend für die entsprechenden Typen stehen.

Definition 3.3 (Mehrsortige Signatur mit Prädikaten) *Die Kategorie der mehr-*

sortigen Signaturen mit Prädikaten **SigPrä** *wird als das Resultat der Basisänderung*

$$\begin{tikzcd}
\mathsf{\mathbf{SigPr\ddot{a}}} \arrow[r] \arrow[d] \arrow[dr, phantom, "\lrcorner", very near start] & \mathsf{Fam}(\mathsf{\mathbf{Set}}) \arrow[d] \\
\mathsf{\mathbf{Set}} \arrow[r, "T \longmapsto T^* \times T \uplus T^*"'] & \mathsf{\mathbf{Set}}
\end{tikzcd}$$

definiert.

3.2 Terme und die klassifizierende Kategorie

In den nächsten beiden Abschnitten werden Typen, Terme und Formeln über einer mehrsortigen Signatur verbal beschrieben. Das Regelsystem, mit dem zulässige Terme und Formeln aus der Signatur abgeleitet werden können, befindet sich in Abbildung 10 (Typen und Terme) und Abbildung 11 (wohldefinierte Formeln) im Anhang A ab Seite 64.

Sei Σ eine mehrsortige Signatur mit Prädikaten. Unter einem **Kontext** Υ über Σ wird eine endliche Folge der Form

$$\Upsilon = v_1 : \tau_1, \ldots, v_n : \tau_n$$

verstanden, wobei $v_1, \ldots, v_n$ Variablen und $\tau_1, \ldots, \tau_n$ zulässige Typen über Σ sind. Es wird davon ausgegangen, daß in einem Kontext die gleiche Variable nicht doppelt vorkommt. Diese Einschränkung läßt sich durch eine geeignete Variablenmenge mit einfachen syntaktischen Bedingungen formalisieren, siehe [Jac].

Die Menge der **zulässigen Typen** über Σ hängt von der betrachteten Typtheorie ab. In dieser Arbeit soll die Typtheorie Produkttypen zulassen. Zulässige Typen sind deshalb solche, für die mit Hilfe der Typregeln aus Abbildung 10 im Anhang A eine Sequenz der Form

$$\langle\rangle \;\rhd\; \tau : \mathsf{Type}$$

abgeleitet werden kann.

Terme bestehen aus Variablen und Funktionssymbolen. Funktionssymbole sind entweder der Signatur entnommen oder konstante Funktionen der Typtheorie, wie zum Beispiel die erste Projektion π_1. Zulässige Terme sind auch hier solche, für die mit den Regeln aus Abbildung 10 eine Ableitung der Sequenz

$$\Upsilon \;\rhd\; t : \tau$$

existiert. t ist hierbei der Term, τ sein Typ und Υ ein Kontext, in dem alle in t freien Variablen enthalten sind. Das anstelle von $\rhd$ übliche Symbol $\vdash$ soll der Trennung von linker und rechter Sequenz in Abschnitt 3.3 vorbehalten bleiben.

Die Menge der freien Variablen wird wie üblich definiert. Die Substitution eines Terms r für eine freie Variable v in t wird mit $t[r/v]$ bezeichnet.

Eine **Typtheorie mit Produkten** über einer Signatur Σ enthält Typen und Terme wie gerade beschrieben, so daß zusätzlich gilt

- Ist $\Upsilon \ \triangleright \ t : \mathbf{1}$ ableitbar, dann gilt $t = ()$
- Sind t und r zulässige Terme mit identischem Kontext, dann gilt
$$\begin{aligned} \pi_1(t,r) &= t \\ \pi_2(t,r) &= r \end{aligned}$$
- Für jeden zulässigen Term $t : \sigma \times \tau$ gilt
$$(\pi_1 t, \pi_2 t) = t$$

In den folgenden Kapiteln wird eine Semantik mittels Fibrationen definiert. Dazu wird eine Basiskategorie benötigt, die Kontexte und Terme des Termkalküls modelliert. Eine Variante einer solchen Kategorie ist die zu einer Signatur gehörende klassifizierende Kategorie $\mathsf{CL}(\Sigma)$.

Definition 3.4 *Sei Σ eine Signatur. Die* ***klassifizierende Kategorie*** *$\mathsf{CL}(\Sigma)$ ist wie folgt aufgebaut:*

Objekte *Objekte sind Kontexte Υ über Σ.*

Morphismen *Seien $\Upsilon = v_1 : \tau_1, \ldots, v_n : \tau_n$ und $\Xi = u_1 : \sigma_1, \ldots, u_m : \sigma_m$ Kontexte über Σ. Ein Morphismus $\Upsilon \xrightarrow{t} \Xi$ besteht aus m Termen $(t_i)_{i \leq m}$ einer Typtheorie mit Produkten, so daß für jeden Term t_i gilt: $\Upsilon \ \triangleright \ t_i : \sigma_i$.*

Komposition
Die Komposition wird mittels Substitution definiert. Sei $\Theta = v_1 : \tau_1, \ldots, v_n : \tau_n$ ein Kontext und $\Theta \xrightarrow{r} \Xi$ und $\Upsilon \xrightarrow{s} \Theta$ zwei Morphismen. Dann ist $(r \circ s)_i = r_i[s_1/v_1, \ldots, s_n/v_n]$ der i-te Term des Morphismus $r \circ s$.

Die klassifizierende Kategorie hat immer Produkte. Wenn Υ und Θ zwei Objekte aus $\mathsf{CL}(\Sigma)$ sind, dann besteht $\Upsilon \times \Theta$ aus der Verkettung von Υ und Θ, wobei in Υ und Θ gemeinsam auftretende Variablen vorher umbenannt werden. Die Terme, die die Projektionen bilden, bestehen nur aus Variablen von Υ bzw. Θ.

Sind zum Beispiel $\Upsilon = u_1 : \sigma_1, \ldots, u_n : \sigma_n$ und $\Theta = v_1 : \tau_1, \ldots, v_m : \tau_m$ Kontexte über einer Signatur Σ. Dann ist $\Upsilon \times \Theta = u_1 : \sigma_1, \ldots, u_n : \sigma_n, v_1 : \tau_1, \ldots, v_m : \tau_m$. Die erste Projektion $\Upsilon \times \Theta \xrightarrow{\pi_1} \Upsilon$ besteht aus den Termen $(u_1, \ldots, u_n)$.

Für eine Typtheorie mit Produkten, so wie sie in dieser Arbeit definiert wurde, ist der Kontext $\langle u_1 : \sigma, u_2 : \tau \rangle$ in der klassifizierenden Kategorie isomorph zum Kontext $\langle v : \sigma \times \tau \rangle$. Die Isomorphismen bestehen aus den Projektionen für Terme in der einen Richtung und der Paarbildung in der anderen.

3.3 Mehrsortige Lineare Logik

In diesem Abschnitt werden Formeln und Sequenzen über einer mehrsortigen Signatur definiert und schließlich der Sequenzenkalkül der mehrsortigen Linearen Logik vorgestellt. Die Modelle dieses Kalküls stehen in den restlichen Kapiteln dieser Arbeit im Mittelpunkt.

Der Formelaufbau erfolgt gemäß den Regeln aus Abbildung 11 in Anhang A. **Atomare Formeln** über einer Signatur Σ werden aus den Relationssymbolen der Signatur und Termen des Termkalküls unter Berücksichtigung der Typen entsprechend der Regel Relationssymbol zusammengesetzt. Zusätzlich gibt es die propositionalen Konstanten $I, \bot, 0$ und $\top$.

Formeln über Σ werden mit Hilfe des einstelligen Konnektors $(-)^\bot$, der zweistelligen Konnektoren $\otimes, ⅋, \multimap, \oplus, \&$ und den Quantoren und $\forall$ und $\exists$ gebildet.

Eine **Sequenz** über Σ besteht aus drei Teilen:

$$\Upsilon \ \triangleright \ \Gamma \ \vdash \ \Delta$$

Zu beiden Seiten von $\vdash$ stehen Listen Γ und Δ von Formeln über Σ. Alle darin vorkommenden freien Variablen sind im Kontext Υ deklariert.

Eine **mehrsortige Lineare Logik** $\mathfrak{L} = (\Sigma, \mathfrak{A})$ ist gegeben durch eine mehrsortige Signatur mit Prädikaten Σ und eine Menge von Sequenzen über Σ mit leerem Kontext, den Axiomen $\mathfrak{A}$.

In der Originalveröffentlichung zu Linearer Logik [Gir87] wird ein Regelsystem ohne Kontexte angegeben. In den Formeln auftretende Variable besitzen keine Typen. Es werden dort auch verschiedene Semantiken vorgeschlagen, zum Beispiel die Phasensemantik oder kohärente Räume.

Das Regelsystem für mehrsortige Lineare Logik ist in den Abbildungen 12 bis 14 im Anhang A gegeben. Es wurde aus dem Regelsystem von [See89] entwickelt. Wesentliche Unterschiede sind erstens die Verwendung von Kontexten für die Deklaration von freien Variablen und zweitens die Verwendung der von Jean–Yves Girard in [Gir87] vorgeschlagenen Konnektoren.

Eine Sequenz $\Upsilon \ \triangleright \ \Gamma \ \vdash \ \Delta$ ist **gültig** in $\mathfrak{L}$, wenn mit Hilfe des Regelsystems aus den Abbildungen 12 bis 14 im Anhang A mit den Axiomen aus $\mathfrak{L}$ ein Ableitungsbaum mit der Wurzelmarkierung $\Upsilon \ \triangleright \ \Gamma \ \vdash \ \Delta$ gebildet werden kann.

Eine Beispielableitung, die die Anwendung des Regelsystems veranschaulicht, ist in Abbildung 7 angegeben.

Cut–Elimination

Eine wesentliche Eigenschaft Linearer Logik ist, die Beweisbarkeit einer entsprechenden Version von Gentzens Cut–Eliminations–Theorem. Das Theorem besagt: Für jede ab-

$$
\dfrac{
\dfrac{\dfrac{\dfrac{}{\langle\rangle \rhd A \vdash B}\ \text{Axiom}}{\langle\rangle \rhd !A \vdash B}\ \text{Dereliction}}{\langle\rangle \rhd !A \vdash !B}\ \text{Promotion}
\qquad
\dfrac{\dfrac{\dfrac{\dfrac{}{\langle\rangle \rhd B \vdash C}}{\langle\rangle \rhd !B \vdash C} \quad \dfrac{\dfrac{}{\langle\rangle \rhd B \vdash D}\ \text{Axiom}}{\langle\rangle \rhd !B \vdash D}\ \text{Dereliction}}{\langle\rangle \rhd !B, !B \vdash C \otimes D}\ \otimes\text{R}}{\langle\rangle \rhd !B \vdash C \otimes D}\ \text{Contraction}
}{\langle\rangle \rhd !A \vdash C \otimes D}\ \text{Cut}
$$

Diese Ableitung ist korrekt, falls die Axiomenmenge $\mathfrak{A}$ wenigstens die drei Sequenzen $\langle\rangle \rhd A \vdash B$, $\langle\rangle \rhd B \vdash C$ und $\langle\rangle \rhd B \vdash D$ enthält.

Abbildung 7: Eine Beispielableitung in Linearer Logik

leitbare Formel einer mehrsortigen Linearen Logik $\mathfrak{L}$ mit leerer Axiomenmenge gibt es eine Ableitung, in der die Regel Cut aus Abbildung 14 nicht vorkommt.

Betrachtet man eine mehrsortige Lineare Logik mit Axiomen, dann existiert für jede ableitbare Sequenz ein Beweisbaum, in dem wenigstens eine Prämisse jeder Anwendung der Regel Cut ein Axiom ist.

Unter dem Gesichtspunkt des Curry–Howard–Isomorphismus, der eine Ableitung als ein funktionales Programm interpretiert, entspricht die Cut–Elimination der Normalisierung oder in anderen Worten der Wertberechnung.

In dieser Darstellung soll das Cut–Eliminations–Theorem nicht formalisiert und bewiesen werden. Der interessierte Leser wird auf [GLT88] und [Gir87] verwiesen. In Abschnitt 5.4 wird Cut–Elimination noch einmal am Beispiel der Interaktionskategorie $\mathcal{SProc}$ diskutiert. Deshalb soll hier zur Illustration für die Sequenz $\langle\rangle \rhd !A \vdash C \otimes D$ aus Abbildung 7 eine Ableitung entwickelt werden, die die Schnittregel nur noch auf Axiome anwendet.

Das Cut–Eliminations–Theorem wird zum Teil konstruktiv bewiesen. Der Beweis liefert einen Algorithmus zur Umwandlung einer Ableitung mit nichttrivialen Anwendungen der Schnittregel.

In jeweils einem Schritt wird eine Schnittregel im Ableitungsbaum um eine Regel nach oben verschoben. Der resultierende Ableitungsbaum kann erheblich größer sein, wenn zum Beispiel Teile verdoppelt werden, oder auch erheblich kleiner, wenn ganze Teilbäume gestrichen werden. Welche Regel anstelle des Schnittes angewendet wird, hängt von den Regeln ab, mit deren Hilfe die Prämissen des Schnittes erzeugt worden.

$$\dfrac{\dfrac{\dfrac{\dfrac{\overline{\langle\rangle \rhd A \vdash B} \qquad \overline{\langle\rangle \rhd B \vdash C}}{\langle\rangle \rhd A \vdash C}}{\langle\rangle \rhd !A \vdash C}\;\text{Dereliction} \qquad \dfrac{\dfrac{\overline{\langle\rangle \rhd A \vdash B} \qquad \overline{\langle\rangle \rhd B \vdash D}}{\langle\rangle \rhd A \vdash D}\;\text{Cut}}{\langle\rangle \rhd !A \vdash D}\;\text{Dereliction}}{\langle\rangle \rhd !A, !A \vdash C \otimes D}\;\text{Cut}}{\langle\rangle \rhd !A \vdash C \otimes D}\;\text{Contraction}$$

Abbildung 8: Ableitungsbaum nach Cut–Elimination

Zum Beispiel wird die Anwendung der Schnittregel aus Abbildung 7

$$\dfrac{\dfrac{\vdots}{\langle\rangle \rhd !A \vdash !B}\;\text{Promotion} \qquad \dfrac{\dfrac{\vdots \qquad \vdots}{\langle\rangle \rhd !B, !B \vdash C \otimes D}\;\otimes\text{R}}{\langle\rangle \rhd !B \vdash C \otimes D}\;\text{Contraction}}{\langle\rangle \rhd !A \vdash C \otimes D}\;\text{Cut}$$

durch den folgenden Ableitungsteilbaum ersetzt:

$$\dfrac{\dfrac{\dfrac{\vdots}{\langle\rangle \rhd !A \vdash !B}\;\text{Promotion} \qquad \dfrac{\dfrac{\vdots}{\langle\rangle \rhd !A \vdash !B}\;\text{Promotion} \qquad \dfrac{\vdots \qquad \vdots}{\langle\rangle \rhd !B, !B \vdash C \otimes D}\;\otimes\text{R}}{\langle\rangle \rhd !A, !B \vdash C \otimes D}\;\text{Cut}}{\langle\rangle \rhd !A, !A \vdash C \otimes D}\;\text{Cut}}{\langle\rangle \rhd !A \vdash C \otimes D}\;\text{Contraction}$$

Danach wird das Verfahren sukzessive auf die verbliebenen Vorkommen der Schnittregel angewendet. Das Verfahren terminiert, weil die Komplexität der Teilbäume, an deren Wurzel eine Schnittregel ist, sinkt. Der resultierende Ableitungsbaum befindet sich in Abbildung 8.

3.4 Diskussion

Wie schon erwähnt und in den nachfolgenden Kapiteln ausführlich dargestellt, können Quantoren einer Prädikatenlogik innerhalb einer geeigneten Fibration modelliert werden. Die Basiskategorie dieser Fibration ist sehr eng mit der verwendeten Typtheorie verbunden. Sie enthält (vereinfacht formuliert) die zulässigen Kontexte als Objekte und Terme als Morphismen. Damit werden in der totalen Kategorie Formeln und Substitution modelliert. Die Faser über einem Kontext korrespondiert zu den Formeln, die mit Variablen aus diesem Kontext gebildet werden können.

Aus der Literatur sind mir außer [Amb91] bislang keine Arbeiten bekannt, die eine Typtheorie für Lineare Logik entwickeln. Es ist unklar, welche Kategorie als Basiskategorie für ein Modell der Linearen Logik geeignet ist.

Aus verschiedenen Gründen ist die klassifizierende Kategorie einer Typtheorie mit Produkten keine befriedigende Lösung.

In dieser Arbeit und in den Veröffentlichungen zu Interaktionskategorien wird angestrebt, eine Formel einer Linearen Logik als Prozeßspezifikation zu interpretieren. Ein Term müßte demzufolge einen Prozeß repräsentieren. Das wird in den Definitionen dieses Kapitels nicht widergespiegelt.

Führt man diese Idee weiter, so erscheint es sinnvoll, als Basiskategorie eine Prozeßkategorie, wie zum Beispiel $\mathcal{SP}roc$, zu benutzen. Im Falle von $\mathcal{SP}roc$ erhebt sich die Frage, mit welcher Struktur die Verkettung von Kontexten interpretiert werden soll. Es stehen mindestens der Tensor $\otimes$ und das Produkt & zur Wahl, welche dem multiplikativen und additiven „und" der Linearen Logik entsprechen. Im Kapitel 6 wird gezeigt, daß sich die zur Interpretation der Quantoren notwendigen Adjunktionen in beiden Fällen konstruieren lassen. Der Ansatz der klassifizierenden Kategorie ist dafür nicht hinreichend allgemein, weil dort die Kontexte immer mit dem Produkt zusammengesetzt werden.

Drittens entspricht die fundamentale Operation in der Kategorientheorie —die Komposition zweier Morphismen— in der klassifizierenden Kategorie der Substitution von Termen. In der Welt der Interaktionskategorien ist die Komposition jedoch die Synchronisation zweier Prozesse. Für den allgemeinen Korrektheitsbeweis der Regeln $\forall$ L und $\exists$ R (siehe Abschnitt 4.4) wiederum ist Substitution notwendig.

Schließlich ist die Frage zu klären, inwieweit die Idee der Linearität auf die Typtheorie übertragen werden sollte. Das in diesem Kapitel in Abbildung 10 vorgestellte Regelsystem, welches über die Regel Weakening die Erzeugung von Ressourcen gestattet, erscheint für Lineare Logik nicht adäquat.

Eine vollständige Untersuchung dieser Probleme konnte im Rahmen dieser Arbeit nicht erfolgen. Deshalb wurde die Entscheidung getroffen, einen bereits bekannten Ansatz möglichst intuitiv und allgemein darzustellen.

4. Kategorientheoretische Modelle für Lineare Logik

> ..., and in true categorcial spirit, one finds that this question was answered long before being put, ...
>
> R. A. G. Seely

Die grundlegende Idee bei der Definition eines Modells für Lineare Logik mittels bereicherter Kategorien besteht darin, eine Korrespondenz zwischen den Formeln der Linearen Logik und den Objekten der Kategorie herzustellen. Diese Korrespondenz soll die Konnektoren der Linearen Logik respektieren. Eine gültige Sequenz $\Gamma \vdash \Delta$ wird zunächst durch einen Morphismus zwischen den zu Γ bzw. Δ gehörenden Objekten modelliert. Die Kategorie muß genügend Struktur enthalten, damit aus gegebenen Morphismen entsprechend den Ableitungsregeln der Linearen Logik neue Morphismen konstruiert werden können.

Zum Beispiel enthalten $*$–autonome Kategorien für jedes Axiom aus dem Regelsystem der Linearen Logik in Abbildung 12 einen Morphismus. Der Konnektor $\otimes$ wird innerhalb der Kategorie durch ein Tensorprodukt modelliert. Um dann die Regel ($\multimap$ R) zu interpretieren, existiert in $*$–autonomen Kategorien eine eineindeutige Beziehung zwischen Morphismen $X \otimes A \xrightarrow{f} B$ und $X \xrightarrow{\Lambda(f)} A \multimap B$ (siehe den folgenden Abschnitt 4.1).

Eine mehrsortige Lineare Logik $\mathfrak{L}$ nach Abschnitt 3 enthält Prädikate und Quantoren. Innerhalb der Formeln können freie Variablen auftreten. Für ein kategorientheoretisches Modell einer solchen Logik ist die Struktur einer $*$–autonomen Kategorie nicht mehr ausreichend.

Der im folgenden beschriebene Ansatz zur Modellierung von Formeln mit freien Variablen und Quantoren ist nicht neu. Er wurde schon in einer Reihe von Veröffentlichungen verwendet, um kategorientheoretische Modelle zu gewinnen (siehe zum Beispiel [Jac, Amb91, Pit87]) und geht auf Lawvere zurück. Als Modell einer Logik erster Ordnung werden Fibrationen mit bestimmten Eigenschaften benutzt. Die Basiskategorie modelliert die zulässigen Kontexte. Wenn die Typtheorie Produkte hat, dann ist die klassifizierende Kategorie aus Definition 3.4 als Basiskategorie verwendbar. Alle Formeln mit der gleichen Menge freier Variablen werden in einer Faser über einem entsprechenden Objekt der Basiskategorie interpretiert.

Die zu den Projektionen $\Gamma, x : \sigma \xrightarrow{\pi} \Gamma$ in der Basiskategorie gehörenden Funktoren π^* interpretieren eine Formel in einem Kontext mit mehr Variablen. Rechte und linke Adjunktionen zu diesem Funktor werden benutzt, um die Regeln für $\forall$ und $\exists$ zu modellieren.

4.1 Abgeschlossene monoidale Kategorien

Das Konzept einer abgeschlossenen monoidalen Kategorie ist hinlänglich aus Veröffentlichungen wie [EK65, ML71, Kel82, Poi92] bekannt. An dieser Stelle werden lediglich die für diese Arbeit wichtigen Konzepte zusammenfassend dargestellt. Auf Beweise wird vollständig verzichtet.

Die folgende Darstellung ist [Gay94] entlehnt.

Eine **monoidale** Kategorie ist eine Kategorie $\mathbb{C}$ zusammen mit einem Bifunktor $\otimes : \mathbb{C} \times \mathbb{C} \longrightarrow \mathbb{C}$, einem ausgezeichneten Objekt I und natürlichen Isomorphismen $\mathsf{unitl}_A : I \otimes A \longrightarrow A$, $\mathsf{unitr}_A : A \otimes I \longrightarrow A$ und $\mathsf{assoc}_{A,B,C} : A \otimes (B \otimes C) \longrightarrow (A \otimes B) \otimes C$. Diese Isomorphismen müssen folgende Bedingungen erfüllen. Die Diagramme

$$
\begin{tikzcd}
A \otimes (B \otimes (C \otimes D)) \arrow[r, "\mathsf{assoc}_{A,B,C\otimes D}"] \arrow[dd, "\mathsf{id}_A \otimes \mathsf{assoc}_{B,C,D}"] & (A \otimes B) \otimes (C \otimes D) \arrow[d, "\mathsf{assoc}_{A\otimes B,C,D}"] \\
 & ((A \otimes B) \otimes C) \otimes D \\
A \otimes ((B \otimes C) \otimes D) \arrow[r, "\mathsf{assoc}_{A,B\otimes C,D} \otimes id_D"'] & (A \otimes (B \otimes C)) \otimes D \arrow[u, "\mathsf{assoc}_{A,B,C} \otimes \mathsf{id}_D"']
\end{tikzcd}
\tag{4.1}
$$

$$
\begin{tikzcd}
A \otimes (I \otimes B) \arrow[rr, "\mathsf{assoc}_{A,I,B}"] \arrow[dr, "\mathsf{id}_A \otimes \mathsf{unitl}_B"'] & & (A \otimes I) \otimes B \arrow[dl, "\mathsf{unitr}_A \otimes \mathsf{id}_B"] \\
 & A \otimes B &
\end{tikzcd}
\tag{4.2}
$$

$$
\begin{tikzcd}
I \otimes I \arrow[rr, "\mathsf{unitl}_I"] \arrow[dr, "\mathsf{unitr}_I"'] & & I \arrow[dl, equal] \\
 & I &
\end{tikzcd}
\tag{4.3}
$$

kommutieren für alle Objekte A, B, C, D aus $\mathbb{C}$. Verschiedene Autoren (zum Beispiel [Poi92]) nennen Diagramm (4.3) nicht. Laut [EK65] folgt (4.3) aus den vorangehenden Axiomen.

Natürliche Isomorphismen heißen **kohärent**, wenn (etwas vereinfacht formuliert) alle Diagramme, die aus ihnen, ihren Inversen und aus Identitäten zusammengesetzt sind, kommutieren. Die Isomorphismen unitl, unitr und assoc sind kohärent (siehe [ML71] für eine ausführlichere Diskussion von Kohärenz und einen Beweis).

Eine **symmetrische monoidale Kategorie** ist eine monoidale Kategorie $\mathbb{C}$ mit einem natürlichen Isomorphismus $\mathsf{symm}_{A,B} : A \otimes B \longrightarrow B \otimes A$ für den gilt: Die Diagramme

$$\begin{tikzcd}
A \otimes (B \otimes C) \arrow[rr, "\mathsf{assoc}_{A,B,C}"] \arrow[d, "\mathsf{id}_A \otimes \mathsf{symm}_{B,C}"'] & & (A \otimes B) \otimes C \arrow[d, "\mathsf{symm}_{A \otimes B,C}"] \\
A \otimes (C \otimes B) \arrow[d, "\mathsf{assoc}_{A,C,B}"'] & & C \otimes (A \otimes B) \arrow[d, "\mathsf{assoc}_{C,A,B}"] \\
(A \otimes C) \otimes B \arrow[rr, "\mathsf{symm}_{A,C} \otimes \mathsf{id}_B"'] & & (C \otimes A) \otimes B
\end{tikzcd}$$

$$\begin{tikzcd}
A \otimes B \arrow[rr, "\mathsf{symm}_{A,B}"] \arrow[dr, equal] & & B \otimes A \arrow[dl, "\mathsf{symm}_{A,B}"] \\
& A \otimes B &
\end{tikzcd}
\qquad
\begin{tikzcd}
A \otimes I \arrow[rr, "\mathsf{symm}_{A,I}"] \arrow[dr, "\mathsf{unitr}_A"'] & & I \otimes A \arrow[dl, "\mathsf{unitl}_A"] \\
& A &
\end{tikzcd}$$

kommutieren für alle Objekte A, B, C aus $\mathbb{C}$. Wie die Ähnlichkeit der Bezeichnung schon vermuten läßt, wird mit dem symmetrischen Tensorprodukt das multiplikative „und“ der Linearen Logik interpretiert.

Eine symmetrische monoidale Kategorie $\mathbb{C}$ heißt **abgeschlossen**, falls für jedes Objekt X aus $\mathbb{C}$ der Funktor $- \otimes X : \mathbb{C} \longrightarrow \mathbb{C}$ einen rechtsadjungierten Funktor $X \multimap -$ besitzt[5]. Mit Hilfe dieses Funktors läßt sich eine Korrespondenz zwischen den Morphismen $X \xrightarrow{f} Y$ und den „Elementen“ des Objektes $X \multimap Y$ herstellen:

$$\begin{array}{c}
X \xrightarrow{\quad f \quad} Y \\
\hline\hline
I \otimes X \xrightarrow{f \circ \mathsf{unitl}_X} Y \\
\hline\hline
I \xrightarrow{(f \circ \mathsf{unitl}_X)^\vee} X \multimap Y
\end{array}
\begin{array}{l}
\mathsf{unitl} \\[1em]
(- \otimes X) \dashv (X \multimap -)
\end{array}
\tag{4.4}$$

Die Doppellinien $=\!=\!=$ sollen andeuten, daß die Äquivalenzen in beide Richtungen gelten: für $I \xrightarrow{g} X \multimap Y$ kann genau ein Morphismus $g^\wedge \circ \mathsf{unitl}_X^{-1} : X \longrightarrow Y$ bestimmt werden.

Das zu einem Morphismus $X \xrightarrow{f} Y$ gehörende Element wird mit $f^\sharp$ bezeichnet. Damit

[5] In anderen nicht zu Linearer Logik in Beziehung stehenden Publikationen wird dieser Funktor gewöhnlich mit $(-)^X$ bezeichnet.

ist $f^\sharp$ der eindeutig bestimmte Morphismus, der das Diagramm

$$\begin{tikzcd}
X \multimap Y & (X \multimap Y) \otimes X \arrow[rr, "\mathsf{eval}_{X,Y}"] & & Y \\
I \arrow[u, "f^\sharp"'] & I \otimes X \arrow[u, "f^\sharp \otimes \mathsf{id}_X"] \arrow[r, "\mathsf{unitl}_I"'] & X \arrow[ur, "f"'] &
\end{tikzcd}$$

kommutativ schließt. Der Morphismus eval ist die Counit der Adjunktion.

Umgekehrt wird der Morphismus, der zu einem Element $I \xrightarrow{g} X \multimap Y$ gehört, mit $g^\flat$ bezeichnet. Da die Regel 4.4 eine Bijektion darstellt, gilt $f^{\sharp\flat} = f$.

Der Funktor $X \multimap -$ läßt sich auf natürliche Weise zu einem Bifunktor $- \multimap - : \mathbb{C} \times \mathbb{C} \longrightarrow \mathbb{C}$ erweitern und wird wegen des eben beschriebenen Zusammenhanges „Internalisierung“[6] genannt.

Eine symmetrische monoidale Kategorie, die abgeschlossen ist, wird auch als **autonome Kategorie** bezeichnet.

Im folgenden werden zusätzliche Isomorphismen eingeführt, mit denen sich die weiteren Konnektoren der Linearen Logik interpretieren lassen. Betrachtet wird zuerst die Lineare Negation: Ein Objekt $\perp$, für das der Funktor $(-)^\perp \stackrel{\text{def}}{=} - \multimap \perp : \mathbb{C}^{\mathsf{op}} \longrightarrow \mathbb{C}$ eine Äquivalenz von Kategorien [ML71] ist, heißt **dualisierendes Objekt**.

Eine autonome Kategorie mit einem dualisierenden Objekt heißt **$*$–autonome Kategorie**.

Zur Interpretation des Konnektors $\parr$ wird ein Bifunktor $\parr : \mathbb{C} \times \mathbb{C} \longrightarrow \mathbb{C}$ mit Hilfe der Dualität definiert: $A \parr B \stackrel{\text{def}}{=} (A^\perp \otimes B^\perp)^\perp$. Ein so definierter Funktor liefert eine zweite monoidale Struktur mit dem neutralen Element $\perp$ auf $\mathbb{C}$. Eine ausführlichere Beschreibung und ein Beweis findet sich in [MOM90]. Damit ist eine $*$–autonome Kategorie ein Modell für das multiplikative Fragment der Linearen Logik, das aus den Konnektoren $\otimes, \parr, (-)^\perp, \multimap$ besteht.

Für eine Modellierung der additiven Konnektoren muß die Kategorie Produkte ($\&$) und Coprodukte ($\oplus$) enthalten. Das terminale Objekt wird mit $\top$ bezeichnet. Zu beachten ist, daß eine $*$–autonome Kategorie mit endlichen Produkten und einem dualisierenden Objekt immer auch Summen beziehungsweise Coprodukte enthält. Diese erhält man mit den Definitionen $A \oplus B \stackrel{\text{def}}{=} (A^\perp \& B^\perp)^\perp$ und $\mathbf{0} \stackrel{\text{def}}{=} \top^\perp$.

4.2 Lineare Kategorien

In der Literatur herrscht keine Einigkeit über die Bezeichnung bereicherter $*$–autonomer Kategorien. Hier soll den Konventionen von [Gay94] gefolgt werden.

[6] In der englischsprachigen Literatur wurde der Begriff „internal hom–functor“ geprägt.

Eine **Lineare Kategorie** ist eine $*$–autonome Kategorie mit endlichen Produkten und einem Comonad $(!, \epsilon, \delta)$, so daß für alle Objekte A aus $\mathbb{C}$ das Objekt $!A$ ein kommutativer Comonoid ist und die zusätzlichen Isomorphismen

$$
\begin{array}{rcl}
!(A \,\&\, B) & \cong & !A \otimes !B \\
!\top & \cong & I
\end{array}
\tag{4.5}
$$

gelten (Siehe Abschnitt 2.2 für eine Definition von Comonads und Comonoiden).

Eine lineare Kategorie ist ein Modell für klassische Lineare Logik. Für die Interpretation von ? kann ein Funktor $? : \mathbb{C} \longrightarrow \mathbb{C}$ mittels Dualität definiert werden: $?A \stackrel{\text{def}}{=} (!(A^{\perp}))^{\perp}$. Die Regeln „weakening", „contraction" und „dereliction" aus Abbildung 14 werden jeweils von den Morphismen η, μ und ϵ interpretiert. Falls $f : !A \longrightarrow B$ ist, dann kann der durch die Regel „promotion" entstehende Morphismus $f^! : !A \longrightarrow !B$ mit $f^! \stackrel{\text{def}}{=} !f \circ \delta$ konstruiert werden.

4.3 Semantik für Quantoren

Robert A. Seely skizziert in [See89] die Konstruktion einer indizierten Kategorie über einer Basiskategorie mit Produkten, um Quantoren erster Ordnung zu modellieren. In [See90] wird eine Hyperdoktrine als Modell polymorpher Linearer Logik vorgeschlagen.

Eine **indizierte lineare Kategorie** ist eine indizierte Kategorie $\Psi : \mathbb{C}^{\mathsf{op}} \longrightarrow \mathbf{Cat}$ über einer Kategorie $\mathbb{C}$ mit endlichen Produkten und der zusätzlichen Eigenschaft, daß

- jedes $\Psi(X)$ eine lineare Kategorie ist,
- alle Funktoren f^* lineare Funktoren sind, das heißt sie erhalten die gesamte lineare Struktur und
- die zu den kanonischen Projektionen π gehörenden Funktoren π^* sowohl linke als auch rechte Adjunktionen: $\forall \vdash \pi^* \vdash \exists$ haben.

Die Basiskategorie modelliert die Typen des Termkalküls. Die Objekte in einer Faser über einem Typ $\sigma_1 \times \cdots \times \sigma_n$ korrespondieren zu Formeln, die n freie Variablen der Typen σ_1 bis σ_n enthalten. Da die Fasern lineare Kategorien sind, lassen sich in ihnen alle Konnektoren der Linearen Logik interpretieren. Die Quantoren werden mit Hilfe der Adjunktionen interpretiert.

4.4 Korrektheit

In diesem Abschnitt wird die Behauptung, indizierte lineare Kategorien seien ein Modell für Lineare Logik, durch eine Beweisskizze untermauert. Eine genaue Ausarbeitung der Details war jedoch im Rahmen dieser Arbeit nicht möglich. Für die Beweisskizze muß zuerst die Interpretation von Formeln und Axiomen einer mehrsortigen Linearen Logik formalisiert werden. Danach folgt der Beweis, daß die Interpretationen die gültigen Ableitungen respektieren.

Interpretationen

Eine indizierte lineare Kategorie enthält genügend Struktur, um alle Konnektoren der mehrsortigen Linearen Logik zu interpretieren. Darauf wurde bei der Beschreibung der $*$–autonomen Struktur in den vorangegangenen Abschnitten hingewiesen. Diese Intuition wird jetzt präzisiert.

Der Begriff der Interpretation wird in Abhängigkeit einer mehrsortigen Signatur (vergleiche Abschnitt 3.2) und einer indizierten linearen Kategorie (siehe Abschnitt 4.3) definiert. Deshalb soll für diesen Teilabschnitt eine mehrsortige Signatur Σ und eine indizierte lineare Kategorie $\Psi : \mathsf{CL}(\Sigma)^{\mathsf{op}} \longrightarrow \mathbf{Cat}$ fixiert werden.

Sei $\mathfrak{V}$ eine Funktion, die jedem Relationssymbol $R : \sigma_1, \ldots, \sigma_n$ aus Σ ein Objekt in einer Faser von Ψ zuordnet, so daß

$$\mathfrak{V}(R) \quad \in \quad \Psi(\sigma_1, \ldots, \sigma_n) \tag{4.6}$$

gilt.

Aus jeder derartigen Funktion $\mathfrak{V}$ läßt sich eine Interpretation $[\![-]\!]_{\mathfrak{V}}$ der Formeln über Σ in den Fasern von Ψ frei erzeugen. Da eine Formel durch die Regel Weakening Interpretationen in unterschiedlichen Fasern haben kann, wird die Interpretation aller Formeln als eine Familie partieller Funktionen $[\![-]\!]_{\mathfrak{V}}^{\Upsilon}$ definiert. Der Index Υ reicht über alle Kontexte der Signatur Σ.

Im folgenden wird $[\![-]\!]_{\mathfrak{V}}^{\Upsilon}$ durch Induktion über den Formelaufbau definiert:

Propositionale Konstanten werden auf die entsprechenden Objekte der Faser über $\mathbf{1}$, dem terminalen Objekt von $\mathsf{CL}(\Sigma)$, abgebildet

$$\begin{aligned} [\![I]\!]_{\mathfrak{V}}^{\langle\rangle} &\stackrel{\text{def}}{=} I \\ [\![\mathbf{0}]\!]_{\mathfrak{V}}^{\langle\rangle} &\stackrel{\text{def}}{=} \mathbf{0} \\ [\![\bot]\!]_{\mathfrak{V}}^{\langle\rangle} &\stackrel{\text{def}}{=} \bot \\ [\![\top]\!]_{\mathfrak{V}}^{\langle\rangle} &\stackrel{\text{def}}{=} \top \end{aligned}$$

Relationssymbole Sei $\Upsilon \;\rhd\; R(t_1, \ldots, t_n) : \mathsf{Prop}$ mit der Regel Relation ableitbar. Dann gibt es in $\mathsf{CL}(\Sigma)$ einen Morphismus $t = (t_1, \ldots, t_n) : \Upsilon \longrightarrow (\sigma_1, \ldots, \sigma_n)$ und einen Pullbackfunktor $t^* : \Psi(\sigma_1, \ldots, \sigma_n) \longrightarrow \Psi(\Upsilon)$. Desweiteren gilt $\mathfrak{V}(R) \in |\Psi(\sigma_1, \ldots, \sigma_n)|$. Damit kann

$$[\![R(t_1, \ldots, t_n)]\!]_{\mathfrak{V}}^{\Upsilon} \quad \stackrel{\text{def}}{=} \quad t^*(\mathfrak{V}(R))$$

definiert werden.

Zweistellige Konnektoren werden mit der linearen Struktur der jeweiligen Faser interpretiert. Es sei $\ominus$ einer der zweistelligen Konnektoren $\otimes, ⅋, \multimap, \&, \oplus$.

$$[\![A \ominus B]\!]^{\Upsilon}_{\mathfrak{V}} \quad \stackrel{\text{def}}{=} \quad [\![A]\!]^{\Upsilon}_{\mathfrak{V}} \ominus [\![B]\!]^{\Upsilon}_{\mathfrak{V}}$$

Negation Wie man erwartet, wird

$$[\![A^{\perp}]\!]^{\Upsilon}_{\mathfrak{V}} \quad \stackrel{\text{def}}{=} \quad \left([\![A]\!]^{\Upsilon}_{\mathfrak{V}} \right)^{\perp}$$

definiert.

Weakening Vorausgesetzt es existiert eine Ableitung der Sequenz $\Upsilon, x : \sigma \;\triangleright\; A : \mathsf{Prop}$, in der die letzte Regel Weakening ist. Dann gilt auch $\Upsilon \;\triangleright\; A : \mathsf{Prop}$. Desweiteren gibt es einen Pullbackfunktor $\pi_1^* : \Psi(\Upsilon) \longrightarrow \Psi(\langle \Upsilon, x : \sigma \rangle)$, wobei π_1 die Projektion $\langle \Upsilon, x : \sigma \rangle \longrightarrow \Upsilon$ in $\mathsf{CL}(\Sigma)$ ist.

$$[\![A]\!]^{\langle \Upsilon, x:\sigma \rangle}_{\mathfrak{V}} \quad \stackrel{\text{def}}{=} \quad \pi_1^*([\![A]\!]^{\Upsilon}_{\mathfrak{V}})$$

$\forall, \exists$ Sei θ einer der Quantoren $\forall, \exists$ und $\Upsilon \;\triangleright\; \theta x : \sigma \,.\, A : \mathsf{Prop}$ mit der Regel Quantoren ableitbar, dann existiert ein Funktor $\theta : \Psi(\langle \Upsilon, x : \sigma \rangle) \longrightarrow \Psi(\Upsilon)$ als rechter bzw. linker Adjoint zu π_1^*, wobei π_1 die Projektion $\langle \Upsilon, x : \sigma \rangle \longrightarrow \Upsilon$ in $\mathsf{CL}(\Sigma)$ ist.

$$[\![\theta x : \sigma \,.\, A]\!]^{\Upsilon}_{\mathfrak{V}} \quad \stackrel{\text{def}}{=} \quad \theta \left([\![A]\!]^{\langle \Upsilon, x:\sigma \rangle}_{\mathfrak{V}} \right)$$

Lemma 4.7 (Formelinterpretation)
Sei Σ eine mehrsortige Signatur, $\Psi : \mathit{CL}(\Sigma)^{\mathsf{op}} \longrightarrow \mathbf{Cat}$ eine indizierte lineare Kategorie und $\mathfrak{V}$ eine Funktion, so daß Gleichung 4.6 erfüllt ist.

Dann gilt für alle Kontexte Υ über Σ:

- *$[\![-]\!]^{\Upsilon}_{\mathfrak{V}}$ ist genau für die Formeln A definiert, für die es eine Ableitung $\Upsilon \;\triangleright\; A : \mathit{Prop}$ mit den Regeln aus den Abbildungen 10 und 11 gibt.*
- *$[\![A]\!]^{\Upsilon}_{\mathfrak{V}}$ ist ein Objekt der Kategorie $\Psi(\Upsilon)$.*

Beweis Der Beweis folgt direkt aus der Definition von $[\![-]\!]^{\Upsilon}_{\mathfrak{V}}$ und den vorausgesetzten Eigenschaften für Σ, Ψ und $\mathfrak{V}$. □

Damit kann die Interpretation einer mehrsortigen Linearen Logik definiert werden.

Definition 4.8 (Interpretationen) *Sei $\mathfrak{L} = (\Sigma, \mathfrak{A})$ eine mehrsortige Lineare Logik und $\Psi : \mathit{CL}(\Sigma)^{\mathsf{op}} \longrightarrow \mathbf{Cat}$ eine indizierte lineare Kategorie. Eine **Interpretation** $\mathfrak{I}$ von $\mathfrak{L}$ in Ψ ist ein Paar indizierter Familien $(\{\mathfrak{V}_R \mid R$ ein Relationssymbol aus $\Sigma\}, \{\mathfrak{S}_s \mid s \in \mathfrak{A}\})$, so daß*

- $\mathfrak{V}_R \in |\Psi(\sigma_1, \ldots, \sigma_n)|$ *wenn* $R : \sigma_1, \ldots, \sigma_n$ *gilt und*
- $\mathfrak{S}_{\langle\rangle \,\triangleright\, \Gamma \,\vdash\, \Delta} : \mathfrak{V}(\Gamma) \longrightarrow \mathfrak{V}(\Delta)$ *in* $\Psi(\mathbf{1})$

Konventionen 4.9 In den einzelnen Beweisschritten müssen häufig sämtliche Formeln einer Seite einer Sequenz durch einen Konnektor verbunden werden. Angenommen, Γ besteht aus den Formeln $A_1, \ldots, A_n$. Zur Vereinfachung werden die folgenden Abkürzungen benutzt:

$$\begin{aligned} \otimes\Gamma &= A_1 \otimes \cdots \otimes A_n \\ \parr\, \Gamma &= A_1 \parr \cdots \parr A_n \\ \Gamma^\perp &= A_1^\perp, \ldots, A_n^\perp \end{aligned}$$

Zur weiteren Vereinfachung werden selbst die Operationssymbole $\otimes$ und $\parr$ vor Listen von Objekten weggelassen. Mit einem Morphismus zwischen zwei Formelgruppen $\Gamma \longrightarrow \Delta$ ist immer ein Morphismus zwischen den Objekten $\otimes\Gamma \longrightarrow \parr\,\Delta$ gemeint. Das heißt: Ist eine Formelgruppe die Domäne eines Morphismus, dann wird implizit angenommen, daß von allen Formeln das Tensorprodukt gebildet ist. Für die Codomäne gilt das gleiche bezüglich $\parr$.

Da die Konnektoren der Linearen Logik mit den gleichen Symbolen bezeichnet werden, wie die sie interpretierenden Formeln, wird im folgenden nicht mehr zwischen einer Formel A und dem ihr zugeordneten Objekt $[\![A]\!]^\Upsilon_\mathfrak{V}$ unterschieden.

Ist der Kontext Υ aus dem Zusammenhang klar, steht $\Gamma \longrightarrow \Delta$ damit für einen Morphismus zwischen $[\![A_1 \otimes \cdots \otimes A_n]\!]^\Upsilon_\mathfrak{V}$ und $[\![B_1 \parr \cdots \parr B_n]\!]^\Upsilon_\mathfrak{V}$, wenn wie oben $\Gamma = A_1, \ldots, A_n$ und $\Delta = B_1, \ldots, B_n$ ist.

Satz 4.10 *Sei* $\mathfrak{L} = (\Sigma, \mathfrak{A})$ *eine mehrsortige Lineare Logik und* $\Psi : \mathsf{CL}(\Sigma)^{\mathsf{op}} \longrightarrow \mathbf{Cat}$ *eine indizierte lineare Kategorie. Dann gibt es für jede Interpretation* $\mathfrak{I} = (\mathfrak{V}, \mathfrak{S})$ *und jede in* $\mathfrak{L}$ *ableitbare Sequenz* $\Upsilon \;\triangleright\; \Gamma \;\vdash\; \Delta$ *einen Morphismus* $[\![\Gamma]\!]^\Upsilon_\mathfrak{V} \longrightarrow [\![\Delta]\!]^\Upsilon_\mathfrak{V}$ *in* $\Psi(\Upsilon)$*.*

Beweisskizze Unter den Voraussetzungen des Satzes läßt sich aus jeder beliebigen Interpretation $\mathfrak{I} = (\mathfrak{V}, \mathfrak{S})$ eine Interpretation der Formeln $[\![-]\!]^\Upsilon_\mathfrak{V}$ erzeugen, die nach Lemma 4.7 (Formelinterpretaion) jeder Formel aus $\mathfrak{L}$ ein Objekt in einer Faser von Ψ zuordnet.

Zu beweisen bleibt, daß für jede gültige Sequenz $\Upsilon \;\triangleright\; \Gamma \;\vdash\; \Delta$ ein Morphismus $[\![\otimes\Gamma]\!]^\Upsilon_\mathfrak{V} \longrightarrow [\![\parr\,\Delta]\!]^\Upsilon_\mathfrak{V}$ in $\Psi(\Upsilon)$ existiert.

Sei für den Rest des Beweises eine mehrsortige Lineare Logik $\mathfrak{L}$ über der Signatur Σ mit der Axiomenmenge $\mathfrak{A}$, eine indizierte lineare Kategorie Ψ und eine Interpretation $\mathfrak{I}$ beliebig, aber fest gewählt.

Da der Beweis im wesentlichen aus der Konstruktion von Morphismen zwischen bestimmten Objekten besteht, wird den Isomorphismen der beiden abgeschlossenen monoidalen Strukturen in den meisten Fällen keine Aufmerksamkeit geschenkt.

Anstatt die Morphismen direkt zu konstruieren, werden Ableitungen folgender Form angegeben:

$$\frac{U \longrightarrow V}{X \longrightarrow Y}\ \text{Eigenschaft}$$

Damit ist gemeint: Aus einem Morphismus $U \longrightarrow V$ läßt sich wegen der genannten Eigenschaft einer indizierten linearen Kategorie auf kanonische Weise ein Morphismus $X \longrightarrow Y$ in möglicherweise einer anderen Faser konstruieren.

Zum Beispiel bezieht sich „Definition $\parr$“ auf die Definition des Bifunktors $\parr$ aus Abschnitt 4.1. Dort wurde $A \parr B$ mit Hilfe von $\otimes$ und $(-)^\perp$ definiert, wobei $(-)^\perp$ selbst mit Hilfe von $\perp$ und $\multimap$ definiert wurde. Damit ergibt sich

$$A \parr B \quad = \quad (A^\perp \otimes B^\perp)^\perp \quad = \quad (A^\perp \otimes B^\perp) \multimap \perp$$

und jeder Morphismus $U \longrightarrow V \parr W$ ist auch ein Morphismus $U \longrightarrow (V^\perp \otimes W^\perp) \multimap \perp$.

Vor der Fortsetzung der Beweisskizze, wird eine vereinfachte Negationsregel bewiesen:

Lemma 4.11 (Negationslemma) *Ist $\Gamma \longrightarrow A, \Delta$ ein Morphismus in einer linearen Kategorie $\mathbb{C}$, dann gibt es einen Morphismus $\Gamma, A^\perp \longrightarrow \Delta$ in $\mathbb{C}$.*

Umgekehrt läßt sich aus jedem Morphismus $\Gamma, A \longrightarrow \Delta$ ein Morphismus $\Gamma \longrightarrow A^\perp, \Delta$ konstruieren.

Beweis Angenommen, $\Gamma \longrightarrow A, \Delta$ ist gegeben, dann gilt

$$\begin{array}{cl}
\Gamma \longrightarrow A \parr \Delta & \\
\hline\hline
 & \text{Definition } \parr \\
\Gamma \longrightarrow A^\perp \otimes (\otimes\Delta^\perp) \multimap \perp & \\
\hline\hline
 & \multimap \vdash \otimes \\
\otimes\Gamma \otimes A^\perp \otimes (\otimes\Delta^\perp) \longrightarrow \perp & \\
\hline\hline
 & \multimap \vdash \otimes \\
\Gamma \otimes A^\perp \longrightarrow (\otimes\Delta^\perp) \multimap \perp & \\
\hline\hline
 & \text{Definition } \parr \\
\Gamma, A^\perp \longrightarrow \Delta &
\end{array}$$

Der zweite Teil des Lemmas ergibt sich aus der Substitution von $A^\perp$ für A in obiger Ableitung und dem Fakt, daß $(-)^\perp$ eine Involution ist. □

Der eigentliche Beweis von Satz 4.10 wird mit Induktion über die Struktur der möglichen korrekten Ableitungen geführt.

Induktionsanfang

Die Morphismen zur Interpretation der Axiome aus Abbildung 12 sind durch Identitäten bzw. initiales und terminales Objekt in jeder $*$–autonomen Kategorie gegeben. Im einzelnen:

Bottom L, Unit R Die leeren Formelmengen werden auf die entspechenden neutralen Elemente abgebildet, so daß beide Axiome durch die Identitäten $\mathsf{id}_\perp$ und id_I interpretiert werden.

id_A wird durch die Identität interpretiert.

Zero L Aus dem initialen Morphismus $\mathbf{0} \longrightarrow \Gamma^\perp, \Delta$ kann mit dem Negationslemma 4.11 und der Symmetrie des Tensorproduktes einen Morphismus $\Gamma, \mathbf{0} \longrightarrow \Delta$ konstruiert werden.

Top R Es wird die gleiche Idee wie für Zero L verwendet, diesmal mit dem terminalen Objekt.

Axiome aus $\mathfrak{A}$ Für die Axiome $\mathfrak{A}$ der Logik $\mathfrak{L}$ werden Morphismen durch die Interpretation $\mathfrak{I}$ gegeben.

Induktionsschritt

Im folgenden wird für jede Regel aus den Abbildungen 12 bis 14 ein Morphismus konstruiert.

Unit L, $\perp$ R Die Konklusion wird jeweils durch den Morphismus der Prämisse interpretiert, weil $[\![-]\!]^{\Upsilon}_{\mathfrak{D}}$ die neutralen Elemente respektiert.

$\otimes$ L, $\parr$ R Auch hier wird der gleiche Morphismus benutzt. Dies folgt direkt aus den Konventionen zur Interpretation der Formellisten von Sequenzen (vergleiche Punkt 4.9).

$\otimes$ R, $\parr$ L Die Morphismen der Prämissen werden mit dem entsprechenden Funktor verknüpft.

$\multimap$ R,L In der klassischen Linearen Logik gilt die Gleichheit von $A \multimap B = A^\perp \parr B$ per Definition. Damit folgen die Regeln $\multimap$ R und $\multimap$ L aus den entsprechenden Regeln für $\parr$. Es muß deshalb nur gezeigt werden, daß in einer linearen Kategorie die Objekte $A \multimap B$ und $A^\perp \parr B$ isomorph sind. Der Beweis erfolgt mittels des Yonedalemmas:

$$
\begin{array}{rcll}
X & \longrightarrow & A \multimap B & \\ \hline\hline
X \otimes A & \longrightarrow & B & \multimap \vdash \otimes \\ \hline\hline
X \otimes A & \longrightarrow & B^{\perp\perp} & \text{Involution } \perp \\ \hline\hline
X \otimes A \otimes B^\perp & \longrightarrow & \perp & \multimap \vdash \otimes \\ \hline\hline
X & \longrightarrow & (A \otimes B^\perp)^\perp & \multimap \vdash \otimes \\ \hline\hline
X & \longrightarrow & A^\perp \parr B & \text{Definition } \parr
\end{array}
$$

Permutation Sowohl $(\otimes, I)$ als auch $(\parr, \perp)$ sind symmetrische monoidale Strukturen in jeder Faser.

Negation

$$\dfrac{\dfrac{\dfrac{\Gamma, A \longrightarrow B, \Delta}{\Gamma, A, B^\perp \longrightarrow \Delta}\ \text{Lemma 4.11}}{\Gamma, B^\perp, A \longrightarrow \Delta}\ \text{Symmetrie}}{\Gamma, B^\perp \longrightarrow A^\perp, \Delta}\ \text{Lemma 4.11}$$

$\&, \oplus$ Die additiven Konnektoren werden durch das Produkt bzw. die Summe in der jeweiligen Faser interpretiert. Für die Regel $\&$ L1 gilt

$$\dfrac{\dfrac{\dfrac{\Gamma, A \longrightarrow \Delta}{A \longrightarrow \Gamma^\perp, \Delta}\ \text{Lemma 4.11}}{A \,\&\, B \longrightarrow \Gamma^\perp, \Delta}\ \pi_1 \circ -}{\Gamma, (A \,\&\, B) \longrightarrow \Delta}\ \text{Lemma 4.11}$$

Der Beweis für die Regel $\&$ L2 ist vollkommen identisch, nur wird die zweite Projektion π_2 benutzt. Für die Regel $\&$ R ergibt sich

$$\dfrac{\dfrac{\text{Lemma 4.11}\ \dfrac{\Gamma \longrightarrow A, \Delta}{\Gamma, \Delta^\perp \longrightarrow A} \qquad \dfrac{\Gamma \longrightarrow B, \Delta}{\Gamma, \Delta^\perp \longrightarrow B}\ \text{Lemma 4.11}}{\Gamma, \Delta^\perp \longrightarrow A \,\&\, B}\ \langle -, - \rangle}{\Gamma \longrightarrow (A \,\&\, B), \Delta}\ \text{Lemma 4.11}$$

Die Regeln für $\oplus$ werden in der gleichen Art bewiesen, nur werden jeweils linke und rechte Seite vertauscht und an die Stelle der kanonischen Morphismen des Produktes treten die des Coproduktes.

Cut Die Regel Cut wird durch die Komposition von Morphismen interpretiert.

$$\dfrac{\dfrac{\text{Lemma 4.11}\ \dfrac{\Gamma \longrightarrow \Delta, A}{\Gamma, \Delta^\perp \longrightarrow A} \qquad \dfrac{A, \Theta \longrightarrow \Psi}{A \longrightarrow \Theta^\perp, \Psi}\ \text{Lemma 4.11}}{\Gamma, \Delta^\perp \longrightarrow \Theta^\perp, \Psi}\ - \circ -}{\Gamma, \Theta \longrightarrow \Delta, \Psi}\ \text{Negation}$$

Regeln zu ! Zum Beweis sämtlicher Regeln für den modalen Operator ! werden die Eigenschaften, des zu jeder Faser gehörenden Comonads $(!, \epsilon, \delta)$ und der Comonoide $!A$ genutzt (vergleiche Abschnitt 2.2). Alle Konstruktionen finden vollständig in der zu dem Kontext Υ gehörenden Faser $\Psi(\Upsilon)$ statt.

Promotion Ist $f : !\Gamma \longrightarrow A$ ein Morphismus in $\Psi(\Upsilon)$, dann ist $f^! = !f \circ \delta : !\Gamma \longrightarrow !A$.

Weakening Sei der Comonoid für $!A$ durch die Morphismen η_A und μ_A gegeben. Für einen Morphismus $f : \Gamma \longrightarrow \Delta$ gilt $\mathsf{unitr}^{-1}_{\parr\Delta} \circ (f \otimes \eta_A) : \Gamma, A \longrightarrow \Delta$.

Dereliction Für $f : A \longrightarrow \Gamma$ gilt $f \circ \epsilon_A :!A \longrightarrow \Gamma$.

Contraction Der die Prämisse interpretierende Morphismus wird mit $\mu_{!A}$ verkettet.

Regeln zu ? Der Funktor ? ist in einer linearen Kategorie identisch mit $(!(-)^\perp)^\perp$. Deshalb folgen alle Regeln zu ? aus den Regeln zu ! in Verbindung mit dem Negationslemma 4.11.

$\forall$ R, $\exists$ L Beide Regeln werden mit Hilfe der Adjunktion zwischen den Fasern über $\Upsilon \times \sigma$ und Υ interpretiert. Hier wird nur $\forall$ R bewiesen, $\exists$ L läßt sich vollkommen analog zeigen.

Durch die Eigenvariablebedingung tritt die Variable x weder in Γ noch in Δ auf, und es gibt Interpretationen G und D der Formeln Γ bzw. Δ in der Faser Υ. Aus der Definition der indizierten linearen Kategorie folgt $\pi^*(G) = \otimes\Gamma$ und $\pi^*(D) = \parr\, \Delta$.

$$
\begin{array}{llcl}
\text{über } \Upsilon \otimes \sigma: & (\otimes\Gamma) \longrightarrow \Delta, A & \\
\hline
& \pi^*(G) \longrightarrow \pi^*(D) \parr A & \\
\hline
& \pi^*(G), \pi^*(D)^\perp \longrightarrow A & \text{Lemma 4.11} \\
\hline
& \pi^*(G \otimes D^\perp) \longrightarrow A & \text{linearer Funktor} \\
\hline
\text{über } \Upsilon \ : & G \otimes D^\perp \longrightarrow \forall A & \pi^* \dashv \forall \\
\hline
& G \longrightarrow D \parr \forall A & \text{Lemma 4.11}
\end{array}
$$

$\forall$ L, $\exists$ R Auch diese Ableitung nutzt im wesentlichen die Struktur der zwischen den Fasern bestehenden Adjunktionen aus. Wie im vorhergehenden Punkt wird nur eine der beiden Regeln bewiesen und zwar $\exists$ R.

Nach Voraussetzung ist t ein Term des Types σ. Damit ist t auch ein Morphismus $\Upsilon \longrightarrow x : \sigma$ in $\mathsf{CL}(\Sigma)$, und unter Hinzunahme der Variablen aus Υ erhält man einen Morphismus $t' : \Upsilon \longrightarrow \Upsilon, x : \sigma$.

$$
\begin{array}{llc}
\hline
\text{über } \Upsilon : & \exists A \longrightarrow \exists A & \mathsf{id} \\
\hline
\text{über } \Upsilon, \sigma : & A \longrightarrow \pi^*\exists A & \exists \dashv \pi^* \\
\hline
& t'^*(A) \longrightarrow t'^*\pi^*\exists A & t'^* \\
\hline
& t'^*(A) \longrightarrow \exists A & (*)
\end{array}
$$

In der Kategorie $\mathsf{CL}(\Sigma)$ gilt $\pi \circ t' = \mathsf{id}_\Upsilon$. Dadurch sind $t'^*\pi^*\exists A$ und $\exists A$ isomorph (Ableitungsschritt $(*)$). Dieses Zwischenergebnis kann mit der Voraussetzung der Regel $\exists$ R komponiert (bzw. geschnitten) werden. Die Verkettung ist möglich, weil die Pullbackfunktoren t^* genau die Substitution für Formeln realisieren.

$$\dfrac{\dfrac{\text{Lemma 4.11}\ \dfrac{\Gamma \longrightarrow \Delta, A[t/x]}{\Gamma, \Delta^\perp \longrightarrow A[t/x]} \qquad t'^*(A) \longrightarrow \exists A}{\Gamma, \Delta^\perp \longrightarrow \exists A}\ -\circ-}{\Gamma \longrightarrow \Delta, \exists A}\ \text{Lemma 4.11}$$

□

Die Beweisskizze belegt, daß spezielle indizierte lineare Kategorien als Modell für mehrsortige Lineare Logik dienen können. Die Definition aus [See89] können für den mehrsortigen Fall erweitert werden.

5. Die Interaktionskategorie $\mathcal{SP}roc$

We propose *Interaction categories* as a new
paradigm for the semantics of computation.
Samson Abramsky

In diesem Kapitel wird die Interaktionskategorie $\mathcal{SP}roc$ [AGN93] kurz vorgestellt.

5.1 Objekte und Morphismen

$\mathcal{SP}roc$ ist eine bereicherte Kategorie, deren Objekte und Morphismen als Typspezifikationen von Prozessen verstanden werden. In Anlehnung an den Curry–Howard–Isomorphismus ist $\mathcal{SP}roc$ auch ein Modell für klassische Lineare Logik.

Ein Objekt der Kategorie $\mathcal{SP}roc$ ist ein Paar $A = (\Sigma_A, S_A)$, wobei Σ_A ein Alphabet und S_A eine nichtleere, präfixabgeschlossene Teilmenge aller endlichen Worte über dem Alphabet Σ_A ist. Die Menge S_A wird als Sicherheitsspezifikation interpretiert.

Objekte von $\mathcal{SP}roc$ werden als Prozeßtypen verstanden. Ein Prozeß p hat den Typ A, geschrieben $p : A$, wenn jeder Pfad des Synchronisationsbaumes von p in S_A enthalten ist: $\mathsf{traces}(p) \subseteq S_A$. Die Menge aller Prozesse vom Typ A wird mit $\mathsf{proc}(A)$ bezeichnet:

$$\mathsf{proc}(A) \quad \stackrel{\text{def}}{=} \quad \{p \mid \mathsf{traces}(p) \subseteq S_A\}$$

Es ist üblich, die Morphismen von $\mathcal{SP}roc$ mit Hilfe der Internalisierung zu definieren. Deshalb hier zunächst die Definition des Tensors $\otimes$: Für gegebene A und B ist das Objekt $A \otimes B$ das Paar $(\Sigma_{A\otimes B}, S_{A\otimes B})$ mit

$$\begin{aligned} \Sigma_{A\otimes B} &= \Sigma_A \times \Sigma_B \\ S_{A\otimes B} &= \{\sigma \in \Sigma^*_{A\otimes B} \mid \pi^*_1(\sigma) \in S_A \wedge \pi^*_2(\sigma) \in S_B\}. \end{aligned}$$

Die Dualität für Objekte wird trivial definiert: $A^\perp \stackrel{\text{def}}{=} A$. Mit der allgemeinen Definition des Bifunktors $\invamp$ folgt $A \invamp B = A \multimap B = A \otimes B$.

Die Komposition zweier Morphismen wird in [GN95] mit Hilfe einer Ableitungsregel definiert. Wenn $p : B \multimap C$ und $q : A \multimap B$ gilt, dann wird der Prozeß $p \circ q : A \multimap C$ mit

$$\frac{p \xrightarrow{(b,c)} p' \qquad q \xrightarrow{(a,b)} q'}{p \circ q \xrightarrow{(a,c)} p' \circ q'} \tag{5.1}$$

definiert.

Um die Identität nach [GN95] definieren zu können, wird die Einschränkung eines Prozesses auf eine Sicherheitsspezifikation benötigt:

$$\frac{P \xrightarrow{a} Q \qquad a \in S}{P \restriction S \xrightarrow{a} Q \restriction (S/a)} \tag{5.2}$$

wobei $S/a \stackrel{\text{def}}{=} \{\epsilon\} \cup \{\sigma \mid a\sigma \in S\}$. Wenn der Prozeß id^A durch die folgende Regel definiert wird,

$$\frac{a \in \Sigma_A}{\mathsf{id}^A \xrightarrow{(a,a)} \mathsf{id}^A} \tag{5.3}$$

dann ist

$$\mathsf{id}_A \quad \stackrel{\text{def}}{=} \quad \mathsf{id}^A \restriction S_{A \multimap A} \tag{5.4}$$

Der Beweis, daß die eben definierte Komposition und die Identitäten die Axiome einer Kategorie erfüllen, ist in [Gay94] zu finden und soll hier nicht wiederholt werden.

5.2 $\mathcal{SP}roc$ als $*$–autonome Kategorie

Damit $\mathcal{SP}roc$ den Axiomen einer $*$–autonomen Kategorie genügt, müssen die Definitionen für $\otimes$ und $(-)^\perp$ zu Funktoren geliftet werden. Wenn $p : A \longrightarrow C$ und $q : B \longrightarrow D$ gilt, dann wird $p \otimes q : A \otimes B \longrightarrow C \otimes D$ und $p^\perp : C^\perp \longrightarrow A^\perp$ mit Hilfe der folgenden Transitionsregeln definiert.

$$\frac{p \xrightarrow{(a,c)} p' \qquad q \xrightarrow{(b,d)} q'}{p \otimes q \xrightarrow{((a,b),(c,d))} p' \otimes q'} \qquad\qquad \frac{p \xrightarrow{(a,c)} p'}{p^\perp \xrightarrow{(c,a)} {p'}^\perp}$$

Das neutrale Element I des Tensors $\otimes$ ist das Paar (Σ_I, S_I) mit

$$\begin{aligned} \Sigma_I &\stackrel{\text{def}}{=} \{*\} \\ S_I &\stackrel{\text{def}}{=} \{*^n \mid n \in \mathbb{N}\} \end{aligned}$$

Die rechte Adjunktion zum Tensor kann mit der folgenden eineindeutigen Beziehung zwischen den Morphismen $p : A \otimes B \longrightarrow C$ und $\Lambda(p) : A \longrightarrow B \multimap C$ konstruiert werden.

$$\frac{p \xrightarrow{((a,b),c)} q}{\Lambda(p) \xrightarrow{(a,(b,c))} \Lambda(q)}$$

Die Definition der anderen Morphismen, zum Beispiel symm oder unitl, kann auf ähnliche Weise durch Umgruppieren von Tupeln oder Vertauschen von Paaren erfolgen.

5.3 $\mathcal{SP}roc$ als lineare Kategorie

Für zwei Objekte A und B aus $\mathcal{SP}roc$ ist das Coprodukt das Paar $(\Sigma_{A\oplus B}, S_{A\oplus B})$, wobei

$$\begin{aligned} \Sigma_{A\oplus B} &\stackrel{\text{def}}{=} \Sigma_A \uplus \Sigma_B \\ S_{A\oplus B} &\stackrel{\text{def}}{=} \{\mathsf{inl}^*(s) |\ s \in S_A\} \uplus \{\mathsf{inr}^*(s) |\ s \in S_B\}. \end{aligned}$$

Die bis auf Eindeutigkeit bestimmten Morphismen $[\_,\_]$ erhält man leicht durch Umbenennung von Transitionen. Für die Universalität der konstruierten Morphismen spielen die Sicherheitsspezifikationen eine wesentliche Rolle. Eine ausführliche Erläuterung mit Gegenbeispielen ist in [Gay94] enthalten.

Zum Beispiel gilt für die linke Injektion inl_A:

$$\frac{\mathsf{id}_A \xrightarrow{(a,a)} \mathsf{id}_{A'}}{\mathsf{inl}_A \xrightarrow{(a,\mathsf{inl}(a))} \mathsf{inl}_{A'}}$$

Da in $\mathcal{SP}roc$ die Dualität durch die Identität definiert ist, fallen auch die additiven Konnektoren $\oplus$ und $\&$ zusammen. Das initiale und terminale Objekt $\mathbf{0}$ ist durch

$$\begin{aligned} \Sigma_{\mathbf{0}} &= \{\} \\ S_{\mathbf{0}} &= \{\epsilon\} \end{aligned}$$

gegeben. Der einzige Prozeß vom Typ $\mathbf{0}$ wird nil genannt. Er ist nicht in der Lage, irgendeine Aktion auszuführen.

Die Kategorie $\mathcal{SP}roc$ weist als $*$–autonome Kategorie besondere Struktur auf. Die Funktoren $\otimes$ und $\invamp$ sind isomorph, das heißt $\mathcal{SP}roc$ ist „compact closed“, und es gibt endliche Biprodukte [ML71]. Für eine $*$–autonome Kategorie mit diesen Eigenschaften ergibt sich die zur Interpretation der modalen Operatoren ! und ? notwendige Struktur aus der rechten Adjunktion zum „forgetful functor“, der die Kategorie der Comonoide auf die Kategorie selbst abbildet.

Da in diesem Abschnitt die Kategorie $\mathcal{SP}roc$ nur vorgestellt werden soll, wird auf die allgemeine Konstruktion nicht eingegangen. Sie ist unter anderem in [AGN93, Gay94] zu finden. Statt dessen werden die Objekte und Morphismen, welche die Regeln zu den modalen Operatoren interpretieren, direkt definiert.

Zunächst werden einige Bezeichnungen benötigt. Multimengen sollen mit den Symbolen $\{\!|$ und $|\!\}$ eingeschlossen werden. $\mathcal{M}_n(X)$ bezeichne die Multimengen der Kardinalität n mit Elementen aus X. Die symmetrische Potenz des Tensors $\otimes_s^n$ wird für Objekte durch

$$\begin{aligned} \Sigma_{\otimes_s^n A} &\stackrel{\text{def}}{=} \mathcal{M}_n(\Sigma_A) \\ S_{\otimes_s^n A} &\stackrel{\text{def}}{=} \{\phi_n^*(s) \mid s \in S_{\otimes^n A}\} \end{aligned}$$

definiert. Die Abbildung $\phi_n : \Sigma_A^n \longrightarrow \mathcal{M}_n(\Sigma_A)$ bildet n–Tupel in Multimengen mit n Elementen ab: $\phi_n(l_1, \ldots, l_n) = \{\!| l_1, \ldots, l_n |\!\}$. Der Morphismenteil des symmetrischen Tensorproduktes soll auch hier durch eine Transitionsregel beschrieben werden.

$$\frac{\otimes^n p \xrightarrow{(l_1, \ldots, l_n)} \otimes^n q}{\otimes_s^n p \xrightarrow{\{\!| l_1, \ldots, l_n |\!\}} \otimes_s^n q}$$

Mit diesen Hilfsmitteln kann ! definiert werden:

$$!A \stackrel{\text{def}}{=} \bigoplus_{n \in \mathbb{N}} \otimes_s^n A \tag{5.5}$$

Der Morphismenteil des Funktors wird in der gleichen Art definiert.

Der Morphismus $\mathsf{der}_A : !A \longrightarrow A$, der die Interpretation der Regel Dereliction ermöglicht, selektiert genau die Komponente aus $!A$, die nur eine Kopie von A enthält:

$$\frac{\mathsf{id}_A \xrightarrow{(a, a)} \mathsf{id}_{A'}}{\mathsf{der}_A \xrightarrow{(\{\!| a |\!\}, a)} \mathsf{der}_{A'}}$$

Zur Struktur des Comonads gehört weiterhin eine Operation $(-)^!$ für Morphismen. Dafür ist eine Hilfskonstruktion notwendig. Sei $!A \xrightarrow{f} B$ ein Morphismus, dann wird $!A \xrightarrow{f_n} \otimes_s^n B$ durch die Regel

$$\frac{\otimes_s^n f \xrightarrow{(\{\!| \{\!| a_{11}, \ldots, a_{1m_1} |\!\}, \ldots, \{\!| a_{n1}, \ldots, a_{nm_n} |\!\} |\!\}, \{\!| b_1, \ldots, b_n |\!\})} \otimes_s^n f'}{f_n \xrightarrow{(\{\!| a_{11}, \ldots, a_{nm_n} |\!\}, \{\!| b_1, \ldots, b_n |\!\})} f'_n}$$

definiert. Daraus wird $f^!$ gebildet:

$$f^! \stackrel{\text{def}}{=} [f_n]_{n \in \mathbb{N}}$$

Das Tripel $(T, \mathsf{der}, (-)^!)$ erfüllt die Gleichungen 2.4 aus Abschnitt 2.2 (siehe [AGN93] für eine ausführlichere Darstellung), und mit Satz 2.3 ist T ein Comonad.

Der Comonoid für das Objekt $!A$ wird durch die beiden Morphismen $!A \xrightarrow{\mathsf{weak}_A} I$ und $!A \xrightarrow{\mathsf{contr}_A} !A \otimes !A$ definiert:

$$\frac{\mathsf{id}_I \xrightarrow{(*,*)} \mathsf{id}_I}{\mathsf{weak}_A \xrightarrow{(\{\!|\ |\!\},*)} \mathsf{weak}_{A'}}$$

Zur Definition von contr_A wird erst der Morphismus

$$\mathsf{contr}_A^{(r,m,n)} : \otimes_s^r A \longrightarrow (\otimes_s^m A) \otimes (\otimes_s^n A)$$

konstruiert:

$$\frac{\mathsf{id}_{\otimes_s^n A} \xrightarrow{(\alpha,\alpha)} \mathsf{id}_{\otimes_s^n A'}}{\mathsf{contr}_A^{(r,m,n)} \xrightarrow{(\alpha,(\beta,\gamma))} \mathsf{contr}_{A'}^{(r,m,n)}} \quad |\beta| = m,\ |\gamma| = n,\ \alpha = \beta \cup \gamma$$

Damit ergibt sich $\mathsf{contr}_A^{(r,m,n)} = \mathsf{nil}$ für $r \neq m + n$ und es gilt

$$\mathsf{contr}_A \stackrel{\text{def}}{=} \bigoplus_{r,n,m \in \mathbb{N}} \mathsf{contr}_A^{(r,n,m)}$$

Als letztes Detail aus der Struktur von $\mathcal{SP}roc$ soll hier noch der Isomorphismus

$$\iota\ : !(A \,\&\, B) \longrightarrow !A \otimes !B$$

illustriert werden:

$$\frac{\mathsf{id}_{!A\otimes !B} \xrightarrow{((\alpha,\beta),(\alpha,\beta))} \mathsf{id}_{!A'\otimes !B'}}{\iota_{A,B} \xrightarrow{(\{\!|\ \mathsf{inl}(a_1),\ldots,\mathsf{inl}(a_n),\mathsf{inr}(b_1),\ldots,\mathsf{inr}(b_m)\ |\!\},(\alpha,\beta))} \iota_{A,B}} \quad \begin{array}{l}\alpha = \{\!|\ a_1,\ldots,a_n\ |\!\} \\ \beta = \{\!|\ b_1,\ldots,b_m\ |\!\}\end{array}$$

Damit gilt, wie von Robert A. Seely in [See89] gefordert,

$$\iota_{A,A} \circ !\triangle \quad = \quad \mathsf{contr}_A \tag{5.6}$$

5.4 $\mathcal{SP}roc$ und Cut–Elimination

Aus Abschnitt 4.2 folgt direkt, daß $\mathcal{SP}roc$ ein Modell für eine beliebige Lineare Logik ohne Quantoren ist. Eine weiterführende Fragestellung ist, ob $\mathcal{SP}roc$ auch den Prozeß der Cut–Elimination modelliert. Damit ist gemeint, ob die beiden Morphismen, die aus einer Ableitung vor und nach dem Prozeß der Cut–Elimination gewonnen werden können, in jedem Fall identisch sind.

Es gilt

Satz 5.7 *Die Kategorie $\mathcal{SP}roc$ modelliert den Prozeß der Cut–Elimination **nicht**.*

Das bedeutet, es gibt einen gültigen Ableitungsbaum $\mathcal{T}$, der durch einen Morphismus f modelliert wird, so daß der Morphismus f', der den durch Cut–Elimination aus $\mathcal{T}$ entstandenen Ableitungsbaum $\mathcal{T}'$ modelliert, nicht mit f identisch ist: $f \neq f'$.

Beweis Für den Beweis genügt die Angabe eines Beispiels. Seien $!Y \otimes !Y \xrightarrow{f} Z$ und $!X \xrightarrow{g} !Y$ die Interpretationen zweier Axiome einer beliebigen Linearen Logik. Dann gibt es einen Ableitungsbaum

$$\dfrac{\dfrac{}{\rhd\, !X \vdash !Y}\,g \qquad \dfrac{\dfrac{}{\rhd\, !Y, !Y \vdash Z}\,f}{\rhd\, !Y \vdash Z}\,\text{Contraction}}{\rhd\, !X \vdash !Z}\,\text{Cut}$$

zu dem der Morphismus

$$!X \xrightarrow{\quad g \quad} !Y \xrightarrow{\text{contr}_Y} !Y \otimes !Y \xrightarrow{\quad f \quad} Z$$

gehört. Wird Cut–Elimination auf den Beweisbaum angewendet, so erhält man

$$\dfrac{\dfrac{\dfrac{}{\rhd\, !X \vdash !Y}\,g \qquad \dfrac{\dfrac{}{\rhd\, !X \vdash !Y}\,g \qquad \dfrac{}{\rhd\, !Y, !Y \vdash Z}\,f}{\rhd\, !Y, !X \vdash !Z}\,\text{Cut}}{\rhd\, !X, !X \vdash !Z}\,\text{Cut}}{\rhd\, !X \vdash !Z}\,\text{Contraction}$$

und den Morphismus

$$!X \xrightarrow{\mathsf{contr}_X} !X \otimes !X \xrightarrow{g \otimes g} !Y \otimes !Y \xrightarrow{\quad f \quad} Z$$

Angenommen, f sei die Identität, dann ist ein Morphismus g zu finden, so daß das folgende Diagramm

$$\begin{tikzcd}
!X \arrow[r, "\mathsf{contr}_X"] \arrow[d, "g"'] & !X \otimes !X \arrow[d, "g \otimes g"] \\
!Y \arrow[r, "\mathsf{contr}_Y"'] & !Y \otimes !Y
\end{tikzcd} \tag{5.8}$$

nicht kommutiert.

Im folgenden wird ein Prozeß g als Transitionssystem konstruiert. Dafür wird angenommen, daß für beide Prozeßtypen X und Y wenigstens zwei Aktionen möglich sind:

$$\begin{aligned} \{a, b\} &\subseteq \Sigma_X \\ \{1, 2\} &\subseteq \Sigma_Y \end{aligned}$$

und die Sicherheitspezifikationen von X und Y wenigstens Folgen gleicher Aktionen der Länge zwei zulassen

$$\begin{aligned} \{\langle aa \rangle, \langle bb \rangle\} &\subseteq S_X \\ \{\langle 11 \rangle, \langle 22 \rangle\} &\subseteq S_Y \end{aligned}$$

Unter diesen Annahmen gilt zum Beispiel $\langle \{\!|\, a,b \,|\!\}\{\!|\, a,b \,|\!\}\rangle \in S_{!X}$. Sei g der Prozeß, der durch das folgende Transitionssystem definiert ist

$$\begin{tikzcd}
 & g \arrow[dl, "{\{\!|\, a,b \,|\!\}, \{\!|\, 1,2 \,|\!\}}"'] \arrow[d, "{\{\!|\, a \,|\!\}, \{\!|\, 1 \,|\!\}}"] \arrow[dr, "{\{\!|\, b \,|\!\}, \{\!|\, 2 \,|\!\}}"] & \\
\bullet \arrow[d, "{\{\!|\, a,b \,|\!\}, \{\!|\, 1,2 \,|\!\}}"] & \widehat{g} \arrow[d, "{\{\!|\, a \,|\!\}, \{\!|\, 1 \,|\!\}}"] & \overline{g} \arrow[d, "{\{\!|\, b \,|\!\}, \{\!|\, 2 \,|\!\}}"] \\
\bullet & \bullet & \bullet \\
\vdots & \vdots & \vdots
\end{tikzcd} \tag{5.9}$$

Dann enthält $\mathsf{contr}_Y \circ g$ einen Pfad

$$\mathsf{contr}_Y \circ g \xrightarrow{\left(\{\!|\, a,b \,|\!\}, (\{\!|\, 1 \,|\!\}, \{\!|\, 2 \,|\!\})\right)} \bullet \xrightarrow{\left(\{\!|\, a,b \,|\!\}, (\{\!|\, 2 \,|\!\}, \{\!|\, 1 \,|\!\})\right)} \bullet \ \cdots$$

Wesentlich ist hier, daß die einelementigen Multimengen die Plätze vertauschen.

Für die andere Kante $(g \otimes g) \circ \mathsf{contr}_X$ des Diagrammes 5.8 gilt: Ist die erste Aktion $\left(\{\!|\, a,b \,|\!\}, (\{\!|\, 1 \,|\!\}, \{\!|\, 2 \,|\!\})\right)$ dann befindet sich die erste Kopie von g im Zustand $\widehat{g}$ und die zweite im Zustand $\overline{g}$. Das Tensorprodukt $g \otimes g$ kann deshalb nach dieser ersten Aktion nur eine Aktion $\left((\{\!|a|\!\}, \{\!|b|\!\}), (\{\!|1|\!\}, \{\!|2|\!\})\right)$ machen. Daraus folgt $\mathsf{contr}_Y \circ g \neq (g \otimes g) \circ \mathsf{contr}_X$.
$\square$

Unmittelbar im Zusammenhang mit dem vorangegangenen Beweis steht das folgende Korollar.

Korollar 5.10 *Die Familie von Morphismen* $\{\mathsf{contr}_X\}$ *ist nicht natürlich.*

Beweis Sei h ein Morphismus, der entweder eine Folge von $(a,1)$–Aktionen oder eine Folge von $(b,2)$–Aktionen durchführt. Die möglichen Aktionen von $!h$ sind Paare von Multimengen gleicher Kardinalität. Innerhalb eines Pfades bleibt diese Kardinalität konstant. Der Prozeß $!h$ enthält den linken und den mittleren Pfad des Transitionssystemes 5.9 als die beiden einzigen Pfade, mit Aktionen der Kardinalität eins. Neben weiteren Pfaden enhält $!h$ auch den rechten Pfad mit der Kardinalität zwei aus obigem Transitionssystem.

Der Beweis von Satz 5.7 stützt sich genau auf diese beiden Eigenschaften: Der Exitenz eines speziellen Pfades mit Aktionen der Kardinalität zwei und dem Fehlen eines dritten Pfades der Kardinalität eins.

Damit ist auch $!h$ ein Morphismus, für den das Diagramm 5.8 nicht kommutiert. $\square$

Es ist auch möglich, zum Beweis des Satzes 5.7 ein Gegenbeispiel mit einem Objekt X mit einem einelementigen Alphabet zu konstruieren. Der Prozeß g ist dann allerdings nicht deterministisch.

6. Prozeßmengen

Für eine mehrsortige Prädikatenlogik kann man auf folgende Weise ein Modell erhalten. Zuerst wird aus den Mengen, die die Sorten interpretieren, und den Funktionen aus der SIG–Struktur eine Basiskategorie, zum Beispiel die klassifizierende Kategorie zusammengesetzt. Für jedes Objekt aus der Basis wird eine Kategorie aus der Potenzmenge und der Teilmengenrelation konstruiert. Der Funktor, der die Fasern auf die Basis abbildet, ist dann eine Fibration. Die totale Kategorie enthält genügend Struktur zur Interpretation von Quantoren.

Die gleiche Idee soll in diesem Kapitel auf die Interaktionskategorie $\mathcal{SP}roc$ angewendet werden. Als Basiskategorie soll $\mathcal{SP}roc$ verwendet werden. Die Faser über einem Objekt A besteht aus allen Mengen, die nur Prozesse des Types A enthalten. Der Einfachheit halber wird zuerst eine indizierte Kategorie konstruiert und dann die Grothendieckkonstruktion ausgeführt.

6.1 Die Fibration ProcSet $\longrightarrow \mathcal{SP}roc$

Sei $\Psi : \mathcal{SP}roc^{\mathsf{op}} \longrightarrow \mathbf{Cat}$ eine indizierte Kategorie, die wie folgt konstruiert ist:

Objekte Jedem Objekt A aus $\mathcal{SP}roc$ wird die Kategorie der Potenzmenge von $\mathsf{proc}(A)$ zugeordnet:

$$\Psi(A) \quad \stackrel{\text{def}}{=} \quad \mathbb{P}(\mathsf{proc}(A))$$

Die einzigen Morphismen, die $\Psi(A)$ enthält, sind die Inklusionen, das heißt, für $M, N \in |\Psi(\mathbb{A})|$ gilt:

$$\exists! f : M \longrightarrow N \quad \Longleftrightarrow \quad M \subseteq N$$

Morphismen Der Pullbackfunktor zu $A \xrightarrow{f} B$ bildet eine Menge $M \in |\Psi(\mathbb{B})|$ auf das Urbild bezüglich der Komposition mit f ab. Dabei werden Prozesse des Types A als Morphismen $I \longrightarrow A$ (vergleiche Ableitung 4.4) aufgefaßt:

$$f^*(M) \quad = \quad \Psi(f)(M) \quad \stackrel{\text{def}}{=} \quad \{p : A \mid f \circ p \in M\} \tag{6.1}$$

Mit dieser Definition folgt aus $M \subseteq N$ direkt $f^*(M) \subseteq f^*(N)$ und f^* ist ein Funktor.

Lemma 6.2 *Die eben vorgestellte indizierte Kategorie ist split, das heißt, Ψ ist ein kontravarianter Funktor.*

Beweis Es ist nachzuweisen, daß $\Psi(\mathsf{id}_A) = \mathsf{Id}_{\mathbb{P}(\mathsf{proc}(A))}$ und $\Psi(g) \circ \Psi(f) = \Psi(f \circ g)$ gilt. Zunächst ist

$$\begin{aligned} \Psi(\mathsf{id}_A)(M) &= \{f \mid \mathsf{id}_A \circ f \in M\} \\ &= M \end{aligned}$$

und

$$\Big(\Psi(g) \circ \Psi(f)\Big)\ (M) \quad = \quad \Psi(g)\ \Big(\{p \mid f \circ p \in M\}\Big)$$
$$= \quad \{q \mid f \circ g \circ q \in M\}$$
$$= \quad \Psi(f \circ g)\ (M)$$

□

Definition 6.3 (ProcSet) *Die Kategorie* **ProcSet** *ist die Kategorie, die aus der eben definierten indizierten Kategorie mittels der Grothendieckkonstruktion gewonnen werden kann.*

Wendet man die Grothendieckkonstruktion auf Ψ an, erhält man:

Objekte sind alle Mengen M, die nur Prozesse einen Typs enthalten:

$$|\mathbf{ProcSet}| \quad = \quad \{M \subseteq \mathsf{proc}(A) \mid A \in |\mathcal{SProc}|\}$$

Falls der Typ der in einer Menge enthaltenen Prozesse von Bedeutung ist, wird die Notation $M : A$ für $M \subseteq \mathsf{proc}(A)$ verwendet.

Morphismen Ein Morphismus $A \xrightarrow{f} B$ in $\mathcal{SProc}$ ist ein Morphismus $M \xrightarrow{f} N$ in **ProcSet**, wenn die Komposition von f mit jedem Prozeß aus M in N liegt:

$$\mathbf{ProcSet}(M : A, N : B) \quad = \quad \{f \in \mathcal{SProc}(A, B) \mid \forall p \in M \,.\, f \circ p \in N\}$$

Komposition Die Komposition wird auf die Komposition in $\mathcal{SProc}$ zurückgeführt. Liegen zwei Morphismen $M : B \xrightarrow{f} N : C$ und $L : A \xrightarrow{g} M : B$ vor, dann ist

$$f \circ_{\mathbf{ProcSet}} g = f \circ_{\mathcal{SProc}} g \tag{6.4}$$

Aus Satz 2.15 folgt, daß der Funktor $\mathbf{ProcSet} \xrightarrow{p} \mathcal{SProc}$ mit $p(M : A) = A$ und $p(f) = f$ eine Fibration mit Cleavage ist. Für einen gegebenen Morphismus $A \xrightarrow{f} B$ in $\mathcal{SProc}$ und eine Menge $M : B$ ist $\overline{f} = f : f^*(M) \longrightarrow M$ ein kartesischer Morphismus.

6.2 Die Struktur der Kategorie ProcSet

In diesem Abschnitt wird die Struktur der Kategorie **ProcSet** untersucht. Die Fasern von **ProcSet** sind Ordnungen und besitzen endliche Produkte und Summen. In den folgenden Teilabschnitten wird gezeigt, daß die Kategorie als ganzes eine reichere Struktur besitzt.

Tensor

Ein Bifunktor $\otimes : \mathbf{ProcSet} \times \mathbf{ProcSet} \longrightarrow \mathbf{ProcSet}$ wird durch:

$$(6.5)\qquad \begin{aligned} M \otimes N &\stackrel{\text{def}}{=} \{(p \otimes q) \circ \mathsf{unitl}_I^{-1} \mid p \in M \text{ und } q \in N\} \\ f \otimes g &\stackrel{\text{def}}{=} f \otimes g \end{aligned}$$

definiert, wobei auf der rechten Seite der Gleichungen immer das Tensorprodukt und die Komposition der Kategorie $\mathcal{SProc}$ gemeint sind.

Das Tensorprodukt von $M : A$ und $N : B$ ist nach dieser Definition ein Objekt in der Faser über $A \otimes B$. In den Fasern von **ProcSet** können deshalb die weder die Regeln zur Bildung wohldefinierter Formeln noch die logischen Formeln für $\otimes$ interpretiert werden. Dieses Problem betrifft auch die noch weiter unten untersuchten Produkte, Coprodukte und die Internalisierung. Es wird im folgenden Abschnitt 6.3 noch einmal aufgegriffen.

Lemma 6.6 *Es gilt*

1. *Die Gleichungen 6.5 definieren einen Bifunktor*
$$\mathbf{ProcSet} \times \mathbf{ProcSet} \longrightarrow \mathbf{ProcSet}$$

2. *Die Kategorie* **ProcSet** *ist eine symmetrische monoidale Kategorie mit*
$$\begin{aligned} I &\stackrel{\text{def}}{=} \{\mathsf{id}_I\} \\ \mathsf{unitl}_{M:A} &\stackrel{\text{def}}{=} \mathsf{unitl}_A \\ \mathsf{unitr}_{M:A} &\stackrel{\text{def}}{=} \mathsf{unitr}_A \\ \mathsf{assoc}_{M:A,N:B,O:C} &\stackrel{\text{def}}{=} \mathsf{assoc}_{A,B,C} \\ \mathsf{symm}_{M:A,N:B} &\stackrel{\text{def}}{=} \mathsf{symm}_{A,B} \end{aligned}$$

Beweis

6.6.1 Für die Objekte von **ProcSet** gilt für $p \in M : A$ und $q \in N : B$, daß
$$(p \otimes q) \circ \mathsf{unitl}_I^{-1} : I \longrightarrow A \otimes B$$
ein Morphismus in $\mathcal{SProc}$ ist. Damit ist $p \otimes q$ ein Prozeß vom Typ $A \otimes B$.

Für zwei Morphismen $(M : A) \xrightarrow{f} (N : B)$ und $(L : C) \xrightarrow{g} (Q : D)$ gilt $(f \otimes g) \circ (p \otimes q) \circ \mathsf{unitl}_I^{-1} = ((f \circ p) \otimes (g \circ q)) \circ \mathsf{unitl}_I^{-1}$. Damit folgt aus $p \in M$ und $q \in L$, daß $(f \otimes g) \circ (p \otimes q) \circ \mathsf{unitl}_I^{-1} \in N \otimes Q$ gilt.

Die Morphismen von **ProcSet** sind Morphismen aus $\mathcal{SProc}$. Eigenschaften, die bestimmte Morphismen in $\mathcal{SProc}$ haben, gelten dadurch auch in **ProcSet**, vorausgesetzt, die Morphismen existieren in **ProcSet** zwischen den betrachteten Morphismen. Die Natürlichkeit des Bifunktors folgt zum Beispiel direkt aus den Eigenschaften des Bifunktors $\otimes$ über $\mathcal{SProc}$ und aus der Definition der Komposition in **ProcSet**.

6.6.2 Zu zeigen ist nur, daß die Isomorphismen im Lemma 6.6.2 wohldefiniert sind, das heißt, daß sie auch in **ProcSet** Morphismen zwischen den entsprechenden Objekten sind.

Angenommen, $\mathsf{id}_I \otimes p$ ist eine beliebiger Prozeß aus $I \otimes M$, dann ist $\mathsf{unitl}_A \circ (\mathsf{id}_I \otimes p) \circ \mathsf{unitl}_A^{-1} = p$ und $p \in M$.

Die gleiche Argumentation gilt für die anderen natürlichen Isomorphismen: Sie sind Familien von Morphismen in **ProcSet**, weil sie in $\mathcal{SP}roc$ natürlich sind.

Abgeschlossenheit

Die Internalisierung $(M : A) \multimap -$ wird für jedes $M : A \in |\mathbf{ProcSet}|$ wie folgt definiert:

$$
\begin{aligned}
(M : A) \multimap (N : B) &\stackrel{\text{def}}{=} \{ f^\sharp \mid f : M \longrightarrow N \} \\
(M : A) \multimap g &\stackrel{\text{def}}{=} A \multimap g
\end{aligned} \tag{6.7}
$$

Auch hier bezeichnet $\multimap$ auf der rechten Seite den entsprechenden Funktor auf $\mathcal{SP}roc$. Der Morphismus $f^\sharp$ ist das Element $I \longrightarrow A \multimap B$, welches zu einem Morphismus $A \xrightarrow{f} B$ gehört (vergleiche die Ableitung 4.4 im Abschnitt 4.1).

Lemma 6.8 (Internalisierung) *Es gilt*

1. *Die Gleichungen 6.7 definieren einen Funktor* **ProcSet** $\longrightarrow$ **ProcSet**.
2. *Der Funktor* $M \multimap -$ *ist rechtsadjungiert zu* $- \otimes M$ *und* **ProcSet** *eine abgeschlossene monoidale Kategorie.*

Beweis

6.8.2 Der Beweis wird mit der „Standarddefinition“ für Adjunktionen geführt. Als Voraussetzung ist nur notwendig, daß $-\otimes M$ ein Funktor ist. Das wurde bereits in 6.6.1 bewiesen. Seien $M : A$, $N : B$ und $Q : C$ beliebige Objekte und $f : Q \otimes M \longrightarrow N$ ein beliebiger Morphismus aus **ProcSet**. Die Situation wird durch folgendes Diagramm illustriert:

$$
\begin{tikzcd}
Q \arrow[d, dashed, "f^\vee"] \\
M \multimap N
\end{tikzcd}
\qquad
\begin{tikzcd}
\cdot \arrow[r, shift left=2, "- \otimes M"] \arrow[r, phantom, "\perp"] & \cdot \arrow[l, shift left=2, "M \multimap -"]
\end{tikzcd}
\qquad
\begin{tikzcd}
Q \otimes M \arrow[d, "f^\vee \otimes \mathsf{id}_M"'] \arrow[dr, "f"] & \\
(M \multimap N) \otimes M \arrow[r, "\epsilon_N"'] & N
\end{tikzcd}
$$

- Aus der Definition 6.7 ist ersichtlich, daß $M \multimap N \subseteq \mathsf{proc}(A \multimap B)$ ist.

- Für die Counit gilt $\epsilon_N = \mathsf{eval}_{A,B}$. Es verbleibt zu zeigen, daß $\mathsf{eval}_{A,B}$ ein Morphismus $(M \multimap N) \otimes M \longrightarrow N$ ist. Sei $h^\sharp \in M \multimap N$ und $q \in M$. Dann gilt:

$$\begin{aligned}
\mathsf{eval}_{A,B} \circ (h^\sharp \otimes q) \circ \mathsf{unitl}_I^{-1} &= \mathsf{eval}_{A,B} \circ \left[(h^\sharp \circ \mathsf{id}_I) \otimes (\mathsf{id}_A \circ q)\right] \circ \mathsf{unitl}_I^{-1} \\
&= \mathsf{eval}_{A,B} \circ (h^\sharp \otimes \mathsf{id}_A) \circ (\mathsf{id}_I \otimes q) \circ \mathsf{unitl}_I^{-1} \\
&= h \circ \mathsf{unitl}_A \circ (\mathsf{id}_I \otimes q) \circ \mathsf{unitl}_I^{-1} \\
&= h \circ \mathsf{unitl}_A \circ \mathsf{unitl}_A^{-1} \circ q \\
&= h \circ q
\end{aligned}$$

Mit diesem Zwischenergebnis läßt sich nun schlußfolgern:

$$\begin{aligned}
h^\sharp \in M \multimap N &\Longrightarrow \forall r \in M \,.\, h \circ r \in N \\
&\Longrightarrow \mathsf{eval}_{A,B} \circ (h^\sharp \otimes q) \circ \mathsf{unitl}_I^{-1} \in N \\
&\Longrightarrow (M \multimap N) \otimes M \xrightarrow{\mathsf{eval}_{A,B}} N
\end{aligned}$$

- Es wird $f^\vee$ konstruiert und nachgewiesen, daß das rechte Dreieck kommutiert. Für $Q \otimes M \xrightarrow{f} N$ gilt

$$\forall q \in Q \,.\, \forall r \in M \,.\, f \circ (q \otimes r) \circ \mathsf{unitl}_I^{-1} \in N \tag{6.9}$$

Die Transposition der Adjunktion über **ProcSet** wird mit Hilfe der Transposition über $\mathcal{SP}roc$ definiert. Sei $q \in Q$ und $r \in M$, dann gilt:

$$\begin{aligned}
(f^\vee \circ q)^\flat \circ r &= (f^\vee \circ q)^\wedge \circ \mathsf{unitl}_A^{-1} \circ r \\
&= f^{\vee\wedge} \circ (q \otimes \mathsf{id}_A) \circ \mathsf{unitl}_A^{-1} \circ r \\
&= f \circ (q \otimes \mathsf{id}_A) \circ (\mathsf{id}_I \otimes r) \circ \mathsf{unitl}_I^{-1} \\
&= f \circ (q \otimes r) \circ \mathsf{unitl}_I^{-1}
\end{aligned}$$

Damit folgt aus 6.9:

$$\begin{aligned}
(f^\vee \circ q)^\flat \circ r \in N &\Longrightarrow f^\vee \circ q \in M \multimap N \\
&\Longrightarrow f^\vee : Q \longrightarrow M \multimap N
\end{aligned}$$

Die Gültigkeit von $\epsilon_N \circ (f^\vee \otimes M) = f$ in **ProcSet** folgt aus der Gleichheit $\mathsf{eval}_{A,B} \circ (f^\vee \otimes \mathsf{id}_A) = f$ in $\mathcal{SP}roc$.

Die Eineindeutigkeit der Transposition ergibt sich ebenfalls aus der Adjunktion über $\mathcal{SP}roc$. Damit ist der Beweis für $- \otimes M \dashv M \multimap -$ vollständig.

6.8.1 Folgt direkt aus 6.8.2. □

Korollar 6.10 *Die Kategorie* **ProcSet** *ist eine autonome Kategorie.*

Beweis Nach den Lemmata 6.6 und 6.8 ist nur noch zu zeigen, daß der Funktor $M \multimap -$ sich zu einem Bifunktor erweitern läßt. Das geschieht mit

$$f \multimap (N : A) \quad \stackrel{\text{def}}{=} \quad f \multimap A$$

□

Die Negation und das dualisierende Element

Die nächste zu untersuchende Frage ist, ob **ProcSet** ein dualisierendes Element enthält. Zur Erinnerung: Ein Objekt X heißt dualisierendes Element, wenn der Funktor

$$- \multimap X \quad : \quad \mathbf{ProcSet}^{\mathsf{op}} \longrightarrow \mathbf{ProcSet}$$

eine Äquivalenz von Kategorien ist (vergleiche Abschnitt 4.1).

Vergleicht man die Definition der Internalisierung mit der Definition des Tensors, bzw. der der Komposition in **ProcSet** kann man folgendes feststellen: Bis auf die Internalisierung ist die gesamte bisher diskutierte Struktur mittels Komposition und Morphismen aus $\mathcal{SP}roc$ definiert. Die Internalisierung jedoch nimmt direkt auf Eigenschaften von **ProcSet** bezug. Vor diesem Hintergrund überrascht das folgende negative Resultat nicht.

Satz 6.11 *Die Kategorie* **ProcSet** *besitzt* kein *dualisierendes Element.*

Beweis Der Beweis wird indirekt geführt. Angenommen, $Q : A$ sei ein Objekt aus **ProcSet** und der Funktor $F = (- \multimap Q) : \mathbf{ProcSet}^{\mathsf{op}} \longrightarrow \mathbf{ProcSet}$ ist eine Äquivalenz von Kategorien. Dann gibt es einen zweiten Funktor $G : \mathbf{ProcSet} \longrightarrow \mathbf{ProcSet}^{\mathsf{op}}$, so daß zwischen der Verkettung $G \circ F$ und dem Identitätsfunktor Id ein natürlicher Isomorphismus existiert (vergleiche die Definition von Äquivalenzen von Kategorien, zum Beispiel in [ML71]).

Für den Rest des Beweises wird ein beliebiges Objekt B aus $\mathcal{SP}roc$ mit nichtleerer Sicherheitsspezifikation und ein Prozeß $p : B$ mit $p \neq \mathsf{nil}$ benötigt. In $\mathcal{SP}roc$ existieren immer ausreichend Objekte mit dieser Eigenschaft.

Angenommen, es gelte $(\mathsf{nil} : A) \notin Q$. Dann ist $F(\{\mathsf{nil}\}) = \{\mathsf{nil}\} \multimap Q = \emptyset$ weil für alle Morphismen f aus $\mathcal{SP}roc$ $f \circ \mathsf{nil} = \mathsf{nil}$ gilt. Desgleichen gilt $F(\{\mathsf{nil}, p\}) = \emptyset$ und damit $GF(\{\mathsf{nil}\}) = GF(\{\mathsf{nil}, p\})$. Das ist ein Widerspruch, weil $\{\mathsf{nil}\}$ und $\{\mathsf{nil}, p\}$ nicht isomorph sein können.

Folglich muß Q den Prozeß $\mathsf{nil} : A$ enthalten. Dann folgt unmittelbar

$$\begin{aligned} \{\mathsf{nil}\} \multimap Q &= \{\mathsf{nil}\} \\ \{\mathsf{nil}, p\} \multimap Q &= \{\mathsf{nil}\} \end{aligned}$$

Damit ist die ursprüngliche Annahme, Q sei ein dualisierendes Element, auf einen Widerspruch geführt. $\square$

In kategorientheoretischen Modellen für Lineare Logik stehen dualisierende Elemente in der Kategorie in engem Zusammenhang mit der Negation. In [MOM90] wird gezeigt, daß ohne ein dualisierendes Element keine Involution $(-)^{\perp}$ definiert werden kann, die die Negation der Linearen Logik interpretiert. Die Argumentation aus [MOM90] trifft auch für **ProcSet** zu. Diese Kategorie ist deshalb als Modell der Linearen Logik nur von sehr begrenztem Interesse.

Cotensor

Mit der gleichen Methode wie in 6.5 ist es möglich einen zweiten Tensor $\parr$ zu definieren. Es ergäbe sich $M \parr N = M \otimes N$ weil in $\mathcal{SProc}$ für beliebige Prozesse $p \parr q = p \otimes q$ gilt. Ohne Struktur für eine Interpretation der Negation ist der Cotensor allerdings kaum von Interesse.

Produkte und Coprodukte

ProcSet besitzt endliche Produkte und binäre Coprodukte. Für $M : A$ und $N : B$ ist das Produkt

$$M \mathbin{\&} N \quad = \quad \{\langle p, q\rangle \mid p \in M, q \in N\}$$

Damit gilt $M \mathbin{\&} N : A \mathbin{\&} B$. Wenn $A \mathbin{\&} B \xrightarrow{\pi_1} A$ und $A \mathbin{\&} B \xrightarrow{\pi_2} B$ die Projektionen sind, dann gilt

$$\begin{aligned} \pi_1 \circ \langle p, q\rangle &= p \\ \pi_2 \circ \langle p, q\rangle &= q \end{aligned}$$

für alle $\langle p, q\rangle \in M \mathbin{\&} N$ und die Projektionen in **ProcSet** sind die Projektionen aus $\mathcal{SProc}$.

Für die Paarbildung gilt das gleiche. Für zwei Morphismen $(Q : C) \xrightarrow{f} (M : A)$ und $(Q : C) \xrightarrow{g} (N : B)$ muß nur gezeigt werden, daß $\langle f, g\rangle$ ein Morphismus zwischen Q und $M \mathbin{\&} N$ ist. Mit $r \in Q$ gilt:

$$\langle f, g\rangle \circ r \quad = \quad \langle f \circ r, g \circ r\rangle$$

und weil f und g Morphismen in **ProcSet** sind, folgt $\langle f \circ r, g \circ r\rangle \in M \mathbin{\&} N$. Die Eindeutigkeit der Konstruktion $\langle -, -\rangle$ folgt aus der gleichen Eigenschaft in $\mathcal{SProc}$.

Das terminale Objekt ist die Menge, die nur aus dem Prozeß $\mathsf{nil} : \mathbf{0}$ besteht.

Für $M : A$ und $M : B$ ist das Coprodukt

$$M \oplus N \quad = \quad \{\mathsf{inl} \circ p \mid p \in M\} \;\cup\; \{\mathsf{inr} \circ q \mid q \in N\}$$

Hier sind $A \xrightarrow{\mathsf{inl}} A \oplus B$ und $B \xrightarrow{\mathsf{inr}} A \oplus B$ die Injektionen aus $\mathcal{SProc}$. Damit ist $M \oplus N : A \oplus B$. Genau wie im Fall der Produkte werden die Injektionen und die Kombination von Morphismen auf die Eigenschaften von $\mathcal{SProc}$ zurückgeführt. Diese Morphismen respektieren die Prozeßmengen. Die Argumentation folgt dem gleichen Muster wie für Produkte und wird hier nicht wiederholt.

Die Eindeutigkeit der Kombination folgt aus den Eigenschaften von $A \oplus B$ in $\mathcal{SProc}$.

Es gibt kein initiales Element. Von der in Betracht kommenden leeren Menge über $\mathbf{0}$ gibt es mehr als einen Morphismus zu fast allen Objekten in **ProcSet**.

Im Unterschied zu $\mathcal{SProc}$ sind in **ProcSet** das Produkt und das Coprodukt voneinander verschieden.

Modalitäten

In diesem Teilabschnitt wird ein spezieller Morphismus $I \xrightarrow{\kappa} !I$ benötigt. Er wird, wie andere Morphismen mit einer Codomäne $!A$ im Abschnitt 5.3 als abzählbare Summe konstruiert. Sei κ_n der Prozeß, der die Marke $* \in \Sigma_I$ genau n mal kopiert:

$$\kappa_n \xrightarrow{\left(*,\{\!|\, \overbrace{*,\cdots,*}^{n \text{ mal}} \,|\!\}\right)} \kappa_n$$

Es gilt $I \xrightarrow{\kappa_n} \otimes_s^n I$. Dann ist

$$\kappa \overset{\text{def}}{=} \bigoplus_{n \in \mathbb{N}} \kappa_n$$

Ein Prozeß, der mit κ verbunden wird, kann bei der ersten Synchronisation die Anzahl der Kopien der Marke $*$ frei wählen. Danach ist die Anzahl für die gesamte Lebensdauer von κ festgelegt. Es werden nur noch Aktionen mit der einmal gewählten Anzahl von Marken angeboten.

Die Eigenschaft eines Prozesses, nach einer ersten Aktion nur noch Aktionen mit Multimengen der gleichen Kardinalität anzubieten, spielt für die Definition von Modalitäten eine wichtige Rolle. Prozesse über einem Alphabet von Multimengen, die diese Eigenschaft haben, werden im folgenden **kardinalitätserhaltend** genannt.

Damit im Zusammenhang steht noch eine zweite Vorüberlegung: Angenommen, x ist ein Morphismus $I \longrightarrow !A$ für ein beliebiges Objekt A aus $\mathcal{SProc}$. Ist x kardinalitätserhaltend, dann kann es in $\overline{x} \circ \kappa$ faktorisiert werden. Die Intuition ist, daß $\overline{x}$ die Summe aller der Prozesse ist, die das Verhalten von x für Aktionen einer festen Kardinalität simulieren.

Zur Interpretation der modalen Operatoren muß es auf **ProcSet** einen Comonad ! geben und jedes Objekt $!M$ muß ein kommutativer Comonoid sein (vergleiche die Abschnitte 2.2 und 5.3). Eine naheliegende Defintion des Comonads für Objekte wäre

$$!(M : A) \overset{\text{def}}{=} \{!p \circ \kappa \mid p \in M\} \tag{6.12}$$

$$\begin{tikzcd}
 & & I \otimes I & \\
I \arrow[urr, "\mathsf{unitl}_I^{-1}"] \arrow[r, "\kappa"] \arrow[dr, "x"'] \arrow[r, phantom, shift left=6ex, "(a)"] & !I \arrow[r, "\mathsf{contr}_I"] \arrow[d, "\overline{x}"] \arrow[dl, phantom, "(c)"] \arrow[dr, phantom, "(d)"] & !I \otimes !I \arrow[d, "\overline{x} \otimes \overline{x}"] \arrow[r, phantom, "(b)"] & I \arrow[ul, "\mathsf{unitl}_I^{-1}"'] \arrow[dl, "(x \otimes x) \circ \mathsf{unitl}_I^{-1}"] \\
 & !A \arrow[r, "\mathsf{contr}_A"'] & !A \otimes !A &
\end{tikzcd}$$

(The vertical arrow $I \otimes I \to\ !I \otimes !I$ is labelled $\kappa \otimes \kappa$.)

Abbildung 9: Zur Definition eines Comonoiden auf $!M$

Mit dieser Definition ist $!M$ ein Objekt in der Faser über $!A$.

Um mit Satz 2.3 zu beweisen, daß mit Gleichung 6.12 ein Comonad definiert wird, braucht nur noch eine Operation $(-)^!$ definiert zu werden. Die im Punkt 2 dieses Satzes geforderte Familie von Morphismen ist durch die Familie der aus $\mathcal{SP}roc$ gegeben.

Ich habe eine Idee für eine Definition von $(-)^!$ und eine Beweisskizze für die Bedingungen 2.4 aus Satz 2.3 entwickelt. Im folgenden werde ich jedoch zeigen, daß die Struktur des Comonoiden $!A$ in $\mathcal{SP}roc$ sich nicht zu einer Struktur eines Comonoiden $!M$ in **ProcSet** heben läßt. Wegen dieses negativen Resultates habe ich den genannten Beweis nicht weiter ausgearbeitet.

Im folgenden wird gezeigt, daß im allgemeinen contr_A kein Morphismus $!M \longrightarrow (!M \otimes !M)$ in **ProcSet** ist.

Angenommen, es gilt $x \in\ !M$. Dann muß $\mathsf{contr}_A \circ x \in (\ !M \otimes !M\ )$ gelten. Da im allgemeinen nicht davon ausgegangen werden kann, daß $!M$ mehr als einen Prozeß enthält, muß contr_A die folgende Gleichung erfüllen:

$$\mathsf{contr}_A \circ x \quad \stackrel{?}{=} \quad (x \otimes x) \circ \mathsf{unitl}_I^{-1} \tag{6.13}$$

Das Diagramm in Abbildung 9 veranschaulicht die Situation. Aus der Definition des Prozesses κ folgt, daß Diagramm (a) kommutiert.

Mir der Definition des Funktors ! über $\mathcal{SP}roc$ (vergleiche Gleichung 5.5) ergibt sich, daß alle Prozesse des Typs $!A$ kardinalitätserhaltend sind. Deshalb kommutieren (b) und (c).

Es verbleibt das Teildiagramm (d). Aus Satz 5.7 und dessen Beweis folgt direkt, daß im allgemeinen Fall (*d*) *nicht* kommutiert.

In diesem Abschnitt bleiben zwei Fragen offen. Erstens, ob es möglich ist, einen kommutativen Comonoid $!M$ in **ProcSet** zu konstruieren, ohne den Morphismus contr zu benutzen. Falls das nicht möglich ist, ergibt sich zweitens die Frage, ob ein Comonad ! auf andere Art definiert werden kann. Beide Probleme konnte ich bislang nicht lösen. Ich vermute, daß in beiden Fällen die Antwort negativ ist, weil sich eine natürliche Struktur auf **ProcSet** nur mit Hilfe der entsprechenden Morphismen aus $\mathcal{SP}roc$ definieren läßt.

Adjunktionen zwischen den Fasern

In diesem Teilabschnitt wird untersucht, ob in der Kategorie **ProcSet** die Quantoren der Linearen Logik modelliert werden können. Robert A. Seely definiert die Funktoren, die die Quantoren modellieren, mit Hilfe der Projektionen des kartesischen Produktes (siehe Abschnitt 4.3). Sowohl in der Definition als auch im Beweis von Satz 4.10 werden die universellen Eigenschaften des Produktes in der Basiskategorie jedoch nicht benutzt. Das ermutigt zum Suchen nach einer schwächeren Struktur in der Basiskategorie, zu deren Pullbackfunktoren es Adjunktionen mit geeigneten Eigenschaften gibt.

Zur Untermauerung dieser Intuition wird in der Folge angenommen, die Verkettung von Kontexten in der Basiskategorie sei mit Hilfe des Tensorproduktes $\otimes$ definiert. Es werden Projektionsmorphismen $A \otimes B \xrightarrow{\varrho} A$ für Objekte definiert, die Struktur der Pullbackfunktoren ϱ^* untersucht und die beiden Adjunktionen $\exists \dashv \varrho^* \dashv \forall$ beschrieben.

In der Kategorie $\mathcal{SP}roc$ gibt es zwischen den Objekten $A \otimes B$ und A bzw. B Morphismen ϱ und ϱ', die sich durch Umbenennung von Transitionen aus der Identität des Objektes $A \otimes B$ gewinnen lassen.

$$(6.14)\quad \frac{\mathsf{id}_{A\otimes B} \xrightarrow{((a,b),(a,b))} \mathsf{id}_{A'\otimes B'}}{\varrho_{A,B} \xrightarrow{((a,b),a)} \varrho_{A',B'}} \qquad\qquad \frac{\mathsf{id}_{A\otimes B} \xrightarrow{((a,b),(a,b))} \mathsf{id}_{A'\otimes B'}}{\varrho'_{A,B} \xrightarrow{((a,b),b)} \varrho'_{A',B'}}\quad (6.15)$$

Es gilt zum Beispiel

$$\varrho_{A,B\otimes C} = \varrho_{A,B} \circ \varrho_{A\otimes B,C} \circ \mathsf{assoc}_{A,B,C}$$

Beide Morphismen sind jedoch keine Projektionen. Im allgemeinen gilt $\varrho \circ (p \otimes q) \neq p \circ \varrho$.

Satz 6.16 (Adjunktionen zu ϱ^*) *Der zum Morphismus ϱ gehörende Pullbackfunktor $\varrho^* : \mathbf{ProcSet}_A \longrightarrow \mathbf{ProcSet}_{A\otimes B}$ besitzt links- und rechtsadjugierte Funktoren $\exists$ und $\forall : \mathbf{ProcSet}_{A\otimes B} \longrightarrow \mathbf{ProcSet}_A$. Diese sind durch*

$$\begin{aligned} \exists(N) &= \{\varrho \circ p \mid p \in N\} \\ \forall(N) &= \Big\{q : A \mid \forall p : A \otimes B \,.\, \big(\varrho \circ p = q \Longrightarrow p \in N\big)\Big\} \end{aligned}$$

gegeben.

Zur Erinnerung: Es gilt nach der Definition der Pullbackfunktoren (siehe Gleichung 6.1)

$$\varrho^*(N) \quad = \quad \{p : A \otimes B \mid \varrho \circ p \in N\}$$

Beweis

$\varrho^* \dashv \forall$ Der Beweis erfolgt nach der gleichen Methode wie der Beweis von Lemma 6.8.2 und bezieht sich hier auf die folgende Situation:

$$\begin{tikzcd}
N \arrow[dd, dashed, "f^\vee"] & & & \varrho^*(N) \arrow[dd, "\varrho^*(f^\vee)"'] \arrow[ddrr, "f"] & & \\
& {} \arrow[r, shift left=2, "\varrho^*"] \arrow[r, phantom, "\perp"] & {} \arrow[l, shift left=2, "\forall"] & & & \\
\forall M & & & \varrho^*(\forall M) \arrow[rr, "\epsilon_N"'] & & M
\end{tikzcd} \tag{6.17}$$

Faser über A Faser über $A \otimes B$

Die Objekte $N : A$ und $M : A \otimes B$ seien beliebig aus **ProcSet** gewählt. Da die dargestellten Morphismen jeweils zu einer Faser gehören, werden sie immer durch Identitäten repräsentiert. Zum Beispiel existiert ein $f^\vee : N \longrightarrow \forall M$ genau dann, wenn $N \subseteq \forall M$ gilt.

- ϱ^* ist ein Funktor.
- Aus den Gleichungen in Satz 6.16 folgt offensichtlich, daß $\forall M \subseteq \mathsf{proc}(A)$ für $M \subseteq \mathsf{proc}(A \otimes B)$ ist.
- Die Counit ϵ ist ein Morphismus in einer Faser und kann deshalb nur $\mathsf{id}_{A \otimes B}$ sein.

$$\begin{array}{rcl}
p \in \varrho^*(\forall M) & \Longrightarrow & \varrho \circ p \in \forall M \\
& \Longrightarrow & \forall p' : A \otimes B \,.\, (\varrho \circ p' = \varrho \circ p \Longrightarrow p' \in M) \\
& \Longrightarrow & p \in M \\
& \Longrightarrow & \mathsf{id}_{A \otimes B} : \varrho^*(\forall M) \longrightarrow M
\end{array}$$

- Angenommen, es existiert ein Morphismus f in Diagramm 6.17, das heißt, $\varrho^*(N) \subseteq M$. Zu zeigen ist, daß $N \subseteq \forall M$ gilt:

$$\begin{array}{rcl}
q \in N & \Longrightarrow & \forall p : A \otimes B \,.\, (\varrho \circ p = q \Longrightarrow p \in \varrho^*(N)) \\
& \Longrightarrow & \forall p : A \otimes B \,.\, (\varrho \circ p = q \Longrightarrow p \in M) \\
& \Longrightarrow & q \in \forall M \\
& \Longrightarrow & N \subseteq \forall M
\end{array}$$

Das Dreieck in Diagramm 6.17 über $A \otimes B$ kommutiert und $f^\vee$ ist der einzige Morphismus, der dieses Diagramm schließt, weil die Fasern in **ProcSet** Ordnungen sind, das heißt, höchstens ein Morphismus zwischen den Objekten existiert.

Damit ist $\varrho^* \dashv \forall$ bewiesen. Ohne Beweis wird vermerkt, daß die Unit dieser Adjunktion die Identität in **ProcSet** ist. Es gilt $M = \forall\varrho^*(M)$.

$\exists \dashv \varrho^*$ Der Beweis wird genau wie der vorangehende geführt. Die Situation wird im folgenden Diagramm dargestellt:

$$\begin{tikzcd} M \arrow[r, "\eta"] \arrow[dr, "f"'] & \varrho^*\exists M \arrow[d, "\varrho^*(f)"] & & \exists M \arrow[d, dashed, "f^\wedge"'] \\ & \varrho^*(N) & & N \end{tikzcd} \qquad \begin{tikzcd} \cdot \arrow[r, shift left, "\exists"] \arrow[r, phantom, "\perp"] & \cdot \arrow[l, shift left, "\varrho^*"] \end{tikzcd} \tag{6.18}$$

Faser über $A \otimes B$ Faser über A

- Aus Satz 6.16 ist auch hier ersichtlich, daß $\exists(M) \subseteq \mathsf{proc}(A)$ für jedes $N : A \otimes B$ ist.
- η ist ein Morphismus $M \longrightarrow \varrho^*\exists M$ weil

$$\begin{aligned} p \in M &\Longrightarrow \varrho \circ p \in \exists M \\ &\Longrightarrow \forall p' : A \otimes B \,.\, (\varrho \circ p' = \varrho \circ p \Longrightarrow p' \in \varrho^*\exists M) \\ &\Longrightarrow p \in \varrho^*\exists M \\ &\Longrightarrow M \subseteq \varrho^*\exists M \end{aligned}$$

- Für die Transposition sei angenommen, daß $M \subseteq \varrho^* N$. Dann gilt:

$$\begin{aligned} q \in \exists M &\Longrightarrow p \in M \,.\, \varrho \circ p = q \\ &\Longrightarrow \exists p \in \varrho^*(N) \,.\, \varrho \circ p = q \\ &\Longrightarrow \exists p \in \varrho^*(N) \,.\, \varrho \circ p = q \\ &\Longrightarrow q \in N \end{aligned}$$

Für die Eineindeutigkeit und die Kommutativität des Diagramms 6.18 gelten die gleichen Bemerkungen wie im vorhergehenden Beweis. □

In der Kategorie **ProcSet** gibt es zwischen den Fasern auch Adjunktionen zu den Pullbackfunktoren π_1^*, wobei π_1 die erste Projektion des kartesischen Produktes in $\mathcal{SP}roc$ ist:

$$\begin{aligned} \exists^\pi(N) &= \{\pi_1 \circ p \mid p \in N\} \\ \forall^\pi(N) &= \Big\{q : A \mid \forall p : A \otimes B \,.\, (\pi_1 \circ p = q \Longrightarrow p \in N)\Big\} \end{aligned}$$

Wie man sieht, unterscheidet sich diese Definition von Satz 6.16 nur durch die Ersetzung von ϱ durch π_1. Das gleiche gilt für einen Beweis von $\exists^\pi \dashv \pi_1^* \dashv \forall^\pi$. Aus diesen Gründen wird auf einer detailliertere Formalisierung dieses Sachverhaltes verzichtet.

6.3 Diskussion

In diesem Abschnitt wurde ein aus der Literatur bekannter Ansatz, Modelle für Prädikatenlogiken zu gewinnen, auf die Interaktionskategorie $\mathcal{SProc}$ angewendet. Die entstandene Struktur, die Kategorie **ProcSet**, ist als Modell für mehrsortige Lineare Logik jedoch nicht geeignet. Es können nur die Konnektoren $\otimes, \multimap, \&, \oplus$ und die Quantoren interpretiert werden.

Im Rahmen dieser Arbeit gelang es nicht, endgültig zu klären, ob **ProcSet** den modalen Operator ! modelliert.

Wesentlichstes Manko ist von **ProcSet** ist, daß diese Kategorie kein dualisierendes Element enthält. Ohne dualisierendes Element ist eine Interpretation der Negation und damit von Linearer Logik nicht möglich (siehe [MOM90] für den Zusammenhang zwischen der Negation der Linearen Logik und dualisierenden Elementen).

Da für das Problem des dualisierenden Elementes ein Beweis vorgelegt werden konnte, halte ich die noch offen Fragen zur Definition eines Comonads für wenig interessant.

Die Konstruktion dieses Kapitels unterscheidet sich noch in einem zweiten Punkt wesentlich, von der von Robert A. Seely in [See89] vorgeschlagenen und im Abschnitt 4.3 beschriebenen Struktur einer indizierten linearen Kategorie: Die Fasern von **ProcSet** sind Ordnungen und nicht, wie von Robert A. Seely gefordert, lineare Kategorien. Dementsprechend läßt sich keiner der Konnektoren $\otimes, \multimap, \&, \oplus$ innerhalb einer Faser interpretieren. Anstelle der Regel für zweistellige Konnektoren, wie sie in Abbildung 11 angegeben ist, modelliert **ProcSet** die Regel

$$\text{Für } \ominus \in \{\otimes, \multimap, \&, \oplus\} \qquad \frac{\Upsilon \;\triangleright\; A : \mathsf{Prop} \qquad \Xi \;\triangleright\; B : \mathsf{Prop}}{\Upsilon, \Xi \;\triangleright\; A \ominus B : \mathsf{Prop}} \text{ Konnektoren}$$

mit der Nebenbedingung, daß die Variablenmengen aus Υ und Ξ disjunkt sind.

Nicht zwingend erforderlich, aber wünschenswert wäre eine Anpassung der logischen Regeln $\otimes$ R, $⅋$ L, $\multimap$ L, $\&$ R und $\oplus$ L.

Welche Auswirkungen eine derartige Änderung des Regelsystems auf die mehrsortige Lineare Logik hat, habe ich nicht untersucht. Ich vermute jedoch, daß der Korrektheitsbeweis aus Abschnitt 4.4 nach einer entsprechenden Anpassung der Interpretation der Formeln $[\![-]\!]_{\mathfrak{V}}$ mit kleineren Änderungen gültig bleibt.

7. Diskussion

Die vorliegende Diplomarbeit ist auf zwei Probleme konzentriert. Das erste ist die Definition einer mehrsortigen Linearen Logik, die Konstruktion eines kategorientheoretischen Modells dafür und dessen Korrektheit. Die zweite behandelte Frage ist, inwieweit die Prozeßmengenfibration über der Interaktionskategorie $\mathcal{SP}roc$ geeignet ist, als ein konkretes Modell für Lineare Logik zu dienen.

Zum ersten Problemkreis gehört vor allem die Definition einer Typtheorie. Eine Bearbeitung dieser Frage mußte aus dieser Arbeit ausgeklammert bleiben. Stattdessen wurde eine einfache Typtheorie der Literatur entnommen. Die einzige Anforderung war, daß sich Kontexte auf natürliche Art in der Typtheorie modellieren lassen.

Die Typtheorie mit Produkttypen ist die schwächste derartige Struktur, die mir bekannt ist. Die Erfahrungen mit **ProcSet** zeigen, daß selbst diese Typtheorie zu spezielle Eigenschaften hat.

In diesem Zusammenhang hat sich die Frage herauskristallisiert, ob eine Typtheorie existiert, die ein Tensorprodukt für Typen besitzt, welches keine Universalitätseigenschaft hat. Zusätzlich müssen sich Typen und Terme in einer Kategorie darstellen lassen, die als Basiskategorie verwendet werden kann. Die klassifizierende Kategorie hat dafür zu starke Eigenschaften: In ihr gibt es immer Produkte, unabhängig von der betrachteten Typtheorie.

Es ist mir keine Veröffentlichung bekannt, die mehrsortige Lineare Logik definiert. In dieser Hinsicht beschreitet Abschnitt 3.3 Neuland. Allerdings konnten alle Fragen, die mit einer mehrsortigen Linearen Logik zusammenhängen im Rahmen dieser Arbeit nicht behandelt werden und mußten unbeantwortet bleiben. Unter anderem ist das Verhältnis zur klassischen Linearen Logik von Girard [Gir87] unbekannt.

Kapitel 4 geht einer dieser offenen Fragen nach: Läßt sich der Ansatz von Robert A. Seely zur Konstruktion von kategorientheoretischen Modellen auf den mehrsortigen Fall übertragen? Wie sich herausstellte ist der Schlüssel zu diesem Problem eine geeignete Definition der Interpretation von Formeln. Die restliche Konstruktion ist vom unterliegenden Typsystem relativ unabhängig.

Der Korrektheitsbeweis in Abschnitt 4.4 enthält damit im wesentlichen bekannte Resultate. Allerdings ist mir keine Veröffentlichung bekannt, in der der Beweis so detailliert ausgearbeitet wurde.

Zum zweiten Problemkreis: Im Kapitel 6 wird die Struktur der Prozeßmengenfibration über $\mathcal{SP}roc$ ausführlich untersucht. Die Kategorie **ProcSet** eignet sich nicht als ein Modell Linearer Logik.

Daneben halte ich die Beweise für die Adjunktion $\exists \dashv \varrho^* \dashv \forall$ als ein wesentliches Ergebnis dieses Kapitels. Sie demonstrieren, daß es für Lineare Logik nicht zwingend ist, die Verkettung von Kontexten durch ein kartesisches Produkt zu modellieren. Die Beweise können als Motivation zur Suche nach einer schwächeren Typtheorie für Lineare

Logik dienen.

Ein unerwartetes Resultat ergab sich bei der Suche nach einer geeigneten Definition für den Funktor ! auf **ProcSet**: Es ist nicht möglich, die Struktur von $\mathcal{SProc}$ zur Definition eines Comonoiden in **ProcSet** zu nutzen, weil die Morphismen der Comonoiden in $\mathcal{SProc}$ nicht natürlich sind.

Damit im Zusammenhang steht Satz 5.7 im Abschnitt 5.4: Die Interaktionskategorie $\mathcal{SProc}$ modelliert Cut–Elimination nicht. Sowohl in den Veröffentlichungen von Samson Abramsky [AGN93] als auch in [Gay94] und in [GN95] wird Cut–Elimination nicht im Zusammenhang mit $\mathcal{SProc}$ betrachtet. Auch in [See89] wird Cut–Elimination nicht erwähnt.

A. Das Regelsystem der Linearen Logik

Typableitung

$$\frac{\tau \in |\Sigma|}{\langle\rangle \ \rhd \ \tau : \mathsf{Type}}\ \text{SigType} \qquad\qquad \frac{}{\langle\rangle \ \rhd \ \mathbf{1} : \mathsf{Type}}\ 1\ \text{Type}$$

$$\frac{\langle\rangle \ \rhd \ \tau : \mathsf{Type} \qquad \langle\rangle \ \rhd \ \sigma : \mathsf{Type}}{\langle\rangle \ \rhd \ \tau \times \sigma : \mathsf{Type}}\ \times\ \text{Type}$$

Terme

$$\frac{\langle\rangle \ \rhd \ \sigma : \mathsf{Type}}{v : \sigma \ \rhd \ v : \sigma}\ \text{Term Id} \qquad\qquad \frac{}{\langle\rangle \ \rhd \ () : \mathbf{1}}\ \text{Term ()}$$

$$\text{Für } F : \sigma_1, \ldots, \sigma_n \longrightarrow \sigma \text{ in } \Sigma \qquad \frac{\Upsilon \ \rhd \ t_1 : \sigma_1 \quad \cdots \quad \Upsilon \ \rhd \ t_n : \sigma_n}{\Upsilon \ \rhd \ F(t_1, \ldots, t_n) : \sigma}\ \text{Symbol}$$

$$\text{Für } v \text{ nicht in } \Upsilon \qquad \frac{\sigma : \mathsf{Type} \qquad \Upsilon \ \rhd \ t : \tau}{\Upsilon, v : \sigma \ \rhd \ t : \tau}\ \text{weakening}$$

$$\frac{\Upsilon, u : \sigma, v : \sigma \ \rhd \ t : \tau}{\Upsilon, u : \sigma \quad \rhd \ t[u/v] : \tau}\ \text{Contraction} \qquad\qquad \frac{\Upsilon, u : \sigma, v : \rho, \Xi \ \rhd \ t : \tau}{\Upsilon, v : \rho, u : \sigma, \Xi \ \rhd \ t : \tau}\ \text{Exchange}$$

$$\frac{\Upsilon, v : \sigma \ \rhd \ t : \tau \qquad \Upsilon \ \rhd \ r : \sigma}{\Upsilon \ \rhd \ t[r/v] : \tau}\ \text{Subst}$$

$$\frac{\Upsilon \ \rhd \ t : \sigma \qquad \Upsilon \ \rhd \ r : \tau}{\Upsilon \ \rhd \ (t, r) : \sigma \times \tau}\ \times\ \text{Term}$$

$$\frac{\Upsilon \ \rhd \ t : \sigma \times \tau}{\Upsilon \ \rhd \ \pi_1 t : \sigma}\ \pi_1\ \text{Term} \qquad\qquad \frac{\Upsilon \ \rhd \ t : \sigma \times \tau}{\Upsilon \ \rhd \ \pi_2 t : \tau}\ \pi_2\ \text{Term}$$

Abbildung 10: Regelsystem des Termkalküls

Atomare Formeln

Für $R : \sigma_1, \ldots, \sigma_n$ ein Relationssymbol aus Σ

$$\frac{\Upsilon \ \rhd \ t_1 : \sigma_1 \quad \cdots \quad \Upsilon \ \rhd \ t_n : \sigma_n}{\Upsilon \ \rhd \ R(t_1, \ldots, t_n) : \mathsf{Prop}} \ \text{Relationssymbol}$$

Für $X \in \{I, \mathbf{0}, \top, \bot\}$

$$\frac{}{\langle\rangle \ \rhd \ X : \mathsf{Prop}} \ \text{Konstante}$$

Abgeleitete Formeln

Für $\ominus \in \{\otimes, ⅋, \multimap, \&, \oplus\}$

$$\frac{\Upsilon \ \rhd \ A : \mathsf{Prop} \qquad \Upsilon \ \rhd \ B : \mathsf{Prop}}{\Upsilon \ \rhd \ A \ominus B : \mathsf{Prop}} \ \text{Konnektoren}$$

Für $\theta \in \{\forall, \exists\}$

$$\frac{\Upsilon, x : \sigma \ \rhd \ A : \mathsf{Prop}}{\Upsilon \ \rhd \ \theta x : \sigma \,.\, A : \mathsf{Prop}} \ \text{Quantoren}$$

$$\frac{\Upsilon \ \rhd \ \sigma : \mathsf{Type} \qquad \Upsilon \ \rhd \ A : \mathsf{Prop}}{\Upsilon, x : \sigma \ \rhd \ A : \mathsf{Prop}} \ \text{Weakening Prop}$$

$$\frac{\Upsilon \ \rhd \ A : \mathsf{Prop}}{\Upsilon \ \rhd \ A^{\bot} : \mathsf{Prop}} \ \text{Negation}$$

Für die Regel Weakening gilt, wie im Text im Abschnitt 3.2 vermerkt, daß x nicht bereits in Υ vorkommen darf.

Abbildung 11: Wohldefinierte Formeln der mehrsortigen Linearen Logik

Axiome

$$\frac{}{\langle\rangle \ \rhd \ \bot \ \vdash \ \langle\rangle} \ \text{Bottom L} \qquad \frac{}{\langle\rangle \ \rhd \ \langle\rangle \ \vdash \ I} \ \text{Unit R}$$

$$\frac{\text{für alle } A \in \Gamma, \Delta \ \ : \ \ \Upsilon \ \rhd \ A : \mathsf{Prop}}{\Upsilon \ \rhd \ \Gamma, 0 \ \vdash \ \Delta} \ \text{Zero L}$$

$$\frac{\text{für alle } A \in \Gamma, \Delta \ \ : \ \ \Upsilon \ \rhd \ A : \mathsf{Prop}}{\Upsilon \ \rhd \ \Gamma \ \vdash \ \top, \Delta} \ \text{Top R}$$

$$\frac{\Upsilon \ \rhd \ A : \mathsf{Prop}}{\Upsilon \ \rhd \ A \ \vdash \ A} \ \mathsf{id}_A$$

Multiplikative Regeln

$$\frac{\Upsilon \ \rhd \ \Gamma \ \vdash \ \Delta}{\Upsilon \ \rhd \ \Gamma, I \ \vdash \ \Delta} \ \text{Unit L} \qquad \frac{\Upsilon \ \rhd \ \Gamma \ \vdash \ \Delta}{\Upsilon \ \rhd \ \Gamma \ \vdash \ \bot, \Delta} \ \bot \ \text{R}$$

$$\frac{\Upsilon \ \rhd \ \Gamma, A, B \ \vdash \ \Delta}{\Upsilon \ \rhd \ \Gamma, A \otimes B \ \vdash \ \Delta} \ \otimes \ \text{L} \qquad \frac{\Upsilon \ \rhd \ \Gamma \ \vdash \ A, B, \Delta}{\Upsilon \ \rhd \ \Gamma \ \vdash \ A \parr B, \Delta} \ \parr \ \text{R}$$

$$\frac{\Upsilon \ \rhd \ \Gamma \ \vdash \ A, \Delta \qquad \Upsilon \ \rhd \ \Theta \ \vdash \ B, \Psi}{\Upsilon \ \rhd \ \Gamma, \Theta \ \vdash \ A \otimes B, \Delta, \Psi} \ \otimes \ \text{R}$$

$$\frac{\Upsilon \ \rhd \ \Gamma, A \ \vdash \ \Delta \qquad \Upsilon \ \rhd \ \Theta, B \ \vdash \ \Psi}{\Upsilon \ \rhd \ \Gamma, \Theta, A \parr B \ \vdash \ \Delta, \Psi} \ \parr \ \text{L}$$

$$\frac{\Upsilon \ \rhd \ \Gamma \ \vdash \ A, \Delta \qquad \Upsilon \ \rhd \ \Theta, B \ \vdash \ \Psi}{\Upsilon \ \rhd \ \Gamma, \Theta, A \multimap B \ \vdash \ \Delta, \Psi} \ \multimap \ \text{L}$$

$$\frac{\Upsilon \ \rhd \ \Gamma, A \ \vdash \ B, \Delta}{\Upsilon \ \rhd \ \Gamma \ \vdash \ A \multimap B, \Delta} \ \multimap \ \text{R}$$

Abbildung 12: Das Regelsystem der Linearen Logik I

Strukturelle Regel

σ, τ zwei beliebige Permutationen

$$\frac{\Upsilon \ \triangleright \ \Theta \vdash \Gamma}{\Upsilon \ \triangleright \ \sigma\Theta \vdash \tau\Gamma} \ \text{Permutation}$$

Negation

$$\frac{\Upsilon \ \triangleright \ \Gamma, A \vdash B, \Delta}{\Upsilon \ \triangleright \ \Gamma, B^{\perp} \vdash A^{\perp}, \Delta} \ \text{Negation}$$

Additive Regeln

$$\frac{\Upsilon \ \triangleright \ \Gamma, A \vdash \Delta \qquad \Upsilon \ \triangleright \ B : \mathsf{Prop}}{\Upsilon \ \triangleright \ \Gamma, A \,\&\, B \vdash \Delta} \ \&\ \text{L1}$$

$$\frac{\Upsilon \ \triangleright \ \Gamma, B \vdash \Delta \qquad \Upsilon \ \triangleright \ A : \mathsf{Prop}}{\Upsilon \ \triangleright \ \Gamma, A \,\&\, B \vdash \Delta} \ \&\ \text{L2}$$

$$\frac{\Upsilon \ \triangleright \ \Gamma \vdash A, \Delta \qquad \Upsilon \ \triangleright \ \Gamma \vdash B, \Delta}{\Upsilon \ \triangleright \ \Gamma \vdash A \,\&\, B, \Delta} \ \&\ \text{R}$$

$$\frac{\Upsilon \ \triangleright \ \Gamma, A \vdash \Delta \qquad \Upsilon \ \triangleright \ \Gamma, B \vdash \Delta}{\Upsilon \ \triangleright \ \Gamma, A \oplus B \vdash \Delta} \ \oplus\ \text{L}$$

$$\frac{\Upsilon \ \triangleright \ \Gamma \vdash A, \Delta \qquad \Upsilon \ \triangleright \ B : \mathsf{Prop}}{\Upsilon \ \triangleright \ \Gamma \vdash A \oplus B, \Delta} \ \oplus\ \text{R1}$$

$$\frac{\Upsilon \ \triangleright \ \Gamma \vdash B, \Delta \qquad \Upsilon \ \triangleright \ A : \mathsf{Prop}}{\Upsilon \ \triangleright \ \Gamma \vdash A \oplus B, \Delta} \ \oplus\ \text{R2}$$

Abbildung 13: Das Regelsystem der Linearen Logik II

Schnittregel

$$\frac{\Upsilon \ \triangleright\ \Gamma \ \vdash\ \Delta, A \qquad \Upsilon \ \triangleright\ A, \Theta \ \vdash\ \Psi}{\Upsilon \ \triangleright\ \Gamma, \Theta \ \vdash\ \Delta, \Psi}\ \text{Cut}$$

Modalitäten

$$\frac{\Upsilon \ \triangleright\ !\Gamma \ \vdash\ A}{\Upsilon \ \triangleright\ !\Gamma \ \vdash\ !A}\ \text{Promotion !} \qquad \frac{\Upsilon \ \triangleright\ \Gamma \ \vdash\ \Delta \qquad \Sigma \ \triangleright\ A : \mathsf{Prop}}{\Upsilon, \Sigma \ \triangleright\ \Gamma, !A \ \vdash\ \Delta}\ \text{Weakening !}$$

$$\frac{\Upsilon \ \triangleright\ A \ \vdash\ \Gamma}{\Upsilon \ \triangleright\ !A \ \vdash\ \Gamma}\ \text{Dereliction !} \qquad \frac{\Upsilon \ \triangleright\ !A, !A \ \vdash\ \Gamma}{\Upsilon \ \triangleright\ !A \ \vdash\ \Gamma}\ \text{Contraction !}$$

$$\frac{\Upsilon \ \triangleright\ A \ \vdash\ ?\Gamma}{\Upsilon \ \triangleright\ ?A \ \vdash\ ?\Gamma}\ \text{Promotion ?} \qquad \frac{\Upsilon \ \triangleright\ \Gamma \ \vdash\ \Delta \qquad \Sigma \ \triangleright\ A : \mathsf{Prop}}{\Upsilon, \Sigma \ \triangleright\ \Gamma \ \vdash\ ?A, \Delta}\ \text{Weakening ?}$$

$$\frac{\Upsilon \ \triangleright\ \Gamma \ \vdash\ A}{\Upsilon \ \triangleright\ \Gamma \ \vdash\ ?A}\ \text{Dereliction ?} \qquad \frac{\Upsilon \ \triangleright\ \Gamma \ \vdash\ ?A, ?A}{\Upsilon \ \triangleright\ \Gamma \ \vdash\ ?A}\ \text{Contraction ?}$$

Quantoren

$$\frac{\Upsilon \ \triangleright\ t : \sigma \qquad \Upsilon \ \triangleright\ \Gamma, A[t/x] \ \vdash\ \Delta}{\Upsilon \ \triangleright\ \Gamma, \forall x : \sigma \,.\, A \ \vdash\ \Delta}\ \forall\ \text{L} \qquad \frac{\Upsilon, x : \sigma \ \triangleright\ \Gamma \ \vdash\ \Delta, A}{\Upsilon \ \triangleright\ \Gamma \ \vdash\ \Delta, \forall x : \sigma \,.\, A}\ \forall\ \text{R}$$

$$\frac{\Upsilon, x : \sigma \ \triangleright\ \Gamma, A \ \vdash\ \Delta}{\Upsilon \ \triangleright\ \Gamma, \exists x : \sigma \,.\, A \ \vdash\ \Delta}\ \exists\ \text{L} \qquad \frac{\Upsilon \ \triangleright\ t : \sigma \qquad \Upsilon \ \triangleright\ \Gamma \ \vdash\ \Delta, A[t/x]}{\Upsilon \ \triangleright\ \Gamma \ \vdash\ \Delta, \exists x : \sigma \,.\, A}\ \exists\ \text{R}$$

Die Regeln $\forall$ R und $\exists$ L gelten unter der gewöhnlichen Voraussetzung der Eigenvariablebedingung: Die Variable $x : \sigma$ darf in beiden Regeln weder in Γ noch in Δ frei vorkommen.
In den Regeln steht $!\Gamma$ bzw. $?\Gamma$ als Abkürzung für die Anwendung von ! bzw. ? auf jede einzelne Formel: $!A_1, \ldots, !A_n$.

Abbildung 14: Das Regelsystem der Linearen Logik III

B. Erklärung

Hiermit erkläre ich, daß ich die vorliegende Arbeit selbständig angefertigt und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet sowie Zitate kenntlich gemacht habe.

Dresden, den 8. November 1995

Hendrik Tews

C. Literaturverzeichnis

[Abr94] ABRAMSKY, SAMSON: *Interaction Categories and communicating sequential processes.* In ROSCOE, A. W. (editor): *A Classical Mind: Essays in Honour of C. A. R. Hoare*, pages 1–15. Prentice Hall International, 1994.

[Acz88] ACZEL, PETER: *Non-Well-Founded Sets.* Center for the Study of Language and Information, Stanford University, 1988. CSLI Lecture Notes, Volume 14.

[AGN93] ABRAMSKY, SAMSON, SIMON J. GAY, and RAJAGOPAL NAGARAJAN: *Interaction categories and the foundations of typed concurrent programming.* In BROY, M. (editor): *Deductive Program Design: Proceedings of the 1994 Marktoberdorf Summer School*, NATO ASI Series F: Computer and System Sciences. Springer–Verlag, 1993. to appear.

[Amb91] AMBLER, S. J.: *First Order Linear Logic in Symmetric Monoidal Closed Categories.* PhD thesis, University of Edinburgh, 1991.

[Bar92] BARENDREGT, HENK P.: *Lambda calculi with types.* In ABRAMSKY, S., DOV M. GABBAY, and T.S.E. MAIBAUM (editors): *Handbook of Logic in Computer Science*, volume 2. Oxford Science Publications, 1992.

[BW84] BARR, MICHAEL and CHARLES WELLS: *Toposes, Triples and Theories.* Springer–Verlag, 1984. Grundlehren der mathematischen Wissenschafter, Volume 278.

[EK65] EILENBERG, SAMUEL and G. MAX KELLY: *Closed categories.* In EILENBERG, S., D. K. HARRISON, S. MACLANE, and H. RÖHRL (editors): *Proceedings of the Conference on Categorical Algebra, La Jolla, 1965*, pages 421–562. Springer–Verlag, 1965.

[FKKK94] FARWER, BERNDT, PETER-ALEXIS KASTEN, ANDREAS KÖSTER und MICHAEL KOPKA: *Nichtklassische Logiken.* Diplomarbeit, Universität Hamburg, Dezember 1994.

[Gay94] GAY, SIMON J.: *Linear Types for Communicating Processes.* PhD thesis, University of London, 1994.

[Gir87] GIRARD, JEAN-YVES: *Linear logic.* Theoretical Computer Science, 50:1–102, 1987.

[GLT88] GIRARD, JEAN-YVES, YVES LAFONT, and PAUL TAYLOR: *Proofs and Types*, volume 7 of *Cambridge Tracts in Theoretical Computer Science 7.* Cambridge University Press, 1988.

[GN95] GAY, SIMON J. and R. NAGARAJAN: *A typed calculus of synchronous processes.* In KOZEN, D. (editor): *Proceedings of the Tenth Annual Symposium on Logic in Computer Science*, San Diego, California, June 1995. To appear.

[Jac] JACOBS, B. P. F.: *Categorical Logic and Type Theory.* Forthcoming.

[Jac91] JACOBS, B. P. F.: *Categorical type theory.* PhD thesis, University of Nijmegen, Nijmegen, The Netherlands, 1991.

[Kel82] KELLY, GREGORY MAXWELL: *Basic Concepts of Enriched Category Theory.* Mathematical Society Lecture Note Series 64. Cambridge University Press, 1982.

[Mil89] MILNER, ROBIN: *Communication and Concurrency.* Prentice Hall, 1989.

[ML71] LANE, SAUNDERS MAC: *Categories for the Working Mathematician.* Springer–Verlag, 1971.

[MOM90] MARTÍ-OLIET, N. and J. MESEGUER: *Duality in closed and linear categories.* Technical Report SRI-CSL-90-01, SRI International, Computer Science Laboratory, February 1990.

[Pie91] PIERCE, BENJAMIN C.: *Basic Category Theory for Computer Scientists.* MIT Press, Cambridge, 1991.

[Pit87] PITTS, ANDREW M.: *Polymorphism is set theoretic, constructively.* In PITT, D. H., A. POIGNÉ, and DAVID E. RYDEHEARD (editors): *Category Theory and Computer Science*, volume 283 of *Lecture Notes in Computer Science*, Berlin, 1987. Springer–Verlag.

[Poi92] POIGNÉ, AXEL: *Basic category theory.* In ABRAMSKY, S., DOV M. GABBAY, and T.S.E. MAIBAUM (editors): *Handbook of Logic in Computer Science*, volume 1, chapter Basic Category Theory. Oxford Science Publications, 1992.

[Rei95] REICHEL, HORST: *Logik für Informatiker.* Vorlesungsskript, Sommersemester 1994/95.

[See89] SEELY, ROBERT A. G.: *Linear logic, ∗-autonomous categories and cofree coalgebras.* In GRAY, JOHN W. and ANDRE SCEDROV (editors): *Categories in Computer Science and Logic*, volume 92 of *Contemporary Mathematics*, pages 371–382, Providence, Rhode Island, 1989. American Mathematical Society.

[See90] SEELY, R. A. G.: *Polymorphic linear logic and topos models.* C.R. Math. Rep. Acad. Sci. Canada - Vol XII, No. 1, February 1990.

[Tew95] TEWS, HENDRIK: *Lineare Logik und Prozeßtheorie.* Großer Beleg, Technische Universität Dresden, 1995.

[Wal92] WALTERS, R. F. C.: *Categories and Computer Science.* Cambridge University Press, August 1992.