Kategorientheorie und Typen

Skript zur Vorlesung Wintersemester 2011/2012

Eva Richter

17. Februar 2012

Inhaltsverzeichnis

Vorwort

Dieses Skript ist als Begleitung zur Vorlesung Kategorietheorie und Logik entstanden und hat vor allem das Anliegen, eine einheitliche Darstellung des Materials zu geben, da die Notation und die Reihenfolge der Themen in den verschiedenen Büchern stark voneinander abweicht. Ich habe verschiedene Bücher als Quellen verwendet, die am Ende in der Literaturliste zusammengefasst sind. Einige der angegebenen Bücher sind in elektronischer Version erhältlich. Dieses Skript hat nicht den Anspruch, Lehrbuchniveau zu erreichen. Insbesondere sind die Texte zur Motivation der einzelnen Teile oft sehr kurz geraten. Dieser Mangel wird in der Vorlesung hoffentlich ausgeglichen.

Mathematik im Allgemeinen und ebenso theoretische Informatik erfordern ein hohes Maß an Mitarbeit auf Seiten der Hörer. Um sich mit den Konzepten vertraut zu machen kann es z.B. helfen, wenn die Teilnehmer die Konstruktionen in verschiedenen Kategorien ausführen. Das kann manchmal mit einem erheblichen Rechenaufwand verbunden sein, was nicht überraschend ist, wenn man sich vor Augen führt, dass man es in diesem Gebiet mit einem hohen Abstraktionsgrad zu tun hat. Dieser Abstraktionsgrad ist oft auch dafür verantwortlich, dass man sehr lange braucht, um wenige Seiten zu lesen (und zu verstehen). Man sollte sich davon nicht entmutigen lassen. Um das Verstehen zu erleichtern, habe ich an verschiedenen Stellen Übungsaufgaben eingestreut. Zusätzliche Aufgaben werden gelegentlich in der Vorlesung gegeben. An einigen Stellen, an denen die Beweise (meiner Meinung nach) keine speziellen Kniffe und Ideen brauchen, habe ich diese nicht angegeben. Man kann das Ausführen eines solchen Beweises als gute Fingerübung ansehen. Wer Schwierigkeiten mit einer Aufgabe hat, ist eingeladen mir seinen Lösungsversuch zu zeigen und sich Hilfe zu holen.

Da das Skript sowieso in ständiger Bearbeitung ist, nehme ich auch gern Hinweise zu Typos oder missverständlichen Formulierungen entgegen.

1 Einleitung

Die Theorie der Kategorien beginnt mit der Beobachtung, dass sich durch die Darstellung mittels Diagrammen von Pfeilen viele Eigenschaften mathematischer Systeme einheitlich erfassen und vereinfachen lassen. Um die Verkettung von zwei Funktionen f und g darzustellen, können wir folgendes Diagramm zeichnen:

$$\begin{tikzcd} X \arrow[rr, "f"] \arrow[dr, "h"'] & & Y \arrow[dl, "g"] \\ & Z & \end{tikzcd}$$

Falls für h gilt, dass $h = g \circ f$, dann sagen wir, dass das Diagramm *kommutativ* ist. Lesen wir das Diagramm so, dass X, Y, Z Elemente einer Menge sind und dass ein Pfeil bedeutet, dass das Element am Anfang des Pfeils kleiner als das am Ende ist, dann bedeutet es $X \leq Y, Y \leq Z$ und $X \leq Z$. Das typische Vorgehen in der Kategorientheorie besteht darin, eine gesuchte mathematische Struktur als universelle Eigenschaft von Diagrammen darzustellen.

Betrachten wir dazu das Beispiel des Kreuzproduktes $X \times Y$ zweier Mengen X und Y. Es besteht aus der Menge der geordneten Paare (x, y) von Elementen $x \in X$ und $y \in Y$. Außerdem haben wir Funktionen $p_1 : X \times Y \to X$ und $p_2 : X \times Y \to Y$ mit $p_1((x, y)) = x$ und $p_2((x, y)) = y$. Das Produkt hat folgende Eigenschaft:

(*) Für Mengen W, X, Y und jedes Paar von Funktionen $f : W \to X$ und $g : W \to Y$ gibt es eine eindeutig bestimmte Funktion $h : W \to X \times Y$ mit $p_1 \circ h = f$ und $p_2 \circ h = g$.

Ebenso gilt die Umkehrung, nämlich dass sich für jede Funktion h mithilfe von p_1 und p_2 die Koordinatenfunktionen f und g berechnen lassen.

Mit (*) haben wir einen Weg gefunden, das Wesentliche eines Produktes zu beschreiben, ohne uns auf die Elemente zu stützen.

Wir wollen nun als Objekte der Beobachtung statt Mengen Elemente einer Halbordnung betrachten und die Existenz eines Pfeils $X \to Y$ als $X \leq Y$ interpretieren, dann bedeutet (*)

(**) Für drei beliebige Elemente W, X, Y mit $W \leq X$ und $W \leq Y$ gilt $W \leq X \times Y$, wobei $X \times Y \leq X$ und $X \times Y \leq Y$

Wenn es existiert, ist das Element $X \times Y$ kleiner als X und als Y und jedes andere Element W, das sowohl kleiner X als auch kleiner Y ist, ist kleiner als $X \times Y$. Mit anderen Worten: $X \times Y$ ist das Infimum von X und Y. Es stellt sich also heraus, dass die Struktur, die wir als kartesisches Produkt für Mengen kennen, eine sinnvolle Entsprechung in einer anderen Domäne hat. Wie jede andere Abstraktion macht die kategoriale Beschreibung solche Gemeinsamkeiten sichtbar.

Die Kategorientheorie wurde ursprünglich als Sprache zur Beschreibung mathematischer Strukturen entwickelt. Eingeführt wurde sie in den 40-er Jahren des 20. Jahrhunderts von Samuel Eilenberg und Saunders MacLane. Ihre Wurzeln liegen in der algebraischen Topologie. Dort wurden Konstruktionen entwickelt, die das Gebiet der Topologie mit dem der Algebra, insbesondere der Gruppentheorie verbinden. Die Kategorientheorie wurde jedoch bald auch für andere Bereiche der Mathematik verwendet und ist inzwischen ein eigener wesentlicher Zweig der reinen Mathematik.

Für uns noch viel wichtiger ist, dass die Kategorientheorie einen bemerkenswerten Einfluss auf die konzeptionellen Grundlagen der Mathematik und die Sprache der mathematischen Praxis hatte. Während die Welt der Mathematik die Kategorie der Mengen ist, d.h. die Mengenlehre als Grundlage der Mathematik betrachtet werden kann, ist diese für die Informatik nicht unbedingt der geeignetste Ausgangspunkt. Der Gegenstand der theoretischen Informatik sind Algorithmen und Funktionen, und es ist wichtig, zwischen Funktionen, die berechenbar sind und solchen, die es nicht sind, zu unterscheiden. Eng damit verknüpft ist die Frage nach rekursiven Mengen, also Mengen, für die es eine berechenbare Funktion gibt, die ihre Elemente konstruiert. Eine wichtige Aufgabe ist es also Kategorien zu untersuchen, deren Objekte z.B. aufzählbare Mengen sind.

Kategorientheoretische Ansätze, die Grundlagen der Mengenlehre zu axiomatisieren, haben zu einer Verallgemeinerung des Mengenbegriffs geführt. Betrachtet wurden dazu Kategorien, die sich „mengenähnlich“ verhalten, wobei zunächst genauer geklärt werden musste, was mit „mengenähnlich“ gemeint sein sollte.

Erste Antworten auf diese Frage hat F.William Lawvere 1964 gegeben. Sein ursprüngliches Ziel war, Mengenlehre kategorial zu axiomatisieren, d.h. Begriffe wie Elementbeziehung, leere Menge, Teilmenge und Potenzmenge mit Hilfe von Universalitätseigenschaften zu beschreiben. Das Ergebnis war jedoch nicht befriedigend, da er als Grundlage seiner Betrachtungen selbst Mengenlehre verwenden musste. Im Jahr 1969 begann Lawvere zusammen mit Myles Tierney, solche Kategorien zu studieren, die über eine bestimmte Art von Pfeilen verfügen, die sie als „subobject classifier“ bezeichneten. Wie sich später herausstellte, war dieser Begriff gerade der Schlüssel zu Lawveres früherem Problem. Das Ergebnis der Untersuchungen war das abstrakte axiomatische Konzept eines elementaren Topos[1], das wir später noch kennenlernen werden. Der Begriff „Topos“ war schon vorher von Grothendieck in der Algebra verwendet worden und wurde von Lawvere und Tierney übernommen, nachdem sich herausstellte, dass die von Grothendieck untersuchten Topoi alle einen solchen subobject classifier besaßen.

Warum sollte sich ein Informatiker für Kategorientheorie interessieren?

Einen der Gründe haben wir schon genannt: über die Kategorientheorie erhält man Zugang zu einem Universum, das sich mengenähnlich verhält, aber doch mit zusätzlicher Struktur ausgerüstet werden kann z.B. um berechenbare von allgemeinen Funktionen zu unterscheiden. Auch in der Herangehensweise kommt die Kategorientheorie der Informatik entgegen, da sie eher Funktionen und Relationen statt Mengenzugehörigkeit in den Mittelpunkt der Aufmerksamkeit stellt.

Eng verknüpft mit der Mengenlehre ist die klassische Logik. Die Prinzipien der klassischen Logik werden als Operationen auf einer bestimmten Menge, nämlich der Boolschen Algebra, dargestellt. Betrachtet man Topoi, dann stellt sich heraus, dass jeder Topos seine eigene Logik trägt. Eine solche Topos-Logik kann von der klassischen Logik ziemlich stark abweichen. Im (allgemeinen) Topos sind die logischen Prinzipien die der intuitionistischen Logik. Intuitionismus ist eine konstruktivistische Philosophie, die sich mit der Natur der mathematischen Objekte, ihrer Bedeutung, sowie der Gültigkeit mathematischer Aussagen beschäftigt. Ihre wichtigstes Paradigma ist die Forderung, dass jede Existenzaussage durch eine direkte Konstruktion bewiesen werden soll, d.h. man kann von einem Objekt nur dann behaupten, dass es existiert, wenn man eine Konstruktion dafür angeben kann. Das Gesetz vom Ausgeschlossenen Dritten ist hier kein Axiom und damit werden indirekte Beweise nicht anerkannt. Überträgt man diese Forderung auf Spezifikationen, dann bedeutet das, dass der Beweis der Existenz eines bestimmten Ergebnisses darin besteht, einen Algorithmus für seine Berechnung anzugeben. Ein Programm ist also seinem Wesen nach ein Beweis. Welche Operationen dabei möglich (also welche Schlüsse dabei zulässig) sind, hängt von der Programmiersprache ab. Dabei geben die Konstruktoren für Typen vor, welche Kategorie die passende Interpretation ist. So braucht man für Case-Anweisungen eine Kategorie, in der direkte Summen existieren. Die Beziehung zwischen Programmen und Beweisen heißt *Curry-Howard-Correspondence*[2].

Eine andere Beziehung zwischen Logik und Informatik, die durch Kategorientheorie sehr gut beschrieben werden kann, ist die zwischen Modallogik und Coalgebren. Während man sich lange hauptsächlich für induktive Datentypen interessiert hat, die kategorial als initiale Algebren interpretiert werden können, ist in letzter Zeit auch das Interesse an den Datentypen gewachsen, die durch (finale) Coalgebren interpretiert werden. Die-

[1]Das Wort stammt aus dem Griechischen und heißt soviel wie „Platz“ oder „Ort“.

[2]benannt nach dem Mathematiker Haskell Curry und dem Logiker William Alvin Howard

se sind nämlich gerade die Objekte (im Sinne der Objektorientierten Programmierung). Kategorial betrachtet sind Coalgebren das zu Algebren duale[3] Konzept. Statt eines Konstruktors hat man hier Destruktoren, die wiederum den Methoden[4] entsprechen. Wie sich herausgestellt hat, ist die logische Entsprechung für coinduktive Datentypen die Modallogik. Die kategoriale Betrachtung kann hier als Werkzeug zur Entwicklung geeigneter Spezifikationssprachen dienen, insbesondere im Bereich von Sicherheitsanalysen.

[3]Dualität ist ein wichtiges Prinzip der Kategorientheorie, das wir in 3.1 kennenlernen werden.
[4]der objektorientierten Programmierung

2 Kategorien

2.1 Graphen

Kategorientheorie ist unter anderem deswegen so beliebt, weil die typische Darstellung abstrakter Zusammenhänge in Form von sogenannten kommutativen Diagrammen sehr anschaulich und intuitiv verständlich ist. Grundlage für die häufige Verwendung solcher Diagramme ist, dass wir Kategorien selbst als Graphen auffassen können. Der zugrundeliegende Graphenbegriff ist der eines gerichteten Multigraphen mit Schleifen. Um einen Graphen zu beschreiben gibt man seine Knoten und (gerichteten) Kanten an. Jede Kante muss einen Anfangsknoten (Quelle) und einen Endknoten (Ziel) haben, der eindeutig bestimmt ist. Mit $f : A \to B$ bezeichnen wir eine Kante mit Quelle A und Ziel B. Dabei kann es mehrere oder keine Kanten zwischen einer bestimmten Quelle und einem bestimmten Ziel geben. Wenn Quelle und Ziel einer Kante übereinstimmen, dann heißt diese Kante Endomorphismus[5] dieses Knotens.

Beispiele 1. *1. Sei $\mathcal{G} = (G_0, G_1)$ mit $G_0 = \{A, B\}$ und $G_1 = \{f, g, h\}$, mit*

$$\mathit{Quelle}(f) = \mathit{Ziel}(f) = \mathit{Quelle}(g) = \mathit{Ziel}(h) = A \text{ und } \mathit{Ziel}(g) = \mathit{Quelle}(h) = B$$

Dieser Graph wird in Abbildung 1 dargestellt.

Abbildung 1:

$$\begin{tikzcd} A \arrow[loop above, "f"] \arrow[r, bend left, "g"] & B \arrow[l, bend left, "h"] \end{tikzcd}$$

2. Relationen lassen sich als Graphen darstellen. Für eine Menge $A = \{1, 2, 3, 4\}$ entspricht die Relation $\alpha = \{(1, 2), (2, 2), (2, 3), (1, 4)\} \subset A \times A$ dem Graphen in Abbildung 15.

Abbildung 2: Relationen

$$\begin{tikzcd} 1 \arrow[r] \arrow[d] & 2 \arrow[loop above] \arrow[r] & 3 \\ 4 & & \end{tikzcd}$$

[5] Die Kanten werden allgemein als Morphismen bezeichnet; „morph“ stammt aus dem Griechischen und bedeutet Form, „endo“ bedeutet innen.

Natürlich haben Graphen, die aus Relationen entstanden sind, nie mehr als einen Pfeil pro Paar aus Quelle und Ziel. Sie heißen einfache Graphen.

3. *Betrachtet man den Graphen, der für jede Menge einen Knoten hat und dessen Kanten die Funktionen zwischen diesen Mengen darstellen, dann bilden die Knoten und die Kanten keine Mengen. Das liegt an Russels Pradoxon. Würde man nämlich erlauben, dass eine Kollektion der Art $\{S \mid S \text{ ist eine Menge und } S \notin S\}$ eine Menge ist, dann führt das zu Widersprüchen. Um diese Widersprüche zu vermeiden, verlangt man bei der Konstruktion, dass die S nur von einem bestimmten Typ (aus einem bestimmten Universum) sein dürfen. Ein Graph, dessen Knoten und Kanten keine Mengen (über demselben Universum wie die Knoten selbst) bilden, heißt großer Graph.*

4. *Graphen eignen sich auch zur Darstellung von Datenstrukturen. In Abbildung 3 sieht man eine Darstellung der Natürlichen Zahlen mit Hilfe der 0 und der Nachfolgerfunktion succ. Der Name 1 für den linken Knoten ist die übliche Notation*

Abbildung 3: Natürliche Zahlen

$$\begin{tikzcd} 1 \arrow[r, "0"] & n \arrow[loop above, "\mathrm{succ}"] \end{tikzcd}$$

für eine Einermenge. Wir werden später noch sehen, wie man aus der formalen mathematischen Beschreibung eine Graphendarstellung ableitet und umgekehrt.

Mindestens ebenso wichtig für die Darstellung mathematischer Strukturen und Eigenschaften wie die Graphen sind Homomorphismen zwischen Graphen. Das sind Abbildungen, die die Gestalt eines Graphen erhalten.

Definition 1. *Ein Graphenhomomorphismus φ von einem Graphen $\mathcal{G}$ zu einem Graphen $\mathcal{H}$ ist ein Paar von Funktionen $\varphi_0 : G_0 \to H_0$ und $\varphi_1 : G_1 \to H_1$, mit der Eigenschaft, dass falls $f : A \to B$ eine Kante in $\mathcal{G}$ ist, $\varphi_1(f) : \varphi_0(A) \to \varphi_0(B)$ eine Kante in $\mathcal{H}$ ist.*

Die Kanten eines Graphen kann man zu Pfaden zusammensetzen.

Definition 2. *Sei $\mathcal{C}$ ein Graph, und sei $k > 0$. Ein Pfad der Länge k von einem Knoten A zu einem Knoten B ist eine Folge von (nicht unbedingt verschiedenen) Kanten $(f_1, \dots, f_k)$ so, dass*

(i) $Quelle(f_k) = A$

(ii) $Ziel(f_i) = Quelle(f_{i-1})$ *für alle* $i = 2, \dots, k$ *und*

(iii) $Ziel(f_1) = B$

$$A \xrightarrow{f_k} \cdot \xrightarrow{f_{k-1}} \dots \xrightarrow{f_2} \cdot \xrightarrow{f_1} B$$

Per Konvention gibt es für jeden Knoten A genau einen Pfad der Länge 0 von A nach A der mit () bezeichnet wird. Dieser Pfad heißt der leere Pfad. Die Menge der Pfade der Länge k im Graphen $\mathcal{G}$ bezeichnet man mit G_k.
Man beachte, dass wenn man die Pfeile wie oben zeichnet, dann steht f_k links und die Indizes werden von links nach rechts kleiner. Wir machen das um Konsistenz mit der Schreibweise für die Komposition zu erreichen(siehe in Definition 3(ii)). Eine Kategorie ist ein Graph mit einer Regel für die Komposition von Kanten.

Definition 3. *Eine Kategorie ist ein Graph* $\mathbf{C}$ *mit zwei Funktionen, einer Komposition* $c : C_2 \to C_1$ *und Identitätsfunktion* $u : C_0 \to C_1$*, so dass (i) – (iv) erfüllt sind.*

(i) $Quelle(c(g, f) = g \circ f) = Quelle(f)$ *und* $Ziel(g \circ f) = Ziel(g)$

(ii) $(h \circ g) \circ f = h \circ (g \circ f)$ *wenn jede der Seiten definiert ist*

(iii) $u(A) = id_A$ *mit* $Quelle(id_A) = Ziel(id_A) = A$

(iv) Wenn $f : A \to B$ *dann* $f \circ id_A = id_B \circ f = f$

Die Elemente von C_0 *heißen Objekte, die Elemente von* C_1 *heißen Pfeile.*

Die Tatsache, dass c auf C_2 definiert ist, sichert, dass nur komponierbare Pfeile verknüpft werden.
Eine Kategorie heißt klein, wenn Objekte und Pfeile Mengen sind, andernfalls heißt sie groß.

Definition 4. *Sei* $\mathbf{C}$ *eine Kategorie, seien* A *und* B $\mathbf{C}$*-Objekte. Die Klasse aller Pfeile mit Quelle* A *und Ziel* B *bezeichnet man als Hom-Menge* [6] *und schreibt* $Hom_{\mathbf{C}}(A, B)$.

Falls aus dem Kontext hervorgeht, welche Kategorie gemeint ist, kann man den Index $\mathbf{C}$ auch weglassen. Die Komposition c induziert eine Funktion $Hom(B, C) \times Hom(A, B) \to Hom(A, C)$. Die Bezeichnung *Hom-Menge* suggeriert, dass $Hom(A, B)$ tatsächlich eine Menge bildet. Kategorien, in denen $Hom(A, B)$ für alle Paare von Objekten eine Menge ist, heißen lokal klein. Im Laufe der Vorlesung werden wir uns nur für lokal kleine Kategorien interessieren.

Beispiele 2. 1. *Die Kategorie* $\mathbf{O}$ *ist die leere Kategorie, sie enthält kein Objekt und keinen Pfeil.* $\mathbf{1}$ *hat genau ein Objekt und einen Pfeil. Andere kleine Kategorien sind* $\mathbf{1+1}$ *und* $\mathbf{2}$*, wie unten abgebildet:*

$$\begin{tikzcd} \cdot \arrow[loop above] & \cdot \arrow[loop above] \end{tikzcd} \qquad \mathbf{1+1}$$

[6] Die Bezeichnung kommt von Homomorphismus, einem aus dem Griechischen abgeleiteten Wort, wobei „homo“ gleich und „morph“ Form bedeutet. Gemeint sind also solche Abbildungen, die die Form erhalten.

$$\begin{tikzcd} \cdot \arrow[loop above] \arrow[r] & \cdot \arrow[loop above] \end{tikzcd} \qquad \mathbf{2}$$

2. *Die Objekte der Kategorie* **SET** *sind Mengen, die Pfeile von* **SET** *sind Funktionen. Für jede Menge gibt es eine Identitätsfunktion, die alle Elemente auf sich selbst abbildet. Die Komposition ist die Hintereinanderausführung von Funktionen. Assoziativität folgt aus der Assoziativität von Funktionen.*

3. *Die Kategorie* **Pfn** *hat als Objekte Mengen und als Pfeile partielle Funktionen. Eine partielle Funktion von einer Menge S in eine Menge T ist eine Funktion mit Definitionsbereich $S_0 \subseteq S$. Die Komposition partieller Abbildungen $f : S \to T$ mit Definitionsbereich S_0 und $g : T \to R$ mit Definitionsbereich T_0 ist auf dem Bereich $\{x \in S_0 \mid f(x) \in T_0\}$ definiert durch $g \circ f(x) = g(f(x))$.*

4. *Wenn α eine Relation auf $S \times T$ ist und β eine Relation auf $T \times U$, dann ist die Komposition $\beta \circ \alpha$ von α und β folgendermaßen definiert: wenn $x \in S$ und $z \in U$, dann ist $(x, z) \in \beta \circ \alpha$, falls ein $y \in T$ existiert mit $(x, y) \in \alpha$ und $(y, z) \in \beta$. Mit dieser Definition hat die Kategorie* **REL** *mit Mengen als Objekten und Relationen als Pfeilen eine Komposition. Die Identität einer Menge A ist die Diagonalrelation $\delta(A, A) = \{(a, a) \mid a \in A\}$.*

5. *Jede partiell geordnete Menge kann als Kategorie betrachtet werden, deren Objekte ihre Elemente sind. Es gibt genau dann einen Pfeil zwischen zwei Objekten, wenn sie in Relation stehen. Die Reflexivität sichert die Existenz der Identität, Transitivität sorgt für Komposition.*

6. *Ein Monoid ist eine Halbgruppe mit einem neutralen Element. Man kann ein Monoid als Kategorie mit einem Objekt auffassen, indem man jedes Element der Trägermenge als Pfeil betrachtet. Der identische Pfeil entspricht dem neutralen Element, die Komposition ist die Monoidoperation.*

7. *Typischerweise erhält man Kategorien durch das Zusammenfassen von Objekten mit der gleichen algebraischen Struktur und mit strukturerhaltenden Operationen als Pfeilen. Beispiele dafür sind* **GRP**, *deren Objekte Gruppen und deren Pfeile Gruppenhomomorphismen sind,* **VECT**, *deren Objekte Vektorräume und deren Pfeile lineare Abbildungen sind oder allgemeiner die Kategorie $\mathbf{ALG}_{\mathbf{\Sigma}}$ der Σ-Algebren mit zugehörigen Algebra-Homomorphismen.*

8. *Mit $\mathbf{G}^*$ bezeichnen wir die durch einen Graphen $\mathcal{G}$ generierte freie Kategorie. Die Objekte von $\mathbf{G}^*$ sind die Knoten des Graphen, die Pfeile von $\mathbf{G}^*$ sind die Pfade von $\mathcal{G}$. Die Komposition ist die Verkettung von Pfaden: $(f_1, \ldots f_{k-1}, f_k) \circ (f_{k+1}, \ldots, f_n) = (f_1, \ldots, f_n)$. Die Komposition ist assoziativ und für jedes Objekt A gibt es einen Identitätspfeil, nämlich den leeren Pfad. Die freie Kategorie über dem Graphen mit einem Knoten und einer Schleife ist das freie Monoid über einem einbuchstabigen Alphabet und ist isomorph zum Monoid der nichtnegativen Ganzen Zahlen mit $+$ als Operation.*

   Die freie Kategorie über dem Graphen in Abbildung 3 auf Seite 8 hat einen Pfeil $id_1 : 1 \to 1$ und für jedes $k \in \mathbb{N}$ gibt es einen Pfeil $succ^k : n \to n$, der dem Pfad

$(succ, succ \dots succ)$ (k-mal) entspricht. $succ^0$ entspricht dabei $()$ und ist id_n. Für jedes k ist $succ^k \circ 0 : 1 \to n$ und für die Komposition gilt $succ^k \circ succ^n = succ^{k+n}$.

Die Bedeutung der nächsten Konstruktion folgt vor allem aus ihrer Verbindung zu indizierten Mengen.

Definition 5. *Sei $\mathbf{C}$ eine Kategorie und $A \in Obj(\mathbf{C})$. Die Pfeilkategorie $\mathbf{C}/A$ ist eine Kategorie, deren Objekte $\mathbf{C}$-Homomorphismen $f : C \to A$ sind, wobei $C \in Obj(\mathbf{C})$. Ein Pfeil in $Hom_{\mathbf{C}/A}(f, f')$ mit $f' : C' \to A$ ist ein $\mathbf{C}$-Pfeil $h : C \to C'$ mit $f = f' \circ_{\mathbf{C}} h$. Die Komposition von $h : f \to f'$ und $h' : f' \to f''$ ist $h' \circ h$. Der identische Pfeil $id_f : f \to f$ ist der $\mathbf{C}$-Identitätspfeil $id_{Quelle(f)}$.*

Man überlegt sich leicht, dass die Definition korrekt ist, das heißt, dass die Komposition $h' \circ h$ wirklich die gewünschte Eigenschaft hat, nämlich $f = h' \circ h \circ f''$. Die Notation von Pfeilen in $\mathbf{C}\,/A$ ist etwas labil, da derselbe $\mathbf{C}$-Pfeil h sowohl $g = g' \circ h$ als auch $f = f' \circ h$ erfüllen kann obwohl $f \neq g$ und damit als Bezeichner für zwei verschiedene $\mathbf{C}\,/$-Pfeile dient.

Beispiele 3.

1. *Sei $(P, \leq)$ eine Halbordnung und $\mathbf{C}(P)$ die zugehörige Kategorie. Die Pfeilkategorie $\mathbf{C}(P)/x$ zu einem Element x hat als Objekte alle $\mathbf{C}(P)$-Pfeile mit Ziel x, sie entspricht also der Menge der Elemente, die kleiner als x sind.*

2. *Eine S-indizierte Menge ist eine Menge X zusammen mit einer Funktion $\tau : X \to S$. Falls $x \in X$ und $\tau(x) = s$, dann sagen wir x ist vom Typ s und sprechen von X als getypter Menge. In der Mathematik wird derselbe Sachverhalt manchmal folgendermaßen ausgedrückt. Eine Familie $\{t^{-1}(s) \mid s \in S\}$ von Teilmengen von X heißt über S indizierte Familie von Mengen. Zum Beispiel ist die Menge $G = G_0 \cup G_1$ von Knoten und Kanten eines Graphen ist eine getypte Menge. Es gilt $G_0 \cap G_1 = \emptyset$ und die Typisierung erfolgt durch $\tau : G \to \{0, 1\}$, wobei die Elemente von G_0 auf 0 und von G_1 auf 1 abgebildet werden.*

3. *Eine Funktion von einer über S getypten Menge X in eine über S getypte Menge X' ist gerade ein Pfeil in der Pfeilkategorie $\mathbf{SET}\,/S$. Eine solche Funktion heißt indizierte oder getypte Funktion. Insbesondere kann man jeden Graphenhomomorphismus als (über $\{0, 1\}$) getypte Funktion ansehen. Umgekehrt ist aber nicht jede getypte Funktion zwischen Graphen ein Graphenhomomorphismus, da das Bild eines Pfeiles nicht notwendig ein Pfeil im Zielgraphen ist.*

2.2 Konstruktionen von Kategorien

Aus der Algebra ist bekannt, wie man aus Gruppen(oder Halbgruppen oder Ringen) deren direktes Produkt bildet, indem man die Operationen koordinatenweise definiert. Das direkte Produkt ist nur ein Spezialfall für eine Struktur, die aus schon vorhandenen Strukturen konstruiert wird. Andere interessante Strukturen, die sich aus schon vorhandenen Strukturen ableiten lassen sind Unterstrukturen, also Teilmengen, die abgeschlossen sind unter den Operationen der Obermenge oder Quotientenstrukturen, die durch Bildung von Äquivalenzklassen bezüglich einer mit den Operationen der Obermenge zusammenpassenden Relation gebildet werden. Eine weitere Konstruktion ist die Bildung einer *freien*

Struktur eines bestimmten Typs über einer gegebenen Menge. Alle diese Konstruktionen können mit Kategorien nachgebildet werden. Einige davon sollen in diesem Abschnitt vorgestellt werden, andere folgen später.

Definition 6. *Sei* $\mathbf{C}$ *eine Kategorie. Eine Kategorie* $\mathbf{D}$ *heißt Teilkategorie, wenn sowohl* $Obj(\mathbf{D}) \subseteq \mathbf{C}$ *als auch für alle* $\mathbf{D}$*-Objekte* A, B *gilt* $Hom_{\mathbf{D}}(A, B) \subseteq Hom_{\mathbf{C}}(A, B)$*. Falls für alle* $A, B \in Obj(\mathbf{D})$ *gilt* $Hom_{\mathbf{D}}(A, B) = Hom_{\mathbf{C}}(A, B)$*, dann heißt* $\mathbf{D}$ *volle* Teilkategorie. *Falls* $Obj(\mathbf{C}) = Obj(\mathbf{D})$*, dann heißt* $\mathbf{D}$ *große Teilkategorie.*

Beispiele 4. *Die Kategorie* $\mathbf{SET}$ *ist eine große Teilkategorie von* $\mathbf{Pfn}$*, der Kategorie, deren Objekte Mengen und deren Pfeile partielle Funktionen sind(siehe Beispiel 2). Die Kategorie der Monoide ist eine Teilkategorie der Kategorie der Halbgruppen, die weder voll noch groß ist. Die Kategorie* $\mathbf{Fin}$ *der endlichen Mengen ist eine volle Teilkategorie von* $\mathbf{SET}$*.*

Definition 7. *Seien* $\mathbf{C}$ *und* $\mathbf{D}$ *Kategorien. Das Produkt* $\mathbf{C} \times \mathbf{D}$ *zweier Kategorien ist die Kategorie, deren Objekte Paare von Objekten aus* $\mathbf{C}$ *und* $\mathbf{D}$ *sind und deren Pfeile* $(A, B) \to (A', B')$ *Paare von Pfeilen* (f, g) *mit* $f \in Hom_{\mathbf{C}}(A, A')$ *und* $g \in Hom_{\mathbf{D}}(B, B')$ *sind. Für die Komposition definiert man:* $(h, k) \circ (f, g) = (h \circ f, k \circ g) : (A, B) \to (A'', B'')$ *falls* $h \in Hom_{\mathbf{C}}(A', A'')$ *und* $g \in Hom_{\mathbf{D}}(B', B'')$*.*

Ein wesentliches Mittel zum Verständnis verschiedener Strukturen ist Dualität.

Definition 8. *Die duale Kategorie* $\mathbf{C}^{op}$ *zu einer Kategorie* $\mathbf{C}$ *ist eine Kategorie, die dieselben Objekte wie* $\mathbf{C}$ *hat und in der für alle Paare* $A, B \in Obj(\mathbf{C})$ *gilt* $Hom_{\mathbf{C}^{op}}(A, B) = Hom_{\mathbf{C}^{op}}(B, A)$

Die zu $\mathbf{C}$ und $\mathbf{C}^{op}$ gehörenden Graphen haben also dieselben Knoten, und genauso viele Kanten. Für jeden Pfeil in $\mathbf{C}$ gibt es einen Pfeil mit entgegengesetzter Richtung in $\mathbf{C}^{op}$

Sowohl das Produkt als auch die duale Kategorie sind zunächst rein formale Konstruktionen. Man betrachte $\mathbf{SET}$. Sei A die Menge der Buchstaben eines Alphabets und $v : A \to \{0, 1\}$ sei die Funktion, die die Konsonanten auf 0 und die Vokale auf die 1 abbildet. In $\mathbf{SET}^{op}$ ist v ein Pfeil von $\{0, 1\}$ nach A, aber keine Funktion.

Nichtsdestotrotz ist es in einigen Fällen möglich zu zeigen, dass das Duale einer Kategorie „grundsätzlich dasselbe“ wie eine anderen wohlbekannten Kategorie ist. So kann man von $\mathbf{FIN}$ der Kategorie der endlichen Mengen und Funktionen sagen, dass sie äquivalent zum Dual der Kategorie der endlichen Boolschen Algebren mit Homomorphismen als Pfeilen ist. Dabei ist ein Homomorphismus von Booleschen Algebren eine monotone Funktion, die $\cap, \cup, \top, \bot$ und $\neg$ erhält. Für $h : B \to B'$ muss also gelten: $f(\top_B) = \top_{B'}$, $f(x \cap y) = f(x) \cap f(y)$ usw. Mit $\mathbf{FBoole}$ wollen wir die Kategorie der endlichen Booleschen Algebren bezeichnen. Dass $\mathbf{FIN}$ „grundsätzlich dasselbe“ ist wie $\mathbf{FBoole}^{op}$ zeigen wir, indem wir zwei Graphenhomomorphismen F und G konstruieren, die den zu $\mathbf{FIN}$ gehörenden Graphen auf den Graphen von $\mathbf{FBoole}$ abbilden und umgekehrt. Dabei werden F und G so gewählt, dass ihre Zusammensetzungen $F \circ G$ und $G \circ F$ die identischen Graphenhomomorphismen sind. Wir definieren $F : \mathbf{FIN}^{op} \to \mathbf{FBoole}$ durch $F(M) = \mathcal{P}(M)$ als Knotenabbildung und $F(f)(A) = f^{-1}(A)$, wobei M eine endliche Menge, $f : M \to N$ eine Funktion und $A \subseteq N$ eine Teilmenge von N(also ein Element von $\mathcal{P}(N)$ ist. Im

$$F: \quad \mathbf{FIN}^{op} \longrightarrow \mathbf{FBoole}$$

$$\begin{tikzcd} M \arrow[d, "f"'] & \mathcal{P}(M) \\ N & \mathcal{P}(N) \arrow[u, "F(f)"'] \end{tikzcd}$$

folgenden Diagramm beachte man vor allem die Richtung der Pfeile: Man überzeugt sich leicht davon, dass $F(f)$ ein Algebrahomomorphismus ist(Übungsaufgabe).

Um den Graphenmorphismus G zu konstruieren brauchen wir noch ein paar Begriffe und Fakten aus der universellen Algebra. Als *Atome* einer Booleschen Algebra B bezeichnet man die Elemente $a \in B$, für die kein Element b von B existiert mit $\perp < b < a$. Die Menge aller Atome von B wird gewöhnlich mit $Atom(b)$ bezeichnet. In endlichen Booleschen Algebren lässt sich jedes Element b darstellen als Supremum aller darunter liegenden Atome, insbesondere $\top_B = \bigcup_a \in Atom(B)a$. Für jeden Algebrahomomorphismus $h : B \to B'$ gilt $\bigcup_a \in Atom(B)h(a) = \top_{B'}$. Außerdem haben wir $h(a_1) \cap h(a_2) = \perp_{B'}$. Falls $b' \in Atom(B')$ dann gibt es ein eindeutig bestimmtes Atom a mit $b' \leq h(a)$. Wir definieren nun $G : \mathbf{FBoole}^{op} \to \mathbf{FIN}$. Für die Knoten(Algebren) sei $G(B) = Atom(B)$. Für jeden Homomorphismus $h : B \to B'$ sei $G(h) : Atom(B') \to Atom(B)$ definiert durch $G(h)(a') = a$ mit $a' \leq h(a)$ da das a eindeutig bestimmt ist, ist $G(h)$ offenbar eine Funktion, also ein Pfeil in **FIN**.

Dass die Zusammensetzung $G \circ F$ tatsächlich einen identischen Graphenhomomorphismen liefert, beweist man mit dem Satz von Stone. Dieser besagt nämlich gerade (in seiner endlichen Variante), dass sich jede Boolesche Algebra darstellen lässt als Potenzmengenalgebra über ihren Atomen. Die andere Richtung sieht man, wenn man sich überlegt, dass die Atome einer Potenzmengenalgebra $\mathcal{P}(A)$ gerade die Elemente von A sind. Damit ist $G(\mathcal{P}(A)) = A$ und wegen $F(A) = \mathcal{P}(A)$ folgt daher $F \circ G(\mathcal{P}(A)) = \mathcal{P}(A)$.

2.3 Funktionale Programmiersprachen als Kategorien

Das Interesse von Informatikern an Kategorientheorie in den letzten Jahrzehnten rührt zum Teil daher, dass man gesehen hat, dass getypte funktionale Programmiersprachen durch die verwendeten typischen Konstruktionen einer Kategorie ziemlich ähnlich sehen. Die Tatsache, dass Deduktionssysteme im Wesentlichen Kategorien sind, hat ebenso dazu beigetragen. Mit Deduktionssystemen werden wir uns in Abschnitt 7.1 beschäftigen, hier soll es erst einmal um funktionale Sprachen gehen. Wir beschränken uns hier auf funktionale Sprachen im Sinne von Funktionen-Ebenen Programmierung nach Backus[**?**]. Im Gegensatz zum imperativen Ansatz, bei dem gegebene Programme auf Werte angewendet werden, um eine Folge von Werten zu produzieren, die im gewünschten Zielwert endet, werden im funktionalen Ansatz Programmbildungsoperationen auf gegebene Programme angewendet, um eine Folge von Programmen zu erzeugen, die im gewünschten Zielprogramm kulminiert. In einer solchen Sprache gibt es atomare Typen und Operationen sowie Konstruktoren, mit deren Hilfe man kompliziertere Typen und Operationen aufbauen kann. Was eine *rein* funktionale Sprache nicht hat, sind Variablen und Wertzuweisungen. Die Kategorie $\mathbf{C}_L$ zu einer rein funktionalen Sprache L sieht aus wie folgt:

FPC-1 Die Typen von L sind die Objekte von $\mathbf{C}_L$,

FPC-2 einfache und abgeleitete Operationen sind Pfeile von $\mathbf{C}_L$[7],

FPC-3 Quelle und Ziel eines Pfeils sind die Ein-und Ausgabetypen einer Operation,

FPC-4 Komposition wird durch einen Kompositionskonstruktor gegeben,

FPC-5 für ein Objekt A ist die Identität id_A die leere Operation mit Ein-und Ausgabe vom Typ A, die alle Daten unverändert lässt.

Fügt man zu L einen Typ 1 hinzu mit der Eigenschaft, dass es für jeden Typ A eine Operation mit Eingabe vom Typ A und Ausgabe vom Typ 1 gibt, dann kann man die Konstanten eines Typs A als Pfeile $1 \to A$ beschreiben.

Beispiele 5. *Eine einfache Sprache soll vier Typen haben: NAT, $BOOLE$, $CHAR$ und 1 wobei NAT eine Konstante $0 : 1 \to NAT$ und eine Operation $succ : NAT \to NAT$ hat. Für Boolean haben wir zwei Konstanten $\top, \bot : 1 \to BOOLE$ und eine Operation $\neg : BOOLE \to BOOLE$ mit $\neg\top = \bot$ und $\neg\bot = \top$. $CHAR$ soll für jedes Zeichen c eine Operation $c : 1 \to CHAR$ haben. $chr : NAT \to CHAR$ (modulo Anzahl der Zeichen) und $ord : CHAR \to NAT$ sind zusätzliche Operationen mit $chr \circ ord = id_{CHAR}$. Dabei steht chr für die Operation, die zu jeder natürlichen Zahl n das n-te Zeichen ausgibt und ord für die Operation, die einem Zeichen seine Nummer zuweist. Das Programm $NEXT$ wird definiert als Komposition $chr \circ succ \circ ord$ und berechnet das nächste Zeichen in der Ordnung. Damit die Komposition wohldefiniert ist, muss, wann immer die Zusammensetzung von zwei Basisoperationen gleich ist, deren Zusammensetzung mit anderen Programmen gleich sein.*

2.4 Monomorphismen und Teilobjekte

Eine Funktion $f : A \to B$ in **SET** ist genau dann injektiv, wenn für beliebige Elemente $x, y \in A$ gilt, wenn $x \neq y$, dann ist $f(x) \neq f(y)$. Ein Monomorphismus ist ein spezieller Typ von Pfeilen in einer Kategorie, der das Konzept von injektiven Funktionen verallgemeinert. Insbesondere sind die Monomorphismen in **SET** gerade die injektiven Funktionen. Die Definition von Monomorphismen in einer Kategorie sieht fast genauso aus wie die von injektiven Funktionen, außer das das Konzept eines Elements dort nicht sinnvoll ist.

Definition 9. *Seien $A, B \in Obj(\mathbf{C})$ Objekte von $\mathbf{C}$. Ein Pfeil $h \in Hom_{\mathbf{C}}(A, B)$ heißt Monomorphismus, gdw. $h \circ g = h \circ f$ impliziert $g = f$.*

[7]Für die Definition mehrstelliger Operationen braucht man zusätzlich Typkonstruktoren, die aus atomaren Typen mehrstellige Produkte aufbauen.

Um zu kennzeichnen, dass f ein Monomorphismus ist, schreibt man $f : A \rightarrowtail B$. In der obigen Definition und vielen ähnlichen wird das Konzept eines Elements von A durch einen beliebigen Pfeil nach A ersetzt. Ein Pfeil $a : T \to A$ heißt auch variables Element von A paramentrisiert über T. Wenn a als variables Element betrachtet wird kann man auch $f(a)$ für $f \circ a$ schreiben. Mit dieser Schreibweise sieht die Definition von Monomorphismen wieder aus wie die in Mengen, nämlich f ist Monomorphismus, wenn für beliebige variable Elemente $x, y : T \to A$ gilt, falls $x \neq y$, dann ist $f(x) \neq f(y)$. Der folgende Satz besagt, dass die Monomorphismen in **SET** tatsächlich gerade die injektiven Funktionen sind.

Satz 1. *Eine Funktion in* **SET** *ist ein Monomorphismus, genau dann, wenn sie injektiv ist.*

Beweis: Sei $f : A \to B$ injektiv, und seien $a, a' : T \to A$ variable Elemente von A. Falls $a \neq a'$, dann gibt es ein Element $t \in T$ mit $a(t) \neq a'(t)$. Dann ist aber auch $f(a(t)) \neq f(a'(t))$, also ist $f \circ a \neq f \circ a'$. Sei umgekehrt f ein Monomorphismus. Für jedes Element $a \in A$ gibt es eine eindeutig bestimmte Funktion $a : \{*\} \to A$ von $\{*\}$ nach A, nämlich die, die a als Funktionswert hat. Für zwei solcher Pfeile a, a' folgt aus $a \neq a'$, dass $f \circ a \neq f \circ a'$, was nichts anderes heißt, als dass f injektiv ist. □

Beispiele 6. *1. In den bekanntesten Kategorien von Mengen mit zusätzlicher Struktur und Funktionen, die diese Struktur erhalten, z.B.* **Mon**, *sind die Monomorphismen gerade die injektiven Homomorphismen.*

2. In einer Kategorie, die durch eine Halbordnung definiert wird, ist jeder Pfeil ein Monomorphismus.

3. Betrachten wir die Kategorie **REL**. *Eine Relation $R : A \to B$ ist ein Monomorphismus, genau dann, wenn folgende Bedingung erfüllt ist.*

(*) *Wenn U und V dieselben Bilder unter R haben, dann ist $U = V$ für beliebige Teilmengen $U, V \subseteq A$*[8].

Dabei ist das Bild UR von U unter R definiert als $UR = \{b \in B \mid \exists a \in U \mid aRb$. Um zu beweisen, dass (*) notwendig und hinreichend ist, nehmen wir zuerst an, dass R die Bedingung (*) erfüllt und zeigen, dass R ein Monomorphismus ist. Seien $S, T \subseteq C \times A$ Relationen, mit $R \circ S = R \circ T$, dann gilt für jedes $c \in C$ auch $cRT = cRS$. Da R die Bedingung (*) erfüllt, folgt daraus, dass $cS = cT$. Da das für jedes c gilt, folgt daraus auch, dass $S = T$.

Umgekehrt nehmen wir an, dass R die Bedingung nicht erfüllt. Es gibt also zwei verschiedene Mengen $U, V \subseteq A$ mit $UR = VR$. Wir definieren Relationen $S, T \subseteq \{1\} \times A$ durch $1Ta$ genau dann, wenn $a \in U$ und $1Sa$ genau dann, wenn $a \in V$. Da 1 das einzige Element von $\{1\}$ ist, sind damit die Relationen vollständig definiert. Offenbar ist $S \neq T$, falls $U \neq V$. Da aber $1(R \circ S) = UR = VR = 1(R \circ T) = VR$, gilt $R \circ S = R \circ T$. Also kann R kein Monomorphismus sein.

Aufgrund der Definition heißen monomorphe Pfeile auch links kürzbar. Das folgende Theorem stellt zwei elementare Eigenschaften von Monomorphismen fest.

[8] Mit anderen Worten: die durch R induzierte Funktion von $\mathcal{P}(A)$ nach $\mathcal{P}(B)$ ist injektiv.

Satz 2. *Seien $f : A \to B$ und $g : B \to C$ zwei Pfeile in einer Kategorie* $\mathbf{C}$. *Dann gilt:*

1. *Falls f und g Monomorphismen sind, dann ist auch $g \circ f$ ein Monomorphismus.*
2. *Wenn $g \circ f$ ein Monomorphismus ist, dann ist auch f ein Monomorphismus.*

Beweis: Übungsaufgabe. □

Teilobjekte sind die kategoriale Analogie zum Konzept einer Teilmenge in der Mengenlehre. Für jede Teilmenge ist die Einbettung in die Obermenge eine Injektion. Mit Monomorphismen habe wir in Kategorien ein passendes Äquivalent zu injektiven Mengenabbildungen. Darum liegt es nahe, Teilobjekte als Monomorphismen zu verstehen. Da es viele verschiedene injektive Abbildungen gibt, mit denen man eine Teilmenge einbetten kann, ist es notwendig, eine vernünftige Äquivalenzrelation zu definieren, die von solchen Unterschieden abstrahiert.

Sei $\mathbf{C}$ eine Kategorie und $f : A \rightarrowtail C$ und $g : B \rightarrowtail C$ zwei Monomorphismen mit Ziel C. Wir sagen f faktorisiert über g und schreiben $f \leq g$ genau dann, wenn es einen Pfeil $h : A \to B$ gibt mit $f = g \circ h$. Nach Satz 2, muss dieser Pfeil ebenfalls ein Monomorphismus sein. Falls sowohl $f \leq g$ als auch $g \leq f$ schreibt man auch $f \cong g$. Die definierte Relation ist eine Äquivalenzrelation auf der Menge der Monomorphismen mit demselben Ziel. Die entstehenden Äquivalenzklassen betrachten wir als Teilobjekte.

Definition 10. *Sei $A \in Obj(\mathbf{C})$. Ein Teilobjekt $[f]$ von A ist die Klasse der zu $f : X \to A$ bezüglich $\cong$ äquivalenten Pfeile. Ein Teilobjekt heißt echt, wenn es id_A nicht enthält.*

Wenn keine Verwechslungsgefahr besteht, schreibt man statt $[f]$ auch einfach nur f, d.h. man unterscheidet nicht zwischen der Klasse und ihren Repräsentanten.

Anders als der normale Teilmengenbegriff erhält der Teilobjektbegriff die Struktureigenschaften einer Kategorie. Auf diese Art ist ein Teilobjekt einer Gruppe(in der Kategorie der Gruppen) eine Untergruppe, da als Monomorphismen nur Gruppenhomomorphismen auftreten.

Manchmal werden wir auch den Ausdruck „A ist ein Teilobjekt von B“ verwenden und damit meinen, dass eine bestimmte Äquivalenzklasse von Monomorphismen mit Ziel B insbesondere einen Monomorphismus von A nach B enthält. Man muss dabei allerdings beachten, dass noch andere Monomorphismen von A nach B existieren können, die nicht zu derselben Äquivalenzklasse gehören.

2.5 Andere Typen von Pfeilen

Betrachten wir nun das Dual zu Monomorphismen: die Epimorphismen.

Definition 11. *Seien $A, B \in Obj(\mathbf{C})$ Objekte von* $\mathbf{C}$. *Ein Pfeil $h \in Hom_{\mathbf{C}}(A, B)$ heißt Epimorphismus*[9]*, gdw. aus $g \circ h = f \circ h$ folgt, dass $g = f$.*

Um einen Epimorphismus zu kennzeichnen, schreiben wir $A \twoheadrightarrow B$.

[9]von griechisch $\varepsilon\pi\iota$ epi- gegen und griechisch $\mu o\varrho\phi\eta$ morph Gestalt, Form

Satz 3. *Eine Pfeil in* **SET** *ist genau dann ein Epimorphismus, wenn er eine surjektive Funktion ist.*

Beweis: Sei $f : A \to B$ surjektiv und $g, h : B \to C$ seien zwei Funktionen. Falls $g \neq h$, dann gibt es ein Element $b \in B$ mit $g(b) \neq h(b)$. Da f surjektiv ist, gibt es ein Element $a \in A$ mit $f(a) = b$. Dann ist aber $g \circ f(a) \neq h \circ f(a)$.

Sei f nicht surjektiv, dann gibt es ein $b \in B$, das kein Bild unter f ist. Wir definieren zwei Funktionen $g, h : B \to \{0, 1\}$ mit $g(x) = h(x)$ für alle $x \in B, x \neq b$ und $g(b) = 1$ und $h(b) = 0$. Offenbar ist $g \circ f = h \circ f$ obwohl $h \neq g$, also ist f kein Epimorphismus. □
Aufgrund der Definition heißen epimorphe Pfeile auch rechts kürzbar.

Anders als beim Verhältnis von Monomorphismen zu injektiven Funktionen, gibt es für Epimorphismen und surjektive Funktionen in Kategorien mit zusätzlicher Struktur keine Äquivalenz. Mit anderen Worten, es gibt Epimorphismen, die nicht surjektiv sind.

Beispiele 7. *In der Kategorie* **Mon** *der Monoide ist die Einbettungsabbildung $incl : \mathbb{N} \to \mathbb{Z}$ der nichtnegativen ganzen Zahlen in die ganzen Zahlen zwar ein Epimorphismus, aber nicht surjektiv. Seien f und g zwei Pfeile mit $f \circ incl(n) = g \circ incl(n)$ für alle n. Offenbar ist $f(z) = g(z)$ für alle nichtnegativen z. Sei n eine positive Zahl, dann gilt $f(0) = f(n + (-n)) = f(n) + f(-n) = g(n) + g(-n) = g(n + (-n)) = g(0)$ da f und g als Pfeile in* **Mon** *Monoidhomomorphismen sind. Wegen $f(n) = g(n)$ folgt daraus $f(-n) = g(-n)$ und damit auch $f = g$.*

Definition 12. *Sei* **C** *eine Kategorie. Ein Pfeil $f : A \to B$ heißt Isomorphismus, wenn es einen inversen Pfeil $g : B \to A$ gibt, mit $f \circ g = id_B$ und $g \circ f = id_A$. Wenn nur die erste Gleichung erfüllt ist, dann heißt g rechtes Inverses von f. Wenn nur die zweite Gleichung erfüllt ist, dann heißt g linkes Inverses von f.*

In **SET** ist ein Pfeil der sowohl Monomorphismus als auch Epimorphismus ist, eine Bijektion und damit auch ein Isomorphismus. In einer beliebigen Kategorie gilt das nicht. Der Pfeil *incl* aus Beispiel 7 ist sowohl Epimorhismus als auch Monomorphismus, aber *incl* hat kein rechtes Inverses.

Definition 13. *Sei f ein Pfeil mit einem rechten Inversen g. Dann heißt f Splitepimorphismus und g heißt Splitmonomorphismus.*

Ein Splitepimorphismus f ist tatsächlich ein Epimorphismus: Aus $h \circ f = k \circ f$ folgt nämlich $h = h \circ f \circ g = k \circ f \circ g = k$, was der Definition für Epimorphismen entspricht. Auf die gleiche Weise zeigt man auch, dass ein Splitmonomorphismus ein Monomorphismus ist.

Bemerkungen Unter Annahme der üblichen Axiome der Mengentheorie ist jede surjektive Funktion $f : A \to B$ in **SET** ein Splitepimorphismus. Um das zu zeigen, konstruiere man das rechte Inverse g wie folgt: man wähle für jedes $b \in B$ ein Element $a \in A$ mit $f(a) = b$ und definiere $g(b) = a$. Wegen der Surjektivität muss ein solches a immer existieren. Das Auswahlaxiom sichert, dass man immer ein a wählen kann. Tatsächlich ist eine mögliche Formulierung des Auswahlaxioms, dass jeder Epimorphismus ein Splitepeimorphismus ist.

Die nächsten Definitionen sind hauptsächlich inspiriert vom Begriff der Reduzierbarkeit in der Rekursionstheorie. Sie spielen eine besondere Rolle in der Kategorie **EN** der aufzählbaren Mengen.

Definition 14. *Seien $A, B \in Obj(\mathbf{C})$.*

- *Ein Pfeil $h \in Hom_{\mathbf{C}}(A, B)$ heißt Hauptpfeil(principal), falls für alle $f \in Hom_{\mathbf{C}}(A, B)$ ein Pfeil $g \in Hom_{\mathbf{C}}(A, A)$ existiert mit $f = h \circ g$.*

- *Ein Retraktionspaar ist ein Paar von Pfeilen $f \in Hom_{\mathbf{C}}(A, B)$ und $g \in Hom_{\mathbf{C}}(B, A)$ mit $g \circ f = id_A$. A heißt Retrakt von B via (f, g), geschrieben als $(A < B)$.*

Beispiele 8. *Die Kategorie* **EN** *der aufzählbaren Mengen hat als Objekte $\underline{A} = (A, e_A)$ Paare von abzählbaren Mengen zusammen mit einer surjektiven Funktion $e_A : \mathbb{N} \to A$. Die Pfeile $f : \underline{A} \to \underline{B}$ in* **EN** *sind Funktionen $f : A \to B$, für die eine total rekursive Funktion $f' : \mathbb{N} \to \mathbb{N}$ existiert, für die gilt: $f \circ e_A = e_B \circ f'$. Das Objekt $\underline{\mathbb{N}}$ ist das Paar $(\mathbb{N}, id)$, das Objekt $\underline{PR} = (PR, \varphi)$ ist die Menge der partiell rekursiven Funktionen mit einer Aufzählung (Gödelnumerierung φ). Ein Pfeil $f \in Hom_{\mathbf{EN}}(\underline{N}, \underline{PR})$ ist eine Funktion $f : \mathbb{N} \to \mathbf{PR}$, für die eine total rekursive Funktion $s : \mathbb{N} \to \mathbb{N}$ existiert mit $f(x)(y) = \varphi_{s(x)}(y)$. f ist ein Hauptpfeil, wenn $f(0) = f_0, f(1) = f_1, \ldots$ eine akzeptable Gödelnumerierung ist, denn dann gibt es für jedes $g \in Hom_{\mathbf{EN}}(\underline{N}, \underline{PR})$ eine total berechenbare Funktion $t : \mathbb{N} \to \mathbb{N}$mit $g(x)(y) = f_{t(x)}(y)$.*

Satz 4. *Sei* **C** *eine Kategorie mit Objekten A, B, dann gilt:*

1. *falls $A < B$ via (f, g), dann ist g ein Epimorphismus und ein Hauptpfeil und f ist ein Monomorphismus,*

2. *falls $h \in Hom_{\mathbf{C}}(A, B)$ ein Hauptpfeil ist und es einen Epimorphismus k in $Hom_{\mathbf{C}}(A, B)$ gibt, dann ist h ein Epimorphismus,*

3. *falls $A < B$ und $g \in Hom_{\mathbf{C}}(B, A)$ ein Hauptpfeil ist, dann existiert ein $f \in Hom_{\mathbf{C}}(A, B)$, so dass $A < B$ via (f, g) ist.*

Beweis:

1. Zuerst zeigen wir, dass g ein Epimorphismus ist: Seien $h, k \in Hom_{\mathbf{C}}(A, C)$ und $h \circ g = k \circ g$. Dann $h \circ g \circ f = k \circ g \circ f$, also $h = k$. Wir beweisen, dass g ein Hauptpfeil ist: wegen $g \circ f = id_A$ gilt für jedes $h : B \to A$ auch $g \circ f \circ h = h$, damit existiert ein $\tilde{h}$ mit $g \circ \tilde{h} = h$, nämlich $\tilde{h} = f \circ h$(siehe Abbildung 4).
   Um zu zeigen, dass f ein Monomorphismus ist, nehmen wir an, wir hätten Pfeile j, k mit $f \circ j = f \circ k$. Dann gilt auch $g \circ f \circ j = g \circ f \circ k$ und damit $j = k$.

2. Sei k ein Epimorphismus und seien i, j Pfeile mit $i \circ h = j \circ h$. Da h Hauptpfeil ist, existiert ein $\tilde{k}$ mit $k = h \circ \tilde{k}$ und $i \circ h \circ \tilde{k} = j \circ h \circ \tilde{k}$. Da k ein Epi ist folgt daraus $i = j$.

3. Sei (i, j) das Paar für $A < B$. Wenn g Hauptpfeil ist, existiert ein $\tilde{j}$ mit $j = g \circ \tilde{j}$. Man definiere f als $f = \tilde{j} \circ i$. Dann gilt $g \circ f = g \circ \tilde{j} \circ i = j \circ i = id_A$.

Abbildung 4:

$$
\begin{tikzcd}
A \arrow[r, bend left, "f"] & B \arrow[l, bend left, "g"] \\
B \arrow[u, "h"] \arrow[ur, dotted, "\tilde{h}"'] &
\end{tikzcd}
$$

$\square$

Aufgaben:

1. Zeigen Sie, dass in der Kategorie der Monoide jeder injektive Homomorphismus ein Monomorphismus ist und umgekehrt.

3 Konstruktionen in Kategorien

In diesem Kapitel geht es um kategoriale Konstruktionen. Das sind Verallgemeinerungen von solchen Konstruktionen, wie sie uns aus der Mengenlehre vertraut sind, etwa die leere Menge als kleinste Menge oder das Produkt zweier Mengen. Kategoriale Konstruktionen werden meist über sogenannte Universalitätseigenschaften definiert. In der Definition wird ein Objekt beschrieben, das eine gewisse Eigenschaft hat und das unter allen Objekten mit dieser Eigenschaft etwas Besonderes (z.B. das „kleinste“) ist. Eine solche Konstruktion ist uns schon begegnet, nämlich der Hauptpfeil. Ein Hauptpfeil ist ein Pfeil eines bestimmten Typs (d.h. mit bestimmter Quelle und Ziel), dessen Besonderheit darin besteht, dass alle anderen Pfeile dieses Typs sich als Zusammensetzung aus diesem Pfeil und einem weiteren (eindeutig bestimmten) beschreiben lassen.

3.1 Initiale und terminale Objekte

Eine der einfachsten kategorialen Konstruktionen ist das Konzept des Initialobjektes.

Definition 15. *Sei* $\mathbf{C}$ *eine Kategorie. Ein Objekt* 0 *heißt Initialobjekt oder initial, wenn es für jedes* $A \in Obj(\mathbf{C})$ *einen eindeutig bestimmten Pfeil* $init_A : 0 \to A$ *gibt.*

Beispiele 9. 1. *In der Kategorie der Mengen ist die leere Menge* $\emptyset$ *initial mit der leeren Funktion als Pfeil in jede Menge.*

2. *Für eine Signatur* Σ *(Menge von Funktionssymbolen zusammen mit zugehörigen Stelligkeiten)*[10] *sei* $\mathbf{Alg_\Sigma}$ *die Kategorie der* Σ*-Algebren mit den zugehörigen* Σ*-Homomorphismen als Pfeilen. Eine* Σ*-Algebra* $\mathbf{S}$*ist ein Paar*

$$\mathbf{S} =< S, \{f_{\mathbf{S}}^{(n)} : S^n \to S \mid f^{(n)} \in \Sigma\} >,$$

wobei S *eine Menge (die sogenannte Trägermenge oder auch Träger von* $\mathbf{S}$*) ist. Wenn die Stelligkeit eines Funktionssymboles* c *die* 0 *ist, dann heißt* c *auch Konstante. Die Interpretation einer Konstanten ist ein Element von* S*. Die Kategorie* $\mathbf{Alg_\Sigma}$ *hat ein Initialobjekt* $\mathbf{T_\Sigma}$*, das auch* Σ*-Wortalgebra oder Herbrand-Universum von* Σ *heißt. Dabei ist die Trägermenge* $T_{\Sigma,s}$ *der Initialen Algebra über der Sorte* $s \in S$ *die Menge der wohlgeformten Ausdrücke, d.h. die Menge aller endlichen Bäume, deren Knoten mit Operationen* f *aus* Σ *benannt werden, so dass jeder Knoten so viele Unterbäume hat, wie die Stelligkeit von* f *angibt. Jede andere* Σ*-Algebra ist eine Interpretation von* T_Σ *und die initiale Abbildung ist die Interpretationsfunktion. Die Trägermenge der Termalgebra für* $(f^{(0)}, f^{(1)})$ *über der Sorte* s *besteht aus* $\{c, f(c), f(f(c)), \ldots\}$*. Wenn* Σ *durch eine kontextfreie Grammatik hergeleitet wurde, d.h. wenn die Sorten Nichtterminale sind und Operationssymbole die Produktionen sind, dann ist* $T_{\Sigma,s}$ *die Menge aller Ableitungsbäume aus dem Nichtterminal* s*. Der Term abb entsteht aus der Grammatik* G *mit* $P_G = \{S \to A, A \to AB|a, B \to b\}$ *durch* $s(ab(ab(a,b),b))$*. Die Signatur der Grammatik ist* $(a^{(0)}, b^{(0)}, s^{(1)} : A \to S, ab^{(2)} : A \times B \to A)$*.*

Lemma 1. *In der Kategorie der Algebren mit der Signatur* $\Sigma = (f^{(0)}, f^{(1)})$ *sind die Natürlichen Zahlen* $< \mathbb{N}, (zero, succ) >$ *initial.*

[10] Gruppen habe zum Beispiel die Signatur $\{f^{(2)}, f^{(0)}\}$

Beweis: *Sei* $< C, (c_0, c_1) >$ *eine beliebige* Σ*-Algebra. Sei* $f : \mathbb{N} \to C$ *eine Funktion, die das folgende Diagramm kommutativ macht.*

Abbildung 5: $\mathbb{N}$ ist initial

$$\begin{tikzcd}
1 \arrow[r, "zero"] \arrow[dr, "c_0"'] & \mathbb{N} \arrow[d, "f"] & \mathbb{N} \arrow[l, "succ"'] \arrow[d, "f"] \\
 & C & C \arrow[l, "c_1"]
\end{tikzcd}$$

Dann gilt $f \circ zero = c_0$ *und* $f \circ succ = h \circ f$*. Diese Gleichungen haben genau eine Lösung, nämlich* f *mit* $f : n \mapsto c_1^n(c_0())$*. Damit haben wir eine Initialabbildung.* □

3. *In der Kategorie der Halbgruppen ist die leere Halbgruppe ein Initialobjekt. In der Kategorie der nichtleeren Halbgruppen gibt es kein Initialobjekt. Das beweist man in dem man die Halbgruppe* 2 *betrachtet, die nur aus zwei Elementen* a *und* e *besteht, wobei* $e * a = a * e = a * a = a$ *und* $e * e = e$*. Für jede Halbgruppe* H *gibt es zwei mögliche Pfeile (Homomorphismen) von* H *nach* 2*: den, der alle Elemente auf* e *abbildet und den, der alle Elemente auf* a *abbildet. Daher kann es keine nichtleere Halbgruppe geben, die initial ist.*

Satz 5. *Je zwei initiale Objekte* 0 *und* $0'$ *sind isomorph.*

Beweis: Seien $init_{0'} : 0 \to 0'$ und $init_0 : 0' \to 0$ die initialen Pfeile. Dann haben wir zwei Pfeile $ini_0 \circ init_{0'}$ und $id_0 \in Hom_{\mathbf{C}}(0, 0')$. Wegen der Eindeutigkeit der Initialabbildung muss gelten $ini_0 \circ init_{0'} = id_0$. Die andere Richtung beweist man analog. □

Wie wir schon bei der Definition des Epimorphismus gesehen haben, ist *Dualität* ein wesentliches Prinzip, um neue Konzepte in der Kategorientheorie herzuleiten. $P(C)$ möge zum Beispiel die Eigenschaft eines Objektes C sein, dass es für jedes Objekt B in der Kategorie genau einen Pfeil $f : C \to B$ gibt. Nach Definition gilt $P(C)$ genau dann, wenn C ein Initialobjekt ist.

Die dazu duale Eigenschaft $P^{op}(C)$ ist dann, dass es für jedes $B \in Obj(\mathbf{C})$ einen eindeutig bestimmten Pfeil $g : B \to C$ gibt. Im allgemeinen wird der Name der dualen Eigenschaft Q^{op} aus dem Namen für Q durch Davorsetzen der Silbe „co- “gebildet. Demnach heißt P^{op} dann co-initial. In einigen Fällen erhalten die Konzepte aber auch eigene Namen. Ein co-initiales Objekt heißt terminal und wird im allgemeinen mit 1 oder dem Buchstaben T bezeichnet; der eindeutig bestimmte Morphismus in das Terminalobjekt wird mit ! bezeichnet.

Da die Eigenschaft ein Isomorphismus zu sein bei Dualisierung erhalten bleibt, erhalten wir die zu Satz 5 duale Aussage:

Satz 6. *Je zwei terminale Objekte* 1 *und* $1'$ *sind isomorph.*

Beispiele 10. 1. *Das einelementige Monoid ist sowohl terminal als auch initial in der Kategorie der Monoide.*

2. *In einer partiell geordneten Menge ist ein Objekt genau dann terminal, wenn es ein absolutes Maximum ist. Da es keine größte natürliche Zahl gibt, hat die Kategorie der natürlichen Zahlen kein Terminalobjekt.*

3. *In der Kategorie der Mengen ist jede Einermenge ein Terminalobjekt.*

Jedes Element x einer Menge A läßt sich darstellen als Bild einer Funktion von einer Einermenge nach A. Tatsächlich gibt es gerade so viele Elemente in A wie Funktionen von $\{*\}$ nach A, oder mit anderen Worten A ist isomorph zu $Hom_{SET}(\{*\}, A)$. Falls $f : A \to B$ eine Funktion und $x : \{*\} \to A$ ein Pfeil ist, dann entspricht die Komposition $f \circ x : 1 \to B$, dem Bild von $x(*)$ unter f. Die Einermengen sind Terminalobjekte in **SET** und wir bezeichnen in einer allgemeinen Kategorie mit Terminalobjekt 1 einen Pfeil $1 \to A$ als Konstante vom Typ A.

In **SET** gibt es für ein Objekt A so viele Konstanten vom Typ A, wie A Elemente hat. in der Kategorie **Mon**, deren Objekte Monoide und deren Pfeile Monoidhomomorphismen sind, gibt es für jedes Objekt M genau eine Konstante vom Typ M, da das Einselement auf das Einselement abgebildet wird. Eine andere Bezeichnung für Konstanten, die aus der Theorie der Garben stammt, ist globales Element von A.

In **SET** gilt das sogenannte Extensionalitätsprinzip. Zwei Funktionen des gleichen Typs sind gleich, wenn sie auf allen Elementen des Definitionsbereiches gleiche Bilder haben. Umgekehrt formuliert: man kann für verschiedene Funktionen ein Element finden, dessen Bilder unter den beiden Funktionen voneinander verschieden sind. Kategorien, in denen man verschiedene Pfeile durch Konstanten des Definitionsbereiches voneinander unterscheiden kann, nennen wir wohlpunktiert.

Definition 16. *Sei* **C** *eine Kategorie.*

1. *Ein* **C***-Objekt T heißt Generator genau dann, wenn für nicht isomorphe Pfeile $f, g \in Hom_{\mathbf{C}}(A, B)$ ein Pfeil $h \in Hom_{\mathbf{C}}(T, A)$ existiert, mit $f \circ h \neq g \circ h$.*

2. **C** *heißt wohlpunktiert oder hat genügend Punkte, wenn es einen Generator in* **C** *gibt, der terminal ist.*

Aufgaben: Zeigen Sie, dass ein Initialobjekt keine echten Teilobjekte hat.

3.2 Produkte und Coprodukte

Die Produktkonstruktion ist eine Verallgemeinerung des kartesischen Produktes auf Kategorien.

Definition 17. *Sei* **C** *eine Kategorie und $A, B \in Obj(\mathbf{C})$. Das Produkt von A und B ist ein Objekt $A \times B$ zusammen mit zwei Morphismen $p_l : A \times B \to A$ und $p_r : A \times B \to B$, so dass für alle Paare $f : C \to A$ und $g : C \to B$ ein eindeutig bestimmter Pfeil $< f, g >: C \to A \times B$ existiert, mit $f = p_l \circ < f, g >$ und $g = p_r \circ < f, g >$.*

Satz 7. *Wenn das Produkt von zwei Objekten existiert, dann ist es (bis auf Isomorphie) eindeutig bestimmt.*

$$\begin{tikzcd}
& A \times B \arrow[dl, "p_l"'] \arrow[d, "< p_l, p_r >"'] \arrow[dr, "p_r"] & \\
A & A \otimes B \arrow[l, "q_l"] \arrow[r, "q_r"'] \arrow[d, "< q_l, q_r >"'] & B \\
& A \times B \arrow[ul, "p_l"] \arrow[ur, "p_r"'] &
\end{tikzcd}$$

Beweis:

Sei $A \otimes B$ mit den Projektionen q_l, q_r ein alternatives Produkt. Dann ist $< q_l, q_r > \circ < p_l, p_r >$ der eindeutig bestimmte Morphismus, der das Diagramm kommutativ macht. Für $id_{A\times B}$ gilt das aber auch, daher ist $< q_l, q_r > \circ < p_l, p_r >= id_{A\times B}$. Aus Symmetriegründen gilt auch $< p_l, p_r > \circ < q_l, q_r >= id_{A\otimes B}$, d.h. $A \times B$ und $A \otimes B$ sind isomorph. □

Aufgaben:

1. Sei $\underline{\mathbb{N}} = (\mathbb{N}, id)$ in **EN**. Dann ist $f \in Hom_{\mathbf{EN}}(\mathbb{N}, \underline{PR})$ genau dann, wenn eine partiell rekursive Funktion g existiert mit $g([n, m]) = f(n)(m)$.

2. Ein Pfeil $f \in Hom_{\mathbf{EN}}(\underline{\mathbb{N}}, \underline{PR})$ ist Hauptpfeil, genau dann, wenn f eine Gödelnumerierung ist.

Satz 8 (Eigenschaften des Produktes). *1. Wenn $A \cong A'$ und $B \cong B'$, dann gilt auch $A \times B \cong A' \times B'$.*

*2. Wenn **C** ein Terminalobjekt enthält, dann gilt (falls die Produkte existieren) für ein **C**-Objekt A, dass $A \cong 1 \times A \cong A \times 1$.*

Das Produkt von mehreren Objekten definiert man als Produkt von Objekten und Produkten. Die Assoziativität einer solchen Konstruktion, nämlich dass $(A \times B) \times C = A \times (B \times C)$ ist nicht schwer zu beweisen und berechtigt dazu, die Klammern wegzulassen.

Definition 18. *Eine Kategorie **C**, die ein Terminalobjekt besitzt und in der für jedes Paar von Objekten A und B das Produkt $A \times B$ existiert, heißt kartesisch.*

Die zum Produkt duale Definition führt zum Coprodukt zweier Objekte.

Definition 19. *Sei **C** eine Kategorie und $A, B \in Obj(\mathbf{C})$. Das Coprodukt von A und B ist ein Objekt $A + B$ zusammen mit zwei Morphismen $i_l : A \to A + B$ und $i_r : B \to A + B$, so dass für alle Paare $f : A \to C$ und $g : B \to C$ ein eindeutig bestimmter Pfeil $[f, g] : A + B \to C$ existiert, mit $f = [f, g] \circ i_l$ und $g = [f, g] \circ i_r$.*

Beispiele 11. *1. In **SET** ist das Coprodukt die disjunkte Vereinigung zusammen mit den Einbettungsabbildungen.*

2. In einer Präordnung P ist das Coprodukt(falls es existiert) die kleinste obere Schranke.

3. Sei **CPO** *die Kategorie der vollständigen partiellen Ordnungen mit stetigen Funktionen bzgl. Scott-Topologie*[11]*. Sei* **CPOS** *deren Teilkategorie mit ausschließlich strikten Funktionen, d.h. Morphismen, die das kleinste Element immer auf das kleinste Element abbilden. Beide Kategorien sind kartesisch. Das Coprodukt in* **CPOS** *ist die Verschmelzungssumme, d.h. die disjunkte Vereinigung, bei der die beiden kleinsten Elemente identifiziert werden.*

$$S + P = \{\perp_{S+P}\} \bigcup_{s \in S \setminus \{\perp_S\}} \{(0, s)\} \bigcup_{p \in P \setminus \{\perp_P\}} \{(1, p)\}$$

In **CPO** *gibt es kein Coprodukt. Das kann man sich folgendermaßen überlegen. Seien $f : S \to Q$ und $g : P \to Q$ zwei stetige Funktionen mit $f(\perp_S) \neq g(\perp_P)$. Wenn man für das Coprodukt die Verschmelzungssumme wählt, dann müsste $in_l(\perp_S) = \perp_{S+P} = in_r(\perp_P)$. Wegen $f(\perp_S) \neq g(\perp_P)$ lässt sich $[f, g](\perp_{S+P})$ nicht definieren.*

3.3 Exponentiale

Für zwei Mengen A und B kann man die Menge B^A definieren durch $B^A = \{f : A \to B\}$. Diese Menge ist dadurch charakterisiert, dass es eine Funktion (nämlich die Auswertungsfunktion) gibt, die jedem Paar (f, a) aus $B^A \times A$ einen Wert in B zuordnet.

$$ev : B^A \times A \to B \text{ wobei } ev(f, a) = f(a)$$

Kategorientheoretisch betrachtet ist dieser Pfeil etwas besonderes, da er eine bestimmte *Universaleigenschaft* hat. Unter allen Pfeilen aus $Hom_{\mathbf{SET}}(C \times A, B)$ für beliebige Objekte C ist er nämlich insofern speziell, als zu jedem (anderen) Pfeil $g \in Hom_{\mathbf{SET}}(C \times A, B)$ ein Pfeil $\tilde{g} : C \to B^A$ existiert, so dass das folgende Diagramm kommutiert. Mit anderen Worten jedes g faktorisiert über ev.

$$\begin{tikzcd}
B^A \times A \arrow[dr, "ev"] & \\
& B \\
C \times A \arrow[uu, dotted, "\tilde{g} \times id_A"] \arrow[ur, "g"'] &
\end{tikzcd}$$

Mit Hilfe von Exponentenmengen B^A lassen sich Funktionen (wie zum Beispiel ev) definieren, die Funktionen als Argumente bekommen.

[11] Die offenen Mengen sind die nach oben abgeschlossenen Mengen, sogenannte upper sets.

Betrachten wir die Situation für aufzählbare Mengen. Sei $\{\varphi_i\}_{i\in\mathbb{N}}$ eine Gödel-Numerierung der Partiell Rekusiven Funktionen und $[,]$ eine Paarungsfunktion. Eine Funktion $f : \mathbb{N}\times\mathbb{N} \to \mathbb{N}$ heißt zweistellige partiell rekursive Funktion, gdw. eine partiell rekursive Funktion f' existiert, mit $f(x,y) = f'([x,y])$. Das $s-m-n$ Theorem besagt, dass eine total berechenbare Funktion g existiert mit $\varphi_{g(x)}(y) = f(x,y)$. Das bedeutet, eine zweistellige Funktion ist berechenbar, falls sie in jedem Argument berechenbar ist und die Funktion $\tilde{f} : x \mapsto f(x,\_)$ berechenbar ist, d.h. wenn eine totale berechenbare Funktion g existiert mit $\varphi_{g(x)} = f(x,\_)$. Anders ausgedrückt: f ist vom Typ $\mathbb{N}\times\mathbb{N} \to \mathbb{N}$ gdw. $\tilde{f} : \mathbb{N} \to (\mathbb{N} \to \mathbb{N})$.

In allgemeinen Kategorien interessieren wir uns für die Existenz eines Morphismus', der dieselbe Aufgabe wie g bzw. $\tilde{f}$ in der Rekursionstheorie erfüllt. Er soll $\Lambda(f)$ heißen.

Definition 20. *Sei* $\mathbf{C}$ *eine kartesische Kategorie und* $A, B \in Obj(\mathbf{C})$. *Das Exponential von A und B ist ein Objekt* B^A *zusammen mit einem Pfeil* $ev_{A,B} : B^A \times A \to B$ *und einer Operation* $\Lambda_C : Hom_{\mathbf{C}}(C \times A, B) \to Hom_{\mathbf{C}}(C, B^A)$ *für jedes Objekt* C *so dass folgendes gilt:*

(i) $ev_{A,B} \circ (\Lambda_C(f) \times id_A) = f$

(ii) $\Lambda_C(ev_{A,B} \circ (h \times id_A)) = h$

Die Betrachtung von B^A als Repräsentanten der Menge $Hom_{\mathbf{C}}(A,B)$ ist gerechtfertigt durch die folgende Überlegung: Für das Terminalobjekt T von $\mathbf{C}$ gelten folgende Isomorphiebeziehungen: $Hom_{\mathbf{C}}(T, B^A) \cong Hom_{\mathbf{C}}(T \times A, B) \cong Hom_{\mathbf{C}}(A,B)$.

Definition 21. *Eine Kategorie* $\mathbf{C}$ *heißt kartesisch abgeschlossen, wenn*

1. *sie kartesisch ist (es existieren alle Produkte und ein Terminalobjekt) und*
2. *für jedes Paar von Objekten* A, B *ein Exponenitial* B^A *existiert.*

Beispiele für kartesisch abgeschlossene Kategorien sind **SET** und **CPO**, die Kategorie der vollständigen partiellen Ordnungen mit stetigen Abbildungen. Für jedes Paar von Objekten A, B ist $Hom_{\mathbf{CPO}}(A,B)$ ebenfalls eine partielle Ordnung, mit der punktweisen Ordnung. Sowohl ev als auch Λ sind mit derselben Definition wie bei **SET** angewendet auf stetige Funktionen wiederum stetig und erfüllen die entsprechenden Bedingungen. Eine für Informatiker interessante Kategorie, die der aufzählbaren Mengen **EN** ist nicht kartesisch abgeschlossen. Man betrachte $\underline{\mathbb{N}} = (\mathbb{N}, id)$. Die Menge $Hom_{\mathbf{EN}}(\underline{\mathbb{N}}, \underline{\mathbb{N}})$ ist offenbar aufzählbar(das ist ja gerade die Menge der partiell rekusiven Funktionen). Gäbe es eine aufzählbare Menge $(\underline{\mathbb{N}}^{\underline{\mathbb{N}}}, \varphi)$ und einen Pfeil ev mit den erforderlichen Eigenschaften, dann wäre $u(x,y) = ev(\varphi(x), y)$ eine universelle Funktion für die total berechenbaren Funktionen, die nicht existiert.

Auch die Kategorie der ω-algebraischen vollständigen partiellen Ordnungen, deren Objekte die vollständigen partiellen Ordnungen sind, die eine abzählbare Menge von kompakten Elementen haben, die alle anderen approximieren und deren Pfeile stetige Abbildungen sind, ist nicht kartesisch abgeschlossen. Allerdings enthält sie einige kartesisch abgeschlossenen Teilkategorien, die eine wesentliche Rolle als Funktionensemantik(denotationale Semantik) von Programmiersprachen und Rekursionstheorie höherer Stufen spielen.

Häufig begegnet man sogenannten *Typstrukturen* in einer kartesisch abgeschlossenen Kategorie. Für $\mathbf{A} \subseteq Obj(\mathbf{D})$ sei $\mathbf{D_A}$ die von $\mathbf{A}$ erzeugte volle kartesisch abgeschlossene Teilkategorie von $\mathbf{D}$, d.h. die kleinste volle Teilkategorie von $\mathbf{D}$, mit $\mathbf{A} \subseteq Obj(\mathbf{D})$ für die gilt, dass wenn $A, B \in Obj(\mathbf{D_A})$, dann sind auch $A \times B$ und A^B Elemente von $\mathbf{D_A}$.

In Abschnitt 14 haben wir den Begriff eines Retraktes eingeführt. Ein Objekt A ist ein Retrakt von B via (i, j) wenn $j \circ i = id_A$. Es ist nicht schwer sich zu überlegen, dass i ein Mono- und j ein Epimorphismus sein muss. Damit ist A ein Teilobjekt von B im Sinne von Definition 10. In **SET** sind Retrakte dasselbe wie nichtleere Teilmengen, da sich immer ein Surjektion von Mengen auf Teilmengen herstellen lässt. In stärker (als **SET**) strukturierten Kategorien führt die Betrachtung von Retrakten als verallgemeinertes Konzept von Teilmengen zu interessanten Beobachtungen. Insbesondere interessieren wir uns für Kategorien, in denen es nichtriviale Objekte A gibt, für die $A^A < A$. Das führt uns zu mathematischen Modellen von Typ-freien Programmiersprachen. In solchen Sprachen werden Programme als Daten betrachtet, bzw. semantisch betrachtet; Exponentiale können in (Quell-oder Ziel-)Objekte zurückgezogen werden.

Zunächst werden wir einige grundlegende Eigenschaften von Retrakten und Exponenten beweisen.

Satz 9. *Sei* $\mathbf{C}$ *eine vollständige kartesisch abgeschlossene Kategorie. Falls* $A < B$ *via* $(vonA, nachA)$ *und* $C < D$ *via* $(vonC, nachC)$*, dann ist* $C^A < D^B$ *via* $(\Lambda(vonC \circ ev \circ (id \times nachA)), \Lambda(nachC \circ ev \circ (id \times vonA)))$.

Beweis:

Um einen Pfeil von C^A nach D^B zu bekommen, konstruieren wir zunächst einen Pfeil $f : C^A \times B \to D$ durch $f := vonC \circ ev \circ (id \times nachA)$ und dessen Bild $\Lambda f : C^A \to D^B$ unter Λ_{C^A}. Analog definiert man einen Pfeil $g : D^B \times A \to C$ durch $g := nachC \circ ev \circ (id \times vonA)$ und den zugehörigen Pfeil $\Lambda g : D^B \to C^A$. Die zugehörigen Diagramme sehen aus wie folgt.

$$
\begin{tikzcd}
 & C^A \times A \arrow[rrrd, "ev_{AC}"] & & & \\
 & D^B \times A \arrow[u, dotted, "\Lambda g \times id_A"'] \arrow[r, "id \times vonA"'] & D^B \times B \arrow[r, "ev_{BD}"'] & D \arrow[r, "nachC"'] & C \\
 & D^B \times B \arrow[u, "id_{D^B} \times nachA"'] \arrow[rrrd, "ev_{BD}"] & & & \\
C^A \times A \arrow[ru, "\Lambda f \times vonA"] \arrow[r, "id_{C^A} \times vonA"'] & C^A \times B \arrow[u, dotted, "\Lambda f \times id_B"'] \arrow[r, "id \times nachA"'] & C^A \times A \arrow[r, "ev_{AC}"'] & C \arrow[r, "vonC"'] & D
\end{tikzcd}
$$

Dass $\tilde{f}$ und $\tilde{g}$ ein Retraktionspaar bilden, verifiziert man wie folgt: Offenbar ist $(\Lambda f \times id_B) \circ (id_{C^A} \times vonA) = \Lambda f \times vonA$. Damit können wir die Zusammensetzung von $\Lambda g \circ \Lambda f$

verstehen als eindeutig bestimmten Pfeil der zu $g \circ (id_B \times nachA) \circ (\Lambda f \times vonA)$. Es folgt:

$$\begin{aligned}
\Lambda g \circ \Lambda f &= \Lambda_{C^A}(nachC \circ ev_{BD} \circ (id_{D^B} \times vonA)) \circ (id_{D^B} \times nachA) \circ (\Lambda_{C^A} f \times vonA)) \\
&= \Lambda_{C^A}(nachC \circ ev_{BD} \circ (id_{D^B} \times vonA)) \circ (\Lambda_{C^A} f \times (nachA \circ vonA)) \\
&= \Lambda_{C^A}(nachC \circ ev_{BD} \circ (id_{D^B} \times vonA)) \circ (\Lambda_{C^A} f \times id_A) \\
&= \Lambda_{C^A}(nachC \circ ev_{BD} \circ (\Lambda_{C^A} f \times vonA)) \\
&= \Lambda_{C^A}(nachC \circ vonC \circ ev_{AC} \circ id_{C^A} \times nachA \circ id_{C^A} \times vonA))
\end{aligned}$$

□

Definition 22. *Sei* **C** *eine kartesisch abgeschlossene Kategorie. Ein Objekt V heißt reflexiv, wenn gilt $V^V < V$.*

Kategorien, in denen es reflexive Objekte gibt, sind deshalb besonders interessant, weil es dort möglich ist, Fixpunktoperatoren zu definieren. Mit Hilfe von Fixpunktoperatoren lassen sich Funktionen rekursiv definieren.
Zunächst beweisen wir eine spezielle Eigenschaft von reflexiven Objekten.

Satz 10. *Sei* **C** *kartesisch abgeschlossen und V ein reflexives Objekt. Dann gibt es Retraktionen $T < V$ und $V \times V < V$*

Beweis: Seien $(nachV, vonV)$ das Retraktionspaar für $V^V < V$.

1. Wir konstruieren einen Pfeil von T nach V. Sei $p_r : T \times V \to V$ die zweite Projektion, dann ist $\Lambda p_r : T \to V^V$. Die Zusammensetzung $\Lambda p_r \circ nachV$ ergibt den gewünschten Pfeil. Dass die Zusammensetzung $! \circ nachV \circ \Lambda p_r = id_T$ liefert, folgt aus der Eindeutigkeit der Terminalabbildung.

2. Um zu zeigen, dass $V \times V < V$ konstruieren wir zuerst eine Retraktion $V \times V < V^V$, durch Zusammensetzung mit dem Paar $(nachV, vonV)$ ergibt sich der Rest. Betrachten wir $vonV$ als Λ, dann gibt uns die Zusammensetzung von $(vonV \times Id)$ mit ev einen Pfeil $app : V \times V \to V$, nämlich

$$app = ev_{VV} \circ (vonV \times id_V).$$

Daraus konstruiert man mit Hilfe von $\alpha_{ABC} : (B \times C) \times A \to (A \times B) \times C)$ durch zweimalige Hintereinanderausführung und anschließende Λ-Abstraktion einen Pfeil $in_1 : (V \times V) \to V^V$

$$in_1 = \Lambda app \circ (app \times id) \circ \alpha$$

Wegen $V^V < V$ und $V < V$ gilt nach Satz 9 auch $V^{V^V} < V^V$ und wir erhalten als zugehöriges Retraktionspaar

$$in_2 = \Lambda(nachV \circ ev \circ (id \times id)) \qquad out_2 = \Lambda(vonV \circ ev \circ (id \times id))$$

.

Aus den beiden Projektionen $p_1, p_2 : V \times V \to V$ erhalten wir durch Λ-Abstraktion Pfeile von $V \times V \to V^V$ und durch Verkettung mit der rechten Projektion $p_r : T \times V \to V$ und anschließende Λ-Abstraktion Pfeile von $T \to V^{V^V}$, nämlich

$$\pi_1 = nachV \circ in_2 \Lambda(\Lambda p_1 \circ p_r) \text{ und } \pi_2 = \Lambda(\Lambda p_2 \circ p_r)$$

Mit Hilfe von π_1 und π_2 definiert man den zweiten Pfeil des Retraktionspaares, $out_1 : V^V \to V \times V$:

$$out_1 =< ev\circ < id, \pi_1 \circ ! >, < ev \circ \pi_2 \circ ! >>$$

Zu zeigen ist, dass $out_1 \circ in_1 = id_{V\times V}$, was äquivalent ist zu

$$p_i \circ out_1 \circ in_1 = id_{V\times V} = p_i \text{ für } i = 1, 2$$

$$\begin{aligned}
& p_i \circ out_2 \circ in_1 = \\
& = ev < id, \pi_i \circ ! > \circ in_1 \\
& = ev < \Lambda app \circ (app \times id) \circ \alpha, \pi_i \circ ! > && \text{(Eindeutigkeit der Terminalabbildung)} \\
& = app \circ (app \times id) \circ \alpha, < id, \pi_i \circ ! > && \text{(Anwendung von } ev) \\
& = app\circ < app\circ < \pi_i \circ !, p_1 >, p_2 > && \text{(Anwendung von } \alpha \text{ und Auflösung von } \times) \\
& = app\circ < ev \circ (vonV \times id_V) \circ < nachV \circ in_2 \circ \Lambda(\Lambda p_i \circ p_r) \circ !, p_1 >, p_2 > \\
& && \text{(Definition von } app \text{ und } \pi_i) \\
& = app\circ < ev < in_2 \circ \Lambda(\Lambda p_i \circ p_r) \circ !, p_1 >, p_2 > \\
& && (vonV \circ nachV = id_V) \\
& = app\circ < ev < \Lambda(nachV \circ ev \circ (id \times id)) \circ \Lambda(\Lambda p_i \circ p_r) \circ !, p_1 >, p_2 > \\
& && \text{(Definition von } in_2) \\
& = app\circ < ev < \Lambda(nachV \circ ev \circ (\Lambda(\Lambda p_i \circ p_r) \times id)) \circ !, p_1 >, p_2 > \\
& && \text{(Reinziehen von } \Lambda) \\
& = app\circ < ev < \Lambda(nachV \circ (\Lambda p_i \circ p_r)) \circ !, p_1 >, p_2 > && \text{(Auswerten mit } ev) \\
& = app \circ (nachV \circ (\Lambda p_i \circ p_1)), p_2 > && (?) \\
& = ev_{VV} \circ (vonV \times id_V) \circ (nachV \circ (\Lambda p_i \circ p_1)), p_2 > \\
& && \text{(Auflösen von } app) \\
& = ev_{VV}(\Lambda p_i \times id_V) && \text{(Retraktionspaar)} \\
& = p_i
\end{aligned}$$

□

3.4 Beispiele kartesisch abgeschlossener Kategorien

Beispiele 12. *1. Eine Boolesche Algebra B ist eine partielle Ordnung. Die zugehörige Kategorie $\mathbf{C}(B)$ ist kartesisch abgeschlossen.*

- *Das Terminalobjekt ist 1, also das größte Element von B.*
- *Da B für je zwei Elemente ein Infimum besitzt, hat $\mathbf{C}(B)$ Produkte.*
- *Das Exponential von $a, b \in B$ ist $b^a = \neg a \vee b$. Um zu zeigen, dass $ev : b^a \times a \to b$ existiert, müssen wir zeigen, dass $(\neg a \vee b) \wedge a \leq b$. Nach dem Distributivgesetz gilt $(\neg a \vee b) \wedge a = (\neg a \wedge a) \vee (b \wedge a) = b \wedge a \leq b$.*

- *Die Existenz von Λf für jedes $f : c \times a \to b$, erfordert zu zeigen, dass $c \wedge a \leq b$ gdw. $c \leq (\neg a \vee b)$. Dazu berechnen wir: $c = c \wedge 1 = c \wedge (\neg a \vee a) = (c \wedge a) \vee (c \wedge \neg a) \leq b \vee (c \wedge \neg a) \leq b \vee \neg a$. Dass für die Zusammensetzung $ev \circ (\Lambda f \times id_a) = f$ gilt, ergibt sich daraus, dass es in einer Halbordnung immer höchstens einen Pfeil gibt.*

2. *Man kann Deduktionssysteme als Kategorien auffassen wie folgt. Ein Deduktionssystem hat Formeln und Beweise. Die Formeln werden als Objekte aufgefasst und der Pfeil $p : A \to B$ steht für einen Beweis, der aus der Annahme von A die Formel B herleitet. Welche Pfeile existieren, hängt von den vorhandenen Ableitungsregeln ab. Offenbar gibt es in einer solchen Kategorie identische Pfeile, nämlich die Beweise $A \to A$. Beweise können zusammengesetzt werden: wenn $p : A \to B$ und $q : B \to C$ dann kann man aus A auch C beweisen mit $q \circ p$. Die Zusammensetzung ist offenbar assoziativ. Wenn A und B Formeln sind, dann ist $A \Rightarrow B$ ebenfalls eine Formel, gelesen als A impliziert B. Sie stellt das Exponential dar. Das Produkt $A \times B$ steht für die Formel $A \wedge B$ und wenn wir einen Beweis $f : C \wedge A \to B$ haben, dann bedeutet das, dass wir aus A und C zusammen B herleiten können. Der Beweis $\Lambda f : C \to (A \Rightarrow B)$ ist dann der Beweis, der aus C die Formel $A \Rightarrow B$ herleitet. Der Zusammenhang zwischen f und Λf heißt in Logiktexten üblicherweise Abtrennungsregel.*

3.4.1 Scott-Domains

Für unser nächstes Beispiel kartesisch abgeschlossener Kategorien benötigen wir noch einige Begriffe: vollständige partielle Ordnungen und Scott Topologie. Ursprünglich definierte Scott [?] seine Topologie für vollständige Verbände, aber die Definition lässt sich einfach auf vollständige partielle Ordnungen verallgemeinern. Wir betrachten hier die allgemeinere Situation, da sie für die Interpretation des λ-Kalküls besser geeignet ist. Wie wir später noch sehen werden, gibt es eine natürliche Interpretation von λ-Termen als eine bestimmte Form von Bäumen und diese bilden eine vollständige partielle Ordnung, aber keinen Verband.

Definition 23. *Sei $(X, \leq)$ eine Halbordnung(partiell geordnete Menge).*

1. *$D \subseteq X$ heißt gerichtet, falls D nicht leer ist und für jedes Paar von Elementen $i, j \in D$ ein Element $k \in D$ existiert, mit $i \leq k$ und $j \leq k$.*

2. *$(X, \leq)$ heißt vollständig(CPO)[12], falls es ein kleinstes Element $\bot \in X$ gibt mit $\bot \leq x$ für alle $x \in X$ und jede gerichtete Menge $D \subseteq X$ eine kleinste obere Schranke $supD \in X$ besitzt (die nicht notwendig in D liegt).*

3. *Ein Element $x \in X$ heißt kompakt oder endlich[13], falls für alle gerichteten Mengen $D \subseteq X$ mit $x \leq supD$ ein Element $d \in D$ existiert, mit $x \leq d$. Die Menge der kompakten Elemente von X wird mit X_0 bezeichnet.*

[12] complete partial order

[13] Wenn X die Potenzmenge einer Menge ist, dann sind die endlichen Mengen gerade die endlichen Elemente.

4. $(X, \leq)$ *heißt algebraisch, falls die Menge* $x\downarrow = \{x_0 \in X_0 \mid x_0 \leq x\}$ *gerichtet ist und* $sup(x\downarrow) = x$.

5. $(X, \leq)$ *heißt beschränkt vollständig, falls jede beschränkte Teilmenge von* X *ein Supremum hat.*

Ein Verband L *ist vollständig, falls jede Teilmenge* $X \subseteq L$ *ein Supremum hat. Jeder vollständige Verband ist eine vollständige Halbordnung, mit* $\perp = sup\emptyset$.

Beispiele 13. *1. Ein Spezialfall für ein algebraische Halbordnung ist ein algebraischer Verband, d.h. ein vollständiger Verband* L*, bei dem jedes Element* $x \in L$ *das Supremum aller kompakten Elemente ist, die kleiner als* x *sind.*
Das folgende Beispiel erklärt den Namen „algebraisch“ :
Sei $Sub(A)$ *die Menge aller Teilalgebren einer Algebra* A *mit derselben Signatur (falls es keine nullstelligen Operationen gibt, gehört auch die leere Algebra dazu).* $(Sub(A), \subseteq)$ *bildet einen Verband mit* A *als größtem Element. Für jedes Paar* S *und* T *ist* $inf\{S, T\} = S \cap T$ *und* $sup\{S, T\}$ *ist die kleinste von* S *und* T *generierte Algebra.* $Sub(A)$ *ist sogar vollständig. Die kompakten Elemente sind die endlich erzeugten Teilalgebren von* A*. Jede Teilalgebra ist die Vereinigung ihrer endlich erzeugten Teilalgebren, also ist* $Sub(A)$ *algebraisch. Es gilt sogar die Umkehrung: jeder algebraischer Verband ist isomorph zu* $Sub(A)$ *für eine Algebra* A.

2. *Ein weiteres Beispiel für einen algebraischen Verband ist* $(Con(A) \subseteq)$*, die Menge aller Kongruenzrelationen auf einer Algebra* A*. Es gilt* $Con(A) \subseteq Sub(A \times A)$*. Das größte Element ist* $A \times A$*, das sich vom konstanten Homomorphismus ableitet, das kleinste ist die Diagonale von* $A \times A$*, das zu den Isomorphismen gehört.* $Con(A)$ *ist vollständig und algebraisch, die kompakten Elemente sind die endlich erzeugten Kongruenzen. Nach einem Satz von G. Grätzer und E.T.Schmidt ist jeder algebraische Verband isomorph zu einem* $Con(A)$ *einer Algebra* A.

Definition 24. *Eine Scott-Domain ist eine beschränkt vollständige, vollständige algebraische Halbordnung.*

Will man auf einer Menge Lagebeziehungen wie „in der Umgebung von“ und oder „konvergiert zu “ ausdrücken können, braucht man eine topologische Struktur, die bestimmte Teilmengen als Umgebungen versteht. Ein topologischer Raum besteht aus einer Trägermenge X zusammen mit einer Menge $\mathcal{O} = \{O_i\}_{i \in I}$ von sogenannten offenen Mengen, mit $X, \emptyset \in \mathcal{O}$. Außerdem $\mathcal{O}$ muss abgeschlossen sein unter endlichen Durchschnitten und beliebigen Vereinigungen.

Definition 25. *Sei* $(X, \leq)$ *eine vollständige Halbordnung. Die Scott-Topologie auf* X *ist wie folgt definiert.* $O \subseteq X$ *ist offen, wenn*

(i) aus $x \in O$ *und* $x \leq y$ *folgt* $y \in O$ *und*

(ii) wenn für eine gerichtete Menge $D \subseteq X$ *gilt, dass* $supD \in O$ *dann folgt* $D \cap O \neq \emptyset$.

Offensichtlich gehören $\emptyset$ und X selbst zu den offenen Mengen. Falls O_1 und O_2 offen sind, dann gilt für jedes gerichtete D mit $supD \in O_1 \cap O_2$ wegen $supD \cap O_1 \neq \emptyset$ und $supD \cap O_2 \neq \emptyset$ auch $supD \cap (O_1 \cap O_2) \neq \emptyset$, also ist $\mathcal{O}$ abgeschlossen unter endlichem

Durchschnitt. Damit ist die Scott-Topologie tatsächlich eine Topologie.

Für jedes $x_0 \in X_0$ bezeichnen wir mit τ_{x_0} die Menge $\tau_{x_0} = \{x \mid x_0 \leq x\}\}$. Die Familie $\{\tau_{x_0}\}_{x_0 \in X_0}$ ist die Basis der Scott-Topologie. Die Mengen τ_{x_0} sind sogenannte upper sets.

Für die folgenden Beweise benötigen wir eine weitere Bezeichnung.

$$U_x = \{z \in X \mid z \not\leq x\}$$

Die U_x sind offene Mengen.

Aufgaben:

1. Beweise für eine endliche Menge M von kompakten Elementen, dass falls $supM$ existiert, dann ist es kompakt.

2. Finde (nichttriviale) Gegenbeispiele für folgende Aussagen: 1. Falls x_0 kompakt ist, dann ist $\{y \mid y \leq x\}$ endlich. 2. Falls x_0 kompakt ist und $y \leq x_0$, dann ist y auch kompakt.

Um Scott-Domains als Kategorien zu betrachten, brauchen wir noch passende Abbildungen, die die Struktur (also die Ordnung und die Topologie) erhalten.

Definition 26. *Seien $(X, \mathcal{O})$ und $(Y, \mathcal{O}')$ topologische Räume. Eine Funktion $f : X \to Y$ heißt stetig, falls $f^{-1}(T) \in \mathcal{O}$ für alle offenen Mengen $T \in \mathcal{O}'$.*

Lemma 2. *Seien $(X, \leq_X)$ und $(Y, \leq_Y)$ Scott-Domains. Eine Funktion $f : X \to Y$ ist genau dann stetig (in Bezug auf die Scott-Topologie), wenn für alle gerichteten Mengen $D \subseteq X$ gilt $f(supD) = sup_D\{f(d)\}$.*

Beweis: „$\Rightarrow$“
Sei f stetig und sei $a \leq_X b$. Wir zeigen zunächst, dass dann $f(a) \leq_Y f(b)$. Falls das nicht so wäre, dann wäre $f(a) \in U_{f(b)}$ und damit wäre $a \in f^{-1}(U_{f(b)})$, was eine offene Menge ist. Wegen $a \leq b$ gilt auch $b \in f^{-1}(U_{f(b)})$ und wir haben einen Widerspruch, da nach Definition $f(b) \notin U_{f(b)}$ und damit $b \notin f^{-1}(U_{f(b)})$. Also ist f monoton. Wegen Monotonie gilt $\forall a \in D$ $f(a) \leq f(supD)$ und damit auch $sup(f(D)) \leq f(supD)$. Wäre nämlich $f(supD) \not\leq sup(f(D))$, dann wäre $f(supD) \in U_{supf(D)}$ und man erhielte einen ähnlichen Widerspruch wie oben, indem man die Bedingung (ii) aus Definition 25 verwendet.
„$\Leftarrow$“
Wir wollen zeigen, dass aus $f(supD) = sup_D\{f(d)\}$ für alle gerichteten Mengen D folgt, dass Urbilder offener Mengen offen sind. Sei $a \leq b$, dann ist $b = sup\{a, b\}$ und $sup\{f(a), f(b)\} = f(sup\{a, b\}) = f(b)$ also auch $f(a) \leq f(b)$. Daher gilt für eine offene Menge in $O \subseteq Y$, dass auch $f^{-1}(O)$ offen in X ist. Damit folgt für jede gerichtete Menge $D \subseteq X$ mit $supD \in f^{-1}(O)$, dass $f(supD) = sup(f(D)) \in O$. Da $f(D)$ gerichtet ist, gilt $f(D) \cap O \neq 0$ und es folgt $D \cap f^{-1}(O) \neq \emptyset$. □

Offenbar ist jede stetige Funktion auch monoton.

Definition 27. *Die Kategorie* **D** *hat als Objekte Scott-Domains und als Pfeile stetige Abbildungen.*

Satz 11. *Die Kategorie* **D** *ist kartesisch abgeschlossen.*

Um zu zeigen, dass das Exponential existiert, müssen wir zeigen, dass die Menge Y^X der stetigen Funktionen von X nach Y selbst eine Scott-Domain ist.

Beweis:

1. Y^X *ist vollständige Halbordnung:* Offenbar induziert die punktweise Ordnung eine Halbordnung auf Y^X mit $f \leq_{Y^X} g$ gdw. $f(x) \leq_Y g(x)$ für alle $x \in X$. Sei $\{f_i\}_{i \in I}$ eine gerichtete Menge von stetigen Abbildungen. Dann ist für jedes x die Menge $\{f_i(x)\}_{i \in I}$ ebenfalls gerichtet. Man definiert $(sup_{i \in I}\{f_i\})(x) = sup_{i \in I}\{f_i(x)\}$. Das Supremum auf der rechten Seite existiert für alle x wegen Vollständigkeit von Y.

2. Y^X *ist beschränkt vollständig:* Sei g eine obere Schranke für eine Menge $\{f_i\}_{i \in I}$, dann ist die Funktion h mit $h(x) = sup\{f_i(x)\}$ wohldefiniert wegen beschränkter Vollständigkeit von Y. Außerdem ist h stetig, da für jede gerichtete Menge $D \subseteq X$ gilt $h(supD) = sup_I\{f_i(D)\} = sup_I\{sup_{x \in D}\{f_i(x)\}\}$ wegen Stetigkeit der f_i. Weiterhin ist $sup_I\{sup_{x \in D}\{f_i(x)\}\} = sup_{x \in D}\{sup_I\{f_i(x)\}\} = sup_{x \in D}\{h(x)\}$.

3. X^Y *ist algebraisch:* Wir geben die kompakten Elemente von Y^X explizit an und definieren dazu zunächst, was eine Schrittfunktion ist.

   Definition 28. *Für kompakte Elemente $a \in X_0, b \in Y_0$ heißt eine Funktion $step_{a,b}$: $X \to Y$ Schrittfunktion, wenn*

   $$step_{a,b}(x) = \begin{cases} b \text{ falls } a \leq x \\ \perp \text{ sonst.} \end{cases}$$

   Wir beweisen, dass ein Element von Y^X genau dann kompakt ist, wenn es das Supremum einer endlichen beschränkten Menge von Schrittfunktionen ist.
   „$\Leftarrow$“ Sei I eine endliche Indexmenge, wir nehmen an, dass $\sup_I\{step_{a_i,b_i}\}$ existiert und betrachten $f = \sup_I\{step_{a_i,b_i}\}$. Nach Definition existiert $\sup_I\{step_{a_i,b_i}\}$ genau dann wenn für jede Teilmenge J, für die $a_J = sup_J\{a_i\}$ existiert, auch $b_J = sup_J\{b_i\}$ existiert. Sei $\{g_k\}_{k \in K}$ eine gerichtete Familie in Y^X mit $f \leq sup_K\{g_k\}$. Dann gilt insbesondere für jede Teilmenge $J \subseteq I$ auch $f(a_J) = b_J \leq (sup_K\{g_k\})(a_J)$. Da b_J das Supremum über eine endliche Menge von kompakten Elementen aus Y ist, ist jedes b_J kompakt und da $\{g_k\})(a_J)$ eine gerichtete Menge ist (da $\{g_k\}$ ja eine ist) muß es einen Index $k_J \in K$ geben, für den $b_J \leq g_{k_J}(a_)$ ist. Da I endlich ist und die Familie $\{g_k\}_{k \in K}$ gerichtet ist, können wir einen Index u finden, so dass $g_J \leq g_k$ für alle $J \subseteq I$. Für dieses u gilt dann $f \leq g_u$, d.h. f ist kompakt.

   „$\Rightarrow$“ Man zeigt leicht, dass für jede stetige Funktion f die Menge $F = \{sup_I\{step_{a_i,b_i}\} \mid b_i \leq f(a_i)\}_{I endlich}$ gerichtet ist und dass $f = \sup F$. Nehmen wir an f wäre kompakt. Wir müssen zeigen, dass es dann eine endliche Indexmenge I gibt mit $f = sup_I\{step_{a_i,b_i}\}$. Da f sich als Supremum der gerichteten Menge F schreiben lässt, muß es auch einen Index I geben für den $f \leq sup_I\{step_{a_i,b_b}\}$ Und wir haben $sup_I\{step_{a_i,b_b}\} \leq supF = f \leq sup_I\{step_{a_i,b_b}\}$ und damit $f = sup_I\{step_{a_i,b_b}\}$.
   Aus „$\Leftarrow$“ und „$\Rightarrow$“ folgt die Behauptung und damit, dass Y^X ein Scott-Domain ist. Der Beweis dass $ev : Y \times X \to Y$ mit $ev(f,x) = f(x)$ stetig ist, bleibt dem Leser überlassen.

$\square$

Bemerkungen Obwohl die Scott-Topologie auf dem Kreuzprodukt von zwei Scott-Domains als Kreuzprodukt der jeweiligen Topologien definiert werden kann, gilt nicht allgemein, dass die Topologie von $X \times Y$ für zwei Halbordnungen das Produkt der Topologien ist. Siehe dazu Aufgabe 1.3.12 in [**?**].

Beispiele 14. *1. Die kartesische Abgeschlossenheit erlaubt es, Scott-Domains auf einfache Weise zu definieren. Man wählt sich Datentypen(z.B. Integer, Bool, Strings) organisiert sie als flache Halbordnungen(man fügt je ein kleinstes Element hinzu, und es gilt $x \leq y$ nur dann, wenn $x = y$ oder $x = \bot$) und erhält damit offenbar Objekte von* $\mathbf{D}$.

2. *Die Menge* $\mathbf{P}$ *der partiellen Funktionen von* $\mathbb{N}$ *nach* $\mathbb{N}$ *mit der Relation* $\sqsubseteq$, *definiert durch* $f \sqsubseteq g$ *genau dann, wenn* $\forall n.[f(n)\downarrow \Rightarrow g(n) = f(n)]$ *ist eine Scott-Domain.*

3. *Auch* $\mathcal{P}(\mathbb{N})$ *ist eine Scott-Domain. Sowohl* $\mathbf{P}$ *als auch* $\mathcal{P}(\mathbb{N})$ *sind auch reflexive Objekte in* $\mathbf{D}$. *Wir konstruieren ein Retraktionspaar* $(graph, fun)$ *für* $\mathcal{P}(\mathbb{N})^{\mathcal{P}(\mathbb{N})}$ *und* $\mathcal{P}(\mathbb{N})$:

   Sei $\{F_n\}_{n\in\mathbb{N}}$ *eine kanonische (bijektive und berechenbare) Aufzählung der endlichen Teilmengen von* $\mathbb{N}$ *und sei* $<,>$ *eine kanonische Paarungsfunktion. Wir definieren einen Pfeil* $graph : Hom_{\mathbf{D}}[\mathcal{P}(\mathbb{N}), \mathcal{P}(\mathbb{N})] \to \mathcal{P}(\mathbb{N})$ *durch*

   $$graph(f) = \{< n, m >\mid m \in f(F_n)\}$$

   und einen Pfeil $fun : \mathcal{P}(\mathbb{N}) \to Hom_{\mathbf{D}}(\mathcal{P}(\mathbb{N}), \mathcal{P}(\mathbb{N}))$ *durch*

   $$fun(A)(B) = \{m \mid \exists F_n \subseteq B. < n, m > \in A\}.$$

   Um sich zu überzeugen, dass $fun \circ graph = id$ *ist, betrachten wir:*

   $$\begin{aligned}(fun \circ graph(f))(B) &= (fun(\{< n, m >\mid m \in f(F_n)\})(B) \\ &= \{k \mid \exists F_i \subseteq B. < i, k > \in \{< n, m >\mid m \in f(F_n)\}\} \\ &= \{k \mid \exists F_i \subseteq B. k \in f(F_i)\} \\ &= f(B)\end{aligned}$$

   Natürlich muss man sich auch noch davon überzeugen, dass fun *und* $graph$ *Pfeile in* $\mathbf{D}$, *also stetige Funktionen sind. Die kanonische Paarungsfunktion* $<,>$ *ist definiert als* $< n, m >= \frac{1}{2}(n^2+2nm+m^2+3n+m)$. *Eine Aufzählung der endlichen Funktionen bekommt man folgendermaßen. Sei* f *die Funktion, die definiert ist durch*

   $$f(n) = (i_1, \ldots, i_s) \text{ wenn } n = \prod_{i=1,\ldots s} p_{k_i}^{j_i} \text{ wobei } p_i \text{ die } i\text{-te Primzahl ist}$$

   und sei g *eine Aufzählung der Zahlen, deren Primfaktoren immer nur in einfacher Potenz vorkommen und die für* 0 *das leere Produkt hat. Dann ist* $f \circ g : \mathbb{N} \to \mathcal{P}_{fin}(\mathbb{N}$ *eine bijektive Aufzählung der endlichen Teimengen der natürlichen Zahlen.*

Interessant im Zusammenhang mit der Theorie der berechenbaren Funktionen ist der folgende Fixpunktsatz.

Satz 12. *Sei X eine vollständige Halbordnung.*

(i) Jede stetige Funktion $f : X \to X$ hat einen Fixpunkt.

(ii) Es gibt eine stetige Funktion $Fix : X^X \to X$ so dass $Fix(f)$ der kleinste Fixpunkt von f für $f \in X^X$ ist.

Beweis: (i) Da $\perp \leq f(\perp)$ folgt aus der Monotonie dass $f(\perp) \leq f(f(\perp))$. Also haben wir $f(\perp) \leq f^2(\perp)$, $f^2(\perp) \leq f^3(\perp)$ usw. Die Folge $f^i(\perp)$ ist gerichtet und daher existiert auch ein Supremum

$$x_f = sup_n f^n(\perp).$$

Wegen der Stetigkeit von f haben wir

$$f(x_f) = f(sup_n f^{n+1}(\perp)) = x_f.$$

Also ist x_f ein Fixpunkt von f.
(ii) Der im Teil (i) konstruierte Fixpunkt ist der kleinste Fixpunkt von f, Für jeden anderen Fixpunkt y folgt aus der Monotonie von f, dass $x_f = f^n(\perp) \leq f^n(y) = y$. Wir können den Fixpunktoperator nun definieren als

$$Fix = \lambda f.x_f = \lambda f.sup_n\{f^n(\perp)\}.$$

□

3.5 Differenzkerne und Pullbacks

Für zwei parallele Pfeile $f, g : A \to B$ in **SET** können wir eine Menge $E \subseteq A$ definieren, mit der Eigenschaft, dass ihre Einschränkungen auf E übereinstimmen. Formal sieht das dann aus wie folgt:

$E = \{x \in A \mid f(x) = g(x)\}$. Wir haben früher schon gesehen, dass man den Teilmengenbegriff durch monomorphe Pfeile ersetzen kann. Für einen Differenzkern verlangen wir folgende Eigenschaft:

Definition 29. *Seien $A, B \in Obj(\mathbf{C})$ und $f, g \in Hom_{\mathbf{C}}(A, B)$. Ein Objekt E zusammen mit einem Pfeil $i : E \to A$ heißt Differenzkern (equalizer), wenn gilt*

1. *$f \circ i = g \circ i$ und*

2. *für jeden Pfeil i' mit der Eigenschaft $f \circ i' = g \circ i'$ existiert ein Pfeil $j : dom(i') \to E$ mit $i' = i \circ j$.*

Häufig interessiert man sich auch für das duale Konzept, den Differenzkokern:

Definition 30. *Seien $A, B \in Obj(\mathbf{C})$ und $f, g \in Hom_{\mathbf{C}}(A, B)$. Ein Objekt F zusammen mit einem Pfeil $k : B \to F$ heißt Differenzcokern (coequalizer), wenn gilt*

1. *$k \circ f = k \circ g$ und*

2. *für jeden Pfeil k' mit der Eigenschaft $k' \circ f = k' \circ g$ existiert ein Pfeil $l : B \to codom(k)$ mit $k' = l \circ k$.*

Man kann sich den Differenzcokern als eine Art Quotientenobjekt unter der von f und g induzierten Ähnlichkeitsrelation vorstellen.

Wir betrachten die Situation in **SET**. Sei

$$R \subseteq B \times B \text{ mit } R = \{(f(a), g(a)) \mid a \in A\}.$$

Für alle $(b_1, b_2) \in R$ gilt $k(b_1) = k(b_2)$. Die Relation R ist nicht notwendigerweise eine Äquivalenzrelation. Sie lässt sich allerdings durch Bilden der reflexiven, transitiven und symmetrischen Hülle zu einer solchen vervollständigen zu $\hat{R}$. Sei C die Menge der Äquivalenzklassen bezüglich $\hat{R}$, dann ist $k : B \to C$ die Funktion, die jedes Element auf die Äquivalenzklasse, in der es enthalten ist, abbildet der Differenzkokern von f und g. (Übungsaufgabe)

Satz 13 (Eigenschaften von Differenzkernen und Differenzkokernen)**.** *1. Jeder Differenzkern ist ein Monomorphismus, jeder Differenzkokern ist ein Epimorphismus.*

2. Jeder Differenzkern, der ein Epimorphismus ist, ist ein Isomorphismus.

Beweis:

1. Sei i der Differenzkern von f und g. Wir müssen zeigen, dass falls $h \circ t = h \circ s$, dann gilt $s = t$. Da i ein Equalizer ist, gilt auch $f \circ h \circ t = g \circ h \circ s$. Dann gibt es aber einen eindeutig bestimmten j mit $i \circ j = i \circ s = i \circ t$. Daraus folgt $j = s = t$.

2. Sei $e : E \to A$ der Differenzkern von f und g. Da e ein Epimorphismus ist folgt aus $f \circ e = g \circ e$ auch dass $f = g$. Da id_A ebenfalls f und g ausgleicht, gibt einen Pfeil $id_A = e \circ \tilde{e}$. Da e ein Monomorphismus ist, ist e linkskürzbar und aus $e \circ id_E = id_A \circ e = e \circ \tilde{e} \circ e$ folgt, dass $id_E = \tilde{e} \circ e$.

□

Der nächste Begriff ist in einem gewissen Sinne eine Verallgemeinerung von Differenzkernen auf Paare von Morphismen mit verschiedenen Quellen.

Definition 31. *Gegeben seien zwei Pfeile $f : B \to A$ und $g : C \to A$. Das Pullback oder Faserprodukt ist ein Objekt $B \times_A C$ zusammen mit zwei Pfeilen $p : B \times_A C \to B$ und $q : B \times_A C \to C$ so dass folgende Bedingungen erfüllt sind:*

1. *$f \circ p = g \circ q$ und*

2. *für jedes andere Paar $p' : D \to B$ und $q' : D \to C$ mit $f \circ p' = g \circ q'$ existiert ein eindeutig bestimmter Pfeil $u : D \to B \times_A C$ mit $p' = p \circ u$ und $q' = q \circ u$.*

Abbildung 6 zeigt das typische Pullbackdiagramm, wobei das Quadrat $(B \times_A C)CBA$ auch Pullback-Quadrat heißt.

Bemerkungen Jede Funktion $f : B \to A$ definiert eine Partitionierung von B, indiziert durch die Elemente von A, nämlich

$$B_a = f^{-1}(a) = \{b \in B \mid f(b) = a\}.$$

Die Mengen B_a sind paarweise disjunkte Teilmengen von B, deren Vereinigung B ergibt. Eine solche Partitionierung wird manchmal als Faserung bezeichnet, jedes der B_a heißt dann Faser über a. Man sieht leicht, dass für ein Pullback in **SET** gilt:

$$B \times_A C = \bigcup_{a \in A} \{B_a \times C_a\},$$

was den Namen Faserprodukt erklärt.

Abbildung 6: Pullbackdiagramm

$$\begin{tikzcd}
D \arrow[dr, dotted, "u"] \arrow[drr, "q'"] \arrow[ddr, "p'"'] & & \\
& B \times_A C \arrow[r, "q"'] \arrow[d, "p"] & C \arrow[d, "g"'] \\
& B \arrow[r, "f"] & A
\end{tikzcd}$$

Beispiele 15. *1. In* **SET** *wird das Pullbackobjekt von f und g gebildet durch*

$$B \times_A C = \{(b, c) \mid f(b) = g(c)\}$$

Die zugehörigen Funktionen p und q sind Projektionen mit $p((b, c)) = b$ und $q((b, c)) = c$. In Kategorien mit Produkten ist ein Faserprodukt stets ein Teilobjekt des Kreuzproduktes.

2. *Im speziellen Fall, wo $C \subseteq A$ eine Teilmenge von A und g die Einbettungsabbildung ist, ist das Pullback von f entlang der Einbettung von C das inverse Bild von f $f^{-1}(C) = \{x \in B \mid f(x) \in C\}$.*

3. *Ein weit verbreiteter Ansatz in der Programmverifikation ist das Hinzufügen von Vor-und Nachbedingungen zu Programmteilen wie z.B.*

   (1) $\{X > 3\}$
   (2) $X := X + 1$
   (3) $\{X < 24\}$

   Dabei ist X eine Intergervariable, Bedingung (1) ist ein Vorbedingung und (3) ist eine Nachbedingung. Der gesamte Ausdruck (1) – (3) ist eine Aussage über das Programm, nämlich dass wenn (1) vor der Ausführung gilt, dann gilt (3) nach der Ausführung. Man kann aber eine stärkere Aussage treffen. Für die gegebene

Nachbedingung ist die schwächste Vorbedingung, die man angeben kann, die Bedingung $\{X < 23\}$. In einer sehr allgemeinen Umgebung gibt es für jede Nachbedingung eine schwächste Vorbedingung. Wir betrachten die kategoriale Darstellung. Sei D eine Menge von möglichen Eingaben und E eine Menge von möglichen Ausgaben. Ein (deterministisches und terminierendes) Programmfragment ist ein Pfeil $f : D \to E$. Jeder Bedingung C an eine Menge X kann als Teilmenge $X_C = \{x \in X \mid x \text{ erfüllt } C\}$ dargestellt werden. Insbesondere ist die Nachbedingung eine Teilmenge S von E und die schwächste Vorbedingung für diese Nachbedingung ist das inverse Bild $f^{-1}(S)$ von S entlang f, welches das eindeutig bestimmte Teilobjekt von D ist, für das Abbildung 7 ein Pullback ist. Im Fall von **SET** *lässt*

Abbildung 7: Schwächste Vorbedingung

$$
\begin{tikzcd}
f^{-1}(S) \arrow[r, hook] \arrow[d, "f\,|_{f^{-1}(S)}"'] & S \arrow[d, hook] \\
D \arrow[r, "f"] & E
\end{tikzcd}
$$

sich das ganz einfach verifizieren, die Konstruktion ist ebenso plausibel für andere Kategorien, mit denen man Daten und Programme darstellt.

Satz 14. *Sei* **C** *eine Kategorie mit Terminalobjekt T. Sei $!_A : A \to T$ der eindeutig bestimmte Morphismus. Wenn in* **C** *für jedes Paar von Morphismen mit dem gleichen Ziel das Pullback existiert, dann existiert auch für jedes Paar von Objekten ein Produkt.*

Beweis: Sei $A \times_T B$ mit $p_A : A \times_T B \to A$ und $p_B : A \times_T B \to B$ das Pullback von $!_A$ und $!_B$. Wegen der Universaleigenschaft des PB ist $A \times_T B$ ein Produkt. $\square$

Satz 15. *Sei* **C** *eine Kategorie mit Terminalobjekt, in der für jedes Paar von Morphismen mit dem gleichen Ziel das Pullback existiert, dann existiert auch für jedes Paar von parallelen Pfeilen ein Differenzkern.*

Beweis: Sei C mit $h_1 : C \to A$ und $h_2 : C \to A$ das Pullback von $< f, id_A >: A \to B \times A$ und $g : A \to B \times A$ in Abbildung 8. Dann ist der Differenzkern von f und g das Objekt C zusammen mit $h_1 = h_2$. Offenbar gilt $f \circ h_2 = p_l \circ < f, id_A > \circ h_2 = p_l \circ < g, id_A > \circ h_1 = g \circ h_1$. Außerdem gilt für jedes C' und Pfeil $h : C' \to A$, für den $f \circ h = g \circ h$ auch $< f, id_A > \circ h =< g, id_A > \circ h$ und nach Definition des PB existiert dann ein $u : C' \to C$ mit $h = h_2 \circ u$. $\square$

Bei Beweisen werden wir häufig die folgenden beiden Eigenschaften von Pullbacks verwenden.

Lemma 3 (Pullbacklemma–PBL)**.** *Gegeben sei ein kommutatives Diagramm wie in Abbildung 9, dann gilt:*

1. *wenn die beiden Quadrate Pullbacks sind, dann ist auch das äußere Rechteck ein Pullback,*

Abbildung 8: Differenzkern als Pullback

$$
\begin{tikzcd}
C \arrow[r, "h_1"] \arrow[d, "h_2"'] & A \arrow[d, "{< g, id_A >}"] \arrow[ddr, bend left, "g"] & \\
A \arrow[r, "{< f, id_A >}"'] \arrow[drr, bend right, "f"'] & B \times A \arrow[dr, "p_l"] & \\
& & B
\end{tikzcd}
$$

Abbildung 9: Pullbacklemma

$$
\begin{tikzcd}
\bullet \arrow[r] \arrow[d] & \bullet \arrow[r] \arrow[d] & \bullet \arrow[d] \\
\bullet \arrow[r] & \bullet \arrow[r] & \bullet
\end{tikzcd}
$$

2. *falls das äußere Rechteck und das rechte Quadrat Pullbacks sind, dann ist auch das linke eines.*

Beweis: Übungsaufgabe. □

Lemma 4. *Falls das Quadrat in Abbildung 6 ein Pullback ist und g ist ein Monomorphismus, dann ist p ebenfalls ein Monomorphismus.*

Beweis: Übungsaufgabe. □

3.6 Teilobjektklassifizierer und Topoi

Mit den Begriffen Teilobjekt und verallgemeinertes Element haben wir bereits zwei Begriffe aus der Mengenlehre in die Sprache der Kategorientheorie übersetzt. In diesem Abschnitt wollen wir auch eine Verallgemeinerung der Potenzmenge und davon abgeleiteter Begriffe darstellen. In **SET** haben wir eine Äquivalenz zwischen der Menge der Teilmengen einer Menge A und der Menge der Funktionen von A in die Menge $\{0,1\}$(oder eine andere zweielementige Menge). Die Äquivalenz wird hergestellt, indem wir eine Teilmenge $B \subseteq A$ abbilden auf ihre charakteristische Funktion $\chi_B : A \to \{0,1\}$. Kategorientheoretisch kann man diese Abbildung als Pullback darstellen wie in Abbildung 10, wobei 1 die Funktion ist, die jedes Argument auf 1 abbildet.

Eine Verallgemeinerung dieses Diagramms erlaubt folgende Definition:

Abbildung 10: Teilmengen als Pullbacks

$$\begin{tikzcd} B \arrow[r, "incl_B"] \arrow[d, "!"'] & A \arrow[d, "\chi_B"] \\ \{*\} \arrow[r, "1"'] & \{0,1\} \end{tikzcd}$$

Definition 32. *Sei* **C** *eine Kategorie mit einem Terminalobjekt T. Ein Teilobjektklassifizierer ist ein Objekt Ω von* **C** *zusammen mit einer Abbildung $true : T \to \Omega$, sodass es für jeden Monomorphismus $f : B \to A$ einen eindeutig bestimmten Pfeil $\chi_f : A \to \Omega$ gibt, so dass das Diagramm in Abbildung 11 ein Pullback ist.*

Abbildung 11: Definition von Ω

$$\begin{tikzcd} B \arrow[r, "f"] \arrow[d, "!"'] & A \arrow[d, "\chi_f"] \\ T \arrow[r, "true"'] & \Omega \end{tikzcd}$$

Die Definition des Teilobjektklassifizierers spielt die zentrale Rolle in der Übersetzung der Mengentheorie in kategoriale Sprache und erlaubt die Definition von Operationen, die Verallgemeinerungen der Mengenoperationen Durchschnitt, Vereinigung und Komplementbildung sind. Für diesen Zweck baut man eine Heyting Algebra von Wahrheitswertmorphismen über Ω auf. Man definiert die verallgemeinerten Elemente $\bot, \top : T \to \Omega$ durch $\top := true$ und $\bot := \chi_!$ im Pullbackdiagramm in Abbildung 13. Die Wahrheits-

Abbildung 12: Definition von $\bot$

$$\begin{tikzcd} 0 \arrow[r, "ini_T"] \arrow[d, "!"'] & T \arrow[d, "\chi_! = \bot"] \\ T \arrow[r, "true = \top"'] & \Omega \end{tikzcd}$$

wertfunktionen in Ω definiert man dann wie folgt:

1. $\neg : \Omega \to \Omega$ ist der eindeutig bestimmte Pfeil $\neg = \chi_\perp$
2. $\wedge : \Omega \times \Omega \to \Omega$ ist der charakteristische Pfeil von $<\top, \top>: T \to \Omega \times \Omega$
3. $\vee : \Omega \times \Omega \to \Omega$ ist der charakteristische Pfeil von $[< T \circ !_\Omega, id_\Omega >, < id_\Omega, T \circ !_\Omega >]$.

Aufgrund ihrer Bedeutung für Interpretationen bekommen Kategorien, in denen sich Teilobjekte klassifizieren lassen einen Namen.

Definition 33. *Eine Kategorie* **C** *heißt Topos, wenn folgende vier Bedingungen erfüllt sind:*

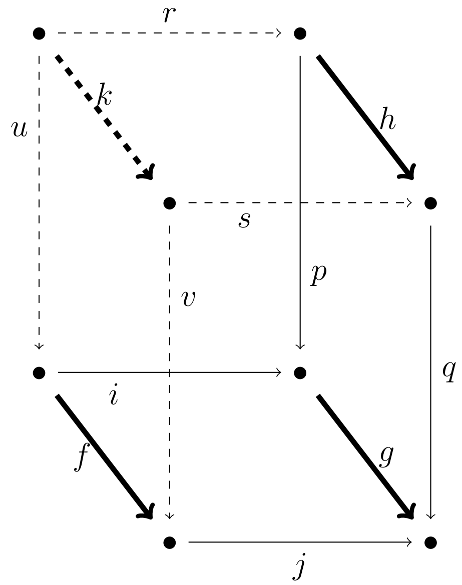
1. **C** *hat ein Terminalobjekt,*
2. *für jedes Paar von Pfeilen mit gemeinsamen Ziel existiert ein Pullback,*
3. **C** *hat einen Teilobjektklassifizierer,*
4. *für alle Paare von Objekten existieren Exponentiale.*

3.6.1 Beispiele für Topoi

1. Offensichtlich ist **SET** ein Topos. Ein Terminalobjekt ist eine Einermenge. Der Teilobjektklassifizierer ist $\Omega = \{0, 1\}$ und $true : \{*\} \to \{0, 1\}$ ist die Abbildung mit $true(*) = 1$.
2. **SET** $\times$ **SET** ist ein Topos. Alle Konstruktionen werden zunächst auf den einzelnen Komponenten in SET ausgeführt und dann zu Paaren zusammengefaßt. Terminalobjekt ist z.B. $(\{*\}, \{*\})$. Seien $< f, f' >: (A, A') \to (B, B')$ und $< g, g' >: (C, C') \to (B, B')$ Pfeile in $\mathbf{SET}^2$ Pfeile mit derselben Codomain. Wir bilden die Pullbacks $(A \times_B C, p, q)$ für f, g und $(A' \times_{B'} C', p', q')$ für f', g' und erhalten als Pullback für $< f, f' >$ und $< g, g' >$ das Tripel $((A \times_B C, A' \times_{B'} C'), < p, p' >, < q, q' >)$. Der Teilobjektklassifizierer ist $\Omega \times \Omega$ und der zugehörige Pfeil ist $< true, true >: (\{*\}, \{*\}) \to \Omega \times \Omega$.
3. $\mathbf{SET}^{\to}$, die Kategorie der Funktionen ist ein Topos.
   (a) *Terminalobjekt* $id_* : \{*\} \to \{*\}$, für jede Funktion $f : A \to B$ ist die Terminalabbildung das Paar $<!_A, !_B>$, denn aufgrund der Eindeutigkeit der Terminalabbildung ist $!_B \circ f = !_A = !_{\{*\}} \circ !_A$.
   (b) *Pullbacks* Für zwei Pfeile in $\mathbf{SET}^{\to}$ also Paare (i, j) und (p, q) von Funktionen bilden wir die **SET**-Pullbacks von i und p und $j$und q. Das liefert uns Pfeile s und v mit $j \circ v = q \circ s$ und Pfeile r und u mit $i \circ u = p \circ r$.

   Wir müssen uns davon überzeugen, dass die Paare $(u.v)$ und (r, s) Pfeile in $\mathbf{SET}^{\to}$ sind und dass es eine Funktion $k : dom(u) \to dom(v)$ gibt, die das Pullbackobjekt für (i, j) und (p, q) ist mit den Projektionen $(u.v)$ und (r, s).

Abbildung 13: Pullbacks in $\mathbf{SET}^{\to}$



$$
\begin{aligned}
j \circ f \circ u &= g \circ i \circ u \text{ weil}(i,j) \in Hom_{\mathbf{SET}^{\to}}(f,g) && (1)\\
&= g \circ p \circ r \text{ weil hinteres Quadrat PB, also kommutativ} && (2)\\
&= q \circ h \circ r \text{ weil}(p,q) \in Hom_{\mathbf{SET}^{\to}}(h,g) && (3)\\
& && (4)
\end{aligned}
$$

Weil s und v die Projektionen im PB von j und q sind, existiert ein Pfeil k mit $v \circ k = f \circ u$ und $s \circ k = h \circ r$, damit ist $(k,h) \in Hom_{\mathbf{SET}^{\to}}(r,s)$ und $(k,f) \in Hom_{\mathbf{SET}^{\to}}(u,v)$. Damit ist der ganze Würfel tatsächlich ein kommutatives Diagramm in $\mathbf{SET}^{\to}$. Wir müssen nun noch zeigen, dass es auch ein PB ist.

Sei also $(k', (r', s'), (u', v'))$ ein anderes Tripel, so dass $(p,q) \circ (r', s') = (i,j) \circ (u', v')$. Da das vordere Quadrat ein PB ist, existiert ein Pfeil β mit $s' = s \circ \beta$. Da das hintere Quadrat ein PB ist, erhalten wir einen Pfeil α mit $v \circ \alpha = v'$ und $r \circ \alpha = r'$. Es bleibt zu zeigen, das (α, β) ein $\mathbf{SET}^{\to}$-Pfeil von k' nach k ist. Nach Konstruktion ist $q \circ s \circ k = j \circ v \circ k$ und daher gilt auch $q \circ s \circ k \circ \beta = j \circ v \circ k \circ \beta$.

$$
\begin{aligned}
q \circ s \circ k \circ \beta &= j \circ v \circ k \circ \beta && (5)\\
&= j \circ f \circ u \circ \beta \text{ weil } (u,v) \in Hom_{\mathbf{SET}^{\to}}(k,f) && (6)\\
&= j \circ f \circ u' \text{nach Konstruktion von} \beta && (7)\\
&= j \circ v' \circ k' \text{nach Voraussetzung ist } (u', v') \in Hom_{\mathbf{SET}^{\to}}(k', f) && (8)\\
&= j \circ v \circ \alpha \circ k' \text{nach Konstruktion von } \alpha && (9)\\
& && (10)
\end{aligned}
$$

Das Paar $(s \circ k \circ \beta, v \circ \alpha \circ k')$ „macht die Pfeile j und q gleich“. Da das vordere Quadrat ein PB ist, stimmen die beiden Pfeile $k \circ \beta$ und $\alpha \circ k'$ überein und wir haben gezeigt, dass $(\alpha,\beta) \in Hom_{\mathbf{SET}^{\to}}(k',k)$. Somit ist $(k,(s,r),(v,u)$ tatsächlich PB in $\mathbf{SET}^{\to}$.

(c) Teilobjektklassifizierer: Falls $f : A \to B$ Subobjekt von $g : C \to D$ in $\mathbf{SET}^{\to}$ ist, dann gibt es einen monomorphen $\mathbf{SET}^{\to}$-Pfeil (i,j) mit $g \circ i = j \circ f$, wobei i, j Monomorphismen in $\mathbf{SET}$, also injektive Funktionen sind. Wenn die Monomorphismen Einbettungen sind, dann ist $A \subseteq C$ und $b \subseteq D$ und f ist die Einschränkung von g auf A, d.h. $f(x) = g(x)$ für $x \in A$.

Ein Element von C kann daher auf drei Arten klassifiziert werden: Entweder ist $x \in A$, oder $x \notin A$ und $g(x) \in B$ oder $x \notin A$ und $g(x) \notin B$. Wir definieren $F = \{0, \frac{1}{2}, 1\}$ und $\psi : C \to \{0, 1/2, 1\}$ mit

$$\psi(x) = \begin{cases} 1 \text{ wenn } x \in A \\ \frac{1}{2} \text{ wenn } x \notin A, g(x) \in B \\ 0 \text{ wenn } x \notin A, g(x) \notin B \end{cases}$$

Die Funktion $t : F \to \{0,1\}$ mit $t(1) = t(\frac{1}{2}) = 1$ und $t(0) = 0$ wird der Teilobjektklassifizierer von $\mathbf{SET}^{\to}$. Das zugehörige Pullback sieht dann aus wie in Abbildung 14.

Abbildung 14: Teilobjektklassifizierer in $\mathbf{SET}^{\to}$

$$\begin{tikzcd}
A \arrow[rr, "i"] \arrow[dr, "f"] \arrow[dd, "!"'] & & C \arrow[dr, "g"] \arrow[dd, "\psi" near end] & \\
& B \arrow[rr, crossing over, "j"' near start] \arrow[dd, "!" near start] & & D \arrow[dd, "\chi_B"] \\
\{*\} \arrow[rr, "t'"' near start] \arrow[dr, "id"'] & & \mathbf{F} \arrow[dr, dashed, "t"] & \\
& \{*\} \arrow[rr, "true"'] & & \{\mathbf{0},\mathbf{1}\}
\end{tikzcd}$$

Der zugehörige Pfeil $true : 1 \to \Omega$ ist das Paar $(t', true) : id_{\{*\}} \to t$ mit $true(*) = t'(*) = 1$. Vorder-und Rückseite des Diagramms sind Pullbacks in $\mathbf{SET}$, das Diagramm zeigt, dass (ψ, χ_B) der charakteristische Pfeil für (i,j) ist.

(d) Exponential: Seien $f : A \to B$ und $g : C \to D$ Objekte in $\mathbf{SET}^{\to}$. Dann muss g^f eine $\mathbf{SET}^{\to}$-Objekt $g^f : E \to F$ sein und $\gamma : g^f \to g$ eine passende Auswertungsabbildung. Dabei ist

(i) $F = D^B$, das Exponential in $\mathbf{SET}$,

(ii) $E = Hom_{\mathbf{SET}^{to}}(f,g) = \{(i,j), i : A \to C\; j : B \to D \text{ mit } j \circ f = g \circ i\}$ und

(iii) $g^f((i,j)) = j$.

Um zu zeigen, dass g^f tatsächlich das Exponentialobjekt ist, betrachten wir eine Funktion α von $h \times f \to g$, wobei $h : M \to N$ eine beliebige Funktion ist.

Der Pfeil α besteht aus einem Paar von Funktionen (α_1, α_2) mit $\alpha_1 : M \times A \to C$ und $\alpha_2 : N \times B \to D$. Wir suchen nun einen Pfeil $\beta = (\beta_1, \beta_2) : h \to g^f$, so dass $\gamma \circ \beta \times id_f = \alpha$.
Damit γ ein Evaluierungspfeil $\gamma = (\gamma_1, \gamma_2) : g^f \times f \to g$ ist, muss gelten

$$g \circ \gamma_1((i,j), a) = \gamma_2 \circ (g^f, f)((i,j), a).$$

Wegen $g^f((i,j)) = j$ bekommen wir $g \circ \gamma_1((i,j), a) = \gamma_2(j, f(a))$. Wählt man $\gamma_2(j, f(a)) = ev(j, f(a)) = j(f(a))$, wobei ev die Auswertungsfunktion in **SET** ist und $\gamma_1((i,j), a) = i(a)$, dann gilt

$$g \circ \gamma_1((i,j), a) = g \circ i(a) = j \circ f(a) = \gamma_2((g^f, f)((i,j), a))$$

und die Bedingung dafür, dass (γ_1, γ_2) ein Pfeil ist, ist erfüllt.
Damit (β_1, β_2) ein Pfeil von h nach g^f ist, muss für alle $a \in A$ gelten

$$\beta_2(h(m)) = g^f(\beta_1(m)) = [\beta_1(m)]_2$$

Wegen der Exponentialeigenschaft von D^B in **SET** und der Tatsache, dass gelten muss, dass $ev(\beta_2(h(m)), f(a)) = \alpha_2(h(m), f(a))$, folgt daraus, dass $\beta_2 = \Lambda\alpha_2$ und $[\beta_1(m)]_2 = \Lambda\alpha_2(h(m))$. Und aus $\gamma_1(\beta_1(m), a) = \alpha_1(m, a)$ folgt $[\beta_1(m)]_1(a) = \alpha_1(m, a)$. Damit ist β vollständig beschrieben und das Exponentialdiagramm kommutiert.

3.7 Potenzobjekte

Aufgrund seiner Ähnlichkeit zu **SET** ist ein Topos geeignet als „Universum", in dem fast alle von Mengen bekannte Konstruktionen ausgeführt werden können. Insbesondere kann man von „Relationen" oder „Potenzmengen" sprechen. In **SET** gibt es eine Isomorphie zwischen der Menge der Relationen auf $A \times B$ und der Menge der Funktionen von B nach $mathcal P(A)$. Zu jeder Relation $R \subseteq A \times B$ definiert man eine Funktion $f_R : B \to \mathcal{A}$, in dem man jedem $b \in B$ die Menge $\{a \in A \mid (b, a) \in R\}$ zuordnet. Umgekehrt erhält man für jede Funktion $f : B \to \mathcal{P}(A)$ eine Relation $R_f = \{(a, b) \mid a \in f(b)\}$. Eine spezielle Relation $\in_A \subseteq A \times \mathcal{P}(A)$ ist gegeben durch $\in_A = \{(a, U) \mid U \subseteq A, a \in U\}$. In **SET** ist dann das folgende Diagramm (Abbildung 15) ein Pullback:

Abbildung 15: Relationen

$$\begin{tikzcd}
R \arrow[r, "incl"] \arrow[d, "id_A \times f_R |_R"'] & A \times B \arrow[d, "id_A \times f_R"] \\
\in_A \arrow[r, "incl"'] & A \times \mathcal{P}(A)
\end{tikzcd}$$

Ganz allgemein ist eine zweistellige Relation auf $A \times B$ ein Monomorphismus nach $A \times B$. Damit können wir Potenzobjekte wie folgt definieren.

Definition 34. *Sei* $\mathbf{C}$ *eine CCC und* $A \in Obj(C)$. *Ein* *Potenzobjekt* *ist ein Paar* $(P(A), \in_A)$ *von* $\mathbf{C}$*-Objekten zusammen mit einem Monomorphismus* $in_A : \in_A \to A \times P(A)$, *für den gilt, dass für jedes Objekt* B *und jeden Monomorphismus* $r : R \to A \times B$ *zwei eindeutig bestimmte Pfeile* $f_R : B \to P(A)$ *und* $f'_R : R \to \in_A$ *existieren, so dass das Diagramm in Abbildung 16 ein Pullback ist.*

Abbildung 16: Potenzobjekt

$$\begin{tikzcd} R \arrow[r, "r"] \arrow[d, "f'_R"'] & A \times B \arrow[d, "id_A \times f_R"] \\ \in_A \arrow[r, "in_A"'] & A \times P(A) \end{tikzcd}$$

Satz 16. *In einem Topos ist für jedes Objekt* A *das Exponential* Ω^A *ein Potenzobjekt.*

Beweis: Wir definieren in_A durch das untere Pullback-Quadrat in Abbildung 17.

Abbildung 17: Potenzobjekt

$$\begin{tikzcd} R \arrow[r, "r"] \arrow[d, dashed, "(\Lambda\chi_r)'"'] & A \times B \arrow[d, "id_A \times \Lambda\chi_r"'] \arrow[dd, bend left=60, "\chi_r"] \\ \in_A \arrow[r, "in_A"] \arrow[d, "!"'] & A \times \Omega^A \arrow[d, "eval"] \\ T \arrow[r, "true"'] & \Omega \end{tikzcd}$$

Zu jedem Monomorphismus $r : R \to B \times A$ existiert ein Pfeil $\chi_r : B \times A \to \Omega$ und wegen der Universalitätseigenschaft des Exponentials gibt es dann auch einen Pfeil $\Lambda\chi_r : B \to \Omega^A$ mit $eval \circ \Lambda\chi_r \times id_A = \chi_r$. Da $eval \circ (\Lambda\chi_r \times id_A \circ r) = true \circ !_R$ erhält man aus der Pullbackeigenschaft von Abbildung 17 einen eindeutig bestimmten Pfeil $(\Lambda\chi_r)' : R \to \in_A$, der das obere Quadrat von Abbildung 17 kommutativ macht. Die Anwendung des Pullbacklemmas ergibt, dass das obere Quadrat dann auch ein Pullback ist. $\square$

4 Funktoren und Natürliche Transformationen

Ein Funktor F von einer Kategorie $\mathbf{C}$ in eine Kategorie $\mathbf{D}$ ist ein Graphenhomomorphismus, der die Identitäten und die Komposition erhält. Er spielt dieselbe Rolle für Kategorien wie ein Monoidhomomorphismus für Monoide und monotone Funktionen für Halbordnungen: er erhält die Struktur einer Kategorie. Der andere Grund, warum Funktoren wichtig sind ist, dass sich viele mathematisch interessante Kategorien mit Hilfe von von mengenwertigen Funktoren beschreiben lassen.

Definition 35. *Ein Funktor $F : \mathbf{C} \to \mathbf{D}$ ist ein Paar von Operationen $F_0 : Obj(\mathbf{C}) \to Obj(\mathbf{D})$ und $F_1 : \mathbf{C}_1 \to \mathbf{D}_1$, so dass für beliebige Objekte A, B, C und Pfeile $f : A \to B$, $g : B \to C$ gilt:*

1. $F_1(f) : F_0(A) \to F_0(B)$
2. $F_1(g \circ f) = F_1(g) \circ F_1(f)$
3. $F_1(id_A) = Id_{F_0(A)}$

Beispiele 16.

1. *Jeder Monoidhomomorphismus $f : M \to N$ bestimmt einen Funktor $F : \mathbf{M} \to \mathbf{N}$ der entsprechenden Kategorien. Auf dem einzigen Objekt M ist $F_0(M) = N$. Da alle Pfeile die gleiche Quelle und Ziel haben, sind sie komopositionierbar und wegen $F_1(e) = f(e)$ sind Bedingung (2)und (3) gerade die Eigenschaft, dass f ein Monoidhomomorphismus ist.*
2. *Für einen Funktor zwischen zwei Halbordnungen $(S, \leq_S)$ und $(T, \leq_T)$ muss wegen Bedingung (1) gelten: $x \leq_S y$ gdw. $F(x) \leq_T F(y)$.*
3. *Sei $\mathbf{C}$ eine kartesisch abgeschlossene Kategorie. Wir definieren $\times : \mathbf{C} \times \mathbf{C} \to \mathbf{C}$ den Kreuzproduktfunktor durch $\times_0(A, B) = A \times B$ und $\times_1(f, g) =< f, g >$.*
4. *Der Potenzmengenfunktor $\mathcal{P} : \mathbf{SET} \to \mathbf{SET}$ ordnet jeder Menge ihre Potenzmenge zu und für jeden Pfeil $f : A \to B$ ist der Pfeil $\mathcal{P}(A) \to \mathcal{P}(B)$ definiert durch $\mathcal{P}(f)(U) = \{b \in B \mid b \in f(U)\}$*
5. *Seien $\mathbf{2} + \mathbf{2}$ und $\mathbf{3}$ die Kategorien wie unten abgebildet. Dann ist der Funktor mit $F(A) = 0, F(B) = F(C) = 1, F(D) = 2$ ein Beispiel dafür, dass das Bild eines Funktors nicht unbedingt eine Teilkategorie der Zielkategorie ist, da der Kompositionspfeil von $0 \to 2$ nicht als Bild auftaucht.*

$$A \longrightarrow B \qquad C \longrightarrow D$$

$$\begin{tikzcd} 0 \arrow[rr] \arrow[dr] & & 1 \arrow[dl] \\ & 2 & \end{tikzcd}$$

6. *Die Kategorie* **Cat** *hat als Objekte kleine Kategorien und als Pfeile Funktoren. Die Komposition von Funktoren ist ihre Komposition als Graphenhomomorphismen.*

7. *Die Einbettungsabbildung einer Teilkategorie ist ein Funktor. Die Objekte und Pfeile einer Teilkategorie müssen nicht notwendigerweise tatsächlich Objekte und Pfeile der Zielkategorie sein, es muss nur einen Monomorphismus von der Teilkategorie in die größere Kategorie geben. So ist z.B.* **SET** *eine Teilkategorie in* **REL**. *Der monomorphe Funktor bildet jede Menge auf sich selbst ab, und ordnet jeder Funktion f ihren Graphen zu. Das etwas seltsame Ergebnis dieses Ansatzes ist, dass bei zwei verschiedene Kategorien jede als Teilkategorie der anderen betrachtet werden kann.(siehe Aufgabe 1 und 2)*

8. *Will man einen Teil der Struktur einer Kategorie vernachlässigen, so kann man diese Kategorie in eine mit weniger Struktur abbilden. Einen Funktor, der soetwas ermöglicht, nennt man einen Vergissfunktor. Zum Beispiel ist der Funktor U :* **Mon** $\to$ **Sem**, *der die Monoide in die Halbgruppen einbettet und dabei „vergisst“, dass Monoide ein neutrales Element haben, ein Vergissfunktor. Ein anderes Beispiel ist der Funktor U :* **GRP** $\to$ **SET**, *der jeder Gruppe die unterliegende Menge zuordnet und* jede *andere Struktur „vergisst“. Ein solcher Funktor ist i.A. nicht injektiv in der Objektabbildung. Wir werden keine formale Definition von Vergissfunktoren angeben. Allgemein darf man aber voraussetzen, dass ein Vergissfunktor treu*[14] *ist.*

9. *Der freie Monoidfunktor von* **SET** *in die Kategorie der Monoide ordnet jeder Menge A das freie Monoid A^* über dieser Menge zu, mit der Konkatenation als Monoidoperation. Das Bild $F(f) = f^*$ einer Funktion f wird definiert durch $f^*(a_1 \circ a_2 \circ \ldots \circ a_n) = f(a_1) \circ \ldots \circ f(a_n)$. Insbesondere ist $f^*() = ()$. Um sich davon zu überzeugen, dass F tatsächlich ein Funktor ist, muss man überprüfen, dass $F(g \circ f) = F(g) \circ F(f)$ für kompositionierbare Pfeile g und f. Das folgt aber unmittelbar aus der Definition. Offenbar ist auch die Bildung der Kleene'sche Hülle selbst ein Funktor. Dieser lässt sich darstellen als Komposition $U \circ F$ aus F und dem Vergissfunktor U :* **Mon** $\to$ **SET**.

4.1 Typen von Funktoren

Definition 36. *Sei F :* **C** $\to$ **D** *ein Funktor.*

1. *F heißt treu, wenn für alle* **C***-Objekte A, B und für alle Pfeile $f, g \in \mathbf{C}[A, B]$ gilt, dass aus $F(f) = F(g)$ folgt $f = g$.*

2. *F heißt voll falls für alle* **C***-Objekte A, B und für alle Pfeile $h \in \mathbf{D}[F(A), F(B)]$ ein Pfeil $g \in \mathbf{C}[A, B]$ existiert, mit $h = F(g)$.*

3. *F heißt volle Einbettung, falls er treu und voll ist.*

[14] siehe Definition 36

4.1.1 Freie Funktoren

Außer dem in Beispiel 9 gegebenem Funktor gibt es noch eine ganze Reihe anderer freie Funktoren. Um zu charakterisieren, was ein freier Funktor ist, betrachten wir zunächst den Funktor $F : \mathbf{Graph} \to \mathbf{Cat}$ dessen Objektabbildung jedem Graphen $\mathcal{G}$ die zu $\mathcal{G}$ gehörende freie Kategorie $\mathbf{G}^*$ zuordnet (vergleiche Beispiel 8). Für die Pfeilabbildung F_1 betrachten wir einen Graphenhomomorphismus $\varphi : \mathcal{G} \to \mathcal{H}$ mit der Knotenabbildung $\varphi_0 : G_0 \to H_0$ und der Kantenabbildung $\varphi_1 : G_1 \to H_1$ und definieren $(F_1(\varphi))_0 = \varphi_0$. Nehmen wir nun an, dass $(f_n, f_{n-1}, \dots f_1)$ ein Pfad in $\mathcal{G}$ ist und damit ein Pfeil in $F_0(\mathcal{G})$. Da Funktoren Quelle und Ziel von Pfeilen erhalten, können wir definieren $(F_1(\varphi))_1((f_n, f_{n-1}, \dots f_1)) = (\varphi_1(f_n), \varphi_1(f_{n-1}), \dots \varphi_1(f_1))$ und wissen, dass wir einen Pfad in $\mathcal{H}$ bekommen. Offenbar erhält F die Komposition von Pfaden. Es gibt einen speziellen Graphenhomomorphismus $\eta_\mathcal{G} : \mathcal{G} \to U(F(\mathcal{G}))$, der den Graphen $\mathcal{G}$ in den unterliegenden Graphen der Kategorie $F_0(\mathcal{G})$ einbettet. Die Abbildung $(\eta_\mathcal{G})_0$ ist die Identität, da die Objekte von $F_0(\mathcal{G})$ gerade die Knoten des Graphen sind und für jeden Pfad f in $\mathcal{G}$ ist $(\eta_\mathcal{G})_1(f)$ der Pfad (f) der Länge eins. Das ist eine Inklusionspfeil im kategorialen Sinn, da f und (f) wirklich zwei verschiedene Dinge sind.

Satz 17. *Sei $\mathcal{G}$ ein Graph und $\mathbf{C}$ eine Kategorie. Dann gibt es für jeden Graphenhomomorphismus $h : \mathcal{G} \to U(\mathcal{G}$ genau einen Funktor $H : F(\mathcal{G}) \to \mathbf{C}$ mit $U(H) \circ \eta_\mathcal{G} = h$*

Beweis: Falls () der leere Pfad im Objekt A, dann setze man $H(()) = id_A$ Für jedes Objekt A von $F(\mathcal{G})$(also jeden Knoten von $\mathcal{G}$) definiert man $H(A) = h(A)$ und das Bild unter H für einen Pfad $(A_n, A_{n-1}, \dots A_1)$ ist die Anwendung von h:

$$H(A_n, A_{n-1}, \dots A_1) = (h(A_n), h(A_{n-1}), \dots h(A_1)).$$

Für jeden Knoten A gibt es einen leeren Pfad. Kombiniert man ihn mit irgendeinem Pfad p von A nach B, erhält man p, die umgekehrte Zusammensetzung geht analog. □

4.1.2 Potenzmengenfunktoren

Jede Menge M hat eine Potenzmenge $\mathcal{P}(M)$. Es gibt drei verschiedene Funktoren, deren Objektabbildung einer Menge ihre Potenzmenge zuordnet; sie unterscheiden sich aber durch ihre Pfeilabbildungen. Derjenige der Funktoren, der besonders für die Topostheorie interessant ist, nennen wir *den* Potenzmengenfunktor.

Definition 37. *Der Potenzmengenfunktor $\mathcal{P} : \mathbf{SET}^{op} \to \mathbf{SET}$ ordnet jeder Menge ihre Potenzmenge zu und für jede Funktion $f : A \to B$ ist $\mathcal{P}(f)$ die Urbildfunktion $f^{-1} : \mathcal{P}(B) \to \mathcal{P}(A)$, wobei für eine Menge $C \subseteq B$ $f^{-1}(C) = \{a \in A \mid f(a) \in C$*

f^{-1} wird in der Literatur häufig auch mit f^* bezeichnet.

Die anderen beiden Funktoren, die die Objekte von **SET** auf deren Potenzmengen abbilden sind covariant. Der erste heißt direktes Bild und bildet $f : A \to B$ ab auf

$$f_* : \mathcal{A} \to \mathcal{P}(B) \text{ mit } f_*(A_0) = \{f(x) \mid x \in A_0\}$$

Das Bild einer Funktion unter dem universellen Bildfunktor erfaßt nur diejenigen Elemente von B, die ausschließlich Bilder aus Elementen von A_0 sind. f wird abgebildet auf $f_!$ mit

$$f_!(A_0) = \{y \in B \mid \text{ aus } f(x) = y \text{ folgt } x \in A_0\} = \{y \in B \mid f^{-1}(y) \subseteq A_0\}.$$

4.1.3 Hom-Funktoren

Definition 38. *Sei* $\mathbf{C}$ *eine Kategorie mit Objekten* A, B, C *und einem Pfeil* $f : A \to B$ *wir definieren einen (covarianten) Hom-Funktor* $Hom(C, -) : \mathbf{C} \to \mathbf{SET}$ *wie folgt:*

1. $Hom(C, -)(A) = Hom(C, A)$

2. $Hom(C, -)(f) = Hom(C, f) : Hom(C, A) \to Hom(C, B)$ *mit* $Hom(C, f)(g) = f \circ g$ *für jedes* $g \in Hom(C, A)$

Wir überzeugen uns, dass wir tatsächlich einen Funktor definiert haben. Für die Identität gilt: $Hom(C, id_A)(g) = id_A \circ g = g$ also ist $Hom(C, id_A) = id_{Hom(C,A)}$. Komposition wird erhalten: für Pfeile $f : A \to B$ und $g : B \to D$ ist $(Hom(C, g) \circ Hom(C, f))(h) = g \circ f \circ h$ für jedes $h \in Hom(C, A)$.
Für jedes Objekt C gibt es einen Hom-Funktor, C ist also ein Parameter in einer ganzen Familie von Funktoren. Der Strich $-$ dient hier als Platzhalter für das Argument. Eine analoge Definition aus der Mathematik wäre $f(-) = (-)^n$.

Ein Funktor $F : \mathbf{C}^{op} \to \mathbf{D}$ ist ein sogenannter kontravarianter Funktor von $\mathbf{C}$ nach $\mathbf{D}$. Ein solcher Funktor wird oft mit Ausdrücken aus $\mathbf{C}$ beschrieben. Man kann auch einen kontravarianten Hom-Funktor definieren wie folgt.

Definition 39. *Für ein Objekt* $D \in \mathbf{C}^{op}$ *wird der kontravariante Hom-Funktor* $Hom(-, D) : \mathbf{C}^{op} \to \mathbf{SET}$ *wie folgt:*

$$Hom(-, D)(A) = Hom(A, D) \text{ für jedes Objekt } A$$

$$Hom(-, D)(f)(g) = Hom(f, D)(g) = g \circ f \text{ für jeden Pfeil } f : A \to B \text{ und } g : B \to D$$

Definition 40. *Der zweistellige Hom-Funktor* $Hom(-, -) : \mathbf{C}^{op} \times \mathbf{C} \to \mathbf{SET}$ *bildet Paare von Objekten* (C, D) *ab auf* $Hom(C, D)$ *und für ein Paar von Pfeilen* $f : C \to A$ *und* $g : B \to D$ *ist* $Hom(f, g) : Hom(A, B) \to Hom(C, D)$ *für ein* $h : A \to B$ *definiert man*

$$Hom(f, g)(h) = g \circ h \circ f$$

was einen Pfeil von C *nach* D *ergibt.*

Aufgaben:

1. Zeige dass die Kategorien **Sem** der Halbgruppen und **Mon** der Monoide Teilkategorien voneinander sind.

2. Zeige, dass **SET** und **Pfn** die Kategorie der Mengen mit partiellen Funktionen Teilkategorien voneinander sind. (Hinweis: Um einen Monomorphismus von **Pfn** nach **SET** zu konstruieren, bilde man jede Menge A auf die Menge $A \cup \{A\}$ ab und die partielle Funktion $f : A \to B$ auf die Funktion f' mit $f'(x) = f(x)$, wo das definiert ist und $f(x) = B$ wo nicht.)

4.2 Aktionen von Monoiden

In diesem Abschnitt geht es um mengenwertige Funktoren, als natürliche Verallgemeinerung von Maschinen mit endlich vielen Zuständen. Mengenwertige Funktoren sind wegen des Yoneda-Lemmas auch aus theoretische Gründen wichtig in der Kategorientheorie.

Definition 41. *Sei M ein Monoid mit neutralem Element 1. Die Aktion von M auf einer Menge S ist eine Funktion $\alpha : M \times S \to S$, die folgende Bedingungen erfüllt:*

1. $\alpha(1, s) = s$ *für alle* $s \in S$.
2. $\alpha(mn, s) = \alpha(m, \alpha(n, s))$ *für alle* $m, n \in M$ *und* $s \in S$

Schreibt man wie in der Mathematik üblich ms anstelle von $\alpha(m, s)$ dann haben die beiden Bedingungen die Form $1s = s$ und $(mn)s = m(ns)$, und man nennt eine solche Menge S auch *$M - Menge$*. Es ist hilfreich, sich die Menge S als eine Menge von Zuständen und die Elemente von M als Aktionen vorzustellen, die den Übergang von einem Zustand zu einem anderen bewirken.

Definition 42. *Sei M ein Monoid und S und T zwei M-Mengen. Eine equivariante Abbildung von S nach T ist eine Abbildung $\phi : S \to T$ mit $m\phi(s) = \phi(ms)$.*

Die Identitätsfunktion ist eine equivariante Abbildung und ebenso die Komposition zweier equivarianten Abbildungen. Daher kann man für jedes Monoid eine Kategorie $\mathbf{M - Act}$ bilden, deren Objekte die M-Mengen und deren Pfeile die equivarianten Abbildungen sind.

Offenbar kann man die Aktionen eines Monoids als Funktor auffassen. Sei $C(M)$ die durch M erzeugte Kategorie. Dann definiert eine Aktion α einen Funktor $F_\alpha : C(M) \to$ **SET** mit $F_\alpha(*) = S$ und $F_\alpha(m) = s \mapsto \alpha(m, s)$.

Die Funktoren sind die passenden Pfeile für eine Kategorie, deren Objekte kleine Kategorien sind. Diese Kategorie der Kategorien wird mit **CAT** bezeichnet.

4.3 Kommutative Diagramme

Definition 43. *Seien $\mathcal{G}$ und $\mathcal{H}$ Graphen. Ein Diagramm in $\mathcal{G}$ vom Schema $\mathcal{H}$ ist ein Homomorphismus $D : \mathcal{H} \to \mathcal{G}$.*

Wenn der Zielgraph eines Diagramms der Graph einer unterliegenden Kategorie ist, kann man das Konzept eines kommutativen Diagramms formulieren, was der kategorielle Weg ist, um Gleichungen auszudrücken. Wir sagen ein Diagramm kommutiert, wenn für alle Pfade zwischen zwei Knoten gilt, dass sie denselben Pfeil beschreiben.

Beispiele 17. *Assoziativität: Die Tatsache, dass die Multiplikation in einem Halbgruppe assoziativ ist, lässt sich dadurch ausdrücken, dass ein bestimmtes Diagramm in* **SET** *kommutiert. Sei S eine Halbgruppe. Wir definieren die folgenden Funktionen:*

1. $mult : S \times S \to S$ *mit* $mult(x, y) = xy$
2. $S \times mult : S \times S \times S \to S \times S$ *mit* $S \times mult(x, y, z) = (x, yz)$
3. $mult \times S : S \times S \times S \to S \times S$ *mit* $mult(x, y, z) \times S = (xy, z)$

$$
\begin{tikzcd}
S \times S \times S \arrow[r, "S \times mult"] \arrow[d, "mult \times S"'] & S \times S \arrow[d, "mult"] \\
S \times S \arrow[r, "mult"'] & S
\end{tikzcd}
$$

Dass das folgende Diagramm kommutiert, ist gerade das Assoziativgesetz.

Normalerweise wird die Assoziativität durch eine Gleichung $(xy)z = x(yz)$ ausgedrückt, die von allen Elementen der Halbgruppe erfüllt werden muss. Dasselbe drückt das Diagramm aus, aber ohne Variablen zu benutzen. Natürlich wurden bei der Definition der Funktionen Variablen verwendet, aber auch das lässt sich vermeiden, wenn man konsequent die Definition des Produktes, wie sie in Definition 17 gegeben wurde, benutzt.

Der sauberste Weg, Diagramme zu definieren ist zu sagen, dass ein Diagramm in einer Kategorie $\mathbf{C}$ ein Funktor $F : \mathcal{E} \to \mathcal{C}$ ist. Da jeder Funktor insbesondere ein Graphenhomomorphismus, zwischen den von den Kategorien erzeugten Graphen ist, stimmen beide Definitionen überein.

4.4 Natürliche Transformationen

Laut Definition ist $F : \mathbf{C} \to \mathbf{D}$ ein Funktor, wenn $F(id_C) = id_{F(C)}$ und $F(f \circ g) = F(f) \circ F(g)$ für alle Objekte C und Pfeile f, g in $\mathbf{C}$. Diese Forderungen entsprechen dem folgenden Diagramm:

$$
\begin{tikzcd}
A \arrow[dr, "f"] \arrow[dd, "g \circ f"'] & & & F(A) \arrow[dr, "F(f)"] \arrow[dd, "F(g \circ f)"'] & \\
& B \arrow[dl, "g"] & & & F(B) \arrow[dl, "F(g)"] \\
C & & & F(C) &
\end{tikzcd}
$$

Betrachtet man nun zwei verschiedene Funktoren $F, G : \mathbf{C} \to \mathbf{D}$, kann man mit der folgenden Skizze eine „Übersetzung“ beschreiben, wobei die gestrichelten Linien kommutative Diagramme herstellen.

$$
\begin{tikzcd}
A \arrow[dd, "g\circ f"'] \arrow[dr, "f"] & \\
& B \arrow[dl, "g"] \\
C &
\end{tikzcd}
\qquad
\begin{tikzcd}
F(A) \arrow[dddd, "F(g\circ f)"'] \arrow[ddr, "F(f)"] \arrow[drr, dotted] & & & \\
& & G(A) \arrow[ddr, "G(f)"] \arrow[dddd, "G(g\circ f)"'] & \\
& F(B) \arrow[ddl, "F(g)"] \arrow[drr, dotted] & & \\
& & & G(B) \arrow[ddl, "G(g)"] \\
F(C) \arrow[drr, dotted] & & & \\
& & G(C) &
\end{tikzcd}
$$

Das heißt, eine „Übersetzung“ der Bilder der Funktoren ineinander kann definiert werden, indem man zu jedem Objekt $C \in Obj(\mathbf{C})$ einen **D**-Pfeil findet, der die entsprechenden Diagramme kommutativ macht.

Definition 44. *Seien $F, G : \mathbf{C} \to \mathbf{D}$ Funktoren. Eine natürliche Transformation $\tau : F \dot{\to} G$ ist eine Familie von **D**-Pfeilen $\tau_C : F(C) \to G(C)$, so dass für alle $f \in Hom_{\mathbf{C}}(A, B)$ gilt $\tau_B \circ F(f) = G(f) \circ \tau_A$.*

Beispiele 18. *Sei **C** eine kleine Kategorie. Für jedes $h \in Hom_{\mathbf{C}}(A', A)$ ist die Menge $\{Hom_{\mathbf{C}}(h, B)\}_B$ der Pfeile von $Hom_{\mathbf{C}}(A, B) \to Hom_{\mathbf{C}}(A', B)$ eine natürliche Transformation $Hom_{\mathbf{C}}(h, -) : Hom(A, -) \dot{\to} Hom_{\mathbf{C}}(A', -)$ zwischen den kontravarianten Hom-Funktoren $Hom(A, -), Hom_{\mathbf{C}}(A', -) : \mathbf{C} \to \mathbf{SET}$ wie das folgende Diagramm zeigt.*

$$A \xleftarrow{\;h\;} A'$$

$$
\begin{tikzcd}
B \arrow[d, "k"'] \\
B'
\end{tikzcd}
\qquad
\begin{tikzcd}
Hom(A,B) \arrow[r, "{Hom(h,B) = - \circ h}"] \arrow[d, "{Hom(A,k) = k \circ -}"'] & Hom(A',B) \arrow[d, "{Hom(A',k) = k \circ -}"] \\
Hom(A,B') \arrow[r, "{Hom(h,B') = - \circ h}"'] & Hom(A',B')
\end{tikzcd}
$$

Dasselbe Diagramm zeigt, dass für ein gegebenes $k : B \to B'$ die Menge der Homomorphismen $\{Hom_{\mathbf{C}}(A, k)\}_A$ eine natürliche Transformation

$$Hom_{\mathbf{C}}(-, k) : Hom\mathbf{C}(-, B) \dot{\to} Hom\mathbf{C}(-, B')$$

definiert.

Natürliche Transformationen lassen sich verknüpfen durch $(\tau \circ \beta)_C = \tau_C \circ \beta_C$. Man nennt diese Verknüpfung auch vertikale Komposition im Gegensatz zur horizontalen Komposition, die wir später noch kennenlernen werden. Da es für jeden Funktor eine identische

Transformation $\tau : F \dot{\to} F$ gibt, mit $\tau_A = id_{F(A)}$ kann man Funktoren in Funktorkategorien zusammenfassen.

Definition 45. *Seien $\mathbf{C}, \mathbf{D}$ Kategorien, dann ist die Funktorkategorie $\mathbf{Func}[\mathbf{C}, \mathbf{D}]$ die Kategorie, deren Objekte Funktoren von $\mathbf{C}$ nach $\mathbf{D}$ und deren Morphismen natürliche Transformationen zwischen zwei solchen Funktoren sind.*

Die Menge $Hom_{\mathbf{Func}[\mathbf{C},\mathbf{D}]}(F, G)$ aller natürlichen Transformationen zwischen zwei Funktoren F und G wird oft auch einfach mit $\mathbf{Nat}(F, G)$ bezeichnet.

Zwei Funktoren sind natürlich isomorph, wenn sie isomorph sind als Objekte der Funktorkategorie. Zum Beispiel weiß man, dass eine Menge A isomorph ist zu $A \times \{*\}$. Für kartesische Kategorien mit Terminalobjekt T bedeutet das, dass der Funktor $- \times T$ isomorph zum identischen Funktor $\mathbf{Id}$ ist.
Wenn $F : \mathbf{C} \to \mathbf{D}$ eine volle Einbettung ist, dann ist $\mathbf{C}$ eine volle Teilkategorie von $\mathbf{D}$.
Der Begriff des natürlichen Isomorphismus von Funktoren erlaubt uns die Definition der „Äquivalenz“ von Kategorien zu geben, die die Vorstellung von „im wesentlichen gleich“ besser beschreibt als die Isomorphie von Kategorien. Zwei Kategorien $\mathbf{C}$ und $\mathbf{D}$ sind äquivalent, genau dann, wenn es zwei Funktoren $F : \mathbf{C} \to \mathbf{D}$ und $G : \mathbf{D} \to \mathbf{C}$ gibt, so dass $F \circ G \cong \mathbf{Id}_\mathbf{D}$ und $G \circ F \cong \mathbf{Id}_\mathbf{C}$. (Erinnerung: $\mathbf{C}$ und $\mathbf{D}$ sind isomorph, wenn $F \circ G = \mathbf{Id}_\mathbf{D}$ und $G \circ F = \mathbf{Id}_\mathbf{C}$.

Beispiele 19. *Sei $\mathbf{C}$ die Kategorie mit zwei Objekten A, B und vier Pfeilen $id_A.id_B, i : A \to B, j : B \to A$, (womit automatisch $i \circ j = id_A, j \circ i = id_B$). Sei $\mathbf{D}$ die Kategorie mit einem Objekt E und dem Identitätspfeil auf E. Der eindeutig bestimmte Funktor $F : \mathbf{C} \to \mathbf{D}$ hat zwei Pseudokomplemente, die das einzige Objekt auf entweder A oder B abbilden.*

Satz 18. *Seien $F, G : \mathbf{C} \to \mathbf{D}$ Funktoren und sei $\tau_A : Hom_\mathbf{D}(F(A), G(A))$ ein Isomorphismus für jedes $A \in Obj(\mathbf{C})$. Dann ist τ ein natürlicher Isomorphismus.*

Beweis: Wir definieren $\tau^{-1} : G \dot{\to} F$ durch $(\tau^{-1})_A = (\tau_A)^{-1}$. Tatsächlich ist τ^{-1} eine natürliche Transformation, da

$$\begin{aligned}(\tau^{-1})_B \circ G(f) &= (\tau_B)^{-1} \circ G(f) \circ \tau_A \circ (\tau_A)^{-1} \\ &= (\tau_B)^{-1} \circ \tau_B \circ F(f) \circ (\tau_A)^{-1} \\ &= F(f) \circ (\tau^{-1})_A\end{aligned}$$

□

Ein Beispiel für natürliche Isomorphien ist das sogenannte Lifting. Die mengentheoretische Idee dahinter ist, dass man ein Objekt(eine Menge) liftet, indem man ein zusätzliches (kleinstes) Element hinzufügt. Dann ist jede Hom-Menge von totalen Funktionen isomorph zur entsprechenden Hom-Menge der partiellen Funktionen.

Beispiele 20. *Sei $\mathbf{C}$ eine Kategorie mit partiellen Abbildungen und $\mathbf{C}_t$ die zugehörige Kategorie mit totalen Abbildungen, und $INT : \mathbf{C}_t \to \mathbf{C}$ der Einbettungsfunktor. Das Lifting eines Objektes $A \in Obj(\mathbf{C})$ ist ein Objekt A° so dass die Funktoren $Hom_\mathbf{C}(-, A) \circ Int, Hom_{\mathbf{C}_t}(-, A^\circ) : C_t \to \mathbf{SET}$ natürlich isomorph sind. Nach Definition einer natürlichen Transformation und des Hom-Funktors, existiert eine Familie $\{\tau_C\}$, so dass das folgende Diagramm für jedes totale $f : C \to B$ kommutiert:*

$$\begin{tikzcd}
Hom_{\mathbf{C}}(B,A) \arrow[r, "\tau_B"] \arrow[d, "- \circ f"'] & Hom_{\mathbf{C}_t}(B,A^\circ) \arrow[d, "- \circ f"'] \\
Hom_{\mathbf{C}}(C,A) \arrow[r, "\tau_B"'] & Hom_{\mathbf{C}_t}(C,A^\circ)
\end{tikzcd}$$

Auch das Lifting selbst kann man als Funktor betrachten. Jedem Objekt aus $\mathbf{C}$ *wird ein Lifting-Objekt* A° *zugeordnet. Mit* $ex_A = \tau_{A^\circ}^{-1}(id)$ *erhalten wir einen Pfeil* $ex_A : A^\circ \to A$.

Der Lifting Funktor für **pSET** *ist offensichtlich. Generell lassen sich Liftings für Kategorien mit partiellen Morphismen definieren.*

Sei $\mathbf{C}$ *eine Kategorie mit Pullbacks. Ein partieller Morphismus zwischen zwei Objekten* A *und* B *wird definiert als ein Paar von* $\mathbf{C}$*-Pfeilen* (h, g) *mit* $h : D \to A$ *Monomorphismus und* $g : D \to B$*. Wenn* $h' : D' \to B$ *und* $g' : D' \to C$ *wird die Komposition von* $f = (h, g)$ *und* $f' = (h', g') : B \to C$ *definiert als* $(h \circ h'', g' \circ g'')$ *wobei* $h'' : D \times_B D' \to D$ *und* $g'' : D\times_B \to D'$ *die Projektionen des Pullbacks von* g *entlang* h' *sind. Oftmals ist es sinnvoll, nicht alle möglichen solche partiellen Morphismen zu betrachten, sondern nur eine Teilmenge davon.*

So ist **pPO** *die Kategorie der Halbordnungen mit partiellen monotonen Funktionen, deren Definitionsbereich nach oben abgeschlossen ist. Für den Liftingfunktor, ergänzt man jedes Objekt um ein neues kleinstes Element. Wegen der Monotonie existiert ein solcher Lifting-Funktor nicht, wenn man nicht annimmt, dass die Domänen nach oben hin abgeschlossen sind. In der Katgorie* **pCPO***, der vollständigen Halbordnungen, deren gerichtete Mengen nicht leer sind, sind die Morphismen partielle stetige Funktionen, deren Definitionsbereich eine offene Menge ist. Der Lifting-Funktor sieht aus wie der für* **pPO**.

Ein komplexeres Beispiel gibt die Kategorie **EN** *der aufzählbaren Mengen ab. Sei* **pEN** *die Kategorie der aufzählbaren Mengen mit partiellen Funktionen. Sei* $\underline{A} = (A, e) \in Obj(\mathbf{pEN})$*. Wir definieren* $\underline{A^\circ} = (A^\circ, e^\circ)$ *durch Hinzufügen eines Elementes* $\perp$ *zur Menge* A *und definieren*

$$e^\circ = \begin{cases} e(\varphi_n(0)) & \textit{falls } \varphi_n(0) \textit{ konvergiert,} \\ \perp & \textit{sonst.} \end{cases}$$

Offenbar ist $e^\circ : \mathbb{N} \to A^\circ$ *surjektiv. Sei nun* $\underline{B} = (B, e_B)$ *und* $f \in Hom_{\mathbf{pEN}}(\underline{B}, \underline{A})$*. Nach Definition existiert eine partiell rekursive Funktion* f' *mit* $f \circ e_B = e_A \circ f'$*. Wir definieren eine Erweiterung* $\underline{f} \in Hom_{\mathbf{EN}}(B, A^\circ)$ *von* f*, indem wir die total rekursive Funktion* $\underline{f'}$ *angeben, die* $\underline{f}$ *definiert. Dazu setzen wir* $\varphi_{\underline{f'(n)}}(0) = f(n)$*. Wegen des* $s-m-n$*-Theorems muss ein solches* $\underline{f'}$ *existieren und wir bekommen:*

$$\underline{f}(e_B)(n)) = e_A(\underline{f'}(n)) = \begin{cases} e_A(\varphi_{\underline{f'(n)}}(0)) & \textit{falls } \underline{f'}(n)(0) \textit{ konvergiert} \\ \perp & \textit{sonst.} \end{cases}$$

Sind schon natürliche Transformationen zwischen Funktoren bekannt, so kann man aus der Komposition mit passenden anderen Funktoren neue natürliche Transformationen konstruieren.

Sei $\tau : F \xrightarrow{\cdot} G$ eine natürliche Transformation zwischen F und G mit $F, G : \mathbf{B} \to \mathbf{C}$. Für jeden Funktor $H : \mathbf{A} \to \mathbf{B}$ ist $\{\tau H\}_{A \in Obj(\mathbf{A})}$ mit $(\tau H)_A = \tau_{H(A)}$ eine natürliche Transformation von $F \circ H \xrightarrow{\cdot} G \circ H$ und Für jeden Funktor $K : \mathbf{C} \to \mathbf{D}$ ist $\{K\tau\}_B$ mit $(K\tau)_B = K(\tau_B)$. Dass es sich dabei tatsächlich um natürliche Transformationen handelt, sieht man an den folgenden Diagrammen. Seien $g : A \to A'$ ein Pfeil in $\mathbf{A}$ und $f : B \to B'$ ein Pfeil in $\mathbf{B}$, dann bekommen wir:

$$
\begin{tikzcd}
A \arrow[d, "g"'] & FH(A) \arrow[r, "\tau_{H(A)}"] \arrow[d, "FH(g)"'] & GH(A) \arrow[d, "GH(g)"] \\
A' & FH(A') \arrow[r, "\tau_{H(A')}"'] & GH(A')
\end{tikzcd}
\qquad
\begin{tikzcd}
B \arrow[d, "f"'] & KF(B) \arrow[r, "K(\tau_B)"] \arrow[d, "KF(f)"'] & KG(B) \arrow[d, "KG(f)"'] \\
B' & KF(B') \arrow[r, "K(\tau'_B)"'] & KG(B')
\end{tikzcd}
$$

Die Kommutativität der Diagramme folgt jeweils aus der Eigenschaft der Funktoren, dass sie die Komposition „übertragen“ also, dass $F(g \circ f) = F(g) \circ F(f)$ für passende Funktoren F und Pfeile f, g.

Betrachten wir nun die folgende Situation. Gegeben sind zwei Funktorpaare $F, G : \mathbf{B} \to \mathbf{C}$ und $H, K : \mathbf{C} \to \mathbf{D}$ sowie zwei natürliche Transformationen $\tau : F \xrightarrow{\cdot} G$ und $\sigma : H \xrightarrow{\cdot} K$. Wegen der Natürlichkeit von σ und da $H(\tau_B)$ und $K(\tau_B)$ Pfeile in $\mathbf{D}$ sind, kommutiert das folgende Diagramm für jedes $B \in Obj(\mathbf{B})$.

$$
\begin{tikzcd}
HF(B) \arrow[r, "\sigma_{F(B)}"] \arrow[d, "H(\tau_B)"'] \arrow[dr, dashed, "(\sigma\tau)_B"] & KF(B) \arrow[d, "\sigma_{G(B)}"] \\
HG(B) \arrow[r, "K(\tau_B)"'] & KG(B)
\end{tikzcd}
$$

Wegen der Kommutativität ist der gestrichelte Pfeil $(\sigma\tau)_B : HF(B) \to KG(B)$ eindeutig bestimmt. Man überzeugt sich leicht davon, dass die Familie dieser Pfeile eine natürliche Transformation $\sigma\tau : HF \xrightarrow{\cdot} KG$ bildet.

$\sigma\tau$ mit $(\sigma\tau)_B = \sigma_{G(B)} \circ H(\tau_B) = K(\tau_B) \circ \sigma_{F(B)}$ heißt horizontale Komposition von τ und σ.

4.5 Noch einmal: kartesische und kartesisch abgeschlossene Kategorien

Nach Definition ist eine Kategorie kartesisch, wenn sie ein Terminalobjekt hat und für jedes Paar (A, B) von Objekten das Kreuzprodukt $A \times B$ existiert. Die Universalitätseigenschaft stellt einen Isomorphismus zwischen Paaren von Pfeilen (f, h) mit $f : C \to A$ und $g : C \to B$ zu Pfeilen $< f, g >: C \to A \times B$ her. Dieser Isomorphismus ist eine natürliche Transformation

$$< -, - >: Hom_{\mathbf{C}\times\mathbf{C}}((-, -), (A, B)) \circ \triangle \dot{\to} Hom_{\mathbf{C}}(-, A \times B).$$

Dabei ist $\triangle$: $\mathbf{C} \to \mathbf{C}\times\mathbf{C}$ mit $\triangle\ (C) = (C, C)$ der sogenannte Diagonalfunktor. Sei $k : C' \to C$, dann folgt die Natürlichkeit aus:

$$< f, g >_C \circ k =< p_1 \circ < f, g >_C \circ k, p_2 \circ < f, g >_C \circ k >_{C'} =< f \circ k, g \circ k >_{C'}$$

Sei τ ein natürlicher Isomorphismus

$$\tau : Hom_{\mathbf{C}}(-, A \times B) \dot{\to} Hom_{\mathbf{C}\times\mathbf{C}}(-, (A, B)) \circ \triangle$$

und sei τ^{-1}, dessen Inverses. Dann gilt:

$$\begin{tikzcd}
Hom_{\mathbf{C}\times\mathbf{C}}((C, C)(A, B)) \arrow[d, "< -, - > \circ (k, k)"'] & Hom_{\mathbf{C}}(C, A \times B) \arrow[l, "\tau_C^{-1}"'] \arrow[d, "- \circ f"] & C \\
Hom_{\mathbf{C}\times\mathbf{C}}((C', C')(A, B)) & Hom_{\mathbf{C}}(C', A \times B) \arrow[l, "\tau_{C'}^{-1}"'] & C' \arrow[u, "k"']
\end{tikzcd}$$

Sei $(q_1, q_2) = \tau^{-1}_{A\times B}(id_{A\times B})$. Dann ist $q_1 : A \times B \to A$ und $q_2 : A \times B \to B$. Wir zeigen, dass q_1 und q_2 Projektionen für das Produkt $A \times B$ sind. Wir wählen für k im obigen Diagramm ein $k = \tau_{C'}(h, g) : C' \to A \times B$. Dann bekommen wir $(q_1, q_2) \circ (\tau_{C'}(h, g), \tau_{C'}(h, g)) = \tau_C^{-1}(id_{A\times B} \circ \tau_{C'}(h, g)) = (h, g)$ und insbesondere $q_1 \circ \tau_{C'}(h, g) = h$ und $q_2 \circ \tau_{C'}(h, g) = g$. Diese Überlegung rechtfertigt die folgende Aussage:

Satz 19. *Eine Kategorie mit Terminalobjekt ist kartesisch, genau dann, wenn es für jedes Paar von Objekten A, B ein Objekt $A \times B$ und einen natürlichen Isomorphismus $< -, - >: Hom_{\mathbf{C}} \times Hom_{\mathbf{C}}(--, (A, B)) \circ \triangle \dot{\to} Hom_{\mathbf{C}}(-, A \times B)$ gibt.*

Ganz ähnlich ist die Situation mit dem Exponential. Ein Exponential für ein Paar A, B von Objekten ist ein Objekt B^A zusammen mit einem Auswertungspfeil $ev : B^A \times A \to B$, so dass für jedes Objekt C und jeden Pfeil $f : C \times A \to B$ ein Pfeil $\Lambda_C f' : C \to B^A$ existiert, mit

$$\begin{array}{ll} [\beta] & ev_{A,B} \circ (\Lambda_C f \times id) = f \\ [\eta] & \Lambda_C(ev_{A,B} \circ (h \times id)) = h \end{array}$$

Es stellt sich heraus, dass Λ_C natürlich in C ist, also dass $\Lambda : Hom_{\mathbf{C}}(,B) \circ \times A \dot{\to} Hom_{\mathbf{C}}(-, B^A)$ ein natürlicher Isomorphismus ist. Der inverse Homomorphismus ist $\Lambda^{-1} = ev \circ (- \times id)$.

Satz 20. *Eine kartesische Kategorie ist kartesisch abgeschlossen, genau dann, wenn es für jedes Paar von Objekten A, B ein Objekt B^A und einen natürlichen Isomorphismus $\Lambda : Hom_{\mathbf{C}}(- \times A, B) \dot{\rightarrow} Hom_{\mathbf{C}}(-, B^A)$ gibt.*

Es ist einfach zu zeigen, dass die folgenden natürlichen Isomorphien in allen CCC-s gelten für beliebige Objekte A, B, C.

1. $A \cong A$
2. $T \times A \cong A$
3. $A \times B \cong B \times A$
4. $(A \times B) \times C \cong A \times (B \times C)$
5. $C^{(A \times B)} \cong (C^B)^A$
6. $(B \times C)^A \cong (B^A) \times (C^A)$
7. $A^T \cong A$
8. $T^A \cong T$

Was sehr viel interessanter ist, ist die Tatsache, dass es außer diesen keine weiteren natürlichen Isomorphismen gibt, die in allen CCC-s gelten. Der Beweis für diese Behauptung ist nichttrivial und ergibt sich durch Anwendung des Lambda-Kalküls auf Kategorien.

4.6 Yoneda-Lemma

Sei $\mathbf{C}$ eine kartesisch abgeschlossene Kategorie. Wir betrachten die beiden mengenwertigen Funktoren $F, G : \mathbf{C} \to \mathbf{SET}$.

$$F = Hom_{\mathbf{C} \times \mathbf{C}}((-, -), (A, B)) \circ \triangle \qquad G = Hom_{\mathbf{C}}(- \times A, B)$$

Im vorigen Abschnitt haben wir gezeigt, dass für beide Funktoren ein Objekt existiert, nämlich $A \times B$ bzw. B^A sodass $F \cong Hom_{\mathbf{C}}(-, A \times B)$ und $G \cong Hom_{\mathbf{C}}(-, B^A)$. Allgemein heißen Funktoren, die isomorph zu Hom-Funktoren sind,(co-)darstellbar.

Definition 46. *Ein Funktor $K : \mathbf{C} \to \mathbf{SET}$ heißt darstellbar, wenn es ein $\mathbf{C}$-Objekt R und einen natürlichen Isomorphismus $\varphi : K \dot{\rightarrow} Hom_{\mathbf{C}}(R, -)$ gibt. R heißt darstellendes Objekt.*

Wie das folgende Diagramm zeigt, ist jede natürliche Transformation

$$\tau : Hom_{\mathbf{C}}(R, -) \dot{\rightarrow} K$$

für einen Funktor $K : \mathbf{C} \to \mathbf{SET}$ vollständig durch $\tau_R(id_R)$ bestimmt.

$$\begin{tikzcd}
Hom_{\mathbf{C}}(R,R) \arrow[r, "\tau_R"] \arrow[d, "f \circ -"'] & K(R) \arrow[d, "K(f)"] & R \arrow[d, "f"] \\
Hom_{\mathbf{C}}(R,D) \arrow[r, "\tau_D"] & K(D) & D
\end{tikzcd}$$

Es gilt nämlich

$$\tau_D(f) = \tau_D(f \circ id_R) = K(f) \circ \tau_R(id_R).$$

Dieser Zusammenhang ist bekannt als *Yoneda-Lemma*:

Lemma 5. *(Yoneda) Für jeden Funktor $K : \mathbf{C} \to \mathbf{SET}$ ist die Abbildung*

$$\psi_{R,K} : Nat[Hom(R,-),K] \to K(R),$$

die jeder natürlichen Transformation $\tau : Hom_{\mathbf{C}}(R,-) \dot{\to} K$ den Wert $\tau_R(id_R) \in K(R)$ zuordnet, ist ein Isomorphismus.

(i) Der Pfeil $\psi_{R,K}$ ist natürlich in R d.h.

$$\begin{tikzcd}
Nat[Hom(R,-),K] \arrow[r, "\psi_{R,K}"] \arrow[d, "- \circ Hom(f,-)"'] & K(R) \arrow[d, "K(f)"] & R \arrow[d, "f"] \\
Nat[Hom(D,-),K] \arrow[r, "\psi_{D,K}"] & K(D) & D
\end{tikzcd}$$

wobei $Hom_{\mathbf{C}}(f,-) : Hom_{\mathbf{C}}(D,-) \dot{\to} Hom_{\mathbf{C}}(R,_)$ die natürliche Transformation ist, die definiert ist durch $Hom_{\mathbf{C}}(f,-)_C = Hom_C(f,C) = - \circ f : Hom_{\mathbf{C}}(D,C) \to Hom_{\mathbf{C}}(R,C)$.

(ii) Der Pfeil $\psi_{R,K}$ ist natürlich in K d.h.

$$\begin{tikzcd}
Nat[Hom(R,-),K] \arrow[r, "\psi_{R,K}"] \arrow[d, "\mu \circ -"'] & K(R) \arrow[d, "\mu_R"] & K \arrow[d, "\mu"] \\
Nat[Hom(R,-),H] \arrow[r, "\psi_{R,H}"] & H(R) & H
\end{tikzcd}$$

Falls K ein Hom-Funktor der Form $K = Hom_{\mathbf{C}}(D,-)$ für ein beliebiges Objekt D ist, dann bedeutet das Lemma, dass es eine natürliche Bijektion zwischen den Pfeilen $g : D \to R$ (also Elementen von $Hom_{\mathbf{C}}(D,-)(R)$) und den natürlichen Transformationen von $Hom_{\mathbf{C}}(R,-)$ nach $Hom_{\mathbf{C}}(D,-)$ gibt.

Definition 47. *Sei $F : \mathbf{C} \to \mathbf{SET}$ ein Funktor und $c \in F(C)$ ein Element. Wenn die durch c induzierte natürliche Transformation ein Isomorphismus von $Hom(C,-)$ nach F ist, dann heißt c universelles Element von F.*

Satz 21 (Yoneda-Einbettung). *(i) Sei $Y : \mathbf{C}^{op} \to Func[\mathbf{C}, \mathbf{SET}]$ der Funktor mit $Y(R) = Hom_{\mathbf{C}}(R, -)$ für Objekte R und $Y(g) = Hom_{\mathbf{C}}(g, -) : Hom_{\mathbf{C}}(R, -) \to Hom_{\mathbf{C}}(D, -)$ für Pfeile $g : D \to R$. Y ist eine volle Einbettung $\mathbf{C}^{op} \to Func[\mathbf{C}, \mathbf{SET}]$.*

(ii) Sei $Y : \mathbf{C}^{op} \to Func[\mathbf{C}, \mathbf{SET}]$ der Funktor mit $Y(R) = Hom_{\mathbf{C}}(-, R)$ für Objekte R und $Y(g) = Hom_{\mathbf{C}}(-, g) : Hom_{\mathbf{C}}(-, D) \to Hom_{\mathbf{C}}(-, R)$ für Pfeile $g : D \to R$. Y ist ein volltreuer kovarianter Funktor $\mathbf{C}^{op} \to Func[\mathbf{C}, \mathbf{SET}]$.

Was man aus dem Theorem lernt, ist, dass jede kleine Kategorie $\mathbf{C}$ eine volle Teilkategorie von $Func[\mathbf{C}^{op}, \mathbf{SET}]$ ist.

Aufgrund ihrer Bedeutung hat diese Kategorie einen eigenen Namen: Ein mengenwertiger Funktor über $\mathbf{C}^{op}$ heißt Prägarbe auf $\mathbf{C}$ und $Func[\mathbf{C}^{op}, \mathbf{SET}]$ heißt dann Kategorie der Prägarben auf $\mathbf{C}$. Diese Kategorie erbt einige interessante Eigenschaften von $\mathbf{SET}$, insbesondere ist sie ein Topos. Es gilt folgender Satz:

Satz 22. *Sei $\mathbf{C}$ eine Kategorie. Dann hat $Func[\mathbf{C}^{op}, \mathbf{SET}]$ Pullbacks für jedes Paar von Morphismen und ist kartesisch abgeschlossen.*

Beweis:(Skizze)

Terminal $T : \mathbf{C}^{op} \to \mathbf{SET}$ mit $T(C) = \{*\}$ definiert einen Funktor, wobei die Pfeilabbildung alles auf Id_* abbildet.

Pullbacks Für zwei gegebene natürliche Transformationen $\eta : F \overset{\cdot}{\rightarrow} H$ und $\tau G \overset{\cdot}{\rightarrow} H$ wird komponentenweise das Pullback in $\mathbf{SET}$ gebildet. Dabei entsteht eine Familie von Paaren $\{(\pi_{1,C} : X_C \to F(C), \pi_{2,C} : X_C \to G(C)\}_{C \in Obj(\mathbf{C})}$. Wir definieren einen Funktor X durch $X(C) = X_C$ und für jeden Pfeil $f : C \to D$ soll $X(f)$ der Pfeil sein, der das folgende Diagramm kommutativ macht.

$$\begin{tikzcd}
X(D) \arrow[rr, "\pi_{2,D}"] \arrow[dr, "\pi_{1,D}"] \arrow[dd, dashed, "X(f)"'] & & G(D) \arrow[dr, "\tau_D"] \arrow[dd, "G(f)"] & \\
& F(D) \arrow[rr, "\eta_D"'] \arrow[dd, "F(f)"] & & H(D) \arrow[dd, "H(f)"] \\
X(C) \arrow[rr, "\pi_{2,C}"'] \arrow[dr, "\pi_{1,C}"'] & & G(C) \arrow[dr, "\tau_C"] & \\
& F(C) \arrow[rr, "\eta_C"'] & & H(C)
\end{tikzcd}$$

Produkte Das Produkt $F \times G$ von zwei Funktoren entsteht als Pullback der Terminalabbildungen $t_F : F \overset{\cdot}{\rightarrow} T$ und $t_G : G \overset{\cdot}{\rightarrow} T$.

Exponential Für die Existenz des Exponentials verwenden wir das Yoneda-Lemma. Wenn G^F das Exponential von F und G ist, dann muss es für jedes H einen natürlichen Isomorphismus $Nat[H \times F, G] \cong Nat[H, G^F]$ geben. Falls $H = Hom_{\mathbf{C}}(-, C)$ für ein $C \in Obj(\mathbf{C})$ muss also $Nat[Hom_{\mathbf{C}}(-, C) \times F, G] = Nat[Hom_{\mathbf{C}}(-, C), G^F]$. Für jeden Funktor K ist aber nach dem Yoneda-Lemma $Nat[Hom_{\mathbf{C}}(-, C), K] \cong K(C)$ und daher können wir G^F definieren durch $G^F(C) = Nat[Hom_{\mathbf{C}}(-, C) \times F, G]$. Für Pfeile $f : C \to D$ müssen wir eine Funktion $G^F(f) : Nat[Hom_{\mathbf{C}}(-, D) \times F, G] \to Nat[Hom_{\mathbf{C}}(-, C) \times F, G]$ definieren durch $G^F(f)(\sigma) = \sigma \circ Hom_{\mathbf{C}}(-, f) \times id_F$. (Überprüfung, dass G^F ein Funktor ist, bleibt dem Leser überlassen.)

Die Evaluierungsabbildung $ev_{F,G} : G^F \times F \to G$ legen wir komponentenweise fest. Für jedes $D \in Obj(\mathbf{C})$ sei $ev_{F,G,D} : Nat[Hom_{\mathbf{C}}(-, D) \times F, G] \times F(D) \to G(D)$ festgelegt durch $ev_{F,G,D}(\sigma, n) = \sigma_D(id_D, n)$. ...

$\square$

6 Monaden, adjungierte Funktoren und Algebren

In diesem Abschnitt geht es darum, wie man mit Hilfe von Funktoren und natürlichen Transformationen weitere Kategorien konstruiert.

6.1 Monaden

Auf der einen Seite sind Monaden eine Verallgemeinerung gewisser Eigenschaften von algebraischen Strukturen. Auf der anderen Seite sind sie Verallgemeinerungen von Eigenschaften adjungierter Funktoren, die wir später noch kenenlernen werden. Monaden-Theorie hat sich insbesondere bei der Untersuchung von Topoi als sehr nützliches Werkzeug erwiesen.

Definition 48. *Eine Monade(manchmal auch Tripel) über eine Kategorie $\mathbf{C}$ ist ein Tripel $\mathbf{T} = (T, \mu, \eta)$, wobei $T : \mathbf{C} \to \mathbf{C}$ ein Funktor und $\mu : T^2 \xrightarrow{.} T, \eta : Id_{\mathbf{C}} \xrightarrow{.} T$ natürliche Transformationen, so dass folgende Diagramme kommutativ sind. Die Transformation η*

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

heißt Einheit, und μ heißt Multiplikation.

Die Bezeichnung T^n in den Diagrammen bedeutet n-fache Anwendung des Funktors T. Die C-Komponenten von μT ist die $T(C)$-Komponente von μ, und die Komponente von $T\mu$ in C ist $T(\mu_C)$.

6.1.1 Beispiele

1. Sei M ein Monoid mit dem Einselement 1. Das Darstellungstripel $\mathbf{T} = (T, \eta, \mu)$ auf der Kategorie der Mengen wird gegeben durch die Definition des Funktors T als $T(S) = M \times S$ für jede Menge S. Die Einheit ist gegeben durch $\eta_S : S \to T(S)$ mit $\eta_S(s) = (1, s)$. Die Multiplikation $\mu : T^2 \xrightarrow{.} T$ ist komponentenweise gegeben durch $\mu_S : T^2(S) \to T(S)$ mit $\mu_S(m_1, m_2, s) = (m_1 * m_2, s)$. Das heißt Einheit und Multiplikation entstehen direkt aus denen des Monoids. Die Einheits- und Assoziativitätsgleichungen können einfach gezeigt werden.

2. Auf ähnliche Weise definiert man für einen kommutativen Ring R und eine assoziative R-Algebra mit Eins eine Monade mit Hilfe eines Funktors T auf den R-Moduln mit $T(M) = A \otimes M$.

3. Sei C ein Objekt in einer Kategorie $\mathbf{C}$ mit direkten Summen. Sei $TX = X + C$. Die Komponente $\eta_X : X \to X + C$ sei als Injektion in die Summe definiert und $\mu_X : X + C + C \to X + C$ soll $id_X + \bigtriangledown$ sein, wobei $\bigtriangledown$ die Co-Diagonale ist.

4. Sei $\mathbf{C}$ eine Kategorie mit Produkten und $D \in Obj(\mathbf{C})$. Man definiert $T(C) = D^{Hom(C,D)}$. Um die Pfeilabbildung des Funktors, sowie die natürlichen Transformationen μ und η zu definieren, führen wir folgende Bezeichnung ein. Für einen Pfeil $u : C \to D$ sei $< u >: D^{Hom(C,D)} \to D$ der Pfeil $ev \circ (-, u)$. Für $f : C' \to C$ ist Tf definiert als
$$Tf : D^{Hom(C',D)} \to D^{Hom(C,D)}$$
Wegen der Universaleigenschaft des Produktes können wir Tf eindeutig definieren, indem wir jede seiner Projektionen [15] angeben: Für $v : C \to D$ ist
$$< v > \circ Tf =< v \circ f >$$
$\eta_C : C \to TC$ ist gegeben durch $< u > \circ\eta_C = u$ und $\mu_C : T^2C \to TC$ ist gegeben durch $< u > \circ\mu_C =<< u >>$.

5. Freie Konstruktionen: Sei $T : \mathbf{SET} \to \mathbf{SET}$ der Funktor, der jeder Menge X die der aus X generierten freien Gruppe unterliegende Menge zuordnet. Damit besteht TX aus der Menge der Äquivalenzklassen von Wörtern aus Symbolen x und x^{-1} für alle $x \in X$. Die Äquivalenzrelation identifiziert Wörter, die die Folge xx^{-1} oder $x^{-1}x$ enthalten, mit Wörtern, die man erhält, wenn man diese Folge weglöscht. Die Äquivalenzklasse eines Wortes w wird dann mit $[w]$ bezeichnet. $\eta_X : x \mapsto [x]$ und μ_X bildet Äquivalenzklassen von Wörtern aus TX also Äquivalenzklassen von Äquivalenzklassen von Wörtern aus X ab, indem es die Klammern weglässt. Sind zum Beispiel $a, b, c \in X$ und $[ab^2c^{-1}]$ und $[c^2a^2]$ in TX dann ist $w = [[ab^2c^{-1}][c^2a^2]] \in TTX$ und $\mu_X(w) = [ab^2ca^2] \in TX$. Es gibt viele ähnliche Beispiele die auf der Konstruktion freier algebraischer Strukturen basieren. Tatsächlich kann sogar jede Monade über $\mathbf{SET}$ im Wesentlichen auf diesem Weg definiert werden, wenn man unendliche Operationen in den algebraischen Strukturen erlaubt.

6. Monaden in der Programmierung: Sei $L : \mathbf{SET} \to \mathbf{SET}$ der Listenfunktor mit $L(X) = \{[x_1, \ldots, x_n] \mid n \in \mathbb{N}, x_i \in X\}$ und $L(f)([x_1, \ldots, x_n]) = [f(x_1), \ldots, f(x_n)]$. Seien $\eta_X : X \to T(X)$ mit $\eta(x) = [x]$ und $\mu_X : T^2(X) \to T(X)$ mit
$$\mu([[x_1^1, \ldots x_n^1], \ldots, [x_1^k, \ldots x_l^k]]) = [x_1^1, \ldots x_n^1 \ldots x_1^k, \ldots x_l^k]$$
zwei natürliche Transformationen. Dann ist das Tripel (T, η, μ) eine Monade über $\mathbf{SET}$.

7. Sei $\mathcal{P}$ der Potenzmengenfunktor, der jeder Funktion f die Funktion f^* zuordnet, wobei $f^*(A) = \{f(x) \mid x \in A\}$ für eine Menge $A \in \mathcal{P}(X)$. Als Einheit definiert man $\eta_X(x) = \{x\}$ und als Produkt $\mu_X(\{A_1, \ldots A_n\}) = A_1 \cup \ldots \cup A_n$. Die Prüfung, dass $(\mathcal{P}, \eta, \mu)$ tatsächlich eine Monade bildet, bleibt dem Leser überlassen.

Die nächste Konstruktion geht auf Kleisli[**?**] zurück und ist besonders für die theoretische Informatik von Bedeutung. Mit ihrer Hilfe kann man einen Monadenfunktor „faktorisieren“.

[15] wenn man $D^{Hom(C,D)}$ auffasst als Produkt, nämlich $\prod_{Hom(C,D)} D$

Definition 49. *Sei* $\mathbf{T} = (T, \eta, \mu)$ *eine Monade auf* $\mathbf{C}$. *Die Kleislikategorie* $\mathbf{C_T}$ *dieser Monade hat als Objekte die Objekte von* $\mathbf{C}$. $f_T : C \to D$ *ist ein Pfeil in* $\mathbf{C_T}$, *wenn es einen* $\mathbf{C}$*-Pfeil* $f : C \to TD$ *gibt. Der identische Pfeil auf einem Objekt* C *ist* $Id_C = \eta_C : C \to TC$. *Die Komposition der Pfeile* $f_T : C \to D$ *und* $g_T : D \to E$ *ist definiert als* $g_T \circ f_T = \mu_E \circ T(g) \circ f$.

Wir können zwei Funktoren $U : \mathbf{C_T} \to \mathbf{C}$ und $F : \mathbf{C} \to \mathbf{C_T}$ definieren durch die Objektabbildung $U(C) = C$ und $U(f_T) = \mu_D \circ T(f)$, wobei D das Ziel von f_T ist. Für F gilt $F(C) = C$ und $F(g) = T(g) \circ \eta_C$ wenn $g : C \to D$. Offenbar ist $T = U \circ F$

Man sagt die beiden Funktoren bilden ein sogenanntes adjungiertes Funktorpaar vergleiche Definition 50.

Eine andere Möglichkeit der Faktorisierung von Monadenfunktoren liefern die sogenannten Eilenberg-Moore-Kategorien. Wieder entsteht ein adjungiertes Funktorpaar. Eilenberg-Moore Kategorien sind hauptsächlich für die Mathematik interessant, weniger für die Informatik. Wir werden die entsprechende Definition nach dem Abschnitt über Algebren zu einem Funktor angeben.

6.2 Adjungierte Funktoren

In Beispiel 16 haben wir den freien Monoidfunktor $F : \mathbf{SET} \to \mathbf{Mon}$ und den zugehörigen Vergissfunktor $U : \mathbf{Mon} \to \mathbf{SET}$ definiert. F ordnet jeder Menge X das freie Monoid $F(X)$ über dieser Menge zu. Dabei wird die Menge als Alphabet betrachtet und das zugehörige Monoid besteht aus der Menge der Wörter, die daraus gebildet werden kann. Das leere Wort ist dabei das neutrale Element und die Komposition die assoziative Monoidoperation. Der Vergissfunktor bildet ein Monoid auf die Menge seiner Elemente ab.

Satz 23. *Für jede Menge* X *ist* η_X *mit* $\eta_X(x) =< x >$ *die Komponente einer natürlichen Transformation* $\eta : Id_{\mathbf{SET}} \to U \circ F$. *Dabei bezeichnet* $< x >$ *das Wört der Länge* 1, *das nur aus* x *besteht.*

Beweis: Seien x und Y Mengen und $X^* = UF(X)$ und $Y^* = UF(Y)$. Wir müssen zeigen, dass das folgende Diagramm in **SET** kommutativ ist:

$$\begin{tikzcd} X \arrow[r, "f"] \arrow[d, "\eta_X"'] & Y \arrow[d, "\eta_Y"] \\ X^* \arrow[r, "f^*"] & Y^* \end{tikzcd}$$

Man wendet η_X nach f an und erhält für jedes x als Ergebnis $< f(x) >$. Wegen $f^* = U(F(f))$ gilt $f^*(< x >) = UF(f(< x >)) = U(< f(x) >) =< f(x) >$, da $F(f)$ ein Monoidhomomorphismus ist. □

Das Verhältnis des freien Monoidfunktors zum Vergissfunktor ist ein Beispiel für ein sehr allgemeines Konzept.

Definition 50. *Seien* $\mathbf{A}$ *und* $\mathbf{B}$ *Kategorien und* $F : \mathbf{A} \to \mathbf{B}$ *und* $G : \mathbf{B} \to \mathbf{A}$ *Funktoren. Man sagt* F *ist* linksadjungiert *zu* G *und* G *ist* rechtsadjungiert *zu* F, *wenn es eine natürliche Transformation* $\eta : Id \dot{\to} GF$ *gibt, so dass für alle* $\mathbf{A}$*-Objekte* A *und* $\mathbf{B}$*-Objekte* B *und jeden Pfeil* $f : A \to GB$ *genau einen Pfeil* $g : FA \to B$ *gibt, so dass das Diagramm*

$$\begin{tikzcd} A \arrow[r, "\eta_A"] \arrow[dr, "f"'] & GFA \arrow[d, "G(g)"] \\ & GB \end{tikzcd}$$

Die Eigenschaft von η, dass das Diagramm kommutiert, heißt *universal mapping property*. Der Pfeil η_A ist das universelle Element des Funktors $Hom(A, U(-))$ (vergleiche Definition 47).

Man schreibt $(F \dashv G)$ wenn F linksadjungiert zu G ist. Das Tripel $(F, G, u\eta)$ heißt Adjunktion, die Transformation η heißt *Einheit*. Neben der Zuordnung $Hom(A, UB) \to Hom(FA, B)$ existiert bei einem adjungierten Funktorpaar auch eine eindeutige Zuordnung der Pfeile in der anderen Richtung. Es gilt der folgende Satz:

Satz 24. *Seien* $F : \mathbf{A} \to \mathbf{B}$ *und* $G : \mathbf{B} \to \mathbf{A}$ *ein adjungiertes Funktorpaar mit* $F \dashv G$. *Dann gibt es eine natürliche Transformation* $\epsilon : FU \to id_{\mathbf{B}}$ *so dass für jeden Pfeil* $g : FA \to B$ *ein eindeutig bestimmter Pfeil* $f : A \to GB$ *existiert, mit* $\epsilon_B \circ F(f) = g$. *Diese Transformation heißt* Co-Einheit.

Beispiele 21. 1. *Sei* $\triangle : \mathbf{C} \to \mathbf{C} \times \mathbf{C}$ *der Diagonalfunktor mit* $\triangle(A) = (A, A)$ *und* $\triangle(f) = (f, f)$. *Wir nehmen an, es gäbe einen rechtsadjungierten Funktor* Π *zu* $\triangle$, *dann müsste es eine Co-Einheit, also einen Pfeil* $\epsilon : \triangle\Pi \to Id_{\mathbf{C}\times\mathbf{C}}$ *geben, so dass das folgende Diagramm für jeden Pfeil* $g : (C, C) \to (A, B)$ *einen Pfeil* $f : C \to \Pi(A, B)$ *kommutativ ist:*

$$\begin{tikzcd} & \triangle\Pi(A,B) \arrow[d, "\epsilon_{(A,B)}"] \\ \triangle(C) \arrow[ur, "\triangle f"] \arrow[r, "g"'] & (A,B) \end{tikzcd}$$

Jeder solche Pfeil g *ist aber ein Paar von Pfeilen* (g_1, g_2) *mit* $g_1 : C \to A$ *und* $g_2 : C \to B$. *Damit haben wir gerade die Universalitätseigenschaft des Produktes beschrieben und wissen, daher dass* Π *der Produktfunktor sein muss mit* $\Pi(A, B) = A \times B$, $\Pi(h, k) =< h, k >$ *und* $\epsilon_{(A,B)}$ *ist dann das Paar der Projektionen* $p_1 : A \times B \to A$ *und* $p_2 : A \times B \to B$.

2. *Wir betrachten den einstelligen Produktfunktor* $\Pi_B : \mathbf{C} \times \mathbf{C} \to \mathbf{C}$ *mit* $\Pi(A) = A \times B$ *und* $\Pi(f) =< f, id_B >$. *Wir suchen einen rechstadjungierten Funktor* $E_B : C \to \mathbf{C} \times \mathbf{C}$ *für* Π_B. *Das entsprechende Diagramm sieht dann aus wie folgt:*

$$\begin{tikzcd}
& \triangle E(C) \times B \arrow[d, "\epsilon_C"] \\
A \times B \arrow[ur, "\Pi f"] \arrow[r, "g"'] & C
\end{tikzcd}$$

Das Bild des Funktors E auf dem Objekt C muss ein Objekt $E(C)$ sein, so dass für jeden Pfeil $g : A \times B \to C$ ein Pfeil $f : A \to E_B(C)$ existiert mit $\epsilon_B \circ < f, id_B >= g$. Das ist gerade die Universalitätseigenschaft des Exponentials. Falls in $\mathbf{C}$ *alle Exponentiale existieren ist der rechstadjungierte Funktor E zu $- \times B$ der Funktor mit $E(C) = C^B$ und $\epsilon_B = ev_B$.*

3. *Wir betrachten die Kategorie* $\mathbf{Sub(S)}$ *für eine Menge S. Ihre Objekte sind die Teilmengen von S und es gibt genau zwische zwei Objekten S_1 und S_2, wenn $S_1 \subset S_2$. Für jede Funktion $f : S \to T$ und jede Teilmenge $T_0 \subseteq T$ sei $f^{-1}(T_0) = \{s \in S \mid f(s) \in T_0\}$ das inverse Bild von T_0 unter f. Da aus $T_0 \subseteq T_1$ folgt, dass $f^{-1}(T_0) \subseteq f^{-1}(T_1)$, definiert f^{-1} einen Funktor vo m Typ $Sub(T) \to Sub(S)$. Wie sich herausstellt hat dieser Funktor sowohl einen Linksadjungierten als auch einen Rechstadjungierten. Der Linksadjungierte ist f_* mit $f_*(S_0) = \{f(s) \mid s \in S_0\}$. Offenbar ist $f_*(S_0) \subseteq T_0$ genau dann, wenn $S_0 \subseteq f^{-1}(T_0)$. Die Rechtsadjungierte des inversen Bildfunktors wird gewöhnlich mit $f_!$ bezeichnet und ist definiert durch $t \in f_!(S_0)$ genau dann, wenn $f^{-1}(\{t\}) \subseteq S_0$.*

4. *Adjunktionen in Halbordnungen: Seien* $\mathbf{P}$ *und* $\mathbf{Q}$ *Halbordnungen, d.h. Kategorien, die genau dann einen Pfeil zwischen zwei Objekten haben, wenn sie in Relation sind. Funktoren zwischen* $\mathbf{P}$ *und* $\mathbf{Q}$ *sind dann monotone Funktionen und $F \dashv G$ bedeutet dann übersetzt in die Sprache der Halbordnungen, dass $x \leq_P (G \circ F)(x)$ für alle $x \in Obj(P)$ und $x \leq G(y)$ impliziert $F(x) \leq y$ für alle $x \in Obj(\mathbf{P})$ und $y \in Obj(\mathbf{Q})$. Zusammen ergeben die beiden Bedingungen:*

   $$F(x) \leq_Q y \text{ genau dann, wenn } x \leq_P G(y).$$

   Eine sehr häufig vorkommende Situation ist die eines Paares von kontravarianten Funktoren, d.h ordnungsumkehrende Funktionen. In diesem Fall betrachtet man die duale Kategorie von $\mathbf{P}$ *oder* $\mathbf{Q}$ *und hat also ein Paar von Funktoren $F : \mathbf{P}^{op} \to \mathbf{Q}$ und $G : \mathbf{Q} \to \mathbf{P}^{op}$. Falls $F \dashv G$ findet man bei Rückübersetzung in die Sprache von* $\mathbf{P}$ *als Bedingung:*

   $$F(x) \leq_Q y \text{ genau dann, wenn } G(y) \leq_P x.$$

   Ein solches Paar von Funktoren heißt Galois-Connection.

Eine andere äquivalente Formulierung für Adjunktionen ist die folgende:

Satz 25. *Seien* $\mathbf{A}$ *und* $\mathbf{B}$ *Kategorien, und $G : \mathbf{B} \to \mathbf{A}$ und $F : F \to \mathbf{B}$ Funktoren, dann gilt $F \dashv G$ genau dann, wenn $Hom(F-, -)$ und $Hom(-, G-)$ als Funktoren von $\mathbf{A}^{op} \times \mathbf{B} \to \mathbf{SET}$ natürlich isomorph sind.*

Beweis: „⇒"Sei $F \dashv G$. Man definiert $\beta_{A,B} : Hom(FA, B) \to Hom(A, UB)$ durch $\beta_{A,B}(g) = Ug \circ \eta_A$ und $\gamma_{A,B} : Hom(A, UB) \to Hom(FA, B)$ dadurch, dass $\gamma_{A,B}(f)$ ist der eindeutig bestimmte Pfeil g so dass $f = Ug \circ \eta_A$. Dann ist $\gamma_{A,B}(\beta_{A,B}(g) = g$ wegen der Eindeutigkeitsbedingung in Definition 50 und $\beta_{A,B}(\gamma_{A,B}(f) = f$ durch die Definition.

„⇐"Wir nehmen an, wir hätten einen natürlichen Isomorphismus. Sei A ein beliebiges $\mathbf{A}$-Objekt und $B = FA$. Dann ist $Hom(FA, FA) \cong Hom(A, UFA)$. Sei $\eta_A \in Hom(A, GFA)$ das Bild von Id_{FA} unter dem natürlichen Isomorphismus. Sei weiterhin $g \in Hom(FA, B)$ das Bild von einem $f \in Hom(A, UB)$ unter diesem Isomorphismus. Wegen dessen Natürlichkeit kommutiert das folgende Diagramm:

$$\begin{tikzcd}
Hom(FA,FA) \arrow[r, "\cong"] \arrow[d, "Hom(FA,g)"'] & Hom(A,GFA) \arrow[d, "Hom(A,Gg)"'] \\
Hom(FA,B) \arrow[r, "\cong"'] & Hom(A,GB)
\end{tikzcd}$$

Folgt man im Diagramm der Identität auf FA im Uhrzeigersinn, dann erhält man $Hom(A, Ug)(\eta_A) = Ug \circ \eta_A$ und in der anderen Richtung $Hom(FA, g)(id_{FA}) = g \circ id_{FA} = g$, was unter dem Isomorphismus f entspricht. Daraus können wir schließen, dass $f = Ug \circ \eta_A$. Wegen Eindeutigkeit gilt das, wenn $f = Uh \circ \eta_A$ und damit sowohl g als auch h unter dem Isomorphismus f entsprechen, dann gilt auch $g = h$. □

Sei $G : \mathbf{B} \to \mathbf{A}$ ein Funktor, dann ist der linksadjungierte Funktor F, falls ein solcher existiert bis auf Isomorphie eindeutig bestimmt. Der Grund dafür ist, dass wenn sowohl F und F' Linksadjungierte zu G wären, dann wäre für jedes A sowohl $Hom_{\mathbf{A}}(FA, -)$ als auch $Hom_A(F'A, -)$ natürlich isomorph zu $Hom_{\mathbf{B}}(A, U-)$ damit auch zueinander.

Obwohl die Begriff Monade und adjungiertes Funktorpaar auf den ersten Blick sehr verschieden sind, gibt es doch einen direkten Zusammenhang. Im Abschnitt über Monaden haben wir schon gesehen, dass sich Monadenfunktoren in gewisser Weise zerlegen lassen. In der anderen Richtung gilt folgender Satz:

Satz 26. *Sei F, G ein adjungiertes Funktorpaar mit Einheit η und Coeinheit ϵ. Dann ist $(T = GF, \eta, \mu = G\epsilon F)$ eine Monade.*

Beweis: Zunächst stellen wir fest, dass GF, η und $G\epsilon F$ die richtigen Typen haben. Wenn $F : \mathbf{A} \to B$ und $G : \mathbf{B} \to \mathbf{A}$, dann ist $T : \mathbf{A} \to \mathbf{A}, \eta : Id_{\mathbf{A}} \dot{\to} GF$ und $\mu : G\epsilon F : GFGF \dot{\to} GF$. Wir müssen zeigen, dass die Gesetze für die Monaden nach Definition 48 gültig sind.

Für die Gesetze zur Einheit haben wir:

$$\mu \circ T\eta = G\epsilon F \circ GF\eta = G(\epsilon F \circ F\eta) = G(id_F) = id_{GF}$$
$$\mu \circ \eta T = G\epsilon F \circ \eta GF = (G\epsilon \circ \eta G)F = id_G(F) = id_{GF}$$

Um zu zeigen, dass auch die Assoziativitätsgesetze gelten, stellen wir zunächst fest, dass gilt: $\epsilon \circ \epsilon FG = \epsilon \circ FG\epsilon$. Dazu betrachten wir ein $\mathbf{B}$-Objekt B und den Pfeil $f = \epsilon_B$:

$FG(B) \to D$. Dann gilt wegen Natürlichkeit von ϵ, dass $\epsilon_D \circ \epsilon_{FG(D)} = f \circ \epsilon_{FG(D)} = \epsilon_D \circ FG(f) = \epsilon_D \circ FG(\epsilon_D)$.

Und für die Assoziativgesetze gilt:

$$\begin{aligned}
\mu \circ \mu T &= G\epsilon F \circ G\epsilon FGF \\
&= G(\epsilon F \circ \epsilon FGF) = G(\epsilon \circ \epsilon FG)F \\
&= G(\epsilon \circ FG\epsilon)F \\
&= (G\epsilon \circ GFG\epsilon)F = G\epsilon F \circ GFG\epsilon F \\
&= \mu \circ T\mu
\end{aligned}$$

□

6.3 Von Funktoren abgeleitete Algebren

In diesem Abschnitt beschreiben wir wie mit Hilfe von Endofunktoren das Konzept einer Algebra verallgemeinert werden kann. Diese Verallgemeinerung soll ohne den Begriff einer Trägermenge auskommen. Wir beginnen dabei mit dem Begriff eines Fixpunktes eines Funktors und es stellt sich heraus, dass das natürliche Konzept dafür eine Verallgemeinerung des Algebra-Begriffs ist. Mit Funktoralgebren hat man eine kategoriale Beschreibung von Transitionssystemen und sie eignen sich um das Konzept eines Listenobjektes zu entwickeln.

Hier noch einmal eine Wiederholung der formalen Definition der Kategorie der Ω-Algebren.

Definition 51. *Die Kategorie Alg_Ω von Ω-Algebren hat als Objekte Paare (A, α), wobei A eine Trägermenge ist und α eine Abbildung, die für jedes n und jedes n-stellige Operationssymbol $\rho_n \in \Omega_n$ eine Funktion $\alpha_\rho : A^n \to A$ liefert.*
Ein Morphismus $f : (A, \alpha) \to (A', \alpha')$ ist eine Funktion $f : A \to A'$, so dass

$$f(\alpha_\rho(a_1, \ldots, a_n)) = \alpha'_\rho(f(a_1, \ldots, a_n))$$

für alle $n \in \mathbb{N}, \rho_n \in \Omega_n$ und $a_1, \ldots a_n \in A$.

Eine Signatur Ω definiert einen Funktor $T : \mathbf{SET} \to \mathbf{SET}$ mit der Objektabbildung

$$T(A) = \{\rho(a_1, \ldots a_n) \mid n \in \mathbb{N}, \rho \in \Omega_n, a_1, \ldots a_n \in A\}$$

Die Menge $T(A)$ besteht aus allen formal gebildeten Termen über einer Menge A. Jede konkrete Ω-Algebra (A, α) liefert dann eine Funktion $f_\alpha : T(A) \to A$ durch

$$f(\rho(a_1, \ldots a_n)) = \alpha_\rho(a_1, \ldots a_n),$$

Der formal gebildete Ausdruck wird entsprechend der Funktion für ρ ausgewertet.

Beispiele 22. *Als Beispiel betrachte man eine Signatur die zwei zweistellige Operationssymbole ρ_1, ρ_2 und zwei nullstellige Operationssymbole ρ_3, ρ_4 enthält. Als konkrete Algebra nehmen wir $(\mathbb{N}, \alpha)$ mit $\alpha_{\rho_1}(a, b) = a * b$ und $\alpha_{\rho_2}(a, b) = a + b$ sowie $\alpha_{\rho_3} = 1$ und $\alpha_{\rho_4} = 0$.*

Um uns beim Algebra-Begriff von der Verwendung einer Träger*menge* zu lösen, wollen wir in Zukunft statt Funktoren über **SET** beliebige Endofunktoren betrachten.

Definition 52. *Sei* $\mathbf{C}$ *eine Kategorie und* $T : \mathbf{C} \to \mathbf{C}$ *ein Funktor. Die Kategorie* **T-Alg** *der* T*-Algebren hat als Objekte Paare* (C, α)*, wobei* $C \in Obj(\mathbf{C})$ *und* $\alpha \in Hom_{\mathbf{C}}(T(C), C)$*. Die Pfeile zwischen zwei Objekten* (C, α) *und* (C', α') *sind Pfeile* $h \in Hom_{\mathbf{C}}(C, C')$ *mit* $\alpha' \circ T(h) = h \circ \alpha$*.*

Beispiele 23.

1. *Wir betrachten ein Menge von Datentypen, die eine Kategorie* $\mathbf{C}$ *bilden und fügen einen weiteren Typ* P *für die Programme über diesen Tatentypen als Objekt hinzu. Man definiere einen Funktor* $T : \mathbf{C} \to \mathbf{C}$ *mit der Objektabildung* $T(C) = P \times C$*. Für jedes* $C \in Obj(\mathbf{C})$ *sei* $\alpha : T(C) \to C$ *so definiert, dass der Ausdruck* $\alpha((i, x))$ *bedeutet: „wende Programm* i *auf Eingabe* x *an“. Dann ergibt jedes Paar* (C, α) *eine* T*-Algebra.*

2. *Wir betrachten eine Quasiordnung* $(P, \leq)$ *(reflexiv und transitiv) als Kategorie* $\mathbf{P}$*. Ein Funktor* $T : \mathbf{P} \to \mathbf{P}$ *ist eine monotone Funktion. In dieser Situation ist eine* T*-Algebra ein Prä-Fixpunkt des Funktors* T*, da die Existenz von* $\alpha : T(p) \to p$ *bedeutet, dass* $T(p) \leq p$*, also* p *ein Prä-Fixpunkt ist. Aus der Algebra weiß man, dass, falls eine monotone Funktion einen kleinsten Prä-Fixpunkt hat, dann ist dieser auch ein Fixpunkt.*

Angewendet auf Kategorietheorie erhalten wir folgendes Resultat:

Satz 27. *Sei* $\mathbf{C}$ *eine Kategorie und* T *ein Endofunktor auf* $\mathbf{C}$*. Falls* (C, α) *eine initiale* T*-Algebra ist, dann ist* α *ein* $\mathbf{C}$*-Isomorphismus von* $T(C)$ *nach* C*.*

Beweis: Wir betrachten das folgende kommutative Diagramm: $(T(C), T(\alpha))$ ist eine

$$\begin{tikzcd}
T(T(C)) \arrow[r, "T(\alpha)"] \arrow[d, "T(\alpha)"'] & T(C) \arrow[d, "\alpha"] \\
T(C) \arrow[r, "\alpha"] & C
\end{tikzcd}$$

T-Algebra und $\alpha \in Hom_{\mathbf{T-alg}}((T(C), T(\alpha))(C, \alpha))$. Sei $\eta \in Hom_{\mathbf{T-alg}}((C, \alpha), (T(C), T(\alpha)))$ der initiale Algebrenhomomorphismus. Dann sind $\alpha \circ \eta$ und id_C zwei Morphismen in $Hom_{\mathbf{T-alg}}((C, \alpha), (C, \alpha))$. Und wegen Initialität sind beide gleich. Außerdem gilt:

$$\begin{aligned}
\eta \circ \alpha &= T(\alpha) \circ T(\eta) \text{ da } \eta \text{ ein Algebrahomomorphismus ist} \\
&= T(\alpha \circ \eta) \\
&= T(id_C) \\
&= id_C
\end{aligned}$$

$\square$

7 Lambda-Kalkül

Definition 53. *Ein getypter λ-Kalkül ist eine formale Theorie, die aus Typen, Termen, Variablen und Gleichungen besteht. Zu jedem Term a gehört ein Typ A (der sogenannte Typ von a). Man schreibt $a : A$ um auszudrücken, dass a vom Typ A ist. Es gelten folgende Regeln:*

TL-1 *Es gibt einen Typ* $\mathbf{1}$.

TL-2 *Wenn A und B Typen sind, dann gibt es Typen $A \times B$ und $[A \to B]$.*

TL-3 *Es gibt einen Term $*$ vom Typ* $\mathbf{1}$.

TL-4 *Für jeden Typ A gibt es eine abzählbare Menge von Termen x_i^A – die Variablen vom Typ A.*

TL-5 *Wenn $a : A$ und $b : B$, dann ist der Term (a, b) vom Typ $A \times B$.*

TL-6 *Wenn $c : A \times B$, dann gibt es Terme $proj_1(c)$ und $proj_2(c)$ vom Typ A bzw. B.*

TL-7 *Falls $f : [A \times B]$ und $a : A$, dann ist $f'a$ ein Term vom Typ B.*

TL-8 *Wenn x eine Variable vom Typ A und $\varphi(x)$ ein Term vom Typ B ist, dann ist $\lambda_{x:A}$ vom Typ $\varphi(x) : [A \to B]$.*

Die Bedeutung der Regeln zur Termbildung sollte unmittelbar verständlich sein, vielleicht mit Ausnahme des Terms $f'a$, der interpretiert werden kann als Anwendung von f auf a. Die Bezeichnung $\varphi(x)$ in (TL-8) bedeutet, dass φ ein Term ist, in dem möglicherweise ein x auftritt. Diese Bezeichnung soll zur Definition von Substitution verwendet werden: wenn wir einen Term $\varphi(x)$ haben, dann soll $[a/x]\varphi(x)$ für den Term stehen, bei dem jedes Vorkommen von x durch a ersetzt wurde.

Um die Gleichungen zu fixieren, benötigen wir noch einige weitere Bezeichnungen. Wenn x eine Variable ist, dann ist x frei im Term x. Wenn das Vorkommen von x in den Termen a und b frei ist, dann ist x auch frei in (a, b). Falls x frei in f und in a vorkommt, dann ist x auch frei in $f'a$. In einem Term $\lambda_{x:A}\varphi(x)$ ist dagegen kein Vorkommen von x frei. Man sagt dann, x ist gebunden. Ein Term a kann x in $\varphi(x)$ ersetzen, falls keine Variable von a beim Einsetzen von a in die Vorkommen von x in φ gebunden wird. a heißt dann *substituierbar*. Ein Term heißt *geschlossen*, wenn darin keine freien Variablen vorkommen.

Die folgenden Gleichungen regeln, welche Terme als gleich anzusehen sind. Die Vorstellung dahinter ist, dass die Terme für Funktionen stehen, die bei gleichen Eingaben gleiche Ausgaben liefern. Das Symbol für Gleichheit $=_X$ soll kennzeichnen, dass es sich nicht um eine syntaktische Gleichheit der Terme handelt. Dabei bezeichnet X eine endliche Menge von Variablen, die alle frei in a oder a' auftretetenden Variablen enthält.

TL-9 $=_X$ ist reflexiv, symmetrisch und transitiv

TL-10 Falls $a : \mathbf{1}$, dann $a =_{\{\}} *$.

TL-11 Falls $X \subset Y$, dann impliziert $a =_X a'$ auch $a =_Y a'$.

TL-12 Falls $a =_X b$, dann gilt $f'a =_X f'b$.

TL-13 Falls $f =_X g$, dann gilt $f'a =_X g'a$.

TL-14 Aus $\varphi(x) =_{X\cup\{x\}} \psi(x)$ folgt $\lambda_{x:A}\varphi(x) =_X \lambda_{x:A}\psi(x)$

TL-15 Für alle $a : A, b : B$ gilt $proj_1(a,b) =_X a$ und $proj_2(a,b) =_X b$.

TL-16 Für alle $c : A \times B$ gilt $(proj_1(c), proj_2(c)) =_X c$.

TL-17 Falls a in $\varphi(x)$ für x einsetzbar ist, dann ist $\lambda_{x:A}\varphi(x)'(a) =_X [a/x]\varphi(x)$.

TL-18 Falls $x \notin X$ dann $\lambda_{x:A}(f'x) =_X f$.

TL-19 $\lambda_{x:A}\varphi(x) =_X \lambda_{x':A}\varphi(x')$ falls x' für x in $\varphi(x)$ einsetzbar ist und x' nicht frei in $\varphi(x)$ vorkommt.

Liest man die obigen Gleichheiten von links nach rechts, dann erhält man ein Rewriting-System, das die operationale Semantik eines λ-Kalküls als Programmiersprache darstellt. Wir können eine Reduktionsrelation $\Rightarrow$ auch explizit definieren, als kleinste Relation auf Termen, so dass:

$\rightarrow\beta$ $(\lambda_{x:A}a)b \Rightarrow [b/x]a$,

$\times\beta_1$ $proj_1(a,b) \Rightarrow a$

$\times\beta_2$ $proj_2(a,b) \Rightarrow b$

falls $a \Rightarrow b$, dann gilt $c'a \Rightarrow c'b$

falls $a \Rightarrow b$, dann gilt $a'c \Rightarrow b'c$

falls $a \Rightarrow b$, dann gilt $\lambda_{x:A}a \Rightarrow \lambda_{x:A}b$

falls $a \Rightarrow b$, dann gilt $proj_1(a) \Rightarrow proj_1(b)$

falls $a \Rightarrow b$, dann gilt $proj_2(a) \Rightarrow proj_2(b)$

$\rightarrow\eta$ $\lambda_{x:A}a(x) \Rightarrow a$ falls $x \notin FV(a)$

$\times\eta$ $(proj_1(a), proj_2(a)) \Rightarrow a$.

Die Regeln $\rightarrow\beta, \times\beta_1, \times\beta_2$ sind die sogenannten β-Reduktionen, die Regeln $\rightarrow\eta$ und $\times\eta$ sind die η-Reduktionen.

Ein Term ist in *Normalform*, wenn er nicht weiter reduzierbar ist. Die Reduktionsrelation $\Rightarrow$ ist konfluent und noethersch.

7.1 Vom λ-Kalkül zu Kategorien

Definition 54. *Sei $\mathcal{L}$ ein getypter λ-Kalkül. $\mathbf{C}(\mathcal{L})$ hat als Objekte die Typen von $\mathcal{L}$. Ein Pfeil von A nach B ist eine Äquivalenzklasse von Termen vom Typ B mit einer freien Variable vom Typ A(Die nicht notwendig in den Termen auftritt).*

Die Äquivalenzrelation ist die kleinste reflexive, transitive und symmetrische Relation, die dadurch induziert wird, dass zwei Terme $\varphi(x)$ und $\psi(y)$ äquivalent sind, wenn φ und ψ vom selben Typ sind, x und y vom selben Typ sind, x für y einsetzbar in ψ ist und $\varphi(x) =_{\{x\}} \psi(y)$, wobei $\psi(x)$ durch Ersetzen aller Vorkommen von y in ψ durch x entsteht. Der Grund aus dem man Äquivalenzklassen braucht, ist dass je zwei Variablen vom selben Typ demselben Pfeil entsprechen müssen, nämlich der Identität des Objektes zu sich selbst. Wenn $\lambda_{x:A}x : 1 \to [A \to A]$ für jede Variable x einen identischen Pfeil auf A bezeichnet, dann ist der Pfeil, der einer Variablen x vom Typ A entspricht die Identität auf A. Die Äquivalenzrelation macht auch zwei Variablen vom Type $\mathbf{1}$ äquivalent und sorgt so dafür, dass $\mathbf{1}$ Terminalobjekt der Kategorie ist.

Sei φ ein Term vom Typ B mit höchstens einer freien Variable x vom Typ A und ψ ein Term vom Typ C mit höchstens einer freien Variable y vom Typ B. Indem man falls notwendig x durch eine Variable die in ψ nicht gebunden ist ersetzt, erreicht man, dass man y in ψ durch φ ersetzen darf. Die Komposition der entsprechenden Pfeile ist dann der Pfeil, der zur Äquivalenzklasse des Terms $\psi(\varphi)$ gehört, des Terms, den man erhält, wenn man x in ψ durch φ ersetzt.

Satz 28. *Die Kategorie $\mathbf{C}(\mathcal{L})$ ist kartesisch abgeschlossen.*

Beweis: siehe [6] $\square$

Die Konstruktion ist die offensichtliche. $A \times B$ mit $proj_1$ und $proj_2$ ist das Produkt von A und B und $[A \to B]$ ist das Exponential. Falls $\varphi(x)$ einem Pfeil $f : C \times A \to B$ entspricht, dann muss x eine Variable vom Typ $C \times A$ sein. Indem wir das Axiom $[TL - 15]$ benutzen, können wir (z, y) für x einsetzen und erhalten $\varphi(z, y)$, wobei $z : C$ und $y : A$. λf ist dann die Äquivalenzklasse von $\lambda_{z:C}\varphi(z, y)$. Man beachte, dass wenn z und y tatsächlich im Term $\varphi(z, y)$ auftreten, dann ist er in keiner Äquivalenzklasse, da er zwei freie Variablen hat.

Beispiele 24. *Als Beispiel dafür, wie die Definition von $\mathbf{C}(\mathcal{L})$ funktioniert, werden wir eine Berechnung durchführen, die eine der Eigenschaften Kartesisch abgeschlossener Kategorien beweist. Sei $\mathcal{L}$ ein getypter λ-Kalkül. Wir definieren:*

$$\Lambda : Hom_{\mathbf{C}(\mathcal{L})}(C \times A, B) \to Hom_{\mathbf{C}(\mathcal{L})}(C, B^A)$$

durch $\Lambda([\varphi(u)]) = \lambda_{x:A}\varphi((z, x))$ für $[\varphi(u)] : C \times A \to B$ und Variablen $z : C, x : A$.

$$\Gamma : Hom_{\mathbf{C}(\mathcal{L})}(C, B^A) \to Hom_{\mathbf{C}(\mathcal{L})}(C \times A, B)$$

durch $\Gamma([\psi(z)]) = \psi(proj_1 u)' proj_2 u$ für $[\psi(z)] : C \to [A \to B]$ und eine Variable $u : A$.

Λ und Γ sind inverse Funktionen. Wir zeigen nur eine Richtung der Zusammensetzung und nutzen dafür $[TL - 18]$ sowie die Tatsache, dass x in $\varphi(z)$ nicht auftritt, da nach der

Definition eines Pfeils in $\mathbf{C}(\mathcal{L})$, der Term $\varphi(z)$ nur eine freie Variable hat, und diese ist nicht x.

$$\begin{aligned}\Lambda(\Gamma(\psi(z))) &= \Lambda(\psi(proj_1 u)' proj_2 u)\\ &= \lambda_x \psi(proj_1(z,x))' proj_2(z,x)\\ &=_X \lambda_x \psi(z)', x\\ &=_X \psi(z)\end{aligned}$$

7.2 Von kartesisch abgeschlossenen Kategorien zum λ-Kalkül

Die beiden Konzepte des getypten λ-Kalküls und der kartesisch abgeschlossenen Kategorien dienen dazu, den Funktionenkalkül in mehreren Variablen zu verstehen. Daher ist es nicht überraschend, dass sie äquivalent sind. Sei $\mathbf{C}$ eine kartesisch abgeschlossene Kategorie. Für jede endliche Menge $\{A_i\}_{i\in I}$ gibt es dann ein Produkt mit den Projektionen $p_i : \prod_{i\in I} \to A_i$. Die *interne Sprache* von $\mathbf{C}$ ist ein getypter λ-Kalkül $L(\mathbf{C})$. Wir beschreiben diesen λ-Kalkül, indem wir der Definition 54 folgen: Die Typen von $L(\mathbf{C})$ sind die Objekte von $\mathbf{C}$. Die Typen, die wir für $(TL-1)$ und $(TL-2)$ brauchen, sind die Objekte $\mathbf{1}$, $A \to B$ und B^A. Wir nehmen an, es gibt eine abzählbare Menge von Variablen x_i^A vom Typ A für jedes Objekt A wie in $TL-4$ gefordert. Die Terme werden durch $TL-3$ bis $TL-8$ definiert und die Gleichheiten durch $TL-9$ bis $TL-19$.

Satz 29. *Sei $\mathbf{C}$ eine kartesisch abgeschlossenen Kategorie mit der internen Sprache $\mathcal{L}$. Dann ist $\mathbf{C}(\mathcal{L})$ eine zu $\mathbf{C}$ äquivalente Kategorie.*

Beweis: siehe Lambek und Scott[6]. □

7.3 Pfeile und Terme

Um zu sehen, wie eine einfache Funktion in den beiden Formalismen aussieht, betrachten wir das folgende Beispiel:

Beispiele 25. *Sei $f : \mathbb{N}\times\mathbb{N} \to \mathbb{N}$, die wir traditionell durch die Gleichung $f(x,y) = x^2 + 3xy$ beschreiben. Im getypten λ-Kalkül entspräche das $f = \lambda x : \mathbb{N}\lambda y : \mathbb{N}x^2 + 3xy$, was unmittelbar dem traditionellen Weg entspricht. Die Darstellung in einer kartesisch abgeschlossenen Kategorie sieht ungefähr so aus:*
Der einzige offensichtliche Unterschied besteht also darin, dass wir p_1 und p_2 anstelle von

$$\mathbb{N}\times\mathbb{N} \xrightarrow{(p_1,p_1,p_1,p_2,3)} \mathbb{N}\times\mathbb{N}\times\mathbb{N}\times\mathbb{N}\times\mathbb{N} \longrightarrow \dots$$

$$\dots \xrightarrow{*\times*\times id} \mathbb{N}\times\mathbb{N}\times\mathbb{N} \xrightarrow{id\times *} \mathbb{N}\times\mathbb{N} \xrightarrow{+} \mathbb{N}$$

wobei $*$ die Multiplikation ist. Allerdings kann die kategoriale Formel als $p_1^2 + 3p_1p_2$ umformuliert werden mit der offensichtlichen Semantik.

x und y schreiben. Der Unterschied besteht darin, ob man die Formel oder den Berechnungsprozess wählt. Man beachte, dass die kategoriale Schreibweise deutlich macht, dass mindestens zwei der Multiplikationen parallel ausgeführt werden können. Tatsächlich hat man durch Betonung des Berechnungsprozesses statt des Ergebnisses mit dem kategorialen Ansatz das bessere Instrument in der Hand um solche Fragen zu untersuchen.

An dieser Stelle muss noch eine Sache geklärt werde, die wir bisher nicht explizit behandelt haben. Wenn ein Term $a : A$ Variablen $a_1 : A_1, \ldots, a_n : A_n$ enthält und $b : B$ ein Term in den Variablen $y_1 : B_1, \ldots, y_k : B_k$ ist, dann haben wir bisher gesagt $A \times B$ ist ein Term in den Variablen $a_1 : A_1, \ldots, a_n : A_n, y_1 : B_1, \ldots, y_k : B_k$. Hier entsteht ein Problem, wenn die Variablen sich überlappen. Sei z.B. $x : A$, dann ist der Term (x, x) vom Typ $A \times A$. In diesem Term gibt es aber nur eine freie Variable. Damit entspricht der Pfeil $< x, x >$ einem Pfeil $A \to A \times A$, nämlich dem Pfeil $< id, id >$. Eine Methode, dieses Problem zu lösen besteht darin, die Variablen in geeigneter Weise umzubenennen. Dazu teilt man sie in drei Gruppen ein, nämlich frei nur in a oder nur in b oder frei in a und b.

Seien $x_1 \ldots x_x$ vom Typ $A_1 \ldots A_n$ nur frei in a und $y_1 \ldots y_m$ nur frei in b und $z_1 \ldots z_k$ vom Typ $C_1 \ldots C_k$ in beiden Termen a und b frei. Dann ist

$$(\tilde{a}, \tilde{b}) : A_1 \times \ldots A_n \times B_1 \times \ldots \times B_m \times C_1 \times \ldots C_k \to A \times B$$

definiert als Zusammensetzung von

$$< a, b > \circ < p_1, \ldots p_n, p_{n+m+1}, \ldots, p_{n+m+k}, p_{n+1}, \ldots p_{n+m}, p_{n+m+1} \ldots >$$

Mit anderen Worten es wird eine verallgemeinerte Diagonale für die mehrfachen Einträge genutzt. Natürlich nur dort, wo tatsächlich dieselben Variablen verwendet werden, nicht nur solche vom selben Typ. Falls x und y verschiedene freie Variablen vom Typ A sind, dann entspricht (x, y) einem Pfeil vom Typ $A \times A \to A \times A$, nämlich dem identischen Pfeil.

Vorteile von CCC-s

Unter allen Vorteilen von CCC-s ist der wichtigste, dass es, da es keine Variablen gibt, auch keine Konflikte mit Variablen gibt. Wir betrachten die Regel, die besagt, dass unter bestimmten Umständen gilt:

$$\lambda_{x:A}\varphi(x) =_X \lambda_{y:A}\varphi(y)$$

Eine kategoriale Entsprechung dieser Regel ist dass, wenn

$$f : A_1 \times \ldots \times A_n \times A \to B$$

dann ist

$$\lambda_{x:A}f = \lambda_{x:A}f : A_1 \times \ldots \times A_n \to (A \to B)$$

was immer wahr ist. Für die Regel im λ-Kalkül müssen noch die näheren Umstände, unter denen sie angewendet werden darf, beschrieben werden. Wenn x und y zwei Variablen vom Typ A sind und $\varphi(x) = y$, dann gilt nicht $\lambda_x\varphi(x) = \lambda_y\varphi(y)$, da die linke Seite y immer noch als freie Variable hat, während die rechte Seite keine freien Variablen hat.

Das Problem rührt daher, dass Variablen anstelle von Platzhaltern verwendet werden. Ein anderer Vorteil der kategorialen Betrachtung zumindest auf lange Sicht, ist der, dass die Komposition schon eingebaut ist. Damit steht die gesamte Maschinerie der kommutativen Diagramme usw. als Werkzeug zu Verfügung.

7.4 Der intuitionistische Sequenzenkalkül

Unter den vielen Varianten der intuitionistischen Logik wie sie sich in Systemen des natürlichen Schließens ausdrücken lassen, spielt Gentzens Sequenzenkalkül für die Informatik eine besondere Rolle. Das hat mehrere Gründe: zum einen ist einer der Hauptaspekte der prozeduralen Intepretation von Beweisen der, dass die Annahmen, von denen das Vorkommen einer Formel in einer Ableitung abhängt explizit gemacht werden können, was genau der Begriff der Sequenz widerspiegelt. Da Sequenzenkalküle außerdem als Metakalküle für die Ableitbarkeitsrelation im entspercehnden Kalkül des natürlichen Schließens verstanden werden kann, schein er das beste Instrument zu sein, um die Struktur und die Eigenschaften von Beweisen zu untersuchen.

Aber auch wenn man mit Sequenzen umgeht, kann es schwierig sein alle Abhängigkeiten einer Formel von ihren Prämissen zu verstehen, wenn man einigen der üblichen Gewohnheiten beim Aufschreiben der Sequenzen folgt, z.B. der Vermeidung von Wiederholungen von Formeln in den Voraussetzungen. Das wird besonders klar, wenn man eine Sequenz als effektiven Prozess zur Transformation von Beweisen der Voraussetzungen in Beweise für die Konklusion ansieht. Es erscheint nicht nur sinnvoll, sondern angemessen die Möglichkeit zu garantieren, dass zwei oder mehr Beweise für dieselbe Formel behandelt werden können. Insbesondere bedeutet das, die Wiederholung von Formeln in der Prämisse zu erlauben, möglicherweise jedesmal mit einem anderen Beweis verbunden. Was man außer den Regeln für das Zusammenführen von Beweisen(also den logischen Regeln) braucht, sind sogenannte *strukturelle Regeln*, eine Regel um Formeln innerhalb einer Sequenz „zu bewegen“ eine Kontraktionsregel, die es erlaubt zwei Beweis für eine Formel als äquivalent anzusehen und daher mehrfache Vorkommen von Prämissen zu streichen, sowie eine Weakening-Regel um zusätzliche Prämissen in einen Beweis einzuführen. Sequenzenkalküle wurden in den 30er Jahren von Gentzen eingeführt. Die elementaren Aussagen sind Sequenzen der Form $\Gamma \vdash B$ mit der Konklusion B und einer (möglicherweise leeren) Liste von Prämissen und wird interpretiert als Aussage, dass es einen Beweis für B gibt, der nur Hypothesen aus Γ verwendet. Wir werden uns nur mit einem Teilkalkül beschäftigen, der als als logische Konnektive nur die Konjunktion und die Implikation hat, den sogenannten „positiven Kalkül“. In unserer Darstellung tritt jede Formel A zusammen mit einem zugehörigen Beweis auf, formal dargestellt durch einen λ-Term $a : A$. Die Verbindung zwischen Beweisen und Formeln wird durch die Inferenzregeln des Kalküls definiert. Wenn es eine Ableitung der Sequenz $\Gamma \vdash b : B$ gibt, dann beschreibt b den Herleitungsbaum. Andererseits können die so modifizierten Inferenzregeln als Metaregeln für die Beweismanipulation betrachtet werden

Diese Ansicht des Sequenzenkalküls als Metakalkül für die Ableitbarkeitsrelation ist der besondere Aspekt dieses formalen Systems.

Definition 55. *Das logische Alphabet besteht aus:*

1. *atomaren Aussagen,* $A, B, \ldots$

2. *logische Symbolen* $\times$, $\to$

Wohlgeformte Formeln (Typen) sind

1. *jede atomare Aussage ist eine Formel*

2. *wenn A und B Formeln sind, dann ist auch* $A \times B$ *eine Formel*

3. *wenn A und B Formeln sind, dann ist auch* $A \to B$ *eine Formel*

Zwischen Formeln und Typen des getypten λ-Kalküls mit expliziten Paaren besteht eine Bijektion. Wir können jeder Formel B einen λ-Term des entsprechenden Typs zuordnen, der deren Beweis repräsentiert. Insbesondere, wenn $x_1 : A_1, \dots x_n : A_n$ freie Variablen in M sind, dann ist $M : B$ ein Beweis von M, der von den Hypothesen $A_1 \dots, A_n$ abhängt.

Dieser Ansatz verträgt sich gut mit dem intuitionistischen Verständnis von Beweisen als effektive Prozedur, die die Gültigkeit von B anerkennt, sowie die Gültigkeit der Prämissen bewiesen ist.

Das heißt, wenn M ein Beweis für B ist, der von einem generischen Beis $x : A$ abhängt, dann bekommt man einen Beweis für $A \to B$ durch Abstraktion von $x : A$, d.h. $\lambda x : A.M : A \to B$. Die Inferenzregeln formalisieren die Konstruktion eines komplexen Beweises aus einfachen Beweisen.

Ein wohlgetypter Term $M : A$ heißt *Beweis*, wenn er unter logischem Aspekt betrachtet wird. Eine Variable $x : A$ heißt *generischer Beweis von A*.

Eine *intuitionistische Sequenz* hat die syntaktische Form

$$x_1 : A_1, \dots x_n : A_n \vdash M : B$$

wobei $x_1 : A_1, \dots x_n : A_n$ eine endliche (möglicherweise leere) Liste von verschiedenen generischen Beweisen ist und $M : B$ ein Beweis für B dessen freie Variablen aus der Menge $\{x_1, \dots x_n\}$ stammen. Zu jeder Formel auf der linken Seite gehört eine bestimmte Variable. Die intuitive Bedeutung der Sequenz $x_1 : A_1, \dots x_n : A_n \vdash M : B$ ist, dass sie den Prozess beschreibt, wie man den Beweis M aus den Beweisen für $A_1, \dots A_n$ aufbaut, das heißt eine Funktion des Typs $A_1 \times A_n \to B$ konstruiert.

Wir verwenden griechische Großbuchstaben $\Gamma, \Delta, \dots$ um endliche Folgen von generischen beweisen auf der linken Seite zu beschreiben.

Eine *Inferenz* ist ein Ausdruck der Form

$$\frac{S_1}{S} \qquad \frac{S_1 \quad S_2}{S}$$

Das logische System beschränkt die legalen Inferenzen auf diejenigen, die durch folgende Regeln beschrieben werden:

Definition 56. *Axiome und Regeln*

$$[Axiom] \qquad x : A \vdash x : A$$

$$[Tausch] \qquad \frac{\Gamma, x : A, y : B, \Gamma_1 \vdash M : C}{\Gamma, y : B, x : A, \Gamma_1 \vdash M : C}$$

$$[Weakening] \qquad \frac{\Gamma \vdash M : B}{\Gamma, x : A \vdash M : B}$$

$$[Kontraktion] \qquad \frac{\Gamma, x : A, y : A \vdash M : B}{\Gamma, z : A \vdash [z/x][z/y]M : B}$$

$$[Schnitt] \qquad \frac{\Gamma \vdash M : A \qquad \Gamma_1, x : A, \Gamma_2 \vdash N : B}{\Gamma_1, \Gamma, \Gamma_2 \vdash [M/x]N : B}$$

$$[\to, r] \qquad \frac{\Gamma, x : A \vdash M : B}{\Gamma \vdash \lambda x : A.M : B}$$

$$[\to, l] \qquad \frac{\Gamma_1 \vdash M : A, \Gamma_2, x : B \vdash N : C}{\Gamma_1, \gamma_2, y : A \to B \vdash [yM/x]N : C}$$

$$[\times, r] \qquad \frac{\Gamma \vdash M : A, \Gamma \vdash N : B}{\Gamma \vdash (M, N) : A \times B}$$

$$[\times, l_1] \qquad \frac{\Gamma, x : A \vdash M : C}{\Gamma, z : A \times B \vdash [proc_1(z)/x]M : C}$$

$$[\times, l_1] \qquad \frac{\Gamma, x : A \vdash M : C}{\Gamma, z : A \times B \vdash [proc_2(z)/x]M : C}$$

Die Formel A in der weakening-Regel heißt *Weakening-Formel*, entsprechend definiert man auch die *Schnittformel*. In der Kontraktionsformel unterscheiden wir die *kontrahierenden* Formeln $x : A, y : A$ und die *kontrahierte Formel* $z : A$. Die Formeln $A \to B$ und $A \times B$ in den Einführungsregeln für die entsprechenden Junktoren heißen *Hauptformeln* der Inferenz; A und B sind *Hilfsformeln*.

Man sieht leicht, dass in jeder Inferenzregel der Beweis auf der rechten Seite der unteren Sequenz wohlgeformt ist, wenn das für den in der oberen Sequenz gilt.

Eine *Ableitung* D ist ein Baum von Sequenzen, der die folgenden Bedingungen erfüllt:

1. Die obersten Sequenzen von D sind Axiome

2. jede Sequenz bis auf die unterste(die Wurzel) ist die obere Sequenz einer Inferenz, deren untere Sequenz ebenfalls in D liegt.

Eine unterste Sequenz in einer Ableitung D heißt *Endsequenz* von D. Ein Pfad in einer Ableitung von D von einem Blatt zur Wurzel heißt *thread*. Eine Ableitung ohne Verwendung der Schnittregel heißt *schnittfrei*.

8 Fixpunkte in kartesisch abgeschlossenen Kategorien

Einer der oft zitierten Gründe, warum Informatiker den ungetypten λ-Kalkül studieren ist die Existenz eines Fixpunktkombinators. Dieser ist ein Element Y mit der Eigenschaft $x'(Y'x) = Y'x$, d.h. $Y'x$ ist ein Fixpunkt für x. Das kann in einem getypten λ-Kalkül kaum Sinn ergeben. Hat man zum Beispiel einen Typ für die natürlichen Zahlen, dann kann man nicht annehmen, dass es dort einen Fixpunkt für die Nachfolgerfunktion *succ* gibt. Andererseits gibt es ohnen eine Fixpunktkombinator keinen Weg eine funktional Form wie

$$f = p \to q; H(f) \tag{11}$$

(Falls p, Dann q, Sonst $H(f)$) zu interpretieren. Dabei soll $H(f)$ eine Funktion in f sein. Wir studieren hier die syntaktische Frage; es gibt keine Garantie, dass der Fixpunktkombinator eine terminierende Funktion ergibt. Im Fall des üblichen Fixpunktkombinators Y ist der Fixpunkt der identischen Funktion die Funktion $(\lambda x.xx)'(\lambda x.xx)$ ein typisches Beispiel für eine nichtdeterminierende Schleife.

Um ein Programm zu schreiben, dass diese Form implementiert müssen wir etwas über die Nator von H wissen. Zum Beispiel ist, wenn $H(f) = g \circ f \circ h$ das fragliche f das (unendliche) Programm:

$$\begin{array}{l}
\ \texttt{IF}\ p(x)\ \texttt{THEN OUTPUT}\ q(x) \\
\texttt{ELSE IF}\ p \circ h(x)\ \texttt{THEN OUTPUT}\ g \circ q \circ h(x) \\
\texttt{ELSE IF}\ p \circ h^2(x)\ \texttt{THEN OUTPUT}\ g \circ q \circ h(x) \\
\vdots \\
\texttt{ELSE IF}\ p \circ h^n(x)\ \texttt{THEN OUTPUT}\ g \circ q \circ h(x) \\
\vdots
\end{array}$$

Natürlich hält das Programm niemals an, wenn die Bedingung nicht erfültt wird. Man kann dasselbe Programm auch kürzer schreiben als:

```
 PROC f(x)
IF p(x)
THEN OUTPUT q(x)
ELSE OUTPUT H(f)(x)
ENDIF ENDPROC
```

Ein Ausweg aus dem Dilemma ist, dass man sich überlegt, dass man nicht für alle Funktionen Fixpunkte haben muss, sondern nur für manche. Dann erhebt sich die Frage, wie man erkennt, welche man braucht und zu zeigen, dass man genügend hat, um alle Schleifen zu beschreiben, die man braucht. Wir bearbeiten zuerst die erste Frage. Sei D ein halbgeordnetes Objekt in einer CCC, das heißt auf jeder Hom-Menge $Hom_{\mathbf{C}}(A, D)$

gibt es eine Halbordnung, so dass für jeden Pfeil $f : B \to A$ und für jedes Paar $g, h \in Hom_{\mathbf{C}}(A, D)$ mit $g \leq h$ gilt auch $f \circ g \leq f \circ h$ (in $Hom_{\mathbf{C}}(B, D)$).

Definition 57. *D heißt ω-vollständige Halbordnungsobjekt oder ω-CPO Objekt, wenn es vollständig ist und alle Hom-Mengen sind ω-vollständige Halbordnungen, d.h. jede abzählbare aufsteigende Kette hat eine kleinste obere Schranke.*

Seien D und D' ω-vollständige Objekte. dann heißt ein Pfeil $f : D \to D'$ ω-stetig, wenn für jedes Objekt A und jede Folge $g_0 \leq g_1 \leq \ldots$ von Pfeilen von A nach D mit supremum g gilt, dass

$$sup\{f \circ g_i\} = f \circ g$$

Ofenbar sind ω-stetige Pfeile ordnungserhaltend. Ein halbgeordnetes Objekt D ist strikt, wenn es einen Pfeil $\perp : 1 \to D$ gibt, so dass für jedes Objekt A und jeden Pfeil $f : A \to D$ gilt $\perp \circ ! \leq f$, mit anderen Worten $\perp$ ist das kleinste Element von D.

Satz 30. *Sei D ein striktes ω-CPO-Objekt, und $f : D \to D$ ein ω-stetiger Pfeil. Dann gibt es ein Element $fix(f) : 1 \to D$ mit $f \circ fix(f) = fix(f)$. Das Element $fix(f)$ ist das kleinste Element in D mit dieser Eiegenschaft.*

Beweis: Da $\perp$ das kleinste Element von D ist, gilt $\perp \leq f \circ \perp$, (der Terminalpfeil von 1 idt id_1). Da f monoton ist, folgt daraus $f \circ \perp \leq f \circ f \circ \perp$. Per Induktion erhalten wir auch $f^n \circ \perp \leq f^{n+1} \circ \perp$. Wir erhalten eine aufsteigende Kette

$$\perp \leq f \circ \perp \leq \ldots f^n \circ \perp \leq f^{n+1} \circ \perp \leq \ldots$$

und definieren $fix(f)$ als deren Supremum. Um zu zeigen, dass dies der kleinste Fixpunkt ist, zeigt man für einen beliebigen anderen Fixpunkt $d : 1 \to D$ durch Induktion beginnend mit $\perp \leq d$, dass $f^n(\perp) \leq d$. $\square$

Es ist nicht unmittelbar ersichtlich, was die obigen Ausführungen über Fixpunkte mit der Interpretation von Programmen zu tun haben. Obwohl die meisten üblichen Datentypen eine partielle Ordnung ahenm und sogar als ω-vollständig vorgestellt werden können, sind die meisten Funktionen nicht ordnungserhaltend(wie z.B. quadrieren). Aber es sind nicht die Basistypen, für die wir die obigen Konstruktionen anwenden wollen, sondern es sind die Pfeiltypen. Wir wollen nicht den Fixpunkt der Nachfolgerfunktion finden, sondern den einer Funktion der f die Funktion $p \to q; H(f)$ wie in Formel 11 zuordnet. Außerdem sieht man dass die entsprechende Ordnungsrelation diejenige in der $f \leq g$, wenn Definitionsbereich von f in dem von g enthalten ist und wenn sie auf diesem übereinstimmen.

Wir betrachten einen Datentyp D, der ein Objekt unserer Kategorie von Typen ist. Man vergesse alles über eine Ordnung, die möglicherweise auf D existiert. Wir definieren einen neuen Datentypen, den wir mit $D_\perp$ bezeichnen. Dieser soll das Objekt $D + \{\perp\}$ sein, die Ordnungsrelation ist einfach so definiert, dass $\perp \leq d$ für alle $d \in D$ und für $d \neq \perp$ und $d \neq d'$ gilt niemals $d \leq d'$ (damit ist $D_\perp$ eine flache CPO).

Wir nehmen an, $D_\perp$ wäre ω-vollständig (unter bestimmten Bedingungen an die Kategorie, die z.B: von **SET** erfüllt werden, gilt das auch). Unter dieser Annahme ist der Typ $[A \to D_\perp]$ ebenfalls ein ω-CPO-Objekt. Dann ist eine wachsende Folge von Pfeilen vom Typ $B \to [A \to D_\perp]$ äquivalent zu einer wachsenden Folge von Pfeilen vom Typ

$B \times A \to D_\perp$ und diese hat nach Voraussetzung ein Supremum. Daher hat jeder stetige Endomorphismus auf $[A \to D_\perp]$ einen Fixpunkt.

Um diese Erkenntnsi auf den Backus-Operator anzuwenden, müssen wir nur zeigen, dass H stetig ist. Man kann einen Pfeil $A \to D_\perp$ verstehen als Teil eines komplementären Teilobjektes von A nach D. (Das Paar A,B mit Injektionen $i : A \to C$ und $j : B \to C$, ist ein komplemetäres Teilobjekt, wenn das Diagramm unten zugleich ein Pullback und Pushout ist. Dieses wird erweitert zu einem Pfeil, der den Wert $\perp$ für das komplementäre

$$\begin{tikzcd} & 0 \arrow[dl] \arrow[dr] & \\ A \arrow[dr, "i"'] & & B \arrow[dl, "j"] \\ & C & \end{tikzcd}$$

Teilobjekt bekommt. Ein Pfeil $\phi : [A \to D] \to [A \to D]$ erhält die Ordnung wenn wann immer g den Pfeil f erweitert, dann erweitert $\phi(g)$ den Pfeil $\phi(f)$.Wie schon erwähnt ist die Ordnungsrelation die der Erweiterung des Definitionsbereiches. Um den Satz 30 anwenden zu können, müssen wir zeigen, dass wenn f die Einschränkung von g auf einen kleineren Definitionsbereich ist, dann ist $H(f)$ eine Einschränkung von $H(g)$. Man muss zeigen, dass wenn f das Supremum einer aufsteigenden Folge $f_0 \leq f_1 \leq \ldots \leq f_n \leq \ldots$ ist, dann ist $H(f)$ das Supremum von $H(f_0) \leq H(f_1) \leq \ldots \leq H(f_n) \leq \ldots$. Nun betrachtet Backus die folgenden funktionalen Formen für H

FF-1 $H(f) = r$ (eine Konstante; r ist keine Konstante, aber H ist eine)

FF-2 $H(f) = f_i$, wobei $f =< f_1, \ldots f_n >$ bei backus gibt es nur Listen

FF-3 $H(f) = \Gamma \circ < E_1(f), \ldots E_n(f) >$, wobei $E_1 \ldots E_n$ (einfachere) funktionale Formen sind un Γ eine Funktion ist.

Alle drei Regeln generieren ω-stetige Funktionen auf $[D \to D]$ und sind damit durch Satz 30 abgedeckt. Natürlich gibt es noch viele andere Möglichkeiten für H. Allerdings muss hier betont werden, dass Backus hauptsächlich mit Konvergenz beschäftigt ist und daher andere Bedingungen an H stellt, während wir uns mit Programmen beschäftigen.

9 Rekursive Domain Gleichungen

Eine der frühen Anwendungen der Kategorietheorie in der Informatik war die Lösung von rekursiven Domänengleichungen. Diese Art von Gleichung ist typisch für jede Sprache, in der explizite oder implizite Formen der Selbstanwendung von Datentypen(wie zum Beispiel einen rekursiven Programmaufruf) erlaubt sind. Um eine Semantik für den reinen λ-Kalkül zu erhalten brauchen wir Domänen, die isomorph zu ihrem eigenen Funktionenraum sind, wenn wir außerdem über einer festen Menge von Atomen verfügen wollen, dann brauchen wir eine Lösung für die Gleichung

$$X \cong A + (X \to X) \tag{12}$$

Allgemein kann man rekursive Spezifikation von Domänen ansehen als Spezialfall der rekursiven Definition von Datentypen, was wohl ein noch wichtigerer Gegenstand der Informatik ist. Zum Beispiel ist der Datentyp aller Listen von Objekten vom Typ A eine Lösung der Gleichung

$$A_{List} = Nil + A \times A_{List} \tag{13}$$

wobei Nil der einelementige Datentyp ist. Auf verschiedenen Wegen hat die allgemeine und einheitliche Theorie, die Arten von Gleichungen, die in der Kategorientheorie entwickelt wurde, dazu geführt, dass in den meisten modernen Programmiersprachen zahlreiche Beschränkungen, die es für die Definition von rekursiven Datentypen in älteren Sprachen gegeben hat, verschwunden sind. Die erste mathematische Schwierigkeit im Verstehen der rekursiven Spezifikationen von Datentypen ist, dass sie nicht immer eine mengentheoretische Lösung haben. Gleichung 12 kann in **SET** schon aus Kardinalitätsgründen keine Lösung haben, wenn man $A \to B$ als Menge der Funktionen von A nach B interpretiert. Ein natürlicher Ausweg ist andere Kategorien zu betrachten, die weniger Pfeile haben, aber immer noch genug, um alle „berechenbaren Funktionen" über den gewünschten Datentypen einzuschließen. Die Relevanz der Morphismen erklärt, warum die kategoriale Betrachtung sich so natürlich in diesem Gebiet etabliert hat. Natürlich sollte die betrachtete Kategorie immer noch kartesisch abgeschlossen sein, um eine korrekte Interpretation des Funktionenraums zu gewährleisten. Wenn $\mathbf{C}$ einmal festgelegt ist, braucht man eine Art von *Fixpunkttechnik*, um rekursive Domain Gleichungen wie 12 zu lösen. Allgemein interessieren wir uns für Gleichungen der folgenden Form:

$$X \cong FX \tag{14}$$

wobei X ein Objekt von $\mathbf{C}$ ist und $F : \mathbf{C} \to \mathbf{C}$ ein kovarianter Endofunktor.

Um herauszufinden in welchen Kategorien und für welche Arten von Funktoren eine solche Fixpunktgleichung gelöst werden kann müssen wir zunächst noch ein paar Begriffe einführen.

In verscheidenen Abschnitten haben wir universelle Konstruktionen wie Produkt, Equalizer, Pullbacks usw. kennengelernt. Die Konstruktionen sind Spezialfälle von sogenannten Limiten bzw. Colimiten.

Definition 58. *Sei $\mathbf{C}$ eine Kategorie und $\mathcal{D}$ ein Diagramm mit Objekten $\{D_i\}, i \in I$. EinKegel über $\mathcal{D}$ ist ein Objekt C und eine Familie von Pfeilen $\{f_i : C \to D_i\}_{i \in I}$, so dass für alle $i, j \in I$ und alle Kanten $e : D_i \to D_j$ aus dem Diagramm gilt $e \circ f_i = f_j$. Cokegel werden dual dazu definiert.*

In einer Halbordnung entsprechen die Kegel unteren Schranken für alle Elemente des Diagramms, Cokegel oberen Schranken.

Für ein gegebenes Diagramm $\mathcal{D}$ bilden die Kegel über $\mathcal{D}$ eine Kategorie $\mathbf{Cones}_{\mathbf{C},\mathcal{D}}$, deren Pfeile von $(C, \{f_i : C \to D_i\}_{i \in I})$ nach $(C', \{f'_i : C' \to D_i\}_{i \in I})$ solche Pfeile $g : C \to C'$ sind mit $f'_i \circ g = f_i$.

Definition 59. *Sei $\mathbf{C}$ eine Kategorie und $\mathcal{D}$ ein Diagramm. Ein Limes von $\mathcal{D}$ ist ein Terminalobjekt in $\mathbf{Cones}_{\mathbf{C},\mathcal{D}}$. Colimiten definiert man dual dazu, als Initialobjekte in $\mathbf{Cocones}_{\mathbf{C},\mathcal{D}}$.*

Betrachtet man zwei Objekte A, B als Diagramm, dann ist jedes Objekt C mit einem Paar von Pfeilen $c_1 : C \to A$ und $c_2 : C \to B$ ein Kegel über diesem Diagramm. Das kartesische Produkt ist gerade der Limes über diesen Kegeln und die Universalitätseigenschaft, dass zu jedem Objekt C' mit Paaren von Pfeilen nach $c'_1 : C' \to A$ und $c'_1 : C' \to B$ (das ist nämlich gerade ein Kegel, also ein Objekt in $\mathbf{Cones}_{\mathbf{C},\mathcal{D}}$) ein eindeutig bestimmter Pfeil $< c'_1, c'_2 >: C' \to C$ existiert mit $c_i \circ < c'_1, c'_2 >= c'_i$ definiert gerade die Terminalabbildung von C'.

Der Limes über einem Diagramm, das aus drei Objekten D_1, D_2, D und Pfeilen $e_1 : D_1 \to D$ und $e_2 : D_2 \to D$ ist das Pullback von e_1 entlang e_2.

Natürlich existieren nicht in jeder Kategorie alle Limiten über allen möglichen Diagrammen. Schon im einfachen Beispiel, dass eine Halbordnung als Kategorie betrachtet wird, muss nicht notwendig ein Supremum von zwei Objekten existieren.

Interessant ist der Zusammenhang zwischen Funktoren und Limiten. Während wegen den Erhaltung der Kommutativität von Diagrammen das Bild eines Kegels unter einem Funktor ein Kegel über dem Bild des Diagramms unter diesem Funktor ist, ist das für Limiten nicht der Fall. Wir von einem Funktor F, dass er Limiten erhält, wenn, falls $(L, \{f_i\}_{i\in I})$ ein Limes über einem Diagramm $\mathcal{D}$ ist, dann ist $(F(L), \{F(f_i)\}_{i\in I})$ ein Limes über $F(\mathcal{D})$.

Um uns dem Thema rekursive Gleichungen wieder zu nähern, brauchen wir noch den Begriff der ω-Vollständigkeit. Man stelle sich vor, man hätte eine unendliche Kette von Pfeilen zwischen einem Objekt und dem Bild dieses Objetes unter einem Funktor als Nachfolger. Jeder dieser Pfeile wäre dann eine genauere Approximation des gewünschten Fixpunktes. Der Fixpunkt selbst soll als Colimes dieser Kette entstehen. Wir sind also an solchen Kategorien interessiert, in denen solche Colimiten immer existieren.

Definition 60. 1. *Ein ω-Diagramm in einer Kategorie $\mathbf{C}$ ist ein Diagramm der folgenden Form:*

$$D_0 \xrightarrow{f_0} D_1 \xrightarrow{f_1} D_2 \longrightarrow \cdots \longrightarrow D_n \xrightarrow{f_n} D_{n+1} \longrightarrow \cdots$$

2. *Eine Kategorie heißt ω-vollständig(ω-covollständig), wenn sie Limiten(Colimiten) für alle Diagramme hat.*

3. *Ein Funktor $F : \mathbf{C} \to \mathbf{C}$ heißt ω-stetig, wenn seine Objektabbildung auf $\mathbf{C}$ stetig ist.*

Beispiele 26. *Wenn $\mathbf{C}$ eine Halbordnung ist, dann ist ein ω-Diagramm eine ω-Kette. Eine Halbordnung $\mathbf{C}$ ist genau dann ω-covollständig, wenn $\mathbf{C}$ eine vollständige Halbordnung ist. Ein Funktor F auf einer Halbordnung ist genau dann ω-stetig, wenn er stetig ist.*

Satz 31. *Sei $\mathbf{C}$ eine Kategorie mit Initialobjekt 0. Sei $F : \mathbf{C} \to \mathbf{C}$ ein ω-stetiger Funktor und sei $z \in Hom_{\mathbf{C}}(0, F(0))$ der initiale Pfeil. Sei $(C, \{t_i : F^i(0) \to C\}_{i\in\mathbb{N}})$ der Colimes für das ω-Diagramm $(\{F^i(0)\}_{i\in\mathbb{N}}, \{F^i(z)\}_{i\in\mathbb{N}})$, wobei $F^0(0) = 0$ und $F^0(z) = z$. Dann ist $C \cong F(C)$.*

Beweis: Nach Voraussetzung ist $(F(C), \{F(t_i) : F^{i+1}(0) \to F(C)\}_{i\in\mathbb{N}})$ ein Colimes für $(\{F^{i+1}(0)\}_{i\in\mathbb{N}}, \{F^{i+1}(z)\}_{i\in\mathbb{N}})$ und $(C, \{t_{i+1} : F^{i+1}(0) \to F(C)\}_{i\in\mathbb{N}})$ ein Cokegel für dasselbe Diagramm. Daher existiert wegen Universalität ein eindeutig bestimmter Pfeil $h : FC \to C$ mit $h \circ F(t_i) = t_{i+1}$ für alle $i \in \mathbb{N}$.

Zu $(F(C), \{F(t_i) : F^{i+1}(0) \to F(C)\}_{i\in\mathbb{N}})$ können wir einen weiteren Pfeil, nämlich $z_{FC} : 0 \to FC$ hinzufügen und damit ist $(F(C), \{z_{FC}\} \cup \{F(t_i) : F^{i+1}(0) \to F(C)\}_{i\in\mathbb{N}})$ ein Cokegel für das ursprüngliche ω-Diagramm. Wegen Universalität des Colimes C muss dann ein Pfeil $k : C \to FC$ existieren mit $k \circ t_{i+1} = F(t_i)$ und $z_{FC} = k \circ t_0$. Also ist $h \circ k \circ t_{i+1} = h \circ F(t_i) = t_{i+1}$ für alle $i \in \mathbb{N}$. Wegen Universalität von C muss daher $h \circ k = id_C$. Analog beweist man $k \circ h = id_{FC}$. Damit haben wir $C \cong FC$. □

Falls $\mathbf{C}$ ω-covollständig ist, dann gibt Theorem 31 eine Lösung für 14. Das Hauptproblem ist, dass nicht immer eine Definition für ein solches F existiert, wie wir im Folgenden zeigen werden.

9.1 Das Problem der kontravarianten Funktoren

Wir betrachten erneut die Gleichung 12. Um Theorem 31 anwenden zu können, müssen wir voraussetzen, dass die Kategorie $\mathbf{C}$ ω-covollständig ist. Außerdem brauchen wir kartesische Abgeschlossenheit und Coprodukte um den in der Gleichung auftretenden Operatoren $\to$ und $+$ die erwartete Interpretation zu geben. Solche Kategorien gibt es, am Ende des Abschnitts werden wir eine angeben. Der nächste Schritt ist, einen kovarianten Funktor anzugeben mit $F(X) = A + (X \to X)$ für jedes Objekt X. Die erste Idee ist, F als Komposition der Funktoren $+$ und Exponentiation, die zu den Abschlusseigenschaften von $\mathbf{C}$ gehören, zu konstruieren.

Sei $A + - : \mathbf{C} \to \mathbf{C}$ der Funktor, der jedes X auf $A + X$ abbildet und jeden Pfeil f auf $id_A + f$. Mit dem Exponentialfunktor EXP und dem Diagonalfunktor Δ könnten wir versuchen, F zu definieren als $F = (A + -) \circ EXP \circ \Delta$.

Allerdings ist das nicht möglich, da EXP in der ersten Komponente kontravariant ist und daher nicht mit dem Diagonalfunktor komponiert werden kann. Mit anderen Worten $EXP : \mathbf{C}^{op} \times \mathbf{C} \to C$ und $\Delta : \mathbf{C} \to \mathbf{C} \times \mathbf{C}$.

Um das Problem zu lösen, müssten wir einen Weg haben, wie man einen Funktor F in einen Funktor F^* verwandeln kann, der sich auf den Objekten wie F verhält, aber auf den Pfeilen kovariant ist in all seinen Komponenten. Unglücklicherweise ist das im Allgemeinen nicht möglich. Allerdings gibt es einen einfachen Weg, einen kontravarianten Funktor $F : \mathbf{C} \to \mathbf{C}$ in einen kovarianten Funktor F^* in einer zugehörigen Kategorie $\mathbf{C}^*$ (die dieselben Objekte wie $\mathbf{C}$ hat und deren Isomorphismen auch Isomorphismen für $\mathbf{C}$ sind) zu transformieren.

Falls ein solches $\mathbf{C}^*$ ω-covollständig wäre und ein Terminalobjekt hätte und der Funktor F^* ω-stetig wäre, dann könnten wir Satz 31 anwenden um eine Lösung in $\mathbf{C}^*$ finden und daraus eine Lösung in $\mathbf{C}$ ableiten. Man muss an dieser Stelle anmerken, dass $\mathbf{C}^*$ weder unbedingt kartesisch abgeschlossen, noch abgeschlossen unter Coprodukten sein muss, da wir schon wissen, welche Bedeutung wir neuen Objekten in $\mathbf{C}$ geben müssen.

Wir definieren eine solche Kategorie $\mathbf{C}^*$ als passende Teilkategorie der unten definierten Kategorie $\mathbf{C}^{+-}$.

Definition 61. *Sei $\mathbf{C}$ eine Katgeorie. Die Kategorie $\mathbf{C}^{+-}$ hat dieselben Objekte wie $\mathbf{C}$, ihre Pfeile $f \in Hom_{\mathbf{C}^{+-}}(A, B)$ sind Paare (f^+, f^-) aus $\mathbf{C}$-Pfeilen mit $f^+ \in Hom_{\mathbf{C}}(A, B)$*

und $f^- \in Hom_{\mathbf{C}}(B, A)$. *Komposition ist definiert durch* $(f^+, f^-) \circ (g^+, g^-) = (f^+ \circ g^+, g^- \circ f^-)$.

Der intuitive Weg, um die Kategorie $\mathbf{C}^{+-}$ zu verstehen, ist der folgende. Wir betrachten die Objekte von $\mathbf{C}$ als Datentypen und es gibt nur dann einen Pfeil von A nach B, wenn es ein Paar von Verbindungen gibt, um von einem zum anderen zu kommen. Offenbar sind zwei Objekte, die in $\mathbf{C}^{+-}$ isomorph sind auch isomorph in $\mathbf{C}$.

Als nächstes wollen wir kovariante Funktoren auf $\mathbf{C}^{+-}$ aus beliebigen Funktoren in $\mathbf{C}$ konstruieren. Der Einfachheit halber machen wir das hier nur für Bifunktoren F, die kovariant in der ersten und kontravariant in der zweiten Komponente sind.

Definition 62. *Sei* $F : \mathbf{C} \times \mathbf{C} \to \mathbf{C}$ *ein Funktor, der kontravariant in der ersten und kovariant in der zweiten Komponente ist. Der kovariante Funktor* $F^{+-} : \mathbf{C}^{+-} \times \mathbf{C}^{+-} \to \mathbf{C}^{+-}$ *ist definiert durch:*

$$F^{+-}(A, B) = F(A, B) \qquad F^{+-}((f^+, f^-), (g^+, g^-)) = (F(f^-, g^+), F(f^+, g^-)).$$

Ein Problem der Kategorie $\mathbf{C}^{+-}$ ist, dass man nicht davon ausgehen kann, dass sie Colimiten für alle ω-Ketten hat. Tatsache ist aber, dass die Idee, auf der die obige Definition basiert in jeder Teilkategorie $\mathbf{C}^*$ von $\mathbf{C}^{+-}$ funktioniert, falls man nur solche Funktoren F betrachtet, bei denen $(F(f^-, g^+), F(f^+, g^-))$ ein Pfeil in $\mathbf{C}^*$ ist.

Unser Ziel ist es, solche Teilkategorien $\mathbf{C}^*$ zu finden, bei denen einfache und übliche Eigenschaften von $\mathbf{C}$ (wie z.B. die Existenz von Limiten für Diagramme)ausreichen um die Existenz von Colimiten von beliebigen ω-Ketten zu garantieren. Um eine solche Kategorie $\mathbf{C}^*$ zu finden, machen wir uns folgende Überlegung zunutze. Als wir in Satz 31 das ω-Diagramm $(\{F^i(0)\}_{i\in\mathbb{N}}, \{F^i(z)\}_{i\in\mathbb{N}})$ betrachtet haben, war die Idee dahinter, dass dies eine Kette von immer feineren Approximationen an den Colimes ist. D.h. in einem gewissen Sinn muss ein Morphismus $F^i(z)$ erklären, *warum* $F^i(0)$ weniger ist als $F^{i+1}(0)$: bei Übergang von $F^i(0)$ zu $F^{i+1}(0)$ darf keine Information verloren gehen. Wir werden versuchen eine Teilkategorie $\mathbf{D}$ von $\mathbf{C}^{+-}$ zu definieren, deren Morphismen gerade diese Art von Beziehungen zwischen Objekten ausdrücken.

Eine der Teilkategorien von $\mathbf{C}^{+-}$, die anscheinend diese Bedingung erfüllen, ist $\mathbf{C}^{Ret}$, deren Morphismen (f^+, f^-) die Eigenschaft haben, dass $f^- \circ f^+ = id$. Diese Kategorie ist besonders deswegen interessant, da *jeder* Funktor F auf $\mathbf{C}$ mittels Definition 62 in einen Funktor auf $\mathbf{C}^{Ret}$ verwandelt werden kann.(Man überlege sich einfach, dass $F(f^+, g^-) \circ F(f^-, g^+) = id$, also ein Retraktionspaar ist.)

Leider ist $\mathbf{C}^{Ret}$ nicht ausreichend für unsere Zwecke. Zum Beispiel ist bis jetzt keine nichttriviale CCC $\mathbf{C}$ bekannt, so dass $\mathbf{C}^{Ret}$ ω-vollständig ist.

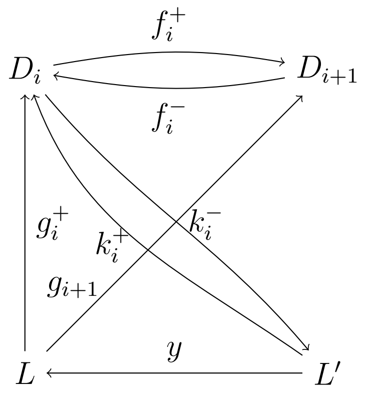
Um zu verstehen, wo die Schwierigkeiten dafür liegen, dass bisher weder eine solche Kategorie angegeben konnte, noch bewiesen werden konnte, dass keine solche Kategorie existiert, betrachten wir folgenden Zusammenhang zwischen Kegeln und Limiten.

Wir nehmen an, wir hätten eine Kategorie $\mathbf{C}$, die Limiten für jedes Diagramm besitzt, sei $\{(D_i, f_i)\}_{i\in\mathbb{N}}$ eine ω-Kette in $\mathbf{C}^{Ret}$. Dann ist $\{D_i, f_i^-\}_{i\in\mathbb{N}}$ eine ω-Kette in $\mathbf{C}$, die einen Limes $(L, \{g_i\})$ besitzt. Das Objekt L scheint ein guter Kandidat für einen Limes für $\{(D_i, f_i)\}_{i\in\mathbb{N}}$ in $\mathbf{C}^{Ret}$ zu sein. Tatsächlich gilt folgendes Theorem:

Satz 32. *Sei* $\{(D_i, f_i)\}_{i\in\mathbb{N}}$ *eine* ω*-Kette in* $\mathbf{C}^{Ret}$. *Wenn* $(L, \{g_i\})$ *ein Limes dafür ist, dann gibt es einen Cokegel* $(L, \{h_i, g_i\}_{i\in\mathbb{N}})$ *für* $\{(D_i, f_i)\}_{i\in\mathbb{N}}$ *in* $\mathbf{C}^{Ret}$, *d.h. jedes* g_i *ist rechter Teil eines Retraktionspaares.*

Beweis: Beweis siehe [1] □

Abbildung 18: L und L' sind Cokegel in $\mathbf{C}^{Ret}$



Allerdings gibt es keinen Weg zu zeigen, dass L universell ist. Nehmen wir an, wir hätten einen zweiten Kegel $(L', \{k_i\})$ für die ω-Kette in $\mathbf{C}^{Ret}$, dann ist $(L, \{k_i^-\})$ ein Kegel in $\mathbf{C}$ für $\{(D_i, f_i^-)\}_{i \in \mathbb{N}}$ und es existiert ein eindeutig bestimmter $\mathbf{C}$-Pfeil $y : L' \to L$ mit $y \circ g_i = k_i^-$ und $g_i \circ y = k_i^+$. Jedoch gibt es im allgemeinen keinen Weg einen Morphismus x von L nach L' zu finden, so dass $y \circ x = id$. Im nächsten Abschnitt sehen wir den Ausweg aus diesem Dilemma.

9.2 0-Kategorien

Wir betrachten die Klasse von Morphismen $\{k_i^+ \circ g_i\}$ von L nach L', wie sie oben definiert wurden. Der Morphismus $k_i^+ \circ g_i : L \to L'$ beschreibt, wie die Approximation von L bis zum i-ten Level innerhalb von L' dargestellt wird. In einem noch zu bestimmenden Sinne fassen wir das intuitiv auf als $k_i^+ \circ g_i \le k_{i+1}^+ \circ g_{i+1}$. Wenn außerdem jede Hom-Menge von $\mathbf{C}$ eine vollständige Halbordnung und die Klasse $\{k_i^+ \circ g_i\}_{i \in \mathbb{N}}$ eine ω-Kette ist, dann können wir x definieren als $x = \cup_{i \in \mathbb{N}} \{k_i^+ \circ g_i\}$ und x würde die am Ende des vorigen Abschnittes definierte Rolle spielen. Wir definieren uns dazu den Begriff einer 0-Kategorie.

Definition 63. *Eine Kategorie* $\mathbf{C}$ *heißt 0-Kategorie, gdw.*

1. *jede Menge $Hom_{\mathbf{C}}(A, B)$ eine vollständige Halbordnung mit kleinstem Element $0_{A,B}$ ist,*

2. *die Komposition von Pfeilen stetig bezüglich der Halbordnung ist,*

3. *für jedes $f \in Hom_{\mathbf{C}}(A, B)$ gilt: $0_{B,C} \circ f = 0_{A,C}$.*

Wir hatten oben gefordert, dass $\{k_i^+ \circ g_i\}$ eine ω-Kette sein soll. Dazu überlegen wir uns zuerst, dass $k_i^+ \circ g_i = k_{i+1}^+ \circ f_i^+ \circ f_i^- \circ g_{i+1}$. Falls außerdem $f_i^+ \circ f_i^- \le id$, dann erhalten wir $k_i^+ \circ g_i \le k_{i+1}^+ \circ g_{i+1}$ und das liefert die Verfeinerung, die wir für $\mathbf{C}^{Ret}$ brauchen.

Definition 64. *Sei* $\mathbf{C}$ *eine 0-Kategorie und seien $i : D \to E$ und $j : E \to D$ zwei* $\mathbf{C}$ *Pfeile. Das Paar (i, j) heißt Projektionspaar(von D nach E), gdw. $j \circ i = id_D$ und $i \circ j \le id_E$. Der Pfeil I heißt dann Einbettung und j heißt Projektion.*

Definition 65. *Sei* $\mathbf{C}$ *eine* 0*-Kategorie, dann ist die Kategorie* C^{Prj} *die Kategorie, deren Objekte die Objekte von* $\mathbf{C}$ *sind und deren Pfeile Projektionspaare sind.*

Satz 33. *Sei* $\mathbf{C}$ *eine* 0*-Kategorie mit allen* ω*-Colimiten. Sei* $\{D_i, (f_i^+, f_i^-)\}_{i\in\mathbb{N}}$ *eine* ω*-Kette in* $\mathbf{C}^{Prj}$ *(und daher auch in* $\mathbf{C}^{Ret}$*) und* $(L, \{(h_i, g_i\}_{i\in\mathbb{N}})$ *sei ein Kegel in* $\mathbf{C}^{Ret}$ *wie in Theorem 32 definiert. Dann ist* $(L, \{(h_i, g_i\}_{i\in\mathbb{N}})$ *auch ein Kegel in* $\mathbf{C}^{Prj}$*. Außerdem ist er in dieser Kategorie universell, also ein Limes.*

Eine nützliche Formulierung für ω-Kolimits in der Kategorie $\mathbf{C}^{Prj}$ ist die folgende:

Satz 34. *Der Cokegel* $(L, \{(h_i, g_i\}_{i\in\mathbb{N}})$ *für die* ω*-Kette* $(\{D_i, (f_i^+, f_i^-)\}_{i\in\mathbb{N}})$ *in* $\mathbf{C}^{Prj}$ *ist universell, falls* $\cup_{i\in\mathbb{N}}\{h_i \circ g_i\} = id$

Bisher haben wir gezeigt, dass wenn $\mathbf{C}$ eine 0-Kategorie mit allen ω-Colimiten ist, dann hat auch $\mathbf{C}^{Prj}$ Colimiten für jede ω-Kette.

Der nächste Schritt ist zu verstehen, was wir in bezug auf die Möglichkeiten der Anwendung der Konstruktion aus Definition 62, als wir kontravariante Funktoren in kovariante transformiert haben, verloren haben. Tatsächlich gibt es keinen Grund anzunehmen, dass der Funktor F^{+-} Projektionspaare in Projektionspaare überführt. Wir erinnern uns, dass ein zweiwertiger Endofunktor, der kontravariant in der ersten und kovariant in der zweiten Komponente ist, den Typ $F : \mathbf{C}^{op} \times \mathbf{C} \to \mathbf{C}$ hat.

Definition 66. *Sei* $\mathbf{C}$ *eine* 0*-Kategorie. Ein Funktor* $F : \mathbf{C}^{op} \times \mathbf{C} \to \mathbf{C}$ *heißt lokal monoton, wenn er monoton auf den Hom-Mengen ist, d.h. für* $f, f' \in Hom_{\mathbf{C}^{op}}(A, B)$ *und* $g, g' \in Hom_{\mathbf{C}}(C, D)$ *gilt*

$$\textit{Falls } f \le f', \quad g \le g' \textit{ dann } F(f, g) \le F(f', g')$$

Satz 35. *Falls* $F : \mathbf{C}^{op} \times \mathbf{C} \to \mathbf{C}$ *lokal monoton ist und* $(f^+, f^-), (g^+, g^-)$ *sind Projektionspaare, dann ist auch* $F^{+-}((f^+, f^-), (g^+, g^-))$ *ein Projektionspaar.*

Der letzte Schritt um unsere Gleichung zu lösen, ist eine einfache Bedingung an den Funktor F in $\mathbf{C}$ zu finden, so dass F^{+-} in $\mathbf{C}^{Prj}$ ω-stetig ist.

Definition 67. *Sei* $\mathbf{C}$ *eine* 0*-Kategorie. Ein Funktor* $F : \mathbf{C}^{op} \times \mathbf{C} \to \mathbf{C}$ *ist lokal stetig, wenn er* ω*-stetig auf den Hom-Mengen ist, d.h. für jede gerichtete Menge* $\{f_i\}_{i\in\mathbb{N}}$ *in* $Hom_{\mathbf{C}^{op}}(A, B)$ *und jede gerichtete Menge* $\{g_i\}_{i\in\mathbb{N}}$ *in* $Hom_{\mathbf{C}}(C, D)$ *gilt*

$$F(\cup_{i\in\mathbb{N}}\{f_i\}, \cup_{i\in\mathbb{N}}\{g_i\}) = \cup_{i\in\mathbb{N}}\{F(f_i, g_i)\}.$$

Wenn ein Funktor lokal stetig ist, ist er offensichtlich auch lokal monoton. Beide Eigenschaften bleiben unter Komposition erhalten(Übungsaufgabe).

Mit dem folgenden Satz fassen wir unsere Bemühungen um die Lösbarkeit von rekursiven Domaingleichungen zusammen. Wir geben Bedingungen an die Kategorie der Typen und an die Funktoren an, so dass nach der Transformation in rein kovariante Funktoren, der Limes einer ω-Kette, also der Fixpunkt berechnet werden kann.

Satz 36. *Sei* $\mathbf{C}$ *eine* ω*-covollständige* 0*-Kategorie. Sei* $F : \mathbf{C}^{op} \times \mathbf{C} \to \mathbf{C}$ *ein lokal stetiger Funktor. Dann ist der Funktor* $F^{+-} : \mathbf{C}^{Prj} \times \mathbf{C}^{Prj} \to \mathbf{C}^{Prj}$ ω*-stetig.*

Damit haben wir eine Klasse von Funktoren charakterisiert, für die eine Lösung von Gleichung 12 möglich ist.

Beispiele 27. *Um eine Lösung für die Gleichung 12 in einer auf $CPO-s$ basierenden Kategorie zu finden, (wie CPO,CPOS,Scott-Domains usw.)muss man i.A. die Interpretation einer der beiden Symbole + oder → abschwächen. Tatsächlich haben aufgrund ihrer Natur alle diese Kategorien für alle Objekte und wir wissen, dass das inkonsistent mit der Forderung nach der Existenz von Coprodukten und gleichzeitig kartesischer Abgeschlossenheit ist. Ein typischer Weg um das Problem zu vermeiden, ist sie bei der Interpretation von + sich auf ein schwaches Coprodukt zu einzuschränken. Ein schwaches Coprodukt wird definiert wie in Definition 19, nur das man nicht verlangt, dass der Pfeil $[f,g]$ eindeutig bestimmt ist.*

Die Kategorie **CPO** *der vollständigen Halbordnungen mit kleinstem Element und stetigen Funktionen als Pfeilen ist eine* 0*-Kategorie bezüglich der punktweisen Ordnung auf Morphismen.* **CPO** *ist ein CCC mit schwachen Coprodukten $A+B$ gegeben durch die Verschmelzungssumme. Sie hat Differenzcokerne und daher auch Colimiten für jedes Diagramm. Die Funktoren $A+-$ und → werden definiert durch:*

$$A+-(B) = A+B, A+-(f) = id_A + f \tag{15}$$
$$\to (A,B) = B^A, \to (f,g) = \lambda h.g \circ h \circ f \tag{16}$$

und sind beide lokal stetig. Der Diagonalfunktor $\Delta : \mathbf{CPO} \times \mathbf{CPO} \to \mathbf{CPO}$ ist ebenfalls lokal stetig. Wir können daher Satz 36 anwenden und erhalten die zugehörigen Funktoren:

$$(A+-)^{+-} : \mathbf{CPO^{Prj}} \to \mathbf{CPO^{Prj}} \tag{17}$$
$$(A+-)^{+-}(f^+,f^-) = (id_A + f^+, id_A + f^-) \tag{18}$$
$$(\to)^{+-} : \mathbf{CPO^{Prj}} \times \mathbf{CPO^{Prj}} \to \mathbf{CPO^{Prj}} \tag{19}$$
$$\to^{+-} ((f^+,f^-),g^+,g^-)) = (\lambda h.g^+ \circ h \circ f^-, \lambda h.g^- \circ h \circ f^+) \tag{20}$$
$$(\triangle)^{+-} : \mathbf{CPO^{Prj}} \to \mathbf{CPO^{Prj}} \times \mathbf{CPO^{Prj}} \tag{21}$$
$$\triangle^{+-}(f^+,f^-) = ((f^+,f^-),(f^+,f^-)) \tag{22}$$
$$\tag{23}$$

Alle Funktoren sind ω-stetig, also auch ist es auch ihre Zusammensetzung,also der Funktor $F = (A+-)^{+-} \circ (\to)^{+-} \circ \triangle^{+-} : \mathbf{CPO^{Prj}} \to \mathbf{CPO^{Prj}}$. Explizit ist dieser Funktor definiert durch $F(X) = A+X^X$ und $F(f^+,f^-) = (id_A+\lambda h.f^+\circ h\circ f^-, id_A+\lambda h.f^-\circ h\circ f^+)$. Daher existiert für jedes A ein X mit $X \cong A+X^X$.

Literatur

[1] Andrea Asperti and Longo Guiseppe. *Categories, Types and Structures.* The MIT Press, Cambridge, Massachusetts and London, England.

[2] Michael Barr and Charles Wells. *Category Theory for Computing Science.* Prentice-Hall International Series in Computer Science. Prentice-Hall International, New York, 1990.

[3] Michael Barr and Charles Wells. Category theory - lecture notes for esslli. 1999.

[4] Robert Goldblatt. *Topoi–The Categorical Analysis of Logic*, volume 98 of *Studies in Logic and the Foundations of Mathematics.* Elsevier Science Publishers, 1984.

[5] John W. Gray and Andre Scedrov. *Categories in Computer Science and Logic*, volume 92 of *Contemporary Mathematics.* American Mathematical Society, 1989.

[6] Joachim Lambek and Philip.J. Scott. *Introduction to Higher Order Categorical Logic*, volume 7 of *Cambridge Studies in Advanced Mathematics.* Cambridge University Press, 1986.

[7] Saunders Mac Lane. *Categories for the Working Mathematician*, volume 5 of *Graduate Texts in Mathematics.* Springer-Verlag, 1971.