Universelle Coalgebra

H. Peter Gumm

Philipps-Universität Marburg
Fachbereich Mathematik und Informatik
35032 Marburg

gumm@mathematik.uni-marburg.de

Erscheint in : Th. Ihringer: *Universelle Algebra*
Heldermann Verlag, Berlin.

Inhaltsverzeichnis

1 Zustandsbasierte Systeme

Universelle Coalgebra ist die Theorie allgemeiner zustandsbasierter Systeme. Diese zeichnen sich dadurch aus, dass ihr Verhalten von einem internen Zustand abhängig ist, welcher nicht unmittelbar beobachtet werden kann. Insbesondere darf die Ausgabe (**output**), welche ein solches System produziert, nicht nur von der Eingabe (**input**) sondern auch von einem inneren **Zustand** abhängen. Ebenso kann jede Eingabe als **Seiteneffekt** den Zustand ändern. Insbesondere können daher zwei aufeinander folgende identische Eingaben verschiedene Ausgaben produzieren.

Beispiel 1.0.1 DIGITALUHR. *Eine Digitaluhr habe drei Knöpfe* SET, MODE, STP. *Die Anzeige ist nicht nur von dem zuletzt gedrückten Knopf abhängig, sondern auch von früheren Eingaben.*

Bei der **Spezifikation** eines zustandsbasierten Systems ist man nur an seinem Ein-Ausgabe-Verhalten interessiert. Die Zustände des Systems sind Teil der Implementierung. Den Benutzer interessiert nur das beobachtbare Verhalten.

Ein Konstrukteur, wird daher zunächst versuchen, ein funktionsfähiges System zu implementieren, das sich korrekt verhält. In einem zweiten Schritt kann das System optimiert werden - etwa durch Verwendung einer minimalen Anzahl von Zuständen. Bei der Minimierung werden überflüssige Zustände entfernt. Zu diesem Zweck bezeichnen wir zwei Zustände als **ununterscheidbar**, wenn sie sich nicht durch das extern beobachtbare Ein-Ausgabe-Verhalten des Systems unterscheiden lassen.

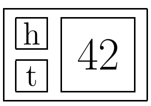
Der mathematische Begriff, den wir für diesen Zweck einführen werden, heißt **Bisimilarität**. Intuitiv sind zwei Zustände s und s' **bisimilar**, wenn sie allein anhand des Ein-Ausgabe-Verhaltens **nicht unterscheidbar** sind. Wir schreiben dafür $s \sim s'$. Die Relation $\sim$ wird sich als reflexiv und symmetrisch herausstellen, zudem ist sie mit dem System-Verhalten in einem noch zu definierenden Sinne "verträglich".

Im Falle, dass $\sim$ sogar transitiv ist, mithin eine verträgliche Äquivalenz-Relation, kann man von **Beobachtungs-Äquivalenz** sprechen. Wir werden allerdings auch Systeme kennen lernen, bei denen $\sim$ nicht transitiv ist. Für solche Fälle erscheint uns der Begriff der **Ununterscheidbarkeit** angemessener.

Bevor wir eine allgemeine mathematische Definition von Coalgebren geben, wollen wir einige prototypische Systeme betrachten und die zugehörigen Verträglichkeits- bzw. Ununterscheidbarkeits-Begriffe entwickeln und motivieren.

1.1 Black Boxes

Wir stellen uns ein Gerät vor, das zwei mit h und t beschriftete Knöpfe besitzt. Drückt man die Taste h, so wird in einem Display eine Zahl oder allgemeiner ein Element d einer Datenmenge D sichtbar. Mit der Taste t kann man den inneren Zustand verändern, so dass ein erneutes Drücken von h ein anderes Datum $d' \in D$ in der Anzeige ergeben kann.



Wir können das System durch ein Paar von Abbildungen beschreiben, wobei S die Menge aller inneren Zustände (engl.: *states*) des Systems darstellt:

$$\begin{aligned} h &: S \to D \\ t &: S \to S. \end{aligned}$$

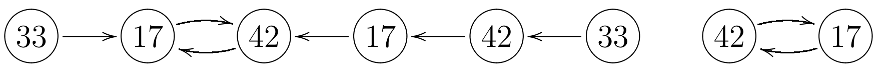
Zwei Zustände mit unterschiedlichen Ausgaben sind offensichtlich unterscheidbar. Allgemeiner sind sie aber auch unterscheidbar, wenn eine Serie identischer Eingaben (Tasten-Kombinationen) in verschiedenen Ausgaben resultieren. $s \sim s'$ bedingt also sowohl $h(s) = h(s')$, als auch $t(s) \sim t(s')$. Diese Bedingung an eine Ununterscheidbarkeits-Relation formulieren als Schluss-Regel in der folgenden Gestalt:

$$\frac{s \sim s'}{h(s) = h(s'),\ t(s) \sim t(s')}$$

Diese Schreibweise bedeutet, dass aus den über dem Bruchstrich geschriebenen Prämissen die darunter aufgeführte Conclusion gefolgert werden darf.

Wir behaupten nicht, dass durch die obige Schlussregel eine Relation $\sim$ eindeutig definiert wäre. Wir wollen daher jede Relation $\sim$, welche die Schlussregel erfüllt, eine **Bisimulation** nennen.

Beispiel 1.1.1 *Wir betrachten eine Black Box mit einer acht-elementigen Zustandsmenge. Die Transitionsfunktion* t *wird in der folgenden Abbildung durch Pfeile dargestellt und die Ausgabe durch die Beschriftung der Zustände.*



Zwei mit verschiedenen Werten beschriftete Zustände sind sofort unterscheidbar. Aber auch die beiden mit 33 *beschrifteten Zustände sind unterscheidbar: Drücken wir nämlich* t *und danach* h*, so erhalten wir in einem Falle die Ausgabe* 17*, im anderen Falle* 42*. Im Gegensatz dazu sind alle Zustände mit Ausgabe* 42 *ununterscheidbar, ebenso alle Zustände mit Ausgabe* 17*.*

1.2 Funktionales Programmieren mit Strömen

Ströme sind unendliche Listen, wie sie in vielen Betriebssystemen zur Verfügung stehen. Das erste Element eines Stromes s heißt der *Kopf* (engl.: **head**) von s. Entfernt man es, so bleibt immer noch ein Strom übrig, den man als *Rest* (engl.: **tail**) von s bezeichnet. Mathematisch kann man einen unendlichen Strom τ von Daten aus D als Abbildung $\tau : \omega \to D$ definieren, wobei ω die geordnete Menge der natürlichen Zahlen bezeichnet und $\tau(k)$ für jedes $k \in \omega$ das k-te Element des Stromes τ.

Beispiel 1.2.1 *In Unix bezeichnet* `yes` *den Strom, der aus einer unendlichen Folge des ASCII-Zeichens* `y` *besteht, also*

$$\texttt{yes} = [\ \texttt{y}, \texttt{y}, \texttt{y},\ \ldots\ ].$$

Mit dem Befehl `remove * | yes`*, zum Beispiel, kann man sämtliche Dateien löschen. Die bei jeder einzelnen Datei* f *zu erwartenden Frage:*

`Do you really want to delete` f `[y/n]?`

wird automatisch - und so oft nötig - mit `y` *beantwortet.*

Einen Strom von Elementen einer Datenmenge D, zusammen mit den Funktionen hd und tl, kann man als spezielle Black Box auffassen, deren Zustandsmenge D^ω aus allen Abbildungen von ω nach D besteht:

$$\begin{aligned} hd &: D^\omega \to D \\ tl &: D^\omega \to D^\omega \end{aligned}$$

Für ein beliebiges $\tau \in D^\omega$, d.h. $\tau : \omega \to D$ ist dabei $h(\tau) := hd(\tau) := \tau(0)$ festgelegt und $t(\tau) := tl(\tau)$ mit $tl(\tau)(k) := \tau(k+1)$. Schreibt man τ als unendliche Liste $[\tau(0), \tau(1), \dots]$, so repräsentiert $hd(\tau) = \tau(0)$ gerade den "head" und $tl(\tau) = [\tau(1), \dots]$ den "tail" der Liste.

Ströme haben eine besondere Eigenschaft, denn je zwei Zustände sind unterscheidbar. Für jede Bisimulation $\sim$ erfüllen sie die folgende Eigenschaft, die wir später unter dem Namen **Coinduktion** als Beweisprinzip einsetzen werden:

$$\frac{s \sim s'}{s = s'}.$$

Moderne **funktionale Programmiersprachen**, wie etwa "Haskell" oder "ML", können mit Strömen als Datentypen umgehen. Unendliche Ströme dürfen als Argumente oder Ergebnisse von Funktionen verwendet werden und erlauben einen sehr eleganten Programmierstil. In dem folgenden Beispiel zeigen wir eine Interaktion mit einer solchen funktionalen Sprache. Nach einer Eingabe-Aufforderung ">" gibt der Programmierer eine (hier kursiv hervorgehobene) Definition ein. In unserem Beispiel handelt es sich um Ströme von ganzen Zahlen oder um Operationen auf solchen Strömen. Ist das Ergebnis ein Strom, so werden die ersten Elemente (die genaue Anzahl ist einstellbar) ausgedruckt.

`>` $ones = [1 : ones]$
    `[ 1, 1, 1, ... ]`
`>` $from\ n = [n : from\ (n+1)]$
`>` $nats = from\ (0)$
    `[ 0, 1, 2, ... ]`
`>` $add\ ([n_1 : l_1], [n_2 : l_2])$ `=` $[(n_1 + n_2) : add\ (l_1, l_2)]$
`>` $add\ nats\ ones == from\ 1$
    `[ true, true, true, ... ]`

Mit `[h : t]` bezeichnet man in vielen funktionalen Sprachen die (endliche oder unendliche) Liste mit erstem Element `h` und Restliste (tail) `t`. In dem gezeigten Beispiel wurde zunächst ein Strom `ones` definiert, der aus einer unendlichen Folge von `1`-en besteht. Das System zeigt jeweils den Anfang des Stromes auf dem Bildschirm an, die Ellipsen "..." stehen für die restlichen Elemente. Nach der Definition der einstelligen Funktion `from` folgt ihr Aufruf mit dem Argument 0. Das Ergebnis wird mit `nats` bezeichnet. Es folgt die Definition einer Additions-Operation `add` für Ströme. Am Ende prüfen wir nach, ob die Gleichung

```
add nats ones == from 1
```

gilt. (Haskell verwendet ein einfaches Gleichheitszeichen für eine Definition und ein doppeltes Gleichheitszeichen für die Gleichheitsrelation.)

Natürlich kann die Antwort nicht überzeugen, da die Gleichheit der Ströme nur auf den ersten Gliedern überprüft wurde – immerhin ist das Ergebnis dort immer wahr (`true`). Wir werden uns später mit der Frage beschäftigen, wie man eine solche Behauptung beweisen kann und wieso Definitionen der obigen Art stets eine – und nur eine – Lösung zulassen.

1.3 Datentypen

Datentypen, wie z.B. der unvermeidliche *Stack* wurden bisher als freie (0-erzeugte) Algebra modelliert. Zu einer Menge D von Daten betrachtete man die freie Algebra zum mehr-sortigen Typ

$$\begin{array}{lll} \mathit{emptyStack} & : & \to \mathit{Stack} \\ \mathit{push} & : D \times \mathit{Stack} & \to \mathit{Stack}. \end{array}$$

Jeder Stack ist dann entweder der leere Stack (*emptyStack*), oder er ist von der Form $\mathit{push}(d_1(\mathit{push}(d_2(\ldots(\mathit{push}(d_n, \mathit{emptyStack})\ldots)$. Die Operationen *top* und *pop* tauchen in dieser Modellierung nicht direkt auf sie entstehen als Umkehrfunktionen von *push* und *pop* auf die folgende Weise: Unter Zuhilfenahme einer ein-elementigen Menge $1 := \{*\}$ und der Notation "+" für die disjunkte Vereinigung zweier Mengen lassen sich *emptyStack* und *push* zu einer Abbildung

$$k : 1 + D \times \mathit{Stack} \to \mathit{Stack}$$

mit $k(*) := \mathit{emptyStack}$ und $k(d, s) := \mathit{push}(d, s)$ kombinieren. Diese Abbildung ist bijektiv. Ihre Umkehrabbildung

$$l : \mathit{Stack} \to 1 + D \times \mathit{Stack}$$

wird durch

$$l(s) = \begin{cases} * & \text{falls } s = \mathit{emptyStack} \\ (d, s') & \text{falls } s = \mathit{push}(d, s'). \end{cases}$$

gegeben, die offensichtlich die partiell definierten Funktionen *top* und *pop* zu einer total definierten Funktion zusammenfasst.

Berücksichtigt man, dass auch die Operationen h und t einer Black Box zu einer einzigen Abbildung

$$h \times t : S \to D \times S$$

zusammengefasst werden können, so lässt sich ein Stack mit der obigen Funktion

$$l : \mathit{Stack} \to 1 + D \times \mathit{Stack}$$

als Black Box deuten, die "kaputtgehen" kann, was durch $l(s) = *$ beschrieben wird. In allen anderen Fällen liefert l ein Paar (d, s') aus einem Ergebnis d und einem neuen Zustand s' zurück. Zwei Zustände sind ununterscheidbar, $s \sim s'$, wenn entweder $l(s) = * = l(s')$ gilt, oder $\mathit{top}(s) = \mathit{top}(s')$ und $\mathit{pop}(s) \sim \mathit{pop}(s')$. Als Schlussregel formuliert hat man also:

$$\frac{s \sim s'}{(\mathit{empty}(s) \wedge \mathit{empty}(s')) \vee (\mathit{top}(s) = \mathit{top}(s') \wedge \mathit{pop}(s) \sim \mathit{pop}(s'))}.$$

Interpretiert man die Elemente von *Stack* als Zustände, so erhält man, ausgehend von einem beliebigen Zustand s_0 durch fortgesetztes Anwenden von l die Zustände s_0, s_1, ..., s_i, ..., wobei immer $s_{i+1} = pop(s_i)$ gilt, und man beobachtet parallel dazu die Folge von Ausgaben $top(s_0), \ldots, top(s_i), \ldots$. Die Folgen brechen ab, sofern irgendwann einmal $l(s_i) = *$ eintritt, falls also der Stack leer ist. In jedem Fall ist eine Beobachtung also durch eine endliche oder unendliche Folge von Elementen aus D gegeben.

Umgekehrt kann man aus der Menge $D^\infty = D^* + D^\omega$ aller endlichen und unendlichen Folgen von Elementen aus D ein System des obigen Typs bauen, indem man setzt:

$$l(\sigma) = \begin{cases} * & \text{falls } \ \sigma = \varepsilon, \\ (hd(\sigma), tl(\sigma)) & \text{sonst.} \end{cases}$$

Man überprüft leicht, dass für dieses System die oben als Co-Induktion bezeichnete Schlussregel gilt:

$$\frac{s \sim s'}{s = s'}.$$

1.4 Automaten

Sei Σ eine Menge (von Zeichen) und D eine beliebige Menge (von Daten). Ein **Automat** über Σ mit Output D besteht aus einer Menge S von *Zuständen*, einer *Transitionsfunktion* δ und einer *Ausgabefunktion* γ mit

$$\begin{aligned} \delta &: S \times \Sigma \to S \\ \gamma &: S \to D. \end{aligned}$$

Wenn der Automat im Zustand $s \in S$ den Input $e \in \Sigma$ erhält, geht er in den neuen Zustand $s' = \delta(s, e)$ über. Außerdem ist jedem Zustand $s \in S$ ein Ausgabewert $\gamma(s) \in D$ zugeordnet.

Man kann das Verhalten des Automaten auch über mehrere Eingaben hinweg verfolgen. Ist der Automat im Zustand s_0 und gibt man nacheinander die Zeichen $e_1, \ldots, e_n$ ein, dann durchläuft der Automat die Zustände $s_0,\ s_1 = \delta(s_0, e_1),\ \ldots\ , s_n = \delta(s_{n-1}, e_n)$. Dabei werden die Ausgaben $d_1 = \gamma(s_1), \ldots, d_n = \gamma(s_n)$ erzeugt. Zwei Zustände s und s' sind also ununterscheidbar, falls sie die gleiche Ausgabe bewirken, und falls identische Eingaben wieder zu ununterscheidbaren Zuständen führen. Daher definieren wir eine Bisimulation $\sim$ eines Automaten durch die folgende Schlussregel:

$$\frac{s \ \sim \ s'}{\gamma(s) = \gamma(s'),\ \forall e \in \Sigma.\ \delta(s, e) \ \sim \ \delta(s', e)}.$$

Worte

Die Inputmenge Σ eines Automaten nennt man auch ein **Alphabet**. Ein Alphabet stellt man sich als (meist endliche) Menge von möglichen Eingabezeichen - etwa Tasten einer Tastatur - vor. Eine endliche Folge von Zeichen aus Σ bezeichnen wir als ein **Wort**. Dabei ist es sinnvoll auch das **leere Wort** ε zuzulassen. Jedes nichtleere Wort w ist dann von der Form $w = e \cdot v$, wobei e der erste Buchstabe von w ist, und v das Restwort, das übrig bleibt, wenn man den ersten Buchstaben entfernt. Das Konkatenations-Symbol "$\cdot$", mit dem man das Anfügen eines Zeichens vor ein Wort kennzeichnet, wird oft weggelassen. Induktiv definiert man die Menge Σ^* aller *Worte über* Σ:

(i) $\varepsilon \in \Sigma^*$,

(ii) ist $e \in \Sigma$ und $w \in \Sigma^*$, so ist $e \cdot w \in \Sigma^*$.

Die Transitionsfunktion δ eines Automaten lässt sich zu einer Funktion $\delta^* : S \times \Sigma^* \to S$ fortsetzen:

$$\delta^*(s, \varepsilon) = s,$$

$$\delta^*(s, e \cdot v) = \delta^*(\delta(s, e), v).$$

Auf einem beliebigen Σ-Automaten mit Output D definiert man die so genannte **Nerode-Kongruenz** durch:

$$s \sim_N s' :\iff \forall w \in \Sigma^*.\gamma(\delta^*(s, w)) = \gamma(\delta^*(s', w)).$$

Hilfssatz 1.4.1 *Die Nerode-Kongruenz ist die größte Bisimulation und gleichzeitig die größte Kongruenz* θ *der mehrsortigen Algebra* $\mathbf{A} = (A, \Sigma, D; \delta, \gamma)$ *mit* $\theta \leq Kern\ \gamma$.

Beweis Aus $s \sim_N s'$ folgt erstens $\gamma(s) = \gamma(\delta^*(s, \varepsilon)) = \gamma(\delta^*(s', \varepsilon)) = \gamma(s')$, und zweitens für jedes $e \in \Sigma$ und beliebiges $w \in \Sigma^*$

$$\begin{aligned} \gamma(\delta^*(\delta(s, e), w)) &= \gamma(\delta^*(s, e \cdot w)) \\ &= \gamma(\delta^*(s', e \cdot w)) \\ &= \gamma(\delta^*(\delta(s', e), w)), \end{aligned}$$

folglich also $\delta(s, e) \sim_N \delta(s', e)$. Damit ist $\sim_N$ als Bisimulation nachgewiesen.

Sei $\sim$ eine beliebige Bisimulation, wir müssen $\sim\ \subseteq\ \sim_N$ zeigen. Durch Induktion über den Aufbau des Wortes w zeigen wir für alle $w \in \Sigma^*$:

$$\forall s, s' \in A.\ s \sim s' \implies \gamma(\delta^*(s, w)) = \gamma(\delta^*(s', w)).$$

Im Falle des leeren Wortes $w = \varepsilon$ folgt die Behauptung direkt aus der Eigenschaft der Bisimulation $\sim$. Falls $w = e \cdot v$ sei die Behauptung für das Wort v bereits bewiesen. Seien $s \sim s'$ beliebig, dann gilt insbesondere auch für das Paar von Zuständen $\delta(s, e)$ und $\delta(s', e)$:

$$\gamma(\delta^*(\delta(s, e), v)) = \gamma(\delta^*(\delta(s', e), v)),$$

also

$$\gamma(\delta^*(s, e \cdot v)) = \gamma(\delta^*(s', e \cdot v)).$$

Jede Bisimulation $\sim$ ist mit den Operationen δ und γ verträglich und außerdem gilt $\sim\ \subseteq Kern\ \gamma$. Als Schnitt von Abbildungskernen ist $\sim_N$ sogar eine Äquivalenzrelation, mithin eine Kongruenz.

Sei $\theta \subseteq Kern\ \gamma$ eine Kongruenzrelation, wir müssen $\theta \subseteq\ \sim_N$ zeigen. Für jedes Paar $(s, s') \in \theta$ gilt $\gamma(s) = \gamma(s')$, also $\delta^*(s, \varepsilon) = \delta^*(s', \varepsilon)$. Sei nun $w = e \cdot u$, wobei wir annehmen können, dass für beliebige $(s, s') \in \theta$ schon gezeigt ist, dass $\delta^*(s, u) = \delta^*(s', u)$. Insbesondere gilt dann auch $(\delta(s, e), \delta(s', e)) \in \theta$ und $\delta^*(\delta(s, e), u) = \delta^*(\delta(s', e), u)$. Es folgt $\delta^*(s, e \cdot u) = \delta^*(\delta(s, e), u) = \delta^*(\delta(s', e), u) = \delta^*(s', e \cdot u)$. $\square$

Akzeptoren

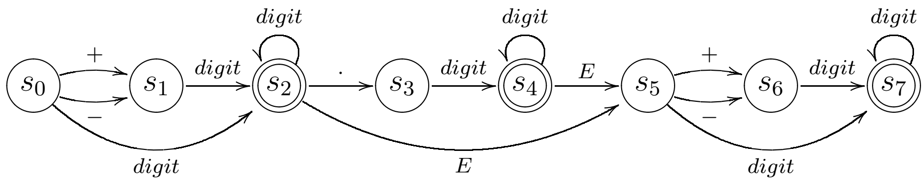
Eine wichtige Rolle spielen Automaten bei der *Erkennung* von Worten (**token**) einer Programmiersprache. Für diese Zwecke benutzt man Automaten mit endlich vielen Zuständen S und einer vorgegebenen Teilmenge $F \subseteq S$ von **akzeptierenden Zuständen akzeptierend**, auch **Endzustände** genannt. Ein Wort $w \in \Sigma^*$ ist genau dann ein Wort der Programmiersprache, wenn es aus einem **Anfangszustand** $s_0 \in S$ in einen akzeptierenden Zustand führt, wenn also $\delta^*(s_0, w) \in F$ gilt.

Solche Automaten heißen *Akzeptoren*. Die Menge der akzeptierenden Zustände kann man mit Hilfe einer Ausgabefunktion $\gamma : S \to \{0, 1\}$ codieren:

$$\gamma(s) = \begin{cases} 1 & s \in F \\ 0 & \text{sonst.} \end{cases}$$

Automaten stellt man gerne durch ihre Graphen dar. Die Knoten entsprechen den Zuständen des Automaten. Man zieht eine mit $e \in \Sigma$ beschriftete Kante von s nach s', wenn $\delta(s, e) = s'$. Akzeptierende Zustände werden durch eine doppelte Umrandung hervorgehoben.

Beispiel 1.4.2 *Der folgende Graph stellt einen Automaten dar, der reelle Zahlkonstanten in der Programmiersprache Pascal akzeptiert.*



"digit" steht in diesem Beispiel für eine beliebige Ziffer. Fehlende Kanten, z.B. die von s_0 ausgehende und mit "." beschriftete Kante, führen alle in einen hier nicht dargestellten Fehlerzustand.

Eine korrekt geschriebene reelle Zahl, muss vom Anfangszustand s_0 in einen der drei Endzustände führen. Für die Eingabe 0.0314159E+02 gilt z.B. $\delta(s_0, 0.0314159E + 02) = s_7 \in F$.

Sprachen

Allgemein bezeichnet man jede Teilmenge $L \subseteq \Sigma^*$ als **Sprache** *über* Σ. Für jedes Zeichen $e \in \Sigma$ definiert man die **Ableitung** *von L nach e* durch

$$L_e := \{w \in \Sigma^* \mid e \cdot w \in L\}.$$

Ordnet man jedem Zustand s eines Akzeptors die Sprache zu, die aus allen Worten besteht, welche von s aus in einen akzeptierenden Zustand führen, also

$$\mathcal{L}(\mathbf{A}, s) := \{w \in \Sigma^* \mid \delta^*(s, w) \in F\},$$

so verifiziert man leicht:

Hilfssatz 1.4.3 $\delta(s, e) = s' :\iff \mathcal{L}(\mathbf{A}, s)_e = \mathcal{L}(\mathbf{A}, s')$.

1.5 Objektorientierte Programme

Im objektorientierten Programmieren versteht man unter einer **Klasse** eine Ansammlung von Datenelementen, sogenannten *Objekten.* Alle Objekte einer Klasse besitzen eine gemeinsame Struktur, welche durch eine Liste von **Attributen** und **Methoden** gegeben ist. Der Benutzer kann auf Objekte nur über die als **public** gekennzeichneten Methoden zugreifen und er kann nur die gleichermaßen gekennzeichneten Attribute verändern. Eine wichtige Eigenschaft von Klassen und Objekten ist, dass sie um Attribute und Methoden *erweitert* werden können.

Wir zeigen hier die Definition einer Klasse, die ein Bankkonto implementieren soll, in der Programmiersprache Java.

```
class Konto{
    private int stand;
    Account(){ stand = 0; }
    public trans(int n){ stand += n; }
    public show(){ return stand; }
    }
```

Wenn ein Konto eröffnet wird, so wird die ganzzahlige Variable `stand` mit `0` initialisiert. Diese als `private` gekennzeichnete Variable ist dem Benutzer der Klasse nicht direkt zugänglich, er kann nur die öffentliche Methode `show` benutzen um sie zu lesen bzw. `trans`, um sie zu verändern.

Der Benutzer muss nicht wissen, wie die Methoden `trans` und `show` implementiert sind. Er will nur sicher sein, dass sie richtig funktionieren. Dazu zählt zum Beispiel die Gewissheit, dass die folgende Gleichung für jedes Konto x und beliebige ganze Zahlen n_1 und n_2 erfüllt ist:

$$x\texttt{.trans(}n_1\texttt{).trans(}n_2\texttt{).show() == }x\texttt{.trans(}n_1+n_1\texttt{).show()}$$

Man beachte, dass Methoden immer von rechts auf ihr Argument angewandt werden. Die Forderung besagt also, dass zwei aufeinander folgende Überweisungen den gleichen Kontostand ergeben wie eine Überweisung der Summe der Beträge. Das Konto `x` kann neben dem Kontostand noch weitere Bestandteile umfassen, auf die der Benutzer keinen Zugriff hat. Aus diesem Grunde wird der Benutzer auch nicht darauf bestehen können, dass die stärkere Forderung

$$x\texttt{.trans(}n_1\texttt{).trans(}n_2\texttt{) == }x\texttt{.trans(}n_1+n_2\texttt{)}$$

für das Konto immer erfüllt ist. So könnte die Bank nachträglich einen Zähler für die Zugriffe auf das Konto einführen. Bei jeder Transaktion würde dieser Zähler erhöht. Nach dieser Änderung wäre in der Tat die letzte Forderung verletzt. Da der Benutzer aber ohnehin nur mittels `show()` sein Konto abfragen kann, würde ihm die erste, schwächere Forderung genügen - und diese bleibt ja erhalten.

1.6 Nichtdeterministische Systeme

Viele informatische Systeme bestehen aus mehreren zusammenwirkenden Komponenten. Große Programme bestehen aus parallel ablaufenden Prozessen, technische Systeme aus vielen zusammenwirkenden Mess-, Auswertungs- und Steuerungskomponenten. Die genaue Abfolge und das zeitliche Ineinandergreifen der Aktionen der einzelnen Komponenten sind nicht vorhersehbar, so dass das komplette System nach außen nicht eindeutig bestimmt (determiniert) ist. Um solche Systeme zu spezifizieren, schränkt man die erlaubten Zustandsübergänge durch eine Übergangsrelation R ein. R muss nicht rechts-eindeutig sein, so dass man hier von einem nicht-deterministischen System spricht. Auf diese Weise gelangt man auch zum Begriff eines nicht-deterministischen Automaten. Die uns hier interessierenden Phänomen können wir bereits anhand nicht-deterministischer Automaten ohne Input studieren, so dass wir uns hier auf solche Modelle konzentrieren. Sie sind in der Logik und in der theoretischen Informatik als *Kripke-Strukturen* bekannt.

Definition 1.6.1 *Sei Φ eine Menge. Eine* **Kripke Struktur** *über Φ besteht aus einer Menge S von Zuständen, einer zweistelligen Relation $R \subseteq S \times S$ und einer Abbildung $v : S \to \mathbb{P}(\Phi)$.*

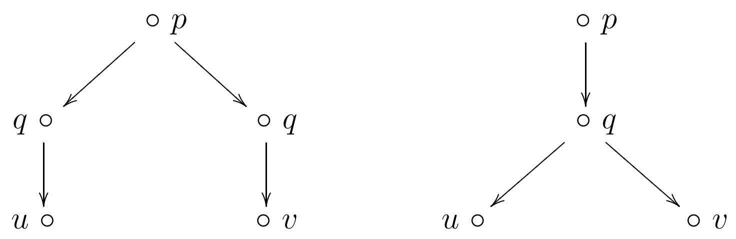
In vielen Anwendungen besteht Φ aus einer Menge von elementaren (atomaren) Aussagen. $v(s)$ ist dann die Menge aller atomaren Aussagen, die im Zustand s wahr sind und sRs' bedeutet, dass das System vom Zustand s in den Zustand s' übergehen kann. Für einen solchen Übergang, auch **Transition** genannt, verwendet man manchmal eine Pfeilnotation: $s \xrightarrow{R} s'$.

Beispiel 1.6.2 *Ein System bestehe aus einer Gruppe von parallelen Prozessen. Jeder Prozess ist ein Programm, in dem einige Variablen deklariert sind. Ein Zustand des Systems ist dann durch eine Belegung der Programmvariablen durch konkrete Werte gegeben.*

Atomare Aussagen werden oft durch Boolesche Ausdrücke beschrieben, in denen diese Variablen vorkommen, etwa "`Hauptstraße = frei`*", "*`Nebenstraße = wartend`*". Zustandsübergänge werden oft durch Ausdrücke beschrieben, welche einen Zustand s zur Zeit t und einen erlaubten Nachfolgezustand s' zur Zeit $t+1$ in Verbindung setzen, etwa:*

```
ampel = rot   ==>   (ampel' = rot OR ampel' = rotgelb).
```

Kripke Strukturen kann man grafisch darstellen, indem man die Zustände durch kleine Kreise und die erlaubten Übergänge durch Pfeile repräsentiert. Die atomaren Aussagen aus $v(s)$ werden neben den Zustand s geschrieben. Das folgende Bild zeigt zwei Kripke Strukturen, auf die wir uns später noch beziehen wollen:



Anhand dieses Beispieles wollen wir untersuchen, wann zwei Zustände unterscheidbar sind. Offensichtlich können wir Zustände unterscheiden, wenn in ihnen unterschiedliche atomare Aussagen gelten. Für eine Ununterscheidbarkeits-Relation (Bisimulation) $\sim$ heißt dies, dass

$$\frac{x \sim y}{v(x) = v(y)}.$$

Schließlich muss es zu jeder Transition aus x eine entsprechende Transition aus y geben, denn sonst wären x und y unterscheidbar. "Entsprechend" heißt hier, dass die erreichten Zustände ununterscheidbar sind. Formal heißt dies:

$$\frac{x \ \sim \ y \ \wedge \ x \to x'}{\exists y'.(y \to y' \wedge x' \sim y')} \quad \text{und} \quad \frac{x \ \sim \ y \ \wedge \ y \to y'}{\exists x'.(x \to x' \wedge x' \sim y')}.$$

Eine Relation $\sim$ zwischen Kripke-Strukturen, die diesen Bedingungen genügt, heißt **Bisimulation**. Die beiden charakterisierenden Bedingungen kann man grafisch darstellen:

$$\begin{tikzcd} x \arrow[r, dash, bend left, "\sim"] \arrow[d] & y \arrow[d, dotted] \\ x' \arrow[r, dash, dotted, bend left, "\sim"] & \exists y' \end{tikzcd} \quad \text{und} \quad \begin{tikzcd} x \arrow[r, dash, bend left, "\sim"] \arrow[d, dotted] & y \arrow[d] \\ \exists x' \arrow[r, dash, dotted, bend left, "\sim"] & y' \end{tikzcd}$$

1.7 Gemeinsamkeit: Der Begriff der Coalgebra

Wir suchen nun Gemeinsamkeiten der bisher betrachteten Systeme. Stets hatten wir eine Menge S von Zuständen und eine oder mehrere Transitionen, die jedem Zustand eine "Kombination von Zuständen und Ausgaben" zuordnet.

Die folgende Aufstellung zeigt, dass alle betrachteten Systeme durch eine einzige Abbildung

$$\alpha : S \to F(S)$$

kodiert werden können, wobei F eine "mengentheoretische Konstruktion" ist, die aus einer Menge S eine neue Menge konstruiert, welche die benötigten Kombinationen von Zuständen und Ausgaben bereitstellt.

Um alle Systeme in diese einheitliche Form zu bringen, nutzen wir aus, dass man ein Paar von Abbildungen $h : S \to \mathbb{N}$ und $t : S \to S$ durch eine einzige Abbildung $\alpha : S \to \mathbb{N} \times S$ ersetzen kann. Analog lässt sich eine Abbildung $\delta : S \times \Sigma \to S$ durch eine Abbildung $\delta' : S \to S^\Sigma$ ersetzen, wenn man $\delta'(s)(e) = \delta(s, e)$ setzt. Ebenso kann man eine zweistellige Relation $R \subseteq A \times B$ auch als Abbildung $R : A \to \mathbb{P}(B)$ verwenden, wenn man $R(a) := \{b \in B \mid a \ R \ b\}$ setzt.

Black Box

$$\left.\begin{array}{l} h : S \to \mathbb{N} \\ t : S \to S \end{array}\right\} \qquad \alpha : S \to \mathbb{N} \times S$$

Bankkonto

$$\left.\begin{array}{l} \texttt{show} \ : S \to \mathbb{Z} \\ \texttt{trans} : S \times \mathbb{Z} \to S \end{array}\right\} \qquad \alpha : S \to \mathbb{Z} \times S^{\mathbb{Z}}$$

Automat

$$\left.\begin{array}{l} \gamma : S \to D \\ \delta : S \times \Sigma \to S \end{array}\right\} \qquad \alpha : S \to D \times S^\Sigma$$

Akzeptor

$$\left.\begin{array}{l} F \subseteq S \\ \delta : S \times \Sigma \to S \end{array}\right\} \qquad \alpha : S \to \{0, 1\} \times S^\Sigma$$

Φ-Kripke Struktur

$$\left.\begin{array}{l} R \subseteq S \times S \\ v : S \to \mathbb{P}(\Phi) \end{array}\right\} \qquad \alpha : S \to \mathbb{P}(\Phi) \times \mathbb{P}(S)$$

Unser abschließendes Beispiel zeigt, dass die Theorie der Coalgebren auch für klassische Gebiete der Mathematik interessant sein kann: Einen *topologischen Raum* (S, τ) repräsentieren wir hier durch die Abbildung $\mathcal{U} : S \to \mathbb{F}(S)$ in die Menge aller sog. *Filter* auf S. Jedem Punkt $s \in S$ wird dabei sein *Umgebungsfilter* $\mathcal{U}(s)$ zugeordnet.

Topologischer Raum

$$\left.\tau \subseteq \mathbb{P}(\mathbb{P}(S))\right\} \qquad \alpha : S \to \mathbb{F}(S)$$

Nachdem wir eine klare strukturelle Gemeinsamkeit der betrachteten Systeme herausgearbeitet haben, können wir uns nun an eine erste Definition für den Begriff der Coalgebra wagen. Für eine feste "mengentheoretische Konstruktion" F definieren wir eine **Coalgebra vom Typ** F als ein Paar, $\mathbf{A} = (A, \alpha_A)$ bestehend aus einer Menge A und einer Abbildung

$$\alpha_A : A \to F(A).$$

Leider ist aber der Begriff der "mengentheoretischen Konstruktion" noch nicht genügend präzise. Wir werden nur solche Konstruktionen zulassen, die sich auf Abbildungen fortsetzen lassen. Dies bedeutet, dass jede Abbildung $f : S \to S'$ "auf natürliche Weise" eine Abbildung $Ff : F(S) \to F(S')$ induziert.

Die Sprache der **Kategorientheorie**, die wir im nächsten Kapitel einführen, stellt uns genau die Begriffe und Definitionen bereit, mit denen wir solche Konstruktionen und ihre Eigenschaften präzise beschreiben können. Anschließend entwickeln wir die Strukturtheorie der Coalgebren zum Typ F. Dabei entdecken wir überraschend viele Gemeinsamkeiten mit der Strukturtheorie Allgemeiner Algebren. Um diese herauszuarbeiten und sichtbar zu machen, ist die Sprache der Kategorientheorie unerlässlich.

1.8 Warum Co-Algebra ?

Zum Abschluss dieses Kapitels wollen wir aber noch die Frage klären, warum wir von "Co"-Algebren sprechen, und was diese mit Algebren gemeinsam haben.

Sei $\mathbf{A}$ eine Algebra vom Typ $F = (f_i)_{i \in I}$, mit Stelligkeiten $\sigma(f_i) = n_i$, dann kann man alle individuellen Operationen $f_i : A^{n_i} \to A$ auch zu einer gemeinsamen Abbildung

$$f : \biguplus_{i \in I} A^{n_i} \to A$$

von der disjunkten Vereinigung der A^{n_i} nach A kombinieren.

Umgekehrt lässt sich jede derartige Abbildung f in eine Familie von n_i-stelligen Operationen aufspalten. Damit ist eine universelle Algebra nichts anderes als ein Paar $\mathbf{A} = (A, f^A)$, bestehend aus einer Menge A und einer Abbildung

$$f^A : F(A) \to A,$$

wenn man $F(X) = \biguplus_{i \in I} X^{n_i}$ als "mengentheoretische Konstruktion" verwendet. Man sieht leicht, wie sich die Konstruktion F auf Abbildungen fortsetzt. Zu einer Abbildung $g : X \to Y$ erhält man eine offensichtliche Abbildung $F(g) : F(X) \to F(Y)$. Ist $h : Y \to Z$ eine weitere Abbildung, so gilt sogar: $F(h \circ g) = F(h) \circ F(g)$, und $F(id_X) = id_{F(X)}$. In der Sprache des folgenden Kapitels bedeutet dies: F ist ein **Funktor**.

1.9 Aufgaben

(i) Seien $\sim$ und $\approx$ zwei Bisimulationen auf Black Boxes. Zeigen Sie, dass das Relationenprodukt $\sim \circ \approx$ und die Vereinigung $\sim \cup \approx$ wieder Bisimulationen sind.

(ii) Zeigen Sie, dass es auf Black Boxes immer eine größte Bisimulation gibt und dass diese eine Kongruenz der mehrsortigen Algebra $\mathbf{B} = (B, D; h, t)$ ist.

(iii) Zeigen Sie, dass Ströme über einer beliebigen Datenmenge D das Coinduktions-Prinzip erfüllen: Für jede Bisimulation $\sim$ zwischen Strömen gilt: $s \sim s' \implies s = s'$.

(iv) Auf Σ^*, der Menge aller Worte über dem Alphabet Σ, bezeichne $\oplus$ die Operation, die zwei Worte konkateniert. Sie ist induktiv über den Aufbau der Worte definiert durch:

$$\begin{aligned} \varepsilon \oplus w &:= w \\ (e \cdot v) \oplus w &:= e \cdot (v \oplus w) \end{aligned}$$

a) Zeigen Sie, dass die resultierende Algebra $\mathbf{\Sigma}^* = (\Sigma^*; \oplus, \varepsilon)$ das freie Monoid über der Erzeugendenmenge Σ ist.

b) Zeigen Sie, dass die Funktion `length`, die einem Wort seine Länge zuordnet, der eindeutige Homomorphismus von $\mathbf{\Sigma}^* = (\Sigma^*; \oplus, \varepsilon)$ nach $\mathbb{N} = (\mathbb{N}; +, 0)$ ist.

2 Grundbegriffe der Kategorientheorie

Eine **Kategorie** besteht aus einer Klasse O von Objekten und einer Klasse M von Morphismen zwischen diesen Objekten. Jeder Morphismus hat genau ein Start- und ein Ziel-Objekt. Ist f ein Morphismus mit Start $\mathbf{A}$ und Ziel $\mathbf{B}$, so schreiben wir $f : \mathbf{A} \to \mathbf{B}$ oder $\mathbf{A} \xrightarrow{f} \mathbf{B}$.

Für unsere Zwecke genügt es, **konkrete Kategorien** zu betrachten, bei denen die Objekte jeweils Mengen mit einer zusätzlichen Struktur sind, und die Morphismen strukturerhaltende Abbildungen zwischen diesen Objekten. Dabei müssen folgende Eigenschaften erfüllt sein:

⋆ die identische Abbildung id_X auf der Grundmenge X eines jeden Objektes ist ein Morphismus, und

⋆⋆ Morphismen sind gegen Hintereinander-Ausführung abgeschlossen. Sind also $\mathbf{A}$, $\mathbf{B}$ und $\mathbf{C}$ Objekte, $f : \mathbf{A} \to \mathbf{B}$ und $g : \mathbf{B} \to \mathbf{C}$ Morphismen, so ist auch $g \circ f : \mathbf{A} \to \mathbf{C}$ ein Morphismus.

Beispiel 2.0.1 *Folgendes sind Beispiele konkreter Kategorien:*

- *die Klasse aller Gruppen mit den Gruppen-Homomorphismen,*
- *die Klasse aller abelschen Gruppen mit den Homomorphismen,*
- *die Klasse aller allgemeinen Algebren eines festen Typs mit ihren Homomorphismen,*

- *die Klasse aller topologischen Räume mit den stetigen Abbildungen,*
- *die Klasse aller Mengen mit ihren Abbildungen,*

Wie sich bald herausstellen wird, bildet auch die Klasse aller Coalgebren eines festen Typs F eine Kategorie $\mathbf{Set}_F$.

Kommutative Diagramme

Mit Diagrammen stellt man Gleichheiten von Morphismen graphisch dar. Die Objekte werden durch Punkte $\bullet$ oder Namen bezeichnet, einen Morphismus von $\mathbf{A}$ nach $\mathbf{B}$ stellt man durch einen Pfeil dar. Ein Weg von einem Objekt $\mathbf{A}$ zu einem Objekt $\mathbf{C}$ entspricht der Komposition der Morphismen entlang des Weges. Ein Diagramm heißt **kommutativ**, wenn für beliebige Objekte $\mathbf{A}$ und $\mathbf{B}$ in dem Diagramm, je zwei Wege von $\mathbf{A}$ nach $\mathbf{B}$ denselben Morphismus ergeben. Beispielsweise sollen die folgenden Diagramme ausdrücken, dass $f = h \circ g$ ist, dass $\psi_1 \circ \varphi = \psi_2 \circ \varphi$ bzw. $\star^H \circ (\varphi \times \varphi) = \varphi \circ \star^G$. Letzteres heißt übrigens, dass $\varphi : \mathbf{G} \to \mathbf{H}$ ein Homomorphismus zwischen den Gruppoiden $(G, \star^G)$ und $(H, \star^H)$ ist.

$$\begin{tikzcd} A \arrow[r, two heads, "g"] \arrow[dr, "f"'] & f(A) \arrow[d, hook, "h"] \\ & B \end{tikzcd} \qquad \begin{tikzcd} \bullet \arrow[r, "\varphi"] & \bullet \arrow[r, shift left, "\psi_1"] \arrow[r, shift right, "\psi_2"'] & \bullet \end{tikzcd} \qquad \begin{tikzcd} G \times G \arrow[r, "(\varphi \times \varphi)"] \arrow[d, "\star^G"] & H \times H \arrow[d, "\star^H"] \\ G \arrow[r, "\varphi"] & H \end{tikzcd}$$

Will man betonen, dass es sich bei einem Morphismus um einen *Epimorphismus*, bzw. einen *Monomorphismus* handelt (siehe die nachfolgende Definition), so benutzt man spezielle Pfeile: $\twoheadrightarrow$ für Epimorphismen, $\rightarrowtail$ oder $\hookrightarrow$ für Monomorphismen.

2.1 Spezielle Morphismen

Definition 2.1.1 *Sei $f : \mathbf{A} \to \mathbf{B}$ ein Morphismus. f heißt*

- **links invertierbar***, falls es einen Morphismus $h : \mathbf{B} \to \mathbf{A}$ gibt mit $h \circ f = id_A$.*
- **rechts invertierbar***, falls es einen Morphismus $g : \mathbf{B} \to \mathbf{A}$ gibt mit $f \circ g = id_B$.*
- **Isomorphismus***, falls es einen Morphismus $g : \mathbf{B} \to \mathbf{A}$ gibt mit $f \circ g = id_B$ und $g \circ f = id_A$.*

Hilfssatz 2.1.2 *Ist $f : \mathbf{A} \to \mathbf{B}$ links- und rechts-invertierbar, so ist f ein Isomorphismus.*

Beweis Es gibt $g, h : \mathbf{B} \to \mathbf{A}$ mit $f \circ g = id_B$ und $h \circ f = id_A$. Es folgt $h = h \circ id_B = h \circ f \circ g = id_A \circ g = g$. $\square$

Insbesondere folgt aus diesem Hilfssatz, dass es für einen Isomorphismus *genau ein* g gibt wie in der Definition gefordert, man schreibt dafür f^{-1}. Gibt es einen Isomorphismus zwischen den Objekten $\mathbf{A}$ und $\mathbf{B}$, so heißen diese **isomorph** und man schreibt: $\mathbf{A} \cong \mathbf{B}$. Morphismen, die nicht notwendigerweise invertierbar, wohl aber **links kürzbar** bzw. **rechts kürzbar** sind, heißen *epi* bzw. *mono*:

Definition 2.1.3 *Ein Morphismus $f : \mathbf{A} \to \mathbf{B}$ heißt*

- **Monomorphismus** *(bzw.* **mono***), falls für alle* $g_1, g_2 : \mathbf{C} \to \mathbf{A}$ *gilt:*

$$f \circ g_1 = f \circ g_2 \implies g_1 = g_2.$$

- **Epimorphismus** *(bzw.* **epi***), falls für alle* $h_1, h_2 : B \to \mathbf{C}$ *gilt:*

$$h_1 \circ f = h_2 \circ f \implies h_1 = h_2.$$

Offensichtlich ist jeder links invertierbare Morphismus mono und jeder rechts invertierbare epi. Die Umkehrung ist i.A. aber nicht richtig., wie das folgende Beispiel zeigt.

Beispiel 2.1.4 *Sei* **Rng** *die Kategorie, mit den Ringen als Objekten und den Ring-Homomorphismen als Morphismen. Die natürliche Einbettung* $\iota : \mathbb{Z} \to \mathbb{Q}$ *ist gleichzeitig mono und epi. Letzteres folgt aus der Tatsache, dass ein beliebiger Ring-Homomorphismus* $\varphi : \mathbb{Q} \to R$ *schon durch die Werte auf den ganzen Zahlen festgelegt ist, denn* $\varphi(\frac{p}{q}) \cdot \varphi(q) = \varphi(\frac{p}{q} \cdot q) = \varphi(p)$.

2.2 Terminale Objekte, Summen, Pushouts

Ein Objekt $\mathbf{T}$ in einer Kategorie heißt **terminal**, falls von jedem Objekt $\mathbf{A}$ genau ein Morphismus $\tau_A : \mathbf{A} \to \mathbf{T}$ existiert.

Terminale Objekte sind, sofern sie existieren, eindeutig bestimmt. Wäre nämlich $\mathbf{Q}$ ebenfalls terminal, so gäbe es jeweils genau einen Morphismus $\tau_Q : \mathbf{Q} \to \mathbf{T}$ und genau einen $\sigma_T : \mathbf{T} \to \mathbf{Q}$. Von $\mathbf{T}$ nach $\mathbf{T}$ hätte man sowohl id_T als auch $\tau_Q \circ \sigma_T$. Es folgt $\tau_Q \circ \sigma_T = id_T$ und analog $\sigma_T \circ \tau_Q = id_Q$. Somit ist τ_Q ein Isomorphismus.

Definition 2.2.1 *Sei* $(\mathbf{A}_i)_{i \in I}$ *eine Familie von Objekten. Ein Objekt* $\mathbf{S}$ *zusammen mit Morphismen* $e_i : \mathbf{A}_i \to \mathbf{S}$ *heißt* **Summe** *der* $\mathbf{A}_i$*, falls für jeden "Konkurrenten", d.h. für jedes andere Objekt* $\mathbf{Q}$ *mit Morphismen* $(q_i : \mathbf{A}_i \to \mathbf{Q})_{i \in I}$ *genau ein Morphismus* $s : \mathbf{S} \to \mathbf{Q}$ *existiert, so dass* $q_i = s \circ e_i$ *für alle* $i \in I$ *gilt.*

$$\begin{tikzcd}
 & \mathbf{S} \arrow[rr, dotted, "\exists!\ s"] & & \mathbf{Q} \\
\mathbf{A}_i \arrow[ur, "e_i"] \arrow[urrr, dashed, "q_i"'] & & \mathbf{A}_j \arrow[ul] \arrow[ur, dashed, "q_j"'] &
\end{tikzcd}$$

Wenn ein solches Objekt $\mathbf{S}$ existiert, dann ist es bis auf Isomorphie eindeutig und man schreibt dafür $\mathbf{S} = \sum_{i \in I} \mathbf{A}_i$. Die Familie der "kanonischen Injektionen" e_i ist in folgendem Sinne **gemeinsam epi**: Für beliebige Morphismen $h_1, h_2 : \mathbf{S} \to \mathbf{C}$ in irgendein Objekt $\mathbf{C}$ gilt

$$(\forall i \in I.\ h_1 \circ e_i = h_2 \circ e_i) \implies h_1 = h_2.$$

Definition 2.2.2 (Co-Equalizer) *Sei* $(f_i : \mathbf{A} \to \mathbf{B})_{i \in I}$ *eine Familie von Morphismen. Ein Morphismus* $g : \mathbf{B} \to \mathbf{C}$ *heißt* **Co-Equalizer** *der* f_i*, falls*

- $g \circ f_i = g \circ f_j$ *für alle* $i, j \in I$ *und*
- *für jedes Objekt* $\mathbf{Q}$ *und jeden Morphismus* $q : \mathbf{B} \to \mathbf{Q}$ *so dass* $q \circ f_i = q \circ f_j$ *für alle* $i, j \in I$ *gilt, gibt es genau ein* $h : \mathbf{C} \to \mathbf{Q}$ *mit* $q = h \circ g$*.*

$$
\begin{tikzcd}
\mathbf{A} \arrow[r, shift left, "f_i"] \arrow[r, shift right, "f_j"'] & \mathbf{B} \arrow[r, "g"] \arrow[dr, dashed, "q"'] & \mathbf{C} \arrow[d, dotted, "h"] \\
 & & \mathbf{Q}
\end{tikzcd}
$$

Definition 2.2.3 (Pushout) *Sei $(f_i : \mathbf{A} \to \mathbf{B}_i)_{i\in I}$ eine Familie von Morphismen. Ein Objekt $\mathbf{P}$ mit einer Familie von Morphismen $(p_i : \mathbf{B}_i \to \mathbf{P})_{i\in I}$ heißt* **Pushout** *der f_i, falls*

(i) $\forall i, j \in I.\ p_i \circ f_i = p_j \circ f_j$, und

(ii) zu jedem "Konkurrenten", d.h. zu jedem anderen Objekt $\mathbf{Q}$, ebenfalls mit einer Familie $(q_i : \mathbf{B}_i \to \mathbf{Q})_{i\in I}$ von Morphismen, so dass $q_i \circ f_i = q_j \circ f_j$ für alle $i, j \in I$ gilt, gibt es genau einen Morphismus $h : \mathbf{P} \to \mathbf{Q}$ mit $h \circ p_i = q_i$ für alle $i \in I$.

$$
\begin{tikzcd}
 & \mathbf{B}_i \arrow[dr, "p_i"] \arrow[drr, dashed, bend left, "q_i"] & & \\
\mathbf{A} \arrow[ur, "f_i"] \arrow[dr, "f_j"'] & & \mathbf{P} \arrow[r, dotted, "h"] & \mathbf{Q} \\
 & \mathbf{B}_j \arrow[ur, "p_j"'] \arrow[urr, dashed, bend right, "q_j"'] & &
\end{tikzcd}
$$

Satz 2.2.4 *Existieren beliebige Summen und Coequalizer, so auch beliebige Pushouts. Für die kanonischen Injektionen $e_i : \mathbf{B}_i \to \sum_{i\in I} \mathbf{B}_i$ sei $g : \sum_{i\in I} \mathbf{B}_i \to \mathbf{P}$ der Coequalizer der Familie $(e_i \circ f_i)_{i\in I}$. Dann ist $(g \circ e_i)_{i\in I}$ der Pushout der $(f_i)_{i\in I}$.*

$$
\begin{tikzcd}
 & \mathbf{B}_i \arrow[d, "e_i"'] \arrow[dr, "p_i"] \arrow[drr, dashed, bend left, "q_i"] & & \\
\mathbf{A} \arrow[ur, "f_i"] \arrow[dr, "f_j"'] & \sum_{i\in I} \mathbf{B}_i \arrow[r, "g"] \arrow[rr, dotted, bend right=15, "s"'] & \mathbf{P} \arrow[r, dotted, "h"] & \mathbf{Q} \\
 & \mathbf{B}_j \arrow[u, "e_j"] \arrow[ur, "p_j"'] \arrow[urr, dashed, bend right, "q_j"'] & &
\end{tikzcd}
$$

Beweis Offensichtlich gilt $(g \circ e_i) \circ f_i = (g \circ e_j) \circ f_j$.

Sei nun $\mathbf{Q}$ mit Abbildungen $q_i : \mathbf{B}_i \to \mathbf{Q}$ ein Konkurrent des Pushout, also $q_i \circ f_i = q_j \circ f_j$ für alle $i, j \in I$. Dann ist $\mathbf{Q}$ automatisch auch ein Konkurrent der Summe, so dass es eine eindeutige Abbildung $s : \sum_{i\in I} \mathbf{B}_i \to \mathbf{Q}$ gibt mit $q_i = s \circ e_i$ für alle $i \in I$. Es folgt $s \circ e_i \circ f_i = s \circ e_j \circ f_j$, so dass s zum Konkurrent des Coequalizers g wird. Damit erhält man den eindeutigen Morphismus $h : \mathbf{P} \to \mathbf{Q}$ mit $s = h \circ g$. Es folgt $q_i = s \circ e_i = h \circ (g \circ e_i)$ für jedes $i \in I$.

Die Eindeutigkeit von h folgert man leicht aus der Tatsache, dass der Coequalizer g epi ist (Aufgabe in Abschnitt 2.7) und die Familie der $(e_i)_{i\in I}$ gemeinsam epi. $\square$

Allgemeiner kann man den Begriff des **Colimes eines Diagramms** definieren. Summen, Coequalizer und Pushouts sind dann Colimites von Diagrammen der folgenden Form:

$$
\circ\ \circ \ldots \circ \qquad
\begin{tikzcd}
\circ \arrow[r, bend left] \arrow[r, phantom, "\vdots"] \arrow[r, bend right] & \circ
\end{tikzcd}
\qquad
\begin{tikzcd}
 & \circ \\
\circ \arrow[ur] \arrow[dr] & \vdots \\
 & \circ
\end{tikzcd}
$$

Man kann wie oben zeigen, dass aus Summen und Coequalizern die Colimites beliebiger Diagramme gewonnen werden können. Für unsere Zwecke werden allerdings Summen, Coequalizer und Pushouts genügen.

2.3 Die Kategorie der Mengen

Die Kategorie **Set** aller Mengen hat als Objekte alle Mengen und als Morphismen alle Abbildungen. Man überlegt sich leicht, dass hier die Epimorphismen gerade die surjektiven Abbildungen sind und die Monomorphismen die injektiven. Jede injektive Abbildung $f : X \to Y$ ist auch links-invertierbar, sofern $X \neq \emptyset$ ist. Zu festem, beliebigem $x_0 \in X$ setzt man

$$f^-_{x_0}(y) := \begin{cases} x \text{ falls } f(x) = y \\ x_0 \text{ falls } y \notin f[X]. \end{cases}$$

Die Frage, ob auch jede surjektive Abbildung eine rechts-inverse besitzt, hat eine vielleicht überraschende Antwort:

Hilfssatz 2.3.1 (Auswahlaxiom) *Das Auswahlaxiom ist äquivalent zu der Aussage, dass jede surjektive Abbildung rechts-invertierbar ist.*

Beweis Eine surjektive Abbildung $f : X \to Y$ liefert eine Familie $(f^{-1}\{y\})_{y \in Y}$ von nicht-leeren Teilmengen von X. Eine Auswahlfunktion für diese Familie ist offensichtlich rechts-invers zu f.

Umgekehrt liefert jede Familie $(X_i)_{i\in I}$ nicht-leerer Mengen eine surjektive Abbildung $f : \bigcup_{i\in I} X_i \to I$. Eine dazu rechts-inverse Abbildung ist eine Auswahlfunktion. □

Diagramm-Lemmata für Mengen

Hilfssatz 2.3.2 (Diagramm-Lemma) *Sei $f : X \twoheadrightarrow Y$ eine surjektive Abbildung und $g : X \to Z$ beliebig. Genau dann wenn $Kern\ f \subseteq Kern\ g$ ist, gibt es eine Abbildung $h : Y \to Z$ mit $h \circ f = g$. Diese ist eindeutig bestimmt.*

$$\begin{tikzcd} X \arrow[r, two heads, "f"] \arrow[dr, "g"'] & Y \arrow[d, dashed, "h"] \\ & Z \end{tikzcd}$$

Beweis Wegen $Kern\ f \subseteq Kern(h \circ f) = Kern\ g$ ist die Notwendigkeit der Bedingung offensichtlich. Umgekehrt rechnet man nach, dass durch

$$h := \{(f(x), g(x)) \mid x \in X\}$$

der Graph einer Abbildung $h : Y \to Z$ mit $h \circ f = g$ gegeben ist. Gäbe es ein weiteres h' mit $h' \circ f = g$, so hätte man $h \circ f = g = h' \circ f$. Man kann den Epimorphismus f rechts kürzen und erhält $h = h'$. □

Sind e, f, g, m Abbildungen mit $f \circ e = m \circ g$, wobei e surjektiv (epi) ist und m injektiv (mono), so nennen wir die Konfiguration ein **E-M-Quadrat**. Eine **Diagonale** eines solchen Quadrates ist ein Morphismus d mit $d \circ e = g$, der also das obere Dreieck kommutativ macht.

$$\begin{tikzcd} X \arrow[r, two heads, "e"] \arrow[d, "g"'] & Y \arrow[d, "f"] \arrow[dl, dashed, "d"'] \\ Z \arrow[r, tail, "m"'] & U \end{tikzcd}$$

Hilfssatz 2.3.3 (Diagonal-Eigenschaft) *In der Kategorie* **Set** *hat jedes E-M-Quadrat eine eindeutige Diagonale. Diese lässt automatisch auch das untere Dreieck kommutieren.*

Beweis Es gilt $Kern\ e \subseteq Kern(f \circ e) = Kern(m \circ g) = Kern\ g$, weil m injektiv ist. Nach dem vorigen Hilfssatz existiert genau eine Abbildung $d : Y \to Z$ mit $d \circ e = g$, also eine Diagonale.

Ist d eine Diagonale, so gilt $m \circ d \circ e = m \circ g = f \circ e$. Wir können e rechts kürzen, also ist $m \circ d = f$. □

Summen in Set

Sei $(X_i)_{i \in I}$ eine Familie von Mengen. Die *Summe der* X_i ist die disjunkte Vereinigung

$$\sum_{i \in I} X_i := \bigcup_{i \in I} \{(i, x) \mid x \in X_i\}$$

mit den durch $e_i(x) := (i, x)$ definierten Abbildungen.

Sei Q ein Konkurrent, also eine Menge mit Abbildungen $q_i : X_i \to Q$, so gibt es genau eine Abbildung $\sigma : \sum_{i \in I} X_i \to Q$ mit $\sigma \circ e_i = q_i$. Diese ist durch $\sigma(i, x) := q_i(x)$ definiert.

Von nun an schreiben wir $X + Y$ für die disjunkte Vereinigung der Mengen X und Y.

Coequalizer in Set

Hilfssatz 2.3.4 *Sind* $(f_i)_{i \in I} : X \to Y$ *Abbildungen, so sei* Θ *die durch die Menge*

$$R := \{(f_i(x), f_j(x)) \mid x \in X, i, j \in I\}$$

erzeugte Äquivalenzrelation. Die kanonische Abbildung $\pi_\Theta : Y \to Y/\Theta$ *mit* $\pi_\Theta(y) = [y]\Theta$ *ist dann der Coequalizer der* f_i*.*

Beweis Offensichtlich gilt für alle $i, j \in I$: $\pi_\Theta \circ f_i = \pi_\Theta \circ f_j$. Für jeden Konkurrenten, also jede Abbildung $q : Y \to Z$ mit $q \circ f_i = q \circ f_j$ für alle $i, j \in I$ muss $\Theta \subseteq Kern\ q$ sein, so dass nach Hilfssatz 2.3.2 genau eine Abbildung $h : A/\Theta \to Z$ existiert mit $h \circ \pi_\theta = q$. □

Pushouts in Set

Aufgrund von Satz 2.2.4 können wir Pushouts aus Summen und Coequalizern konstruieren. Als Coequalizer einer Familie $(f_i : X \to Y_i)_{i \in I}$ von Abbildungen, ergibt sich auf diese Weise der Pushout als $(\pi_\Theta \circ e_i)_{i \in I}$, wobei $e_i : X_i \to \sum_{i \in I} Y_i$ die kanonischen Einbettungen sind und Θ die Äquivalenzrelation auf $\sum_{i \in I} Y_i$, die von allen Paaren $(f_i(x), f_j(x))$ mit $x \in X$ und $i, j \in I$ erzeugt wird. Für uns ist der folgende Spezialfall relevant:

Hilfssatz 2.3.5 *Sei* $(\theta_i)_{i \in I}$ *eine Familie von Äquivalenz-Relationen auf einer Menge* X *und sei* $\Theta := \bigvee_{i \in I} \theta_i$*. Der Pushout der* $\pi_{\theta_i} : X \to X/\theta_i$ *ist* X/Θ *mit der Familie von Abbildungen* $\pi_i : X/\theta_i \to X/\Theta$*, die durch* $\pi_i([x]\theta_i) := [x]\Theta$ *definiert sind.*

Beweis Zunächst gilt $\pi_i \circ \pi_{\theta_i} = \pi_\Theta = \pi_j \circ \pi_{\theta_j}$ für alle $i, j \in I$.

Sei Q mit Abbildungen $q_i : X/\theta_i \to Q$ ein Konkurrent, d.h. $g := q_i \circ \pi_{\theta_i} = q_j \circ \pi_{\theta_j}$ für alle $i, j \in I$, dann gilt $\theta_i \subseteq Kern\ g$ für alle $i \in I$, also $\Theta \subseteq Kern\ g$. Lemma 2.3.2 liefert die eindeutige Abbildung $h : X/\Theta \to Q$ mit $h \circ \pi_\Theta = g$. Für jedes i gilt $q_i \circ \pi_{\theta_i} = g = h \circ \pi_\Theta = h \circ \pi_i \circ \pi_{\theta_i}$, daher gilt $q_i = h \circ \pi_i$. □

2.4 Funktoren

Funktoren stellen Beziehungen zwischen Kategorien her. Ein Funktor F zwischen Kategorien $\mathbb{C}$ und $\mathbb{D}$ ordnet

- jedem Objekt $\mathbf{A} \in \mathbb{C}$ ein Objekt $F(\mathbf{A}) \in \mathbb{D}$ zu,
- jedem $\mathbb{C}$-Morphismus $f : \mathbf{A} \to \mathbf{B}$ einen $\mathbb{D}$-Morphismus $Ff : F(\mathbf{A}) \to F(\mathbf{B})$,

so dass die folgenden Gleichungen gelten:

$$\begin{aligned} Fid_{\mathbf{A}} &= id_{F(\mathbf{A})} \\ F(f \circ g) &= Ff \circ Fg \end{aligned}$$

Ein Funktor verhält sich also wie ein Paar von Abbildungen. Eine davon, die Objekt-Abbildung, ist zwischen den Objekten definiert, die andere zwischen den Morphismen.

$$\begin{tikzcd} \mathbf{A} \arrow[r, mapsto] \arrow[d, dotted, "f"] & F(\mathbf{A}) \arrow[d, dotted, "Ff"'] \\ \mathbf{B} \arrow[r, mapsto] & F(\mathbf{B}) \end{tikzcd}$$

Formal handelt es sich nur deswegen nicht wirklich um “Abbildungen”, weil die Definitionsbereiche echte Klassen (die Klasse aller Mengen, bzw. aller Mengen-Abbildungen) sind. Dennoch schreiben wir $F : \mathbb{C} \to \mathbb{D}$.

Für unsere Zwecke spielen nur **Mengenfunktoren** $F : \mathbf{Set} \to \mathbf{Set}$ eine Rolle. Es handelt sich um die im vorigen Kapitel angesprochenen “mengentheoretischen Konstruktionen” $X \to F(X)$, die sich auch auf Abbildungen $f : X \to Y$ übertragen lassen als $Ff : F(X) \to F(Y)$.

Beispiel 2.4.1 *Der* **Identitäts-Funktor** $\mathcal{I}$ *mit* $\mathcal{I}(X) = X$ *für jede Menge* X *und* $\mathcal{I}f = f$ *für jede Mengen-Abbildung* f *ist ein Funktor.*

Beispiel 2.4.2 *Sei* C *eine fest vorgegebene Menge. Mit* C *bezeichnen wir auch den* **konstanten Funktor** *mit* $C(X) = C$ *für jede Menge* X *und* $Cf = id_C$ *für jede Abbildung* $f : X \to Y$.

Beispiel 2.4.3 *Sei* C *wieder eine fest gewählte Menge. Der* **Potenzfunktor** $(-)^C$ *ordnet jeder Menge* X *die Menge* X^C *aller Abbildungen von* C *nach* X *zu und jeder Abbildung* $f : X \to Y$ *die Abbildung die Abbildung* f^C *mit* $f^C(u) = f \circ u$.

Beispiel 2.4.4 *Der* **Listenfunktor** $(-)^*$ *ordnet einer Menge* X *die Menge* X^* *aller endlichen Listen von Elementen aus* X *zu. (Die Elemente von* X^* *kann man auch als Worte über dem Alphabet* X *deuten). Eine Abbildung* $f : X \to Y$ *lässt sich elementweise zu einer Abbildung* $f^* : X^* \to Y^*$ *fortsetzen.*

Beispiel 2.4.5 *Der* **Potenzmengen-Funktor** $\mathbb{P}$ *ordnet einer Menge* X *die Menge* $\mathbb{P}(X)$ *aller Teilmengen von* X *zu. Einer Abbildung* $f : X \to Y$ *zwischen zwei beliebigen Mengen* X *und* Y *ordnen wir die Abbildung* $\mathbb{P}(f) : \mathbb{P}(X) \to \mathbb{P}(Y)$ *mit* $(\mathbb{P}f)(U) := f[U]$ *zu.*

Beispiel 2.4.6 *Sind $F, G : \mathbf{Set} \to \mathbf{Set}$ Funktoren, dann auch $F \circ G$, $F \times G$ und $F + G$ wenn man für ein beliebiges $X \in \mathbf{Set}$ und eine beliebige Abbildung $f : X \to Y$ setzt:*

(i) $(F \circ G)(X) := F(G(X))$ *und* $(F \circ G)f := F(Gf)$,

(ii) $(F \times G)(X) := F(X) \times G(X)$ *und* $(F \times G)(f)(u, v) := ((Ff)(u), (Ff)(v))$,

(iii) $(F + G)(X) := F(X) + G(X)$ *und* $(F + G)(f)(u) = \begin{cases} (Ff)(u), & \text{falls } u \in F(X) \\ (Fg)(u), & \text{falls } u \in G(X). \end{cases}$

Beispiel 2.4.7 (3-2-Funktor) *Für eine beliebige Menge X sei*

$$(X)^3_2 := \{(x_1, x_2, x_3) \in X^3 \mid x_1 = x_2 \vee x_1 = x_3 \vee x_2 = x_3\}.$$

Für eine beliebige Abbildung $f : X \to Y$ sei $(f)^3_2 : (X)^3_2 \to (Y)^3_2$ komponentenweise erklärt, also $(f)^3_2(x_1, x_2, x_3) := (f(x_1), f(x_2), f(x_3))$. Auf diese Weise erhalten wir einen Funktor $(-)^3_2$.

Beispiel 2.4.8 *Jeder universell algebraische Typ definiert auch einen Funktor. Seien zum Beispiel die Operations-Symbole $\cdot$, $^{-1}$ und e mit den Stelligkeiten 2,1 und 0 gegeben. Dieser Typ bestimmt einen Funktor F mit $F(X) = (X \times X) + X + \{*\}$ für jede Menge X. Für eine Abbildung $f : X \to Y$ setzen wir:*

$$(Ff)(u) = \begin{cases} (f(x_0), f(x_1)), & \text{falls } u = (x_0, x_1) \in X \times X \\ f(u), & \text{falls } u \in X \\ * & \text{falls } u = * \end{cases}$$

Beispiel 2.4.9 *Sei $\Delta = (\mathcal{F}, \sigma)$ ein Typ von universellen Algebren. Dann wird durch $\Delta(X) = \sum_{f \in \mathcal{F}} X^{\sigma(f)}$ die Objekt-Abbildung eines Funktors definiert.*

Beispiel 2.4.10 *Für eine beliebige Menge X definieren wir $\mathcal{P}(X) := \mathbb{P}(X)$, und für eine Abbildung $f : X \to Y$ sei $\mathcal{P}f : \mathbb{P}(Y) \to \mathbb{P}(X)$ durch $(\mathcal{P}f)(V) := \{x \in X \mid f(x) \in V\}$ definiert. Diese Konstruktion ergibt keinen Funktor, da die Richtung der Abbildungen umgedreht wird. Man erkennt aber leicht, dass die Hintereinander-Ausführung $\mathcal{P} \circ \mathcal{P}$ ein Funktor ist.*

2.5 Eigenschaften von Mengen-Funktoren

Hilfssatz 2.5.1 *Sei $F : \mathbf{Set} \to \mathbf{Set}$ ein Mengen-Funktor, $X \neq \emptyset$ und $f : X \to Y$ eine Abbildung. Ist f injektiv, dann auch Ff.*

Beweis Sei g links-invers zu f, dann gilt $g \circ f = id_X$, also auch $F(g) \circ F(f) = F(g \circ f) = F(id_X) = id_{F(X)}$. Somit hat $F(f)$ ein links-inverses und ist daher selber injektiv. □

Der nächste Hilfssatz erlaubt uns, stets $F(X) \neq \emptyset$ anzunehmen:

Hilfssatz 2.5.2 *Falls $F(X) = \emptyset$ für eine Menge $X \neq \emptyset$, dann folgt $F(Y) = \emptyset$ für alle Mengen Y und $Ff = \emptyset$ für jede Abbildung f.*

Beweis Ist $X \neq \emptyset$, so gibt es für jede Menge Y eine Abbildung $f : Y \to X$, also auch eine Abbildung $Ff : F(Y) \to F(X)$. □

2.6 Natürliche Transformationen

Wenn wir einen **Set**-Funktor als "Mengenkonstruktion" deuten, so erhebt sich die Frage, wie man "natürliche" Beziehungen zwischen zwei solchen Konstruktionen beschreiben kann. Zum Beispiel kann man aus jeder Liste eine Menge machen, indem man Duplikate entfernt. Diese Transformation ist im Sinne der folgenden Definition "natürlich" und mit den mit den Abbildungen verträglich:

Definition 2.6.1 *Seien* $F, G : \mathbb{C} \to \mathbb{D}$ *Funktoren. Eine* **natürliche Transformation** η ***von*** F ***nach*** G *ordnet jedem Objekt* $X \in \mathbb{C}$ *einen* $\mathbb{D}$*-Morphismus* $\eta_X : F(X) \to G(X)$ *zu, so dass für jeden* $\mathbb{C}$*-Morphismus* $f : X \to Y$ *gilt*

$$Gf \circ \eta_X = \eta_Y \circ Ff.$$

$$\begin{tikzcd} X \arrow[d, "f"] \\ Y \end{tikzcd} \qquad \begin{tikzcd} F(X) \arrow[r, "\eta_X"] \arrow[d, "Ff"'] & G(X) \arrow[d, "Gf"] \\ F(Y) \arrow[r, "\eta_Y"] & G(Y) \end{tikzcd}$$

Man schreibt $\eta : F \dot{\to} G$ *falls* η *eine natürliche Transformation von* F *nach* G *ist.*

Beispielsweise hat man eine natürliche Transformation η vom Listen-Funktor $(-)^*$ zum endlichen Potenzmengenfunktor $\mathbb{P}_\omega$. Einer Menge X ordnet η die Abbildung $\eta_X : X^* \to \mathbb{P}_\omega(X)$ zu, die jede endliche Liste $l \in X^*$ in die Menge $|l|$ aller Elemente von l transformiert.

Eine natürliche Transformation zwischen Mengenfunktoren F und G nennen wir **surjektive natürliche Transformation**, falls η_X für jede Menge $X \neq \emptyset$ surjektiv ist.

2.7 Aufgaben

(i) Jeder Coequalizer ist epi.

(ii) Die Familie der kanonischen Injektionen $e_i : \mathbf{A}_i \to \sum_{i \in I} \mathbf{A}_i$ in eine Summe ist *gemeinsam epi* (siehe Definition 2.2.1).

(iii) Eine Algebra vom Typ $(2, 1, 0)$ ist eine Abbildung $\gamma_X : F(X) \to X$, wobei F der Funktor aus Beispiel 2.4.8 ist. Ein Homomorphismus zwischen zwei Algebren $\mathbf{A} = (A, \gamma_A)$ und $\mathbf{B} = (B, \gamma_B)$ des Typs $(2, 1, 0)$ ist eine Abbildung $\varphi : A \to B$ so dass das folgende Diagramm kommutiert:

$$\begin{tikzcd} F(A) \arrow[r, "F\varphi"] \arrow[d, "\gamma_A"'] & F(B) \arrow[d, "\gamma_B"] \\ A \arrow[r, "\varphi"] & B \end{tikzcd}$$

(iv) $f : \mathbf{A} \to \mathbf{B}$ ist genau dann epi, wenn das folgende ein Pushout-Diagramm ist:

$$\begin{tikzcd} A \arrow[r, "f"] \arrow[d, "f"'] & B \arrow[d, "id_B"] \\ B \arrow[r, "id_B"] & B \end{tikzcd}$$

(v) ("*Pushouts von epis sind epi*") Ist f in dem folgenden Pushout-Diagramm epi, dann auch das "gegenüberliegende" f':

$$\begin{tikzcd} A \arrow[r, two heads, "f"] \arrow[d, "g"'] & B \arrow[d, "g'"] \\ C \arrow[r, "f'"] & P \end{tikzcd}$$

(vi) Ein Objekt $\mathbf{I}$ in einer Kategorie $\mathbb{C}$ heißt **initial**, wenn es in jedes Objekt $\mathbf{A} \in \mathbb{C}$ genau einen Morphismus $\iota : \mathbf{I} \to \mathbf{A}$ gibt. Bestimmen Sie das initiale und das terminale Objekt in der Kategorie aller Algebren eines festen Typs.

(vii) In der Kategorie aller Algebren eines festen Typs ist die Summe $\mathbf{T}(\{x\}) + \mathbf{T}(\{y\})$ der Termalgebren $\mathbf{T}(\{x\})$ und $\mathbf{T}(\{y\})$ isomorph zu $\mathbf{T}(\{x, y\})$

(viii) Sei $\mathbf{1}$ der ein-elementige Verband. Bestimmen Sie $\mathbf{1} + \mathbf{1}$ in der Kategorie aller Verbände.

(ix) Bestimmen Sie die Summe $G_1 + G_2$ zweier Gruppen G_1 und G_2 in der Kategorie aller Gruppen.

3 Coalgebren

Definition 3.0.1 *Ein* **Typ** *ist ein Mengenfunktor. Eine* **Coalgebra vom Typ** F *ist ein Paar* $\mathcal{A} = (A, \alpha_A)$ *bestehend aus einer Menge* A *und einer Abbildung*

$$\begin{tikzcd} A \arrow[d, "\alpha_A"] \\ F(A) \end{tikzcd}$$

Offensichtlich deckt diese Definition alle aufgeführten Beispiele zustandsbasierter Systeme ab. Wichtig ist zunächst der Homomorphie-Begriff:

Definition 3.0.2 *Seien* $\mathbf{A} = (A, \alpha_A)$ *und* $\mathbf{B} = (B, \alpha_B)$ *Coalgebren. Eine Abbildung* $\varphi : A \to B$ *heißt* **Homomorphismus** *von* $\mathbf{A}$ *nach* $\mathbf{B}$*, falls*

$$\alpha_B \circ \varphi = F\varphi \circ \alpha_A,$$

falls also das folgende Diagramm kommutiert:

$$\begin{tikzcd} A \arrow[r, dashed, "\varphi"] \arrow[d, "\alpha_A"'] & B \arrow[d, "\alpha_B"] \\ F(A) \arrow[r, dashed, "F\varphi"'] & F(B) \end{tikzcd}$$

Die folgenden Eigenschaften ergeben sich sofort aus den entsprechenden Bedingungen an einen Funktor:

Satz 3.0.3 $\mathbf{A} = (A, \alpha_A)$, $\mathbf{B} = (B, \alpha_B)$ *und* $\mathbf{C} = (C, \alpha_C)$ *seien Coalgebren vom Typ* F.

(i) $id_A : \mathbf{A} \to \mathbf{A}$ *ist ein Homomorphismus.*

(ii) Sind $\varphi : \mathbf{A} \to \mathbf{B}$ *und* $\psi : \mathbf{B} \to \mathbf{C}$ *Homomorphismen, dann auch* $\psi \circ \phi : \mathbf{A} \to \mathbf{C}$

Beweis (i): $\alpha_A \circ id_A = \alpha_A = id_{F(A)} \circ \alpha_A = F(id_A) \circ \alpha_A$. Für (ii) rechnen wir nach:

$$\alpha_C \circ (\psi \circ \varphi) = (\alpha_C \circ \psi) \circ \varphi = F(\psi) \circ \alpha_B \circ \varphi = F(\psi) \circ F(\varphi) \circ \alpha_A = F(\psi \circ \varphi) \circ \alpha_A$$

□

Folgerung 3.0.4 *Die Klasse aller Coalgebren zu einem festen Typ* F *mit den oben definierten Homomorphismen als Morphismen bildet eine Kategorie, die wir mit* $\mathbf{Set}_F$ *bezeichnen.*

Isomorphismen, Diagramm-Lemma für Coalgebren

Satz 3.0.5 *Ein bijektiver Homomorphismus ist ein Isomorphismus.*

Beweis Wir rechnen nach, dass die Umkehrabbildung von φ, ein Homomorphismus ist:

$$\begin{aligned} \alpha_A \circ \varphi^{-1} &= F(\varphi^{-1}) \circ F(\varphi) \circ \alpha_A \circ \varphi^{-1} \\ &= F(\varphi^{-1}) \circ \alpha_B \circ \varphi \circ \varphi^{-1} \\ &= F(\varphi^{-1}) \circ \alpha_B. \end{aligned}$$

□

Hilfssatz 3.0.6 *Seien* $\mathbf{A}$, $\mathbf{B}$ *und* $\mathbf{C}$ *Coalgebren und* $f : A \to B$, $g : B \to C$ *Mengenabbildungen, so dass* $\varphi := g \circ f : \mathbf{A} \to \mathbf{C}$ *ein Homomorphismus ist. Dann gilt:*

(i) Ist f *ein surjektiver Homomorphismus, dann ist auch* g *ein Homomorphismus.*

(ii) Ist g *ein injektiver Homomorphismus, dann ist auch* f *ein Homomorphismus.*

Beweis

$$\begin{tikzcd} A \arrow[r, "f"'] \arrow[rr, bend left, "\varphi"] \arrow[d, "\alpha_A"'] & B \arrow[r, "g"'] \arrow[d, "\alpha_B"'] & C \arrow[d, "\alpha_C"] \\ F(A) \arrow[r, "F(f)"] \arrow[rr, bend right, "F(\varphi)"'] & F(B) \arrow[r, "F(g)"] & F(C). \end{tikzcd}$$

(i): Wenn f und $\varphi = g \circ f$ Homomorphismen sind, gilt:

$$\begin{aligned} \alpha_C \circ g \circ f &= \alpha_C \circ \varphi \\ &= F(\varphi) \circ \alpha_A \\ &= F(g \circ f) \circ \alpha_A \\ &= F(g) \circ F(f) \circ \alpha_A \\ &= F(g) \circ \alpha_B \circ f \end{aligned}$$

f ist surjektiv, also rechts-kürzbar, so dass $\alpha_C \circ g = F(g) \circ \alpha_B$ folgt.

Der Beweis von (ii) ist analog, allerdings müssen wir hier unter Verwendung von Hilfssatz 2.5.1 ausnutzen, dass mit g auch $F(g)$ injektiv, somit links-kürzbar ist. $\square$ Wir verbinden dieses Ergebnis mit dem Diagramm-Lemma für Mengen:

Folgerung 3.0.7 (Diagramm-Lemma für Coalgebren) *Seien $\varphi : \mathbf{A} \to \mathbf{B}$ und $\psi : \mathbf{A} \to \mathbf{C}$ Homomorphismen und φ surjektiv. Genau dann gibt es einen Homomorphismus $\chi : \mathbf{B} \to \mathbf{C}$ mit $\chi \circ \varphi = \psi$, wenn $Kern\ \varphi \subseteq Kern\ \psi$.*

3.1 Unterstrukturen

Definition 3.1.1 *Sei $\mathbf{A} = (A, \alpha_A)$ eine Coalgebra. Eine Teilmenge $S \subseteq A$ heißt* **offen**, *falls es eine Struktur-Abbildung $\alpha_S : S \to F(S)$ gibt, so dass $\subseteq^A_S : S \to A$ ein Homomorphismus ist. $\mathbf{S} = (S, \alpha_S)$ heißt dann Unter-Coalgebra von $\mathbf{A}$ und wir schreiben $\mathbf{S} \leq \mathbf{A}$.*

Hilfssatz 3.1.2 *Auf jeder offenen Teilmenge $S \subseteq A$ gibt es eine eindeutige Struktur-Abbildung $\alpha_S : S \to F(S)$, mit der $\mathbf{S} = (S, \alpha_S)$ eine Unter-Coalgebra von $\mathbf{A}$ wird.*

Beweis Für $S = \emptyset$ ist die Behauptung klar. Ansonsten nehmen wir an, es gäbe zwei Struktur-Abbildungen $\sigma_1, \sigma_2 : S \to F(S)$ für die die kanonische Einbettung $\subseteq^A_S$ ein Homomorphismus ist. Es folgt:

$$F(\subseteq^A_S) \circ \sigma_1 = \alpha_A \circ \subseteq^A_S = F(\subseteq^A_S) \circ \sigma_2.$$

Wegen Hilfssatz 2.5.1 können wir $F(\subseteq^A_S)$ links kürzen, so dass $\sigma_1 = \sigma_2$ folgt. $\square$

Dieser Hilfssatz erlaubt uns, die Begriffe "offene Teilmenge" und "Unter-Coalgebra" synonym zu verwenden.

3.2 Homomorphe Bilder, Faktorisierungen

Der folgende Hilfssatz und sein Beweis sind analog zu 3.1.2:

Hilfssatz 3.2.1 *Ist $\varphi : \mathbf{A} \to \mathbf{B}$ ein surjektiver Homomorphismus, so ist die Coalgebra-Struktur α_B auf B eindeutig durch φ und α_A bestimmt. Wir können die Struktur auf B sogar direkt angeben als:*

$$\alpha_B = \{(\varphi(a), (F\varphi)(\alpha_A(a)) \mid a \in A\}.$$

Aus diesem Grunde definieren wir:

Definition 3.2.2 *Ist $\varphi : \mathbf{A} \twoheadrightarrow \mathbf{B}$ ein surjektiver Homomorphismus, so heißt* **B homomorphes Bild** *von* **A**.

Satz 3.2.3 (Faktorisierungssatz) *Sei $\varphi : \mathbf{A} \to \mathbf{B}$ ein Homomorphismus, sei $\varphi = g \circ f$ eine Zerlegung von φ in eine surjektive Abbildung $f : A \twoheadrightarrow Q$ gefolgt von einer injektiven Abbildung $g : Q \rightarrowtail B$. Dann gibt es auf Q eine eindeutige F-Coalgebra-Struktur α_Q, so dass f ein Homomorphismus ist. Bezüglich dieser Struktur ist g automatisch ein Homomorphismus.*

Beweis

$$\begin{tikzcd}
A \arrow[r, two heads, "f"'] \arrow[rr, bend left, "\varphi"] \arrow[d, "\alpha_A"'] & Q \arrow[r, tail, "g"'] \arrow[d, dashed, "\alpha_Q"] & B \arrow[d, "\alpha_B"] \\
F(A) \arrow[r, "Ff"] \arrow[rr, bend right, "F\varphi"'] & F(Q) \arrow[r, tail, "Fg"] & F(B)
\end{tikzcd}$$

Weil $\varphi = g \circ f$ ein Homomorphismus ist, hat man

$$\begin{aligned}
\alpha_B \circ (g \circ f) &= F(g \circ f) \circ \alpha_A \\
&= F(g) \circ F(f) \circ \alpha_A.
\end{aligned}$$

Mit g ist auch $F(g)$ injektiv, so dass f, $\alpha_B \circ g$, $F(f) \circ \alpha_A$ und $F(g)$ ein E-M-Quadrat bilden. Die aufgrund von Hilfssatz 2.3.3 eindeutig existierende Diagonale ist die gesuchte Struktur-Abbildung α_Q auf Q. $\square$

Ist $f : A \to B$ eine Abbildung, so sei

$$f[A] := \{f(a) \mid a \in A\}$$

das **Bild** von A unter f und $f' : A \to f[A]$ sei die Bild-Restriktion von f. Da wir f als

$$f \ = \subseteq^B_{f[A]} \circ f'$$

surjektiv-injektiv zerlegen können, folgt aus 3.2.3:

Folgerung 3.2.4 *Ist $\varphi : \mathbf{A} \to \mathbf{B}$ ein Homomorphismus, so ist $\varphi[A]$ ein homomorphes Bild von $\mathbf{A}$ und eine Unter-Coalgebra von $\mathbf{B}$.*

Ist $\mathbf{U} \leq \mathbf{A}$ eine Unter-Coalgebra von $\mathbf{A}$ und $\varphi : \mathbf{A} \to \mathbf{B}$ ein Homomorphismus, so liefert die surjektiv-injektiv-Zerlegung von $\varphi \circ \subseteq^A_U$:

Folgerung 3.2.5 *$\varphi[U]$ ist eine Unter-Coalgebra von $\mathbf{B}$.*

Kongruenzen, Faktor-Coalgebren

Wie in der Universellen Algebra definieren wir Kongruenzen als Äquivalenz-Relationen, die die Struktur respektieren. In diesem Falle bedeutet dies:

Definition 3.2.6 *Eine Kongruenz auf einer Coalgebra $\mathbf{A}$ ist ein Kern eines Homomorphismus $\varphi : \mathbf{A} \to \mathbf{B}$ für irgendein $\mathbf{B}$.*

Ist θ eine Kongruenz auf $\mathbf{A}$, dann gibt es einen Homomorphismus $\varphi : \mathbf{A} \to \mathbf{B}$ mit *Kern* $\varphi = \theta$. Wir können φ surjektiv wählen. Es lässt sich dann als $f \circ \pi_\theta$ schreiben, mit f bijektiv. Satz 3.2.3 liefert eine eindeutige Coalgebra-Struktur α_θ auf A/θ. Die Coalgebra $\mathbf{A}/\theta = (A/\theta, \alpha_\theta)$ nennen wir **Faktor-Coalgebra**. In Verbindung mit Satz 3.0.5 folgt, dass f sogar ein Isomorphismus ist. Wir erhalten also:

Satz 3.2.7 *Ist $\varphi : \mathbf{A} \to \mathbf{B}$ ein surjektiver Homomorphismus mit Kern θ, so gilt $\mathbf{A}/\theta \cong \mathbf{B}$.*

Insbesondere ist also jede Kongruenz θ Kern eines Homomorphismus $\pi_\theta : \mathbf{A} \to \mathbf{A}/\theta$. Daher können wir auch direkt eine Bedingung dafür formulieren, dass eine Äquivalenz-Relation eine Kongruenz ist. Aus der Homomorphie-Bedingung 3.0.2 und dem Diagramm-Lemma 2.3.2 erhalten wir nämlich $\theta \subseteq Kern(F\pi_\theta) \circ \alpha_A$, was bedeutet:

Folgerung 3.2.8 *Eine Äquivalenzrelation θ auf einer Coalgebra $\mathbf{A}$ ist genau dann eine Kongruenzrelation, wenn für je zwei Elemente $a, b \in A$ gilt:*

$$a\theta b \implies (F\pi_\theta)(\alpha_A(a)) = (F\pi_\theta)(\alpha_A(b)).$$

3.3 Colimiten in $\mathbf{Set}_F$

Ist $\mathbf{A}_i = (A_i, \alpha_i)$ für jedes $i \in I$ eine Coalgebra, so können wir auf der disjunkten Vereinigung $\sum_{i\in I} A_i$ eine Coalgebra-Struktur definieren. Mit den kanonischen Injektionen $e_i : A_i \to \sum_{i\in I} A_i$ und den Abbildungen $F(e_i) \circ \alpha_i$ wird nämlich $F(\sum_{i\in I} A_i)$ zum Konkurrenten der Summe in der Kategorie **Set**, so dass man eine eindeutige Struktur-Abbildung α auf $\sum_{i\in I} A_i$ gewinnt, die augenscheinlich alle e_i zu Homomorphismen macht. Da diese auch injektiv sind, folgt, dass jedes $\mathbf{A}_i$ sogar isomorph zu einer Unter-Coalgebra von $\sum_{i\in I} \mathbf{A}_i$ ist.

$$\begin{tikzcd} A_i \arrow[r, "e_i"] \arrow[d, "\alpha_i"'] & \sum_{i\in I} A_i \arrow[d, dotted, "\alpha"] \\ F(A_i) \arrow[r, "F(e_i)"] & F(\sum_{i\in I} A_i) \end{tikzcd}$$

Man kann α auch direkt angeben als

$$\alpha(i, a) := F(e_i) \circ \alpha_i(a).$$

$\sum_{i\in I} \mathbf{A}_i = (\sum_{i\in I} A_i, \alpha)$ heißt **Summe der $\mathbf{A}_i$**, $i \in I$.

Salopp ausgedrückt ist die disjunkte Vereinigung von Coalgebren wieder eine Coalgebra. Man vergleiche dazu auch das Beispiel der Kripke-Strukturen auf Seite 11. Man kann dieses entweder als zwei disjunkte Kripke-Strukturen auffassen, oder als eine Summe von zwei disjunkten Komponenten.

Satz 3.3.1 *Die disjunkte Summe der Coalgebren $\mathbf{A}_i$ ist die Summe der $\mathbf{A}_i$ in der Kategorie $\mathbf{Set}_F$.*

Beweis Sei $\mathbf{Q} = (Q, \gamma)$ mit Homomorphismen $\varphi_i : \mathbf{A}_i \to \mathbf{Q}$ ein Konkurrent von $\sum_{i\in I} \mathbf{A}_i$ in der Kategorie $\mathbf{Set}_F$, d.h. $\mathbf{Q}$ ist eine Coalgebra und jedes φ_i ein Homomorphismus.

$$\begin{tikzcd} Q \arrow[d, "\gamma"'] & A_i \arrow[l, dashed, "\varphi_i"] \arrow[r, "e_i"'] \arrow[d, "\alpha_i"] & \sum_{i\in I} A_i \arrow[d, "\alpha"] \arrow[ll, dotted, bend right, "\sigma"'] \\ F(Q) & F(A_i) \arrow[l, dashed, "F(\varphi_i)"'] \arrow[r, "F(e_i)"] & F(\sum_{i\in I} A_i) \arrow[ll, dotted, bend left, "F(\sigma)"] \end{tikzcd}$$

Betrachten wir nur die obere Zeile des Diagramms, dann ist Q mit den Abbildungen φ_i auch ein Konkurrent von $\sum_{i\in I} A_i$, der Summe der A_i in der Kategorie der Mengen. Somit existiert eine eindeutige Abbildung σ mit $\varphi_i = \sigma \circ e_i$ für alle $i \in I$. Es bleibt zu zeigen, dass σ sogar ein Homomorphismus ist. Für jedes i gilt

$$\begin{aligned} \gamma \circ \sigma \circ e_i &= \gamma \circ \varphi_i \\ &= F(\varphi_i) \circ \alpha_i \\ &= F(\sigma) \circ F(e_i) \circ \alpha_i \\ &= F(\sigma) \circ \alpha \circ e_i \end{aligned}$$

Da die e_i gemeinsam epi sind, folgt $\gamma \circ \sigma = F(\sigma) \circ \alpha$. □

Eine Anwendung dieses Satzes ist:

Satz 3.3.2 *Sei $(U_i)_{i\in I}$ eine Familie von Unter-Coalgebren von $\mathbf{A}$, dann ist auch die Vereinigung $\bigcup_{i\in I} U_i$ eine Unter-Coalgebra von $\mathbf{A}$.*

Beweis Da $\subseteq^A_{U_i} : \mathbf{U}_i \to \mathbf{A}$ für jedes i ein Homomorphismus ist, wird $\mathbf{A}$ zum Konkurrenten der disjunkten Summe $\mathbf{S} = \sum_{i\in I} \mathbf{U}_i$. Es gibt also (genau) einen Homomorphismus $\varphi : \mathbf{S} \to \mathbf{A}$ mit $\varphi \circ e_i = \subseteq^A_{U_i}$ für jedes i. Das Bild von $\mathbf{S}$ unter φ

$$\varphi[\mathbf{S}] = \{\varphi(i,u) \mid i \in I, u \in U_i\} = \bigcup_{i\in I} (\varphi \circ e_i)[U_i] = \bigcup_{i\in I} U_i.$$

ist nach Folgerung 3.2.4 eine Untercoalgebra von $\mathbf{A}$. □

Ist $\mathbf{A} = (A\alpha_A)$ eine Coalgebra, so existiert aufgrund dieses Satzes zu jeder Teilmenge $S \subseteq A$ eine größte Unter-Coalgebra, die in S enthalten ist. Wir bezeichnen diese als die von S **coerzeugte** Untercoalgebra und schreiben dafür $[S]$.

Folgerung 3.3.3 *Die Menge aller Unter-Coalgebren einer Coalgebra $\mathbf{A}$ bildet einen Verband $Sub(\mathbf{A})$ mit kleinstem Element $\emptyset$ und größtem Element A.*

Dass Untercoalgebren unter Vereinigungen abgeschlossen sind, erscheint nicht verwunderlich, da Unter*algebren* in der dualen Theorie der Universellen Algebra ja gegen beliebige Schnitte abgeschlossen sind. Überraschend ist daher folgendes Ergebnis aus [GS00]:

Satz 3.3.4 *Der Schnitt von endlich vielen Unter-Coalgebren ist eine Untercoalgebra.*

Beweis Seien U und V Untercoalgebren von $\mathbf{A} = (A, \alpha_A)$. Für $U \cap V = \emptyset$ ist die Behauptung trivial, da die leere Menge immer eine Untercoalgebra ist. Sei von jetzt ab also $w \in U \cap V$ gewählt. Wir definieren Abbildungen $p_w : U \to U \cap V$ und $q_w : A \to V$ als

$$p_w(u) := \begin{cases} u & \text{falls } u \in U \cap V \\ w & \text{sonst} \end{cases} \qquad \text{und} \qquad q_w(a) := \begin{cases} a & \text{falls } a \in V \\ w & \text{sonst.} \end{cases}$$

Wir benötigen die Eigenschaften

$$\subseteq^V_{U\cap V} \circ\, p_w = q_w \circ \subseteq^A_U \qquad \text{und} \qquad q_w \circ \subseteq^A_V = id_V.$$

Damit zeigen wir, dass

$$\gamma := F(p_w) \circ \alpha_U \circ \subseteq^U_{U\cap V}$$

eine Coalgebra-Struktur auf $U \cap V$ definiert, die $U \cap V$ zur Untercoalgebra von $\mathbf{A}$ macht:

$$
\begin{aligned}
F(\subseteq^A_{U\cap V}) \circ \gamma &= F(\subseteq^A_V) \circ F(\subseteq^V_{U\cap V}) \circ F(p_w) \circ \alpha_U \circ \subseteq^U_{U\cap V} \\
&= F(\subseteq^A_V) \circ F(q_w) \circ F(\subseteq^A_U) \circ \alpha_U \circ \subseteq^U_{U\cap V} \\
&= F(\subseteq^A_V) \circ F(q_w) \circ \alpha_A \circ \subseteq^A_U \circ \subseteq^U_{U\cap V} \\
&= F(\subseteq^A_V) \circ F(q_w) \circ \alpha_A \circ \subseteq^A_V \circ \subseteq^V_{U\cap V} \\
&= F(\subseteq^A_V) \circ F(q_w) \circ F(\subseteq^A_V) \circ \alpha_V \circ \subseteq^V_{U\cap V} \\
&= F(\subseteq^A_V) \circ \alpha_V \circ \subseteq^V_{U\cap V} \\
&= \alpha_A \circ \subseteq^A_V \circ \subseteq^V_{U\cap V} \\
&= \alpha_A \circ \subseteq^A_{U\cap V}
\end{aligned}
$$

□

Sub($\mathbf{A}$) bildet somit einen topologischen Raum. Umgekehrt wurde in [Gum01] gezeigt, dass man auf jedem topologischen Raum eine Coalgebra eines bestimmten Typs definieren kann, so dass die offenen Mengen genau die Unter-Coalgebren werden. Dies belegt, dass das System der Unter-Coalgebren i.A. nicht unter beliebigen Schnitten abgeschlossen ist.

Coequalizer in $\mathbf{Set}_F$

Hilfssatz 3.3.5 *Sei* $(\varphi_i : \mathbf{A} \to \mathbf{B})_{i\in I}$ *eine Familie von Coalgebra-Homomorphismen und* $\pi_\Theta : B \to B/\Theta$ *der Coequalizer der Abbildungen* φ_i *in der Kategorie* $\mathbf{Set}$. *Dann gibt es auf* B/Θ *eine eindeutige Struktur-Abbildung* α_Θ, *so dass* $\pi_\Theta : \mathbf{B} \to \mathbf{B}/\Theta = (B/\Theta, \alpha_\Theta)$ *der Coequalizer der Homomorphismen* φ_i *in* $\mathbf{Set}_F$ *ist.*

Beweis Für beliebige $i, j \in I$ gilt:

$$
\begin{aligned}
F(\pi_\Theta) \circ \alpha_B \circ \varphi_i &= F(\pi_\Theta) \circ F(\varphi_i) \circ \alpha_A \\
&= F(\pi_\Theta \circ \varphi_i) \circ \alpha_A \\
&= F(\pi_\Theta \circ \varphi_j) \circ \alpha_A \\
&= F(\pi_\Theta) \circ \alpha_B \circ \varphi_j.
\end{aligned}
$$

Damit ist aber $F(\pi_\Theta) \circ \alpha_B$ ein Konkurrent, in der Kategorie der Mengen, des Coequalizers π_Θ. Dies liefert eine Struktur-Abbildung α_Θ auf B/Θ, bezüglich der π_Θ ein Homomorphismus ist.

$$
\begin{tikzcd}
A \arrow[r, shift left, "\varphi_i"] \arrow[r, shift right, "\varphi_j"'] \arrow[d, "\alpha_A"'] & B \arrow[r, "\pi_\Theta"'] \arrow[d, "\alpha_B"] & B/\Theta \arrow[d, dotted, "\alpha_\Theta"] \\
F(A) \arrow[r, shift left, "F(\varphi_i)"] \arrow[r, shift right, "F(\varphi_j)"'] & F(B) \arrow[r, "F(\pi_\Theta)"] & F(B/\Theta)
\end{tikzcd}
$$

Sei jetzt die Coalgebra $\mathbf{Q} = (Q, \alpha_Q)$ mit dem Homomorphismus $\psi : \mathbf{B} \to \mathbf{Q}$ ein Konkurrent von $\mathbf{B}/\Theta$, d.h. $\psi \circ \varphi_i = \psi \circ \varphi_j$ für alle $i, j \in I$. Es folgt $\Theta \subseteq \mathit{Kern}\ \psi$, so dass nach Hilfssatz 3.0.7 genau ein Homomorphismus $\chi : \mathbf{B}/\Theta \to \mathbf{Q}$ existiert mit $\psi = \chi \circ \pi_\Theta$. □

Pushouts in $\mathbf{Set}_F$

Aus Satz 2.2.4 in Verbindung mit den Hilfssätzen 3.3.1 und 3.3.5 folgt:

Hilfssatz 3.3.6 *Sei $(\varphi_i : \mathbf{A} \to \mathbf{B}_i)_{i \in I}$ eine Familie von Homomorphismen. Sei $\psi_i : B_i \to P$ der Pushout der Abbildungen φ_i in* **Set**. *Dann gibt es auf P eine eindeutige Coalgebra-Struktur α_P, so dass die ψ_i Homomorphismen nach $\mathbf{P} = (P, \alpha_P)$ sind und $\mathbf{P}$ mit den ψ_i der Pushout der φ_i in $\mathbf{Set}_F$ ist.*

Allgemeiner kann man dieses Lemma sogar für beliebige Colimiten formulieren: *Colimiten in $\mathbf{Set}_F$ werden genauso gebildet wie in* **Set**, *d.h. sie haben die gleiche Grundmenge und die kanonischen Abbildungen sind Homomorphismen.*

3.4 Bisimulationen

Der Begriff der Bisimulation ist zentral für die Theorie der Coalgebren. Bisimulationen axiomatisieren den Begriff der Ununterscheidbarkeit.

Definition 3.4.1 *Seien* **A** *und* **B** *Coalgebren und $R \subseteq A \times B$ eine zweistellige Relation. R heißt* **Bisimulation**, *wenn man auf R eine Coalgebra-Struktur ρ definieren kann, so dass die Projektionen $\pi_A : R \to A$ und $\pi_B : R \to B$ Homomorphismen sind.*

$$\begin{tikzcd} A \arrow[d, "\alpha_A"'] & R \arrow[l, "\pi_A"'] \arrow[r, "\pi_B"] \arrow[d, dashed, "\rho"'] & B \arrow[d, "\alpha_B"] \\ F(A) & F(R) \arrow[l, "F(\pi_A)"'] \arrow[r, "F(\pi_A)"] & F(B) \end{tikzcd}$$

Ein wichtiger Spezialfall einer Bisimulation ist der Graph eines Homomorphismus. Es gilt sogar:

Satz 3.4.2 *Eine Abbildung $f : A \to B$ zwischen Coalgebren $\mathbf{A} = (A, \alpha_A)$ und $\mathbf{B} = (B, \alpha_B)$ ist genau dann ein Homomorphismus, wenn ihr Graph*

$$G(f) := \{(a, f(a)) \mid a \in A\}$$

eine Bisimulation zwischen **A** *und* **B** *ist.*

Beweis Die Abbildung $\pi_A : G(f) \to A$ ist bijektiv mit inverser Abbildung π_A^{-1}. Ist $G(f)$ eine Bisimulation, so sind π_A und π_B Homomorphismen und, aufgrund von Satz 3.0.5, auch π_A^{-1}. Daher ist $f = \pi_B \circ \pi_A^{-1}$ ein Homomorphismus.

Umgekehrt, sei f als Homomorphismus vorausgesetzt Wir definieren eine Struktur-Abbildung auf $G(f)$ durch

$$\rho(a, f(a)) := F(\pi_A^{-1}) \circ \alpha_A(a),$$

das heißt

$$\rho = F(\pi_A^{-1}) \circ \alpha_A(a) \circ \pi_A.$$

Man rechnet nach:

$$F(\pi_A) \circ \rho = F(\pi_A) \circ F(\pi_A^{-1}) \circ \alpha_A \circ \pi_A = \alpha_A \circ \pi_A,$$

und

$$\begin{aligned} F(\pi_B) \circ \rho &= F(\pi_B) \circ F(\pi_A^{-1}) \circ \alpha_A \circ \pi_A \\ &= F(f) \circ \alpha_A \circ \pi_A \\ &= \alpha_B \circ f \circ \pi_A \\ &= \alpha_B \circ \pi_B. \end{aligned}$$

□

Beispiel 3.4.3 *Eine Φ-Kripke-Struktur $\mathbf{A} = (A, \xrightarrow{A}, v_A)$ fassen wir als Coalgebra vom Typ $\mathbb{P}(\Phi) \times \mathbb{P}(-)$ mit $\alpha_A(a) = (v_A(a), \xrightarrow{A}(a))$ auf. Ist $\mathbf{B} = (B, \xrightarrow{B}, v_B)$ eine weitere Kripke-Struktur und R eine Bisimulation zwischen $\mathbf{A}$ und $\mathbf{B}$, so muss für alle $(a, b) \in R$ eine Menge $\Gamma \subseteq \Phi$ von Propositionen und eine Menge $M \subseteq R$ existieren mit*

(i) $v_A(a) = \Gamma = v_B(b)$,

(ii) $\pi_A[M] = \xrightarrow{A}(a)$ *und*

(iii) $\pi_B[M] = \xrightarrow{B}(b)$.

Falls $a\ R\ b$ besagt die erste Bedingung gerade

$$v_A(a) = v_B(b),$$

während "$\supseteq$" in Bedingung (ii) kombiniert mit "$\subseteq$" in (iii) besagt:

$$a \xrightarrow{A} a' \implies \exists b'. b \xrightarrow{B} b' \wedge a' R b'$$

analog erhält man aus den umgekehrten Inklusionen:

$$b \xrightarrow{B} b' \implies \exists a'. a \xrightarrow{A} a' \wedge a' R b'.$$

Somit deckt sich der gerade eingeführte Begriff der Bisimulation mit dem auf Seite 12 motivierten Begriff.

Für den folgenden Charakterisierungs-Satz benötigen wir – zum ersten und einzigen Mal in diesem Kapitel – das Auswahlaxiom (siehe Hilfssatz 2.3.1):

Satz 3.4.4 *Seien $\mathbf{A}$ und $\mathbf{B}$ Coalgebren. Für eine Coalgebra $\mathbf{P}$ seien Homomorphismen $\varphi_A : \mathbf{P} \to \mathbf{A}$ und $\varphi_B : \mathbf{P} \to \mathbf{B}$ gegeben. Dann ist*

$$(\varphi_A, \varphi_B)[P] := \{(\varphi_A(p), \varphi_B(p)) \mid p \in P\}$$

eine Bisimulation zwischen $\mathbf{A}$ und $\mathbf{B}$, und jede Bisimulation ist von dieser Form.

Beweis Die letzte Behauptung ist klar, denn jede Bisimulation R erlaubt eine Coalgebra-Struktur $\mathbf{R}$ so dass π_A^R und π_A^R Homomorphismen sind. Offensichtlich gilt $(\pi_A^R, \pi_B^R)[R] = R$.

Sei jetzt $\mathbf{P}$ mit Homomorphismen $\varphi_A : \mathbf{P} \to \mathbf{A}$ und $\varphi_B : \mathbf{P} \to \mathbf{B}$ gegeben. Für die surjektive Abbildung $(\varphi_A, \varphi_B) : P \to (\varphi_A, \varphi_B)[P]$ liefert das Auswahlaxiom eine rechtsinverse Abbildung μ. Auf $(\varphi_A, \varphi_B)[P] \subseteq A \times B$ definieren wir die Struktur-Abbildung

$$\rho := F((\varphi_A, \varphi_B)) \circ \alpha_P \circ \mu.$$

Wir müssen zeigen, dass π_A und π_B Homomorphismen sind:

$$\begin{aligned}
F(\pi_A) \circ \rho &= F(\pi_A) \circ F((\varphi_A, \varphi_B)) \circ \alpha_P \circ \mu \\
&= F(\pi_A \circ (\varphi_A, \varphi_B)) \circ \alpha_P \circ \mu \\
&= F(\varphi_A) \circ \alpha_P \circ \mu \\
&= \alpha_A \circ \varphi_A \circ \mu \\
&= \alpha_A \circ \pi_A \circ (\varphi_A, \varphi_B) \circ \mu \\
&= \alpha_A \circ \pi_A.
\end{aligned}$$

Analog folgt: $F(\pi_B) \circ \rho = \alpha_B \circ \pi_B$, somit ist $(\varphi_A, \varphi_B)[P]$ eine Bisimulation. $\Box$

In der universellen Algebra sind verträgliche Relationen gerade die Unteralgebren des kartesischen Produktes. Im Falle der Coalgebren haben wir auf dem kartesischen Produkt keine kanonische Struktur zur Verfügung, dennoch verhalten sich Bisimulationen in vielen Fällen wie "zweidimensionale Unter-Coalgebren". So sind Bisimulationen auch gegen Vereinigungen abgeschlossen.

Satz 3.4.5 *Die Vereinigung $\bigcup_{i\in I} R_i$ von Bisimulationen R_i zwischen $\mathbf{A}$ und $\mathbf{B}$ ist eine Bisimulation.*

Beweis Sei $(R_i)_{i\in I}$ eine Familie von Bisimulationen zwischen $\mathbf{A}$ und $\mathbf{B}$ und $\mathbf{R} := \sum_{i\in I} \mathbf{R}_i$ die Summe der Coalgebren $\mathbf{R}_i$. Für jedes $i \in I$ sind $\pi_A^i : \mathbf{R}_i \to \mathbf{A}$ und $\pi_B^i : \mathbf{R}_i \to \mathbf{B}$ Homomorphismen. Damit werden $\mathbf{A}$ und $\mathbf{B}$ zu Konkurrenten der Summe und es gibt eindeutige Homomorphismen $\pi_A : \mathbf{R} \to \mathbf{A}$ und $\pi_B : \mathbf{R} \to \mathbf{B}$ mit

$$\pi_A \circ e_i = \pi_A^i \text{ und } \pi_B \circ e_i = \pi_B^i$$

für alle $i \in I$. Es gilt nun:

$$\begin{aligned}
(\pi_A, \pi_B)[R] &= \{(\pi_A(i,x), \pi_B(i,x)) \mid i \in I, x \in R_i\} \\
&= \bigcup_{i\in I} \{((\pi_A \circ e_i)(x), (\pi_B \circ e_i)(x)) \mid x \in R_i\} \\
&= \bigcup_{i\in I} \{(\pi_A^i(x), \pi_B^i(x)) \mid x \in R_i\} \\
&= \bigcup_{i\in I} R_i
\end{aligned}$$

Nach Satz 3.4.4 ist $(\pi_A, \pi_B)[R]$ eine Bisimulation, also auch $\bigcup_{i\in I} R_i$. $\Box$

Folgerung 3.4.6 *Seien* $\mathbf{A}$ *und* $\mathbf{B}$ *Coalgebren und* $R \subseteq A \times B$ *eine Relation. Dann gibt eine größte Bisimulation* $[R]$*, welche in* R *enthalten ist. Wir nennen diese den* **Bisimulationskern** *von* R*.*

Wichtig ist der Spezialfall $R = A \times B$:

Folgerung 3.4.7 *Zwischen zwei Coalgebren* $\mathbf{A}$ *und* $\mathbf{B}$ *gibt es immer eine größte Bisimulation.*

Definition 3.4.8 *Die größte Bisimulation zwischen zwei Coalgebren* $\mathbf{A}$ *und* $\mathbf{B}$ *bezeichnen wir mit* $\sim_{A,B}$*, bzw. mit* $\sim_A$*, falls* $\mathbf{A} = \mathbf{B}$*. Zwei Punkte* $a \in A$*,* $b \in B$ *heißen* **bisimilar***, falls* $(a, b) \in \sim_{A,B}$*.*

Dei größte Bisimulation stellt gerade die Ununterscheidbarkeits-Relation dar, die in der Einführung zu diesem Kapitel mehrfach angesprochen wurde. Für diese Interpretation können wir jetzt aufgrund von Satz 3.4.4 charakterisieren, wann zwei Punkte ununterscheidbar sind:

Satz 3.4.9 *Zwei Punkte* $a \in A$ *und* $b \in B$ *zweier* F*-Coalgebren* $\mathbf{A}$ *und* $\mathbf{B}$ *sind genau dann bisimilar, wenn es eine* F*-Coalgebra* $\mathbf{P}$ *gibt, Homomorphismen* $\varphi : \mathbf{P} \to \mathbf{A}$ *und* $\psi : \mathbf{P} \to \mathbf{A}$*, sowie einen Punkt* $p \in P$ *mit* $\varphi(p) = a$ *und* $\psi(p) = b$*.*

Kurz gesagt: Zwei Punkte sind ununterscheidbar, wenn sie von einem gemeinsamen Punkt als "homomorphe Bilder" gewonnen werden können.

3.5 Epis und Monos in $\mathbf{Set}_F$

In jeder Kategorie gilt, dass ein Morphismus $\varphi : \mathbf{A} \to \mathbf{B}$ genau dann epi ist, wenn das folgende ein Pushout-Diagramm ist (siehe Seite 22):

$$\begin{tikzcd} A \arrow[r, "\varphi"] \arrow[d, "\varphi"'] & B \arrow[d, "id_B"] \\ B \arrow[r, "id_B"'] & B \end{tikzcd}$$

Da der Pushout in der Kategorie $\mathbf{Set}_F$ gleichzeitig der Pushout in der Kategorie $\mathbf{Set}$ ist, gilt auch in $\mathbf{Set}_F$:

Folgerung 3.5.1 *Ein Homomorphismus ist genau dann epi wenn er surjektiv ist.*

Ein analoges Resultat für Monos kann man nicht erwarten. Dies zeigt das folgende Beispiel:

Beispiel 3.5.2 *Auf der zwei-elementigen Menge* $\{0, 1\}$ *betrachten wir eine Coalgebra-Struktur zum Funktor* $(-)^3_2$ *von Beispiel 2.4.7. Wir setzen:* $\alpha(0) := (0, 0, 1)$ *und* $\alpha(1) = (1, 0, 0)$*. Die Abbildung* $\tau : \{0, 1\} \to \{\star\}$ *ist ein surjektiver Homomorphismus auf die ein-elementige* $(-)^3_2$*-Coalgebra. Wir behaupten, dass* τ *mono ist.*

Angenommen, es gäbe Homomorphismen $\varphi, \psi : \mathbf{A} \to \{0, 1\}$ *mit* $\varphi \neq \psi$*, dann gäbe es ein* $a \in \mathbf{A}$ *mit* $\varphi(a) = 0$ *und* $\psi(a) = 1$*. Für* $(x, y, z) := \alpha_A(a)$ *rechnet man nun aus:* $(\varphi(x), \varphi(y), \varphi(z)) = ((\varphi)^3_2 \circ \alpha)(a) = (\alpha \circ \varphi)(a) = (0, 0, 1)$ *und analog* $(\psi(x), \psi(y), \psi(z)) = (1, 0, 0)$*. Es folgt* $x \neq z$*,* $y \neq z$ *und* $x \neq y$*, im Widerspruch zu* $(x, y, z) \in (A)^3_2$*.*

Eine Charakterisierung von Monomorphismen gelingt mit dem folgenden Ergebnis:

Satz 3.5.3 *Ein Homomorphismus* $\varphi : \mathbf{A} \to \mathbf{B}$ *ist genau dann mono, wenn* $[Ker\ \varphi] = \Delta$, *d.h. wenn der Kern von* φ *keine nicht-triviale Kongruenz enthält.*

Beweis $[Ker\ \varphi]$ ist eine Bisimulation, d.h. die Projektionen $\pi_1, \pi_2 : [Ker\ \varphi] \to \mathbf{A}$ sind Homomorphismen und $\varphi \circ \pi_1 = \varphi \circ \pi_2$. Ist φ mono, so folgt $\pi_1 = \pi_2$, also $[Ker\ \varphi] = \Delta$.

Sei nun $[Ker\ \varphi] = \Delta$ und seien $\psi_1, \psi_2 : \mathbf{Q} \to \mathbf{A}$ Homomorphismen mit $\varphi \circ \psi_1 = \varphi \circ \psi_2$. Dies bedeutet $(\psi_1, \psi_2)[Q] \subseteq Ker\ \varphi$. Aus Satz 3.4.4 folgt sogar $(\psi_1, \psi_2)[Q] \subseteq [Ker\ \varphi]$. Wegen $[Kern\ \varphi] = \Delta$ somit $\psi_1 = \psi_2$. $\Box$

3.6 Kongruenzen

Kongruenzen sind als Kerne von Homomorphismen definiert. Insbesondere sind sie Äquivalenzrelationen. Wie in der universellen Algebra ist Δ_A immer eine Kongruenz auf $\mathbf{A}$, nicht aber die All-Relation $A \times A$, wie man sich schon an dem Beispiel der folgenden Kripke-Struktur leicht überlegen kann.

Auch der mengentheoretische Schnitt zweier Kongruenzen muss keine Kongruenz sein. Glücklicherweise existiert aber stets das Supremum einer Menge von Kongruenzen, und dieses Supremum ist identisch mit dem Supremum im Verband aller Äquivalenzrelationen: Es ist die transitive Hülle der Vereinigung, $(\bigcup_{i\in I} \theta_i)^*$. Daher bilden auch die Kongruenzen einer Coalgebra $\mathbf{A}$ einen Verband $Con(\mathbf{A})$.

Satz 3.6.1 *Sei* $(\theta_i)_{i\in I}$ *eine Familie von Kongruenzen auf der Coalgebra* $\mathbf{A}$. *Dann ist* $\Theta := (\bigcup_{i\in I} \theta_i)^*$ *das Supremum der* θ_i.

Beweis Nach Hilfssatz 2.3.5 ist die Familie $(\pi_i : A/\theta_i \to A/\Theta)_{i\in I}$ der Pushout der $(\pi_{\theta_i})_{i\in I}$. Somit ist es auch der Pushout in der Kategorie $\mathbf{Set}_F$. Es folgt dass $\pi_\Theta = \pi_i \circ \pi_{\theta_i}$ ein Homomorphismus ist, sein Kern Θ also eine Kongruenz. $\Box$

Folgerung 3.6.2 *Die Kongruenzen einer Coalgebra bilden einen vollständigen Verband. Das größte Element dieses Verbandes nennen wir* ∇_A *oder einfach* ∇. *Im Allgemeinen kann* $\nabla_A \neq A \times A$ *sein.*

Auf dem Faktor $\mathbf{A}/\nabla$ gibt es keine echte Kongruenz mehr, man könnte $\mathbf{A}/\nabla$ daher als **einfach** bezeichnen. Es gilt insbesondere:

Hilfssatz 3.6.3 *Jeder Homomorphismus* $\varphi : \mathbf{A}/\nabla \to \mathbf{B}$ *ist injektiv.*

Definition 3.6.4 *Punkte* a *und* b *einer Coalgebra heißen* **beobachtungs-äquivalent**, *falls es Homomorphismen* φ, ψ *gibt mit* $\varphi(a) = \psi(b)$.

Hilfssatz 3.6.5 *Zwei Punkte* a *und* b *einer Coalgebra* $\mathbf{A}$ *sind genau dann beobachtungs-äquivalent, wenn sie kongruent modulo* ∇_A *sind.*

Beweis Zu $\varphi, \psi : \mathbf{A} \to \mathbf{B}$ sei $\phi : \mathbf{B} \to \mathbf{C}$ der Coequalizer. Aus $\varphi(a) = \psi(b)$ folgt $(a, b) \in Kern\ \phi \circ \varphi \subseteq \nabla$. $\Box$

Man kann den Begriff der Beobachtungs-Äquivalenz auch für Punkte $a \in \mathbf{A}$, $b \in \mathbf{B}$ verschiedener Coalgebren ausdehnen. Es zeigt sich, dass a und b genau dann beobachtungs-äquivalent sind, wenn $(a, b) \in \nabla_{\mathbf{A}+\mathbf{B}}$.

Kongruenzen und Bisimulationen

Es liegt nahe, die größte Bisimulation mit der größten Kongruenz zu vergleichen:

Hilfssatz 3.6.6 *Ist R eine Bisimulation auf $\mathbf{A}$, so ist die kleinste R umfassende Äquivalenzrelation schon eine Kongruenz.*

Beweis Auf R gibt es eine Struktur-Abbildung α_R, so dass $\mathbf{R} = (R, \alpha_R)$ eine Coalgebra ist und $\pi_1, \pi_2 : \mathbf{R} \to \mathbf{A}$ Homomorphismen sind. Als Abbildungen von Mengen haben diese den Coequalizer $\pi_\theta : A \to A/\theta$, wobei θ nach Hilfssatz 2.3.4 die von den Paaren $(\pi_1(r), \pi_2(r))$, also von R, erzeugte Äquivalenzrelation ist.

Wegen Hilfssatz 3.3.5 ist π_θ ein Homomorphismus, θ daher eine Kongruenz. $\square$

Da die größte Bisimulation $\sim_A$ auf einer Coalgebra $\mathbf{A}$ automatisch reflexiv und symmetrisch ist, folgt für die transitive Hülle $(\sim_A)^*$:

Folgerung 3.6.7 $(\sim_A)^*$ *ist eine Kongruenz, insbesondere* $\sim_A \subseteq \nabla_A$.

3.7 Covarietäten

Sei $\mathcal{K}$ eine Klasse von F-Coalgebren. Definiere

$I(\mathcal{K})$ – die Klasse aller isomorphen Kopien

$H(\mathcal{K})$ – die Klasse aller homomorphen Bildern

$S(\mathcal{K})$ – die Klasse aller Untercoalgebren

$\Sigma(\mathcal{K})$ – die Klasse aller Summen

von Coalgebren aus $\mathcal{K}$. Eine **Covarietät** sei eine Klasse von Coalgebren, die unter den Operatoren H, S und Σ abgeschlossen ist.

Alle eingeführten Operatoren sind offensichtlich Hüllenoperatoren und für jede Klasse $\mathcal{K}$ hat man schnell die Inklusionen:

$$HS(\mathcal{K}) \subseteq SH(\mathcal{K})$$

$$H\Sigma(\mathcal{K}) \subseteq \Sigma H(\mathcal{K})$$

$$\Sigma S(\mathcal{K}) \subseteq S\Sigma(\mathcal{K})$$

Im letzten Falle kann man sogar die Gleichheit $\Sigma S(\mathcal{K}) = S\Sigma(\mathcal{K})$ zeigen, wir werden diese Tatsache aber nicht benötigen. Im Vergleich zur Universellen Algebra drehen sich die Operatoren S und H um:

Satz 3.7.1 *Zu jeder Klasse $\mathcal{K}$ von Coalgebren ist $SH\Sigma(\mathcal{K})$ die kleinste $\mathcal{K}$ umfassende Covarietät.*

3.8 Aufgaben

(i) Sei $\varphi : \mathbf{A} \to \mathbf{B}$ nicht surjektiv. Finden Sie eine Coalgebra $\mathbf{C}$ und Homomorphismen $\psi, \phi : \mathbf{B} \to \mathbf{C}$ mit $\psi \neq \phi$, aber $\psi \circ \varphi = \phi \circ \varphi$.

(ii) Sei ∇ die grösste Kongruenz auf $\mathbf{A}$. Für jede Coalgebra $\mathbf{Q}$ gibt es höchstens einen Homomorphismus $\varphi : \mathbf{Q} \to \mathbf{A}/\nabla$.

(iii) Für die zwei-elementige $(-)^3_2$-Coalgebra von Beispiel 2.4.7 gilt: $\sim\; \neq\; \nabla$.

(iv) Finden Sie ein Beispiel zweier Bisimulationen R und S, so dass $R \circ S$ keine Bisimulation ist.

(v) Für jede Klasse $\mathcal{K}$ gilt: $S\Sigma(\mathcal{K}) = \Sigma S(\mathcal{K})$.

(vi) Definieren Sie das Produkt von Coalgebren, indem Sie die Pfeile in Definition 2.2.1 umdrehen. Seien $\mathbf{A}$, $\mathbf{B}$ die $\mathbb{P}$-Coalgebren

$$\mathbf{A} = \bullet \circlearrowright \qquad \mathbf{B} = \bullet$$

Zeigen Sie, dass das Produkt $\mathbf{A} \times \mathbf{B}$ in $\mathbf{Set}_{\mathbb{P}}$ die leere $\mathbb{P}$-Coalgebra ist.

4 Terminale Coalgebren

In der universellen Algebra spielen die freien Algebren eine zentrale Rolle. Von der freien Algebra über der leeren Erzeugendenmenge gibt es genau einen Homomorphismus in jede Algebra des gleichen Typs. Sie ist daher ein *initiales Objekt*.

In der universellen Coalgebra haben die *terminalen* Coalgebren, sofern sie existieren, eine ähnlich herausragende Bedeutung. Man kann ihre Elemente als "Verhalten" interpretieren. Jedes mögliche "Verhalten" eines beliebigen Punktes einer beliebigen Coalgebra ist genau einmal in der terminalen Coalgebra vorhanden.

Die terminale Coalgebra kann man als *cofreie Coalgebra* über einer ein-elementigen Menge von Covariablen verstehen. Die Elemente beliebiger cofreier Coalgebren entsprechen dann "Verhaltensmustern".

Definition 4.0.1 *Eine F-Coalgebra $\mathbf{T}$ heißt* **terminal**, *wenn für jede F-Coalgebra $\mathbf{A}$ genau ein Homomorphismus $\tau : \mathbf{A} \to \mathbf{T}$ existiert.*

Offensichtlich sind terminale Coalgebren, sofern sie existieren, bis auf Isomorphie eindeutig bestimmt.

Verhalten

Die zentrale Rolle, die terminale Coalgebren spielen, wird an dem folgenden Satz deutlich:

Satz 4.0.2 *Sei $\mathbf{T}$ die terminale F-Coalgebra. Zu jeder Coalgebra $\mathbf{A} \in \mathbf{Set}_F$ und jedem $a \in A$ gibt es genau ein Element $t \in T$ mit $a \sim t$.*

Beweis Zu $\mathbf{A}$ gibt es einen Homomorphismus $\tau : \mathbf{A} \to \mathbf{T}$. Nach Satz 3.4.2 ist $a \sim \tau(a) \in T$. Angenommen es gäbe ein weiteres Element $t' \in T$ mit $a \sim t'$. Nach 3.4.9 würde dann eine Coalgebra $\mathbf{P}$ existieren zwei Homomorphismen $\varphi : \mathbf{P} \to \mathbf{A}$ und $\psi : \mathbf{P} \to \mathbf{T}$ sowie ein Element $p \in P$ mit $\varphi(p) = a$ und $\psi(p) = t'$. Mit ψ und $\tau \circ \varphi$ haben wir dann aber zwei Homomorphismen von $\mathbf{P}$ nach $\mathbf{T}$. Es folgt $\psi = \tau \circ \varphi$, also auch $t' = \psi(p) = \tau \circ \varphi(p) = \tau(a) = t$. $\square$

Wenn wir also bisimilaren Zuständen das gleiche Verhalten zuschreiben, kann man die terminale Coalgebra als die Menge aller "Verhaltensweisen" auffassen. Zu allen denkbaren Zuständen enthält sie bis auf Bisimilarität genau einen Repräsentanten. Allerdings soll diese Sprechweise nicht verdecken, dass Bisimilarität im Allgemeinen keine Äquivalenzrelation sein muss.

Ist R irgendeine Bisimulation auf der terminalen Coalgebra, so folgt für $\pi_1, \pi_2 : \mathbf{R} \to \mathbf{T}$, dass $\pi_1 = \pi_2$ ist, also $R = \Delta$. Dies kann man als Extensionalitätsprinzip deuten: "Zwei Elemente, die nicht unterscheidbar sind, sind gleich." Als Beweisregel formuliert bezeichnet man es als **Coinduktionsprinzip**:

$$\frac{x \sim y}{x = y}.$$

4.1 Terminale Automaten

Zu fest gewählten Mengen D und Σ betrachten wir den Funktor $F(-) := D \times (-)^\Sigma$. Es handelt sich um das Produkt des konstanten Funktors D mit dem Potenz-Funktor $(-)^\Sigma$, siehe Seite 20. F-Coalgebren sind Automaten mit Output-Datenmenge D und Input-Alphabet Σ (Seiten 7, 12). Einer Coalgebra-Struktur $\alpha : S \to D \times S^\Sigma$ entspricht gerade der Automat $\mathbf{S} = (S, \delta, \gamma)$ mit

(i) $\gamma(s) = \pi_1(\alpha(s))$, und

(ii) $\delta(s, e) := (\pi_2(\alpha(s)))(e)$.

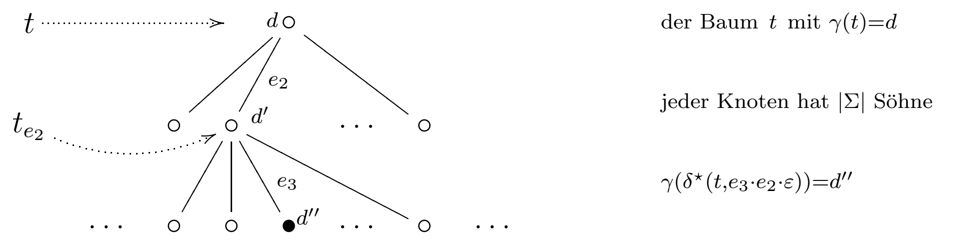
Mit dieser Übersetzung stimmen auch Automaten-Homomorphismen, die gewöhnlich durch die Bedingungen

(i) $\gamma(\varphi(a)) = \gamma(a)$

(ii) $\delta(\varphi(a), e) = \varphi(\delta(a, e))$.

definiert werden, mit den Coalgebra-Homomorphismen überein.

Der terminale Automat $\mathbf{T}$ hat als Zustands-Menge alle unendlichen Bäume t, deren Knoten mit Elementen aus D beschriftet sind, und jeweils genau $|\Sigma|$ viele Nachfolger haben. Dann ist $\gamma(t)$ die Beschriftung der Wurzel von t und $\delta(t, e)$ der Unterbaum t_e von t, dessen Wurzel, gerade der e-te Sohn von t ist.



Jeder Knoten k ist eindeutig durch den Pfad beschrieben, der von k zur Wurzel führt. Dieser Weg entspricht einem Wort $w \in \Sigma^*$. Daher entsprechen die D-beschrifteten Σ-verzweigenden Bäume genau den Abbildungen $\tau : \Sigma^* \to D$. Die Grundmenge von T ist also D^{Σ^*}.

Satz 4.1.1 *Der terminale Automat mit Eingabe-Alphabet Σ und Ausgabe-Menge D ist $\mathbf{T} = (D^{\Sigma^*}, \delta, \gamma)$ mit $\delta(\tau, e)(w) = \tau(e \cdot w)$ und $\gamma(\tau) = \tau(\varepsilon)$.*

Beweis Sei $\mathbf{A} = (A, \delta', \gamma')$ ein beliebiger Automat mit Input-Alphabet Σ und Output-Menge D. Ein Homomorphismus $\varphi : \mathbf{A} \to \mathbf{T}$ muss

(i) $\varphi(a)(\varepsilon) = \gamma(\varphi(a)) = \gamma'(a)$ und

(ii) $\varphi(a)(e \cdot w) = \delta(\varphi(a), e)(w) = \varphi(\delta'(a, e))(w)$

erfüllen. Umgekehrt wird durch (i) und (ii) eine eindeutige Abbildung $\varphi : A \to T$ für alle a induktiv über den Aufbau der Worte $w \in \Sigma^*$ definiert. $\square$

Im Spezialfall $D = \mathbf{2} = \{0, 1\}$ kann man die Ausgabefunktion γ als charakteristische Funktion für die Menge der akzeptierenden Zustände ansehen. Die Elemente von $\mathbf{2}^{\Sigma^*}$ sind gerade die Teilmengen von Σ^*, also die Sprachen über Σ. Jedes $t \in T$ entspricht der Sprache $\mathcal{L}(\mathbf{T}, t)$, siehe Seite 9 und die Transitionsfunktion δ des terminalen Automaten entspricht genau der Ableitung, d.h.

$$\delta(t, e) = t' \iff \mathcal{L}_e(\mathbf{T}, t) = \mathcal{L}(\mathbf{T}, t').$$

4.2 Existenz terminaler Coalgebren

Nicht für jeden Funktor F existiert eine terminale F-Coalgebra. Dies ist eine direkte Konsequenz aus dem folgenden Satz, der als “Lambek’s Lemma” [Lam68] bekannt ist:

Hilfssatz 4.2.1 (Lambek) *Ist $\mathbf{T}$ terminale F-Coalgebra, so ist die Struktur-Abbildung $\alpha : T \to F(T)$ bijektiv.*

Beweis Auf der Menge $F(T)$ haben wir eine natürliche F-Coalgebra-Struktur $F(\alpha) : F(T) \to F(F(T))$. Trivialerweise ist α auch ein Homomorphismus von $\mathbf{T}$ in diese Coalgebra $F(\mathbf{T}) = (F(T), F(\alpha))$.

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

Da $\mathbf{T}$ terminal ist, existiert genau ein Homomorphismus $\beta : F(\mathbf{T}) \to \mathbf{T}$. Von $\mathbf{T}$ nach $\mathbf{T}$ haben wir jetzt aber zwei Homomorphismen, $\beta \circ \alpha$ und id_T, so dass $\beta \circ \alpha = id_T$ aus der Eindeutigkeit folgt. Weil β ein Homomorphismus ist, gilt

$$\alpha \circ \beta = F(\beta) \circ F(\alpha) = F(\beta \circ \alpha) = F(id_T) = id_{F(T)}.$$

$\square$

Da es keine Bijektion zwischen einer Menge T und ihrer Potenzmenge $\mathbb{P}(T)$ gibt, folgt insbesondere:

Folgerung 4.2.2 *Es gibt keine terminale Kripke-Struktur.*

4.3 Schwach Terminale Coalgebren

Definition 4.3.1 *Eine F-Coalgebra $\mathbf{W}$ heißt* **schwach terminal***, falls es zu jeder F-Coalgebra $\mathbf{A}$ mindestens einen Homomorphismus $\varphi : \mathbf{A} \to \mathbf{W}$ gibt.*

Hilfssatz 4.3.2 *Sei $\mathbf{W}$ schwach terminal, dann ist $\mathbf{W}/\nabla$, der Faktor nach der größten Kongruenz, terminal.*

Beweis Für jede F-Coalgebra $\mathbf{A}$ haben wir einen Homomorphismus $\varphi : \mathbf{A} \to \mathbf{W}$, also auch einen Homomorphismus $\pi_\nabla \circ \varphi : \mathbf{A} \to \mathbf{W}/\nabla$.

Seien $\varphi_1, \varphi_2 : \mathbf{A} \to \mathbf{W}/\nabla$ Homomorphismen, so bilden wir den Coequalizer ψ von φ_1 und φ_2. Nach Hilfssatz 3.6.3 ist ψ injektiv. Wegen $\psi \circ \varphi_1 = \psi \circ \varphi_2$ folgt $\varphi_1 = \varphi_2$. $\square$

$$\begin{tikzcd}
& W \arrow[d, "\pi_\nabla"] & \\
A \arrow[r, shift left, "\varphi_1"] \arrow[r, shift right, "\varphi_2"'] & W/\nabla \arrow[r, "\psi"] & B
\end{tikzcd}$$

Hilfssatz 4.3.3 *Seien $F, G : \mathbf{Set} \to \mathbf{Set}$ Mengenfunktoren und $\eta : G \dot{\twoheadrightarrow} F$ eine surjektive natürliche Transformation. Ist $\mathbf{T} = (T, \pi)$ schwach terminale G-Coalgebra, so ist $\mathbf{T}_\eta = (T, \eta_T \circ \pi)$ schwach terminale F-Coalgebra.*

Beweis Sei $\mathbf{A} = (A, \alpha)$ eine nicht-leere F-Coalgebra. Das Auswahlaxiom liefert ein $h : F(A) \to G(A)$ mit $\eta_A \circ h = id_{F(A)}$. Von der G-Coalgebra $\mathbf{A}_G := (A, h \circ \alpha_A)$ gibt es dann einen G-Homomorphismus $\varphi : \mathbf{A}_G \to \mathbf{T}$. Es folgt

$$F(\varphi) \circ \alpha = F(\varphi) \circ \eta_A \circ (h \circ \alpha) = \eta_T \circ G(\varphi) \circ (h \circ \alpha) = (\eta_T \circ \pi) \circ \varphi.$$

$$\begin{tikzcd}
A \arrow[rr, "\varphi"] \arrow[dr, dashed] \arrow[dd, "\alpha"'] & & T \arrow[dr, "\pi"] \arrow[dd, dashed] & \\
& G(A) \arrow[rr, "G\varphi"] \arrow[dl, "\eta_A"] & & G(T) \arrow[dl, "\eta_T"] \\
F(A) \arrow[ur, dotted, bend left, "h"] \arrow[rr, "F\varphi"] & & F(T) &
\end{tikzcd}$$

so dass $\varphi : \mathbf{A} \to \mathbf{T}_\eta$ ein F-Homomorphismus ist. $\square$

4.4 Beschränkte Funktoren

Schränkt man den Potenzmengen-Funktor ein, indem man nur die Teilmengen unterhalb einer festen Kardinalität κ betrachtet

$$\mathbb{P}_\kappa(X) := \{U \subseteq X \mid |U| < \kappa\},$$

so gibt es wieder eine terminale Coalgebra.

Für die Praxis ist der Fall $\kappa = \omega$, die Kardinalzahl der natürlichen Zahlen, von Bedeutung: Kripke-Strukturen, in denen jeder Punkt nur endlich viele Nachfolger hat, heißen

bildendlich. Im Zusammenhang mit verteilten Programmen spricht man von "beschränktem Nondeterminismus". Bild-endliche Kripke-Strukturen sind sind die Coalgebren des Funktors $F(-) := \mathbb{P}(\Phi) \times \mathbb{P}_\omega(-)$. Die Existenz einer terminalen bild-endlichen Kripke-Struktur, allgemeiner einer terminalen $\mathbb{P}_\kappa$-Coalgebra wird aus den Ergebnissen dieses Abschnittes folgen.

Definition 4.4.1 *Sei κ eine Kardinalzahl. Ein Mengen-Funktor F heißt* **κ-beschränkt** *, falls es zu jedem Element a einer F-Coalgebra $\mathbf{A}$ eine Untercoalgebra $\mathbf{U} \leq \mathbf{A}$ gibt mit $a \in U$ und $|U| < \kappa$.*

F heißt **schwach κ-beschränkt***, wenn jede nicht-leere F-Coalgebra A eine nicht-leere Untercoalgebra U mit $|U| < \kappa$ besitzt.*

Wir sagen, dass F **beschränkt***, bzw.* **schwach beschränkt** *ist, wenn es ein κ gibt, so dass F κ-beschränkt, bzw. schwach κ-beschränkt ist.*

Hilfssatz 4.4.2 *Der Funktor $(-)^\Sigma$ ist $|\Sigma|^*$-beschränkt. Sei $F : \mathbf{Set} \to \mathbf{Set}$ κ-beschränkt, dann gilt:*

(i) für jede feste Menge D ist $D \times F(-)$ ebenfalls κ-beschränkt.

(ii) Ist $\eta : F \stackrel{\cdot}{\to} G$ eine surjektive natürliche Transformation, dann ist G auch κ-beschränkt.

Bemerkung 4.4.3 *$\mathbb{P}_\kappa(-)$ ist κ^*-beschränkt mit $\kappa^* := max\{\kappa, \omega\}$; aufgrund des Hilfssatzes auch der Funktor $\mathbb{P}(\Psi) \times \mathbb{P}(-)_\kappa$. Dessen Coalgebren sind Ψ-Kripke-Strukturen, in denen jeder Punkt a weniger als κ viele Nachfolger hat. Ist R^* die reflexiv transitive Hülle von R, so ist $R^*(a) := \{b \mid aR^*b\}$ eine Unter-Coalgebra der Mächtigkeit κ^*.*

Satz 4.4.4 ([GS00]) *Für einen Funktor $F : \mathbf{Set} \to \mathbf{Set}$ sind äquivalent:*

(i) F ist beschränkt.

(ii) F ist schwach beschränkt.

(iii) Es gibt Mengen D und Σ und eine surjektive natürliche Transformation $\eta : D \times (-)^\Sigma \stackrel{\cdot}{\twoheadrightarrow} F$.

Beweis Jeder beschränkte Funktor ist schwach beschränkt. Ist $\eta : D \times (-)^\Sigma \stackrel{\cdot}{\twoheadrightarrow} F$ eine surjektive natürliche Transformation, so ist F aufgrund von Hilfssatz 4.4.2 beschränkt.

Es bleibt daher nur $(ii) \to (iii)$ zu zeigen. Sei F also schwach κ-beschränkt. Wir setzen $D := F(\kappa)$ und $\Sigma = \kappa$. Der Funktor $G := D \times (-)^\kappa$ transformiert eine Abbildung $f : X \to Y$ in die Abbildung $Gf : D \times X^\kappa \to D \times Y^\kappa$ mit $(Gf)(d, \tau) := (d, f \circ \tau)$.

Für jede Menge X setzen wir $\eta_X(d, \tau) := (F\tau)(d)$. Man rechnet nach, dass η eine natürliche Transformation wird:

$$(Ff \circ \eta_X)(d, \tau) = (Ff)((F\tau)(d)) = (F(f \circ \tau))(d) = \eta_Y(d, f \circ \tau) = (\eta_Y \circ (Gf))(d, \tau).$$

Die Beschränktheit des Funktors F benötigen wir nur, um die Surjektivität von η_X für $X \neq \emptyset$ zu zeigen. Sei dazu $w \in F(X)$ beliebig gewählt. Wir benötigen ein $u \in F(\kappa)$ und ein $\tau \in X^\kappa$ mit $\eta_X(u, \tau) = w$.

Die konstante Abbildung $\alpha_w : X \to F(X)$ mit $\alpha_w(x) := w$ für alle $x \in X$ definiert eine Coalgebra $\mathbf{X} = (X, \alpha_w)$ auf X. Da F schwach κ-beschränkt ist, finden wir eine nichtleere Unteralgebra $S \leq \mathbf{X}$ und eine surjektive Abbildung $\gamma : \kappa \twoheadrightarrow S$. Setze $\tau := \subseteq_S^X \circ \gamma$.

$$\begin{tikzcd}
\kappa \arrow[r, two heads, "\gamma"] & S \arrow[r, hook] \arrow[d, "\alpha_S"] & X \arrow[d, "\alpha_w"] \\
F(\kappa) \arrow[r, two heads, "F\gamma"] & F(S) \arrow[r, hook] & F(X)
\end{tikzcd}$$

Für ein beliebiges $s \in S$ sei $v := \alpha_S(s) \in F(S)$. Mit Hilfe des Auswahlaxioms ist $F(\gamma)$ surjektiv (Hilfssatz 2.3.1), wir finden daher ein $u \in F(\kappa)$ mit $(F\gamma)(u) = v$. Wir rechnen nach:

$$\eta_X(u, \tau) = (F\tau)(u) = (F(\subseteq_S^X \circ \gamma))(u) = ((F \subseteq_S^X) \circ (F\gamma))(u) = (F \subseteq_S^X)(v) = w.$$

□

Wenn wir dieses Ergebnis mit Satz 4.1.1 und Lemma 4.3.3 kombinieren, erhalten wir:

Satz 4.4.5 *Ist F ein (schwach) κ-beschränkter Funktor, so ist die Menge T_F^κ aller unendlichen κ-verzweigenden Bäume, deren Knoten mit Elementen aus $F(\kappa)$ beschriftet sind, die Grundmenge einer schwach terminalen F-Coalgebra. Die Struktur-Abbildung ist gegeben durch*

$$\alpha(t) = (F(\vartheta(t)))(\gamma(t))$$

wobei $\gamma(t)$ die Beschriftung der Wurzel von t ist, und $\vartheta(t)$ die Abbildung, die jedem $e \in \kappa$ den e-ten Sohn t_e von t zuordnet.

Folgerung 4.4.6 *Ist F (schwach) κ-beschränkt, so existiert die terminale F-Coalgebra.*

Als Beispiel betrachten wir den Funktor $\mathbb{P}_\omega$. Eine schwach terminale Coalgebra besteht aus allen ω-verzweigenden unendlichen Bäumen $T_{\mathbb{P}_\omega}^\omega$, in denen jeder Knoten durch eine endliche Teilmenge $E \subseteq \omega$ beschriftet ist. Sei t ein solcher Baum und $\{e_1, \ldots, e_n\}$ die Beschriftung der Wurzel von t, dann ist $\{t_{e_1}, \ldots, t_{e_n}\}$ die Nachfolgermenge von t, d.h.

$$t \xrightarrow{R} \{t_{e_1}, \ldots, t_{e_n}\} \iff t(\varepsilon) = \{e_1, \ldots, e_n\}.$$

Man erhält die terminale Coalgebra, wenn man nach der größten Kongruenz faktorisiert. Für beliebige Kripke-Strukturen ist die Komposition $R \circ S$ zweier Bisimulationen wieder eine Bisimulation, so dass die grösste Bisimulation transitiv ist, also mit der grössten Kongruenz übereinstimmt. $T_{\mathbb{P}}^\omega / \sim$ ist folglich terminal.

4.5 Cofreie Coalgebren

Wir verallgemeinern jetzt den Begriff der terminalen Coalgebra zu dem der *cofreien* über einer Menge X von "Covariablen".

Definition 4.5.1 *Sei X eine Menge und $\mathbf{T}_X$ eine Coalgebra mit einer Abbildung $\varepsilon_X : T_X \to X$. Das Paar $(\mathbf{T}_X, \varepsilon_X)$ heißt* **cofrei über** *X, falls für jede F-Coalgebra A und jede Abbildung $g : A \to X$ genau ein Homomorphismus $\tilde{g} : \mathbf{A} \to \mathbf{T}_X$ existiert mit $g = \varepsilon_X \circ \tilde{g}$.*

Die Menge X fasst man auch als *Farbmenge* auf und die Abbildungen g und ε_X als *Färbungen.* So gesehen gibt es also für jede Färbung g von $\mathbf{A}$ durch Farben aus X genau einen Homomorphismus $\tilde{g} : \mathbf{A} \to \mathbf{T}_X$, der mit der Färbung ε_X von $\mathbf{T}_X$ verträglich ist.

$$\begin{tikzcd} & X \\ \mathbf{A} \arrow[ur, dashed, "g"] \arrow[r, dotted, "\tilde{g}"'] & \mathbf{T}_X \arrow[u, "\varepsilon_X"'] \end{tikzcd}$$

Aus der Eindeutigkeitsforderung folgt sofort:

Folgerung 4.5.2 *Für jede Coalgebra* $\mathbf{A}$ *gilt:*

(i) Jeder Homomorphismus $\varphi : \mathbf{A} \to \mathbf{T}_X$ *ist von der Form* $\varphi = \tilde{g}$ *mit* $g = \varepsilon_X \circ \varphi$.

(ii) Sind $\varphi_1, \varphi_2 : \mathbf{A} \to \mathbf{T}_X$ *Homomorphismen mit* $\varepsilon \circ \varphi_1 = \varepsilon \circ \varphi_2$*, dann folgt* $\varphi_1 = \varphi_2$.

Offensichtlich sind cofreie Coalgebren, sofern sie existieren, bis auf Isomorphie eindeutig bestimmt. Die Coalgebra über einer ein-elementigen Menge ist gerade die terminale Coalgebra. Da man ein Paar, bestehend aus einer F-Coalgebra und einer X-Färbung auch als Coalgebra des Funktors $X \times F(-)$ auffassen kann, gilt auch:

Hilfssatz 4.5.3 *Die über der Farbmenge* X *cofreie* F*-Coalgebra* $\mathbf{T}_X$ *ist die terminale Coalgebra für den Funktor* $X \times F(-)$.

Weil mit F auch der Funktor $X \times F(-)$ beschränkt ist, folgt:

Satz 4.5.4 *Ist* F *beschränkt, so existieren cofreie* F*-Coalgebren über jeder Farbmenge* X.

Seien nun X und Y Mengen und $f : X \to Y$ eine beliebige Abbildung. Wenn die cofreien Coalgebren $\mathbf{T}_X$ und $\mathbf{T}_Y$ existieren, so induziert f einen Homomorphismus $T(f) := \widetilde{f \circ \varepsilon_X} : T_X \to T_Y$. Man prüft leicht nach:

Satz 4.5.5 *Existiert für jede Farbmenge* X *die cofreie Coalgebra, so ist* T *ein* **Set**-*Endofunktor.*

Verhaltensmuster

Wenn wir die Elemente der terminalen Coalgebra als Verhalten interpretieren, dann liegt es nahe, die Elemente der cofreien Coalgebra als **Verhaltensmuster** aufzufassen. Verhaltensmuster lassen sich heranziehen, um Klassen von Coalgebren zu definieren.

Verhaltensmuster sind also Elemente cofreier Coalgebren. Sie übernehmen die Rolle, welche Gleichungen in der Universellen Algebra spielen. Allerdings bietet es sich hier an, Klassen von Algebren nicht durch das Erfüllen, sondern das *Vermeiden* von Verhalten zu definieren. Statt Verhaltensmuster sagt man daher auch **Cogleichung**.

Diese Vorgehensweise kennt man aus vielen Gebieten der Mathematik. In der Verbandstheorie, z.B., lässt sich die Klasse der modularen Verbände durch das Nichtenthalten des Verbandes $\mathbf{N_5}$ charakterisieren, die Klasse der distributiven Verbände durch das Vermeiden von $\mathbf{N_5}$ und $\mathbf{M_3}$. Die Klasse aller planaren Graphen kann man durch das Nichtenthalten des 5-elementigen vollständigen Graphen und des Kuratowski-Graphen charakterisieren.

Definition 4.5.6 *Sei $p \in \mathbf{T}_X$ ein Verhaltensmuster, $\mathbf{A}$ eine Coalgebra und $g : A \to X$ eine Färbung. Ein Element $a \in A$* **erfüllt** *das Verhaltensmuster p unter der Färbung g, falls $\tilde{g}(a) \neq p$ ist. Wir schreiben in diesem Fall:*

$$\mathbf{A}, a \models_g p.$$

Wir sagen weiter, dass

$$\begin{array}{lll} \mathbf{A}, a & \models p, & :\iff \forall\, g : A \to X.\ \mathbf{A}, a \models_g p, \\ \mathbf{A} & \models p & :\iff \forall\, a \in A.\ \mathbf{A}, a \models p. \end{array}$$

Für eine Klasse $\mathcal{K}$ von Coalgebren und eine Menge P von Verhaltensmustern definieren wir in nahe liegender Weise:

$$\mathcal{K} \models P :\iff \forall\ \mathbf{A} \in \mathcal{K}. \forall\ p \in P. \mathbf{A} \models p.$$

Hilfssatz 4.5.7 *Sei $p \in T_X$ ein Verhaltensmuster und $X \subseteq Y$. Für eine beliebige Coalgebra $\mathbf{A}$ hat man*

$$\mathbf{A} \models p \iff \mathbf{A} \models T(\subseteq^Y_X)(p).$$

Beweis Zur injektiven Abbildung $\subseteq^Y_X$: $X \to Y$ sei π irgendeine links-inverse. Es folgt $T(\pi) \circ T(\subseteq^Y_X) = T(id_X) = id_{T_X}$, insbesondere $T(\pi)(T(\subseteq^Y_X)(p)) = p$. Für ein $a \in A$ gibt es also genau dann einen Homomorphismus φ mit $\varphi(a) = p$, wenn es einen Homomorphismus ψ gibt mit $\psi(a) = T(\subseteq^Y_X)(p)$. $\square$

Hilfssatz 4.5.8 *Sei P ein Menge von Verhaltensmustern, dann gibt es eine feste Menge X und eine Menge $P' \subseteq T_X$, so dass für jede Coalgebra $\mathbf{A}$ gilt:*

$$\mathbf{A} \models P \iff \mathbf{A} \models P'.$$

Beweis Für jedes $p \in P$ gibt es eine Menge X_p, so dass $p \in T_{X_p}$. Sei $X := \Sigma_{p \in P} X_p$ die disjunkte Vereinigung mit den kanonischen Einbettungen $e_p : X_p \to X$, dann folgt aus dem vorigen Hilfssatz, dass man $P' = \{\, e_p(p) \mid p \in P\}$ wählen kann. $\square$

Aufgrund von Hilfssatz 4.5.3 und der Beschreibung schwach terminaler und terminaler Coalgebren (Satz 4.4.5 und Folgerung 4.4.6) lassen sich im Falle eines beschränkten Funktors F dessen Verhaltensmuster als unendliche Bäume deuten, deren Knoten mit Elementen aus $F(\kappa)$ beschriftet sind, und die eine Farbe aus X tragen. Somit haben Verhaltensmuster in der Tat eine nahe Verwandtschaft zu unendlichen Termen.

4.6 Musterdefinierte Klassen sind Covarietäten

Sei P eine Menge von Cogleichungen. Die Klasse aller Coalgebren, die P erfüllt, heißt die **Modellklasse** von P und wird mit $\mathcal{M}od(P)$ bezeichnet.

Satz 4.6.1 *$\mathcal{M}od(P)$ ist eine Covarietät.*

Beweis Wir zeigen, dass $\mathcal{M}od(P)$ unter homomorphen Bildern, Summen und Untercoalgebren abgeschlossen ist. Sei $p \in T_X$ eine Cogleichung.

H: Sei $\varphi : \mathbf{A} \twoheadrightarrow \mathbf{B}$ epi. Angenommen, $\mathbf{B}$ erfülle p nicht, dann gibt es ein $b \in B$ und einen Homomorphismus $\psi : \mathbf{B} \to \mathbf{T}_X$ mit $\psi(b) = p$. Weil φ surjektiv ist, gibt es ein $a \in A$ mit $\varphi(a) = b$, also $(\psi \circ \varphi)(a) = p$, also gilt auch $\mathbf{A} \not\models p$.

Σ: Zu jedem $a \in \sum_{i \in I} A_i$ gibt es ein i und ein $a_i \in A_i$ mit $e_i(a_i) = a$. Ein Homomorphismus $\psi : \sum_{i \in I} A_i \to T_X$ mit $\psi(a) = p$ liefert einen Homomorphismus $\psi \circ e_i : A_i \to \mathbf{T}_X$ mit $\psi \circ e_i(a_i) = p$.

S: Sei $\mathbf{U} \leq \mathbf{A}$ und $u \in U$. Wir nehmen an, dass es einen Homomorphismus $\psi : \mathbf{U} \to \mathbf{T}_X$ gibt mit $\psi(u) = p$ und zeigen, dass wir ψ zu einem Homomorphismus $\psi' : \mathbf{A} \to \mathbf{T}_X$ *fortsetzen* können, so dass $\psi = \psi' \circ \subseteq_U^A$ gilt, mithin $\psi'(u) = p$.

Mit $g = \varepsilon_X \circ \psi$, gilt $\psi = \tilde{g}$. Wir setzen $g : U \to X$ zu einer Abbildung $g' : A \to X$ fort mit $g' \circ \subseteq_U^A = g$. Für $\psi' := \widetilde{g'}$ gilt dann $\varepsilon_X \circ \psi' \circ \subseteq_U^A = g' \circ \subseteq_U^A = g = \varepsilon_X \circ \psi$. Folgerung 4.5.2 erlaubt uns, ε_X links zu kürzen, so dass wir $\psi' \circ \subseteq_U^A = \psi$ erhalten. $\square$

$$\begin{tikzcd}
A \arrow[rr, dashed, "g'"] \arrow[drr, dotted, "\psi'" near end] & & X \\
U \arrow[u, "\subseteq"] \arrow[urr, "g" near start] \arrow[rr, "\psi"'] & & T_X \arrow[u, "\varepsilon_X"']
\end{tikzcd}$$

4.7 Der Co-Birhoffsche Satz

Um einen Birkhoffschen Satz zu gewinnen, definieren wir für eine beliebige Klasse $\mathcal{K}$ von Coalgebren und eine feste Menge X von Covariablen die X-**Cogleichungs-Klasse** von $\mathcal{K}$:

$$\mathcal{C}eq_X(\mathcal{K}) := \{p \in T_X \mid \forall \mathbf{A} \in \mathcal{K}.\mathbf{A} \models p\}.$$

Wir können nun die coalgebraische Variante des Birkhoffschen Satzes beweisen:

Satz 4.7.1 (Co-Birkhoff) *Sei F ein durch κ beschränkter Funktor. Für jede Klasse $\mathcal{K}$ von F-Coalgebren gilt:*

$$\mathcal{M}od(Ceq_\kappa(\mathcal{K})) = SH\Sigma(\mathcal{K}).$$

Beweis Wegen Satz 4.6.1 ist $Mod(Ceq_\kappa(\mathcal{K}))$ eine Covarietät, umfasst also $SH\Sigma(\mathcal{K})$.

Sei jetzt $\mathbf{A} \in Mod(Ceq_\kappa(\mathcal{K}))$. Wir finden zu jedem $a \in A$ eine Untercoalgebren $\mathbf{U}_a \leq \mathbf{A}$ mit $a \in U_a$ und $|U_a| < \kappa$. Es folgt $A = \bigcup_{a \in A} U_a$, daher erhalten wir einen surjektiven Homomorphismus $\sum_{a \in A} \mathbf{U}_a \twoheadrightarrow \mathbf{A}$. Es genügt nun, zu zeigen, dass jedes $\mathbf{U}_a$ in $SH\Sigma(\mathcal{K})$ ist.

Wir können von jetzt ab $|A| < \kappa$ annehmen, daher gibt es eine injektive Abbildung $g : A \to \kappa$. Es folgt, dass $\tilde{g} : \mathbf{A} \to \mathbf{T}_\kappa$ ebenfalls injektiv ist, somit ist $\mathbf{A}$ isomorph zu der Untercoalgebra $\mathbf{C} := \tilde{g}[\mathbf{A}]$ von $\mathbf{T}_\kappa$.

Jedes Element von $c \in C$ ist eine Cogleichung, welche von $\mathbf{A}$ offensichtlich nicht erfüllt wird. Es muss also ein $\mathbf{B}_c \in \mathcal{K}$ geben mit $\mathbf{B}_c \not\models c$, d.h. einen Homomorphismus $\psi_c : \mathbf{B}_c \to \mathbf{T}_\kappa$ mit $c \in \psi_c[B_c]$. Jedes $\psi_c[B_c]$ ist eine Untercoalgebra von $\mathbf{T}_\kappa$, folglich ist auch die Vereinigung $B := \bigcup_{c \in C} \psi_c[B_c]$ eine Untercoalgebra von $\mathbf{T}_\kappa$. Weil alle B_c in $\mathcal{K}$ sind, folgt $B \in H(\Sigma(\mathcal{K})) \subseteq SH\Sigma(\mathcal{K})$. Es gilt andererseits $\mathbf{A} \cong \mathbf{C} \leq \mathbf{B}$, so dass $\mathbf{A} \in SH\Sigma(\mathcal{K})$ folgt. $\square$

4.8 Programmieren mit terminalen Coalgebren

Wir kehren nun zu den Beispielen aus der Informatik zurück, mit denen wir unseren Exkurs über Coalgebren begonnen haben. Ist D eine Menge von Daten, dann haben wir "black boxes" als Coalgebren des Funktors $F(-) := D \times (-)$ kennen gelernt.

Die terminale Coalgebra $\mathbf{T}$ dieses Funktors hat als Grundmenge D^ω, die Menge aller unendlichen Ströme. Die Strukturabbildung setzt sich aus den Abbildungen

(i) $hd : D^\omega \to D$

(ii) $tl : D^\omega \to D^\omega$

zusammen. Zu jeder anderen F-Coalgebra $\mathbf{S} = (S, \alpha)$ mit Struktur-Abbildungen

(i) $\alpha_1 : S \to D$

(ii) $\alpha_2 : S \to S$

gibt es nun genau einen Homomorphismus $\tau : \mathbf{S} \to \mathbf{T}$. Dies bedeutet aber:

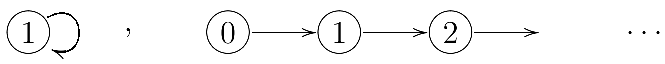
(i) $hd(\tau(s)) = \alpha_1(s)$

(ii) $tl(\tau(s)) = \tau(\alpha_2(s))$.

Durch Induktion kann man zeigen, dass durch diese Bedingungen genau ein Homomorphismus $\tau : \mathbf{S} \to \mathbf{T}$ definiert wird, und dass das k-te Element des Stromes $\tau(s)$ gerade $\alpha_1(\alpha_2^{k-1}(s))$ ist. Diesen Homomorphismus kann man aber als Programm auffassen, das man in der funktionalen Notation folgendermaßen schreiben kann:

$$\tau(s) = [\alpha_1(s) : \tau(\alpha_2(s))].$$

Insbesondere werden die eingangs diskutierten funktionalen Programme `ones`, `from` und `add` auf streams durch die folgenden black boxes definiert:



und $\mathbf{A} := (\mathbb{N} \times \mathbb{N}, (+, suc))$, wobei $+$ die Addition auf den natürlichen Zahlen ist und $suc(n_1, n_2) := (n_1 + 1, n_2 + 1)$.

Die Tatsache, dass die Stream-Coalgebra D^ω terminal ist garantiert also, dass durch die Definitionen in Abschnitt 1.2 eindeutig Programme gegeben sind.

4.9 Beweise durch Coinduktion

Coinduktion können wir verwenden, wenn wir Eigenschaften der eben erwähnten funktionalen Programme herleiten wollen. Das Coinduktionsprinzip gilt für terminale Coalgebren. Es besagt:

$$\frac{s \sim s'}{s = s'}.$$

Jede Bisimulation ist in $\sim$ enthalten. Um zu zeigen, dass zwei Elemente s und s' gleich sind, genügt es also, irgendeine Bisimulation R zu finden, die das Paar (s, s') enthält. Das Finden einer geeigneten Bisimulation für solche Zwecke ist eine kreative, gelegentlich nicht-triviale Aufgabe.

Funktionale Programme

Hier wollen wir ein einfaches Beispiel vorführen. Wir wollen zeigen, dass die in Abschnitt 1.2 aufgeworfene Frage, nach der Gleichheit der Programme

$$add\ nats\ ones\ ==\ from\ 1$$

einfach durch Coinduktion gezeigt werden kann. Die Vorgehensweise ist einfach:

1. Finde eine Bisimulation R, die das Paar $(add\ nats\ ones,\ from\ 1)$ enthält,
2. Schließe $(add\ nats\ ones,\ from\ 1) \in \sim$
3. Schließe mittels Coinduktion $add\ nats\ ones\ =\ from\ 1$.

In unserem Falle wählen wir

$$R := \{(add\ (from\ n)\ ones,\ from(n+1)) \mid n \in \mathbb{N}\}.$$

Um zu zeigen, dass R eine Bisimulation ist, müssen wir die Bisimulationsregel für Black Boxes nachprüfen:

$$\frac{sRs'}{hd(s) = hd(s')\ ,\ tl(s)\ R\ tl(s')}$$

$$\begin{aligned}
hd(add\ (from\ n)\ ones) &= hd(from\ n) + hd(ones) \\
&= n+1 \\
&= hd(from\ (n+1))
\end{aligned}$$

$$\begin{aligned}
tl(add\ (from\ n)\ ones) &= add(tl(from\ n), tl(ones)) \\
&= add(from(n+1), ones) \\
&\mathrel{R} from(n+2) \\
&= tl(from\ (n+1)).
\end{aligned}$$

Sprachen

Akzeptoren sind Automaten mit Eingabemenge Σ und Ausgabemenge $D = \{0,1\}$, also Coalgebren zum Funktor $F(-) = \{0,1\} \times (-)^{\Sigma}$. Die terminale Coalgebra besteht aus allen Sprachen $L \subseteq \Sigma^*$ über dem Alphabet Σ. L ist ein akzeptierender Zustand falls $\varepsilon \in L$ ist, und die Transitionsfunktion wird durch die Ableitung gegeben:

$$\delta(L, e) = L_e = \{w \in \Sigma^* \mid e \cdot w \in L\}.$$

Auf den Sprachen über dem Alphabet Σ hat man andererseits auch klassische Operationen wie z.B. die Vereinigung

$$L + M := L \cup M,$$

die Konkatenation

$$L \cdot M := \{u \cdot w \mid u \in L, v \in M\},$$

oder den Kleene-Stern

$$L^* := \{u_1 \cdot u_2 \cdot \ldots \cdot u_n \mid n \in \mathbb{N}, u_1, u_2, \ldots, u_n \in L\}.$$

Nützlich sind auch noch die speziellen Sprachen $\mathbf{0} := \emptyset$ und $\mathbf{1} := \{\varepsilon\}$.

Relativ leicht kann man nun die Differentiationsregeln herleiten:

$$\begin{array}{rcl} \mathbf{1}_e & = & \mathbf{0} \\ (L+M)_e & = & L_e + M_e \\ (L \cdot M)_e & = & \begin{cases} (L_e \cdot M) + M_e & \text{falls } \varepsilon \in L \\ (L_e \cdot M) & \text{sonst.} \end{cases} \\ (L^*)_e & = & L_e \cdot L^*. \end{array}$$

Rutten zeigt in [Rut98], wie man z.B. die Gleichung $(1 + L \cdot L^*) = L^*$ mittels Coinduktion gewinnen kann. Es genügt zu zeigen, dass

$$R := \{(1 + LL^*, L^*) \mid L \subseteq \Sigma^*\} \cup \{(L, L) \mid L \subseteq \Sigma^*\}$$

eine Bisimulation ist. Wir müssen also die Bedingung

$$\frac{s \ R \ s'}{\gamma(s) = \gamma(s'),\ \forall e \in \Sigma.\delta(s,e) \ R \ \delta(s',e)}$$

nachprüfen. In unserem Falle wird daraus

$$\frac{S \ R \ S'}{\varepsilon \in S \iff \varepsilon \in S',\ \forall e \in \Sigma.\ S_e \ R \ S'_e}.$$

Für die gegebene Relation ist dies rasch verifiziert. Es gilt $\varepsilon \in \mathbf{1} + L \cdot L^*$ und auch $\varepsilon \in L^*$. Für die zweite Bisimulationsbedingung rechnen wir nach:

$$\begin{array}{rcl} (1 + LL^*)_e & = & \mathbf{1}_e + (LL^*)_e \\ & = & \mathbf{0} + L_e \cdot L^* \\ & = & L_e \cdot L^* \\ & R & L_e \cdot L^* \\ & = & (L^*)_e. \end{array}$$

4.10 Aufgaben

Sei F ein Typ, so dass die terminale F-Coalgebra $\mathbf{T}$ existiert.

(i) Eine F-Coalgebra $\mathbf{A}$ erfüllt das Coinduktionsprinzip (siehe Seite) gdw. der Morphismus von $\mathbf{A}$ in die terminale Coalgebra ist mono.

(ii) Für eine F-Coalgebra $\mathbf{A}$ gilt $\frac{x \ \nabla \ y}{x \ = \ y}$ gdw. $\mathbf{A} \leq \mathbf{T}$.

(iii) Für den Identitätsfunktor $\mathcal{I}$ besteht die cofreie $\mathcal{I}$-Coalgebra über der Farbmenge X aus allen unendlichen Folgen von Elementen aus X, also X^ω. Die Strukturabbildung ist die *tail*-operation: $tail(a_k)_{k\in\omega} = (a_{k+1})_{k\in\omega}$

(iv) Zeigen Sie: Die folgende Cogleichung über der Farbmenge $\{schwarz, weiss\}$

$$\circ \longrightarrow \bullet \circlearrowright$$

definiert genau die Klasse aller $\mathcal{I}$-Coalgebren (A, α) mit $\forall s \in A. \exists n \in \omega. \alpha^n(s) = s$.

4.11 Anmerkungen

Coalgebren, als duale Konstruktionen zu klassischen Universellen Algebren, wurden schon von Drbohlav [Drb69] betrachtet. Allerdings hatte die zu enge Definition einer "Coalgebra vom Typ $(n_i)_{i\in I}$", als Menge A mit einer Familie von Abbildungen $f_i : A \to n_i \cdot A$ von A in die n_i-fache disjunkte Summe, keine überzeugenden Anwendungen. Erst durch die Verallgemeinerung des Typs zu einem beliebigen **Set**-Endofunktor F und die Spezialisierung auf den Potenzmengen-Funktor $\mathbb{P}(-)$, bzw. den Potenz-Funktor $(-)^\Sigma$ erhält man relevante Beispiele, wie z.B. Kripke-Strukturen und Automaten.

Automaten wurden bereits von Arbib und Manes [AM82] coalgebraisch beschrieben. In Aczel und Mendler [AM89] findet sich zum ersten Mal die abstrakte Definition von Bisimulationen. Damit wird auch formal ein Zusammenhang hergestellt zwischen ähnlichen Phänomenen - der Nerode Kongruenz der Automatentheorie und der Bisimulation in Kripke-Strukturen. Es wuchs die Einsicht, dass für zustandsbasierte Systeme eine coalgebraische Beschreibung Vorteile bieten könnte. Einflussreich ist in diesem Zusammenhang das Buch [BM96] von J. Barwise und L. Moss. Anwendungen für die objektorientierte Programmierung werden von H. Reichel ([Rei95] und B. Jacobs ([Jac96]) aufgezeigt.

Eine systematische Entwicklung einer allgemeinen mathematischen Theorie der Coalgebren findet sich zum ersten Mal bei J. Rutten in [Rut96]. Allerdings verwendet Rutten, wie zuvor auch Aczel und Mendler, die zusätzliche Hypothese, dass der Typfunktor "schwache Pullbacks" und beliebige Schnitte erhält. Unter diesen Umständen sind Untercoalgebren auch unter beliebigen Schnitten abgeschlossen und jede Coalgebra ist "konjunkte Summe" (siehe [GS01]) von eins-erzeugten. Bisimulationen sind unter dem Relationenprodukt abgeschlossen und die größte Bisimulation ist eine Kongruenz. Es folgt, dass die Begriffe von Beobachtungs-Äquivalenz und Ununterscheidbarkeit zusammenfallen. Unter dieser Voraussetzung wurde in [Gum99b] zunächst auch der Satz von Birkhoff formuliert und bewiesen.

Die allgemeine Theorie, also Coalgebren für einen beliebigen Funktor, wurde zum ersten Male in [Gum99a] systematisch entwickelt. Auch der Birkhoffsche Satz lässt sich in dem allgemeinen Rahmen für den Fall beschränkter Funktoren gewinnen. Die Charakterisierung beschränkter Funktoren aus [GS00] führte sogar auf eine syntaktische Beschreibung von Cogleichungen als Äquivalenzklassen unendlicher Bäume.

Viele Punkte konnten wir in der Kürze nicht ansprechen - dazu gehören die genaueren syntaktischen Beschreibungen von Cogleichungen, der alternative Zugang zu einer Syntax mittels der terminalen Sequenzen ([AK95], [Wor99]) die coalgebraischen Logiken von Moss [Mos99] bzw. von Pattinson [Pat02] (siehe auch [KP02]), sowie Verallgemeinerungen, in der die Basiskategorie **Set** durch andere Kategorien ersetzt wird. In der angegebenen Literatur findet der Leser zahlreiche Verweise.

Literatur

[AK95] J. Adámek and V. Koubek, *On the greatest fixed point of a set endofunctor*, Theoretical Computer Science (1995), no. 150, 57–75.

[AM82] M.A. Arbib and E.G. Manes, *Parametrized data types do not need highly constrained parameters*, Information and Control **52** (1982), 139–158.

[AM89] P. Aczel and N. Mendler, *A final coalgebra theorem*, Proceedings category theory and computer science (D.H. Pitt et al, ed.), Lecture Notes in Computer Science, Springer, 1989, pp. 357–365.

[BM96] J. Barwise and L. Moss, *Vicious circles*, CSLI Lecture Notes, 1996.

[Drb69] K. Drbohlav, *On coalgebras*, Summer Session on the Theory of Ordered Sets and General Algebras, 1969, University of J.E. Puryne, Brno, pp. 81–87.

[GS00] H.P. Gumm and T. Schröder, *Coalgebras of bounded type*, (to appear in Math. Structures in Computer Science), 2000.

[GS01] ———, *Covarieties and complete covarieties*, Theoretical Computer Science **260** (2001), 71–86.

[Gum99a] H.P. Gumm, *Elements of the general theory of coalgebras*, LUATCS 99, Rand Afrikaans University, Johannesburg, South Africa, 1999.

[Gum99b] ———, *Equational and implicational classes of coalgebras*, Theoretical Computer Science **260** (1999), 57–69.

[Gum01] ———, *Functors for coalgebras*, Algebra Universalis (2001), no. 45, 135–147.

[Jac96] B. Jacobs, *Objects and classes, co-algebraically*, Object-Orientation with Parallelism and Persistence (B. Freitag et al., ed.), Kluwer Acad. Publ., 1996, pp. 83–103.

[KP02] A. Kurz and D. Pattinson, *Definability, canonical models, compactness for finitary coalgebraic modal logic*, Coalgebraic Methods in Computer Science (CMCS'02) (L.S. Moss, ed.), ETAPS, 2002.

[Lam68] J. Lambek, *A fixpoint theorem for complete categories*, Mathematische Zeitschrift (1968), no. 103, 151–161.

[Mos99] L.S. Moss, *Coalgebraic logic*, Ann. Pure Appl. Logic **96** (1999), 277–317.

[Pat02] D. Pattinson, *Coalgebraic modal logic: Soundness, completeness and decidability*, Tech. report, LMU München, 2002.

[Rei95] H. Reichel, *An approach to object semantics based on terminal co-algebras*, Math. Struct. in Comp. Sci. (1995), no. 5, 129–152.

[Rut96] J.J.M.M. Rutten, *Universal coalgebra: a theory of systems*, Tech. Report CS-R9652, CWI, Amsterdam, 1996.

[Rut98] ———, *Automata and coinduction (an exercise in coalgebra)*, Proceedings of CONCUR '98 (D. Sangiorigi and R. de Simone, eds.), LNCS, vol. 1466, Springer, 1998, pp. 194–218.

[Wor99] J. Worrell, *Terminal sequences for accessible endofunctors*, Coalgebraic Methods in Computer Science (J. Rutten et al B. Jacobs, ed.), Electronic Notes in Theoretical Computer Science, vol. 19, Elsevier Science, 1999.

Symbolverzeichnis

Mengen, Relationen, Funktionen

$\rightarrowtail$ – injektive Abbildung

$\subseteq_X^Y$ – kanonische Teilmengen-Inklusion

$\twoheadrightarrow$ – surjektive Abbildung /epi

π_i, π_A – Projektionen auf Komponenten

$f[A]$ – Bild von A unter f

$\xrightarrow{R}$ – Pfeilnotation für Relation R

X^* – die endlichen Folgen von Elementen aus X

X^M – die Abbildungen von M nach X

$\mathbf{2}$ – Menge $\{0, 1\}$

ω – kleinste unendliche Kardinalzahl

κ – Kardinalzahl

Kategorien, Funktoren

Set – die Kategorie der Mengen

$\mathbb{P}$ – Potenzmengen-Funktor

$\mathbb{P}_\kappa$, $\mathbb{P}_\omega$ – beschränkte Version von $\mathbb{P}$

$F \dot{\rightarrow} G$ – natürliche Transformation

$F \dot{\twoheadrightarrow} G$ – surjektive nat. Transformation

$\mathcal{P}$ – kontravarianter Potenzmengen-Funktor

Coalgebren

$\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, ... – Coalgebren

$\mathbf{Set}_F$ – Kategorie der F-Coalgebren

α_A – Strukturabbildung auf $\mathbf{A}$

$\leq$ – Untercoalgebra

$[X]$ – von X co-erzeugte Untercoalgebra

θ, Θ – Kongruenzen

$\bigvee_{i \in I} \theta_i$ – Supremum der θ_i

∇, ∇_A – größte Kongruenz auf $\mathbf{A}$

Δ, Δ_A – kleinste Kongruenz auf $\mathbf{A}$

π_θ – Projektion auf θ-Klassen

φ, ψ – Homomorphismen

$\sum_{i \in I} \mathbf{A}_i$ – Summe der $\mathbf{A}_i$

e_i – kanonische Einbettung in die Summe

$[R]$ – Bisimulationskern

$\sim$, $\sim_{\mathbf{A}}$, $\sim_{\mathbf{A},\mathbf{B}}$ – grösste Bisimulationen

$S(\mathcal{K})$ – Klasse aller Untercoalgebren

$\Sigma(\mathcal{K})$ – Klasse aller Summen

$H(\mathcal{K})$ – Klasse aller homomorphen Bilder

$\mathcal{C}eq_X$ – Cogleichungen in X

$\mathcal{M}od$ – Modelle

$\mathbf{T}_X$ – cofreie über X

ε_X – kanonische Färbung der cofreien $\mathbf{T}_X$

Worte, Sprachen, Automaten

ε – leeres Wort

$c \cdot w$ – Anfügen von Zeichen c an Wort w

Σ^* – Menge aller Worte über X

$\mathbf{1}$ – ein-elementige Sprache $\{\varepsilon\}$

L_e – Ableitung der Sprache L nach Buchstaben e

$\mathcal{L}(\mathbf{A}, s)$ – vom Automaten $\mathbf{A}$ mit Anfangszustand s erkannte Sprache

δ – Übergangsfunktion von Automaten

γ – Ausgabefunktion von Automaten

t_e – e-ter Sohn des Baumes t

Index