Koalgebren, Monaden und Semantik

Stefan Milius

Institut für Theoretische Informatik
Technische Universität Carolo Wilhelmina
zu Brauschweig
milius@iti.cs.tu-bs.de

Abstract: Dies ist eine Zusammenfassung der Dissertation [Mi05b] des Autors, welche ihrerseit eine kumulative Arbeit ist, die aus den wissenschaftlichen Artikeln [AMV03, AMV06a, AMV06b, Ac03, Mi05a, Mi02, MM06] besteht. Wir untersuchen mathematische Strukturen, die in der Theorie der Koalgebren auftauchen und die für die Semantik rekursiver Definitionen nützlich sind. Zunächst werden verschiedene neue Konzepte eingeführt und mathematische Ergebnisse bewiesen, die klassische Arbeiten über iterative Theorien von Elgot und Nelson verallgemeinern und erweitern. Dann werden diese neuen Erkenntnisse angewandt, um eine konzeptionell einfache und allgemeine Semantik rekursiver Programmschemas zu erarbeiten. Verschiedene Anwendungen demonstrieren die Stärke unserer neuen Theorie. So ergeben sich die klassischen Ansätze zur Semantik der Rekursion, die geordnete oder metrisierte Strukturen benutzen, als Spezialfälle. Darüber hinaus zeigen wir Anwendungen, die mit klassischen Methoden nicht erhalten werden. Dies betrifft insbesondere rekursiv definierte Funktionen, die zusätzliche Eigenschaften erfüllen, oder rekursive Definitionen von Fraktalen.

1 Einleitung

Rekursion ist eines der wichtigsten Beschreibungsmittel in den mathematisch orientierten Wissenschaften. Sie erlaubt es, komplizierte und (potentiell) unendliche Systeme oder Phänomene in endlicher, klarer und knapper Art und Weise zu beschreiben. So können beispielsweise gewisse Funktionen durch eine rekursive Definition, also implizit oder durch Selbstreferenz, angegeben werden. Paradigmatisch ist hier sicherlich die Fakultätsfunktion $f(n)$, die das Produkt der Zahlen von 1 bis n berechnet und wie folgt rekursiv definiert werden kann:

$$f(n) = \begin{cases} 1 & \text{falls } n = 0 \\ n \cdot f(n-1) & \text{sonst.} \end{cases}$$

Manche irrationalen Zahlen können auf elegante Weise rekursiv präsentiert werden, wie zum Beispiel der berühmte Goldene Schnitt φ, der durch den Kettenbruch

$$1 + \cfrac{1}{1 + \cfrac{1}{\ddots}}$$

gegeben werden kann und der somit die einzige Lösung der rekursiven Gleichung

$$\varphi = 1 + \frac{1}{\varphi}\,.$$

ist. Fraktale sind rekursiv beschriebene Figuren. Ein einfaches Beispiel ist die bekannte Cantor-Menge. Sie ist die einzige nicht leere abgeschlossene Teilmenge des Intervalls $[0,1]$, die die Gleichung

$$c = \frac{1}{3}c \cup \left(\frac{2}{3} + \frac{1}{3}c\right)\,,$$

erfüllt, wobei $\frac{1}{3}c = \{\, \frac{1}{3}x \mid x \in c \,\}$ definiert ist.

In der Informatik ist Rekursion in der einen oder anderen Form natürlich in jeder Programmiersprache und in jedem Formalismus, der Abläufe in einem Informationssystem beschreibt, vorhanden. So wird ein Programmierer etwa

$$\mathsf{fac}(n) := \mathsf{if}\ n = 0\ \mathsf{then}\ 1\ \mathsf{else}\ n * \mathsf{fac}(n-1) \tag{1.1}$$

schreiben, um die Fakultätsfunktion zu programmieren. Bei der Spezifikation von Systemverhalten mittels Prozessalgebren benutzt man Rekursion, um potentiell unendliche Abläufe im System zu beschreiben. Beispielsweise spezifiziert der Prozessterm

$$P = a.P + b \tag{1.2}$$

ein System, das die Aktion a wiederholt und beliebig oft ausführen kann, ohne jemals zu terminieren, oder alternativ wird die Aktion b ausgeführt, wonach das System terminiert. Zum Schluss seien noch rekursiv spezifizierte Datentypen wie

$$\mathsf{list} ::= \mathsf{element} \mid \mathsf{element.list} \tag{1.3}$$

genannt, wobei element hier ein gegebener Datentyp und list der rekursiv definierte Datentyp der linearen Listen mit Einträgen vom Typ element ist.

Schreibt man eine rekursive Spezifikation oder Gleichung hin, stellt sich natürlich sofort die Frage, was ihre Bedeutung sein soll. Solche Fragen werden in der *Semantik* beantwortet, die ein wichtiges Teilgebiet der theoretischen Informatik darstellt. Grundsätzlich gibt es verschiedene Möglichkeiten, um rekursiven Spezifikationen eine Bedeutung zu verleihen. In diesem Text interessieren wir uns für die denotationelle Semantik. Hierbei wird einer rekursiven Definition ein bestimmtes mathematisches Objekt zugeordnet, das als seine Semantik fungiert, ihm also Bedeutung verleiht. Solch eine Semantik ist unabhängig von Details verschiedener Implementierungen eines Programms und lässt sich deshalb zur Grundlage formaler Methoden zur Verifikation rekursiver Programme oder rekursiv spezifizierter Informationssysteme machen.

In der Literatur gibt es verschiedene Ansätze für die denotationelle Semantik rekursiver Definitionen. Die in dieser Dissertation enthaltenen Forschungsergebnisse bauen auf den klassischen Arbeiten [El75, BE74, EBT78] der Gruppe von Elgot auf. Elgots Ziel war eine Semantik rekursiver Definitionen, die rein algebraisch ist, also nicht auf zusätzliche

Strukturen wie Halbordnungen oder Metriken angewiesen ist. In [El75] hat Elgot *iterative Theorien* eingeführt. Dies sind algebraische Theorien im Sinne von Lawvere [La63] mit der zusätzlichen Eigenschaft, dass gewisse endliche Systeme gegenseitig rekursiver Gleichungen eine eindeutige Lösung besitzen. Bloom und Elgot bewiesen in [BE74], dass für jede Signatur Σ von Operationssymbolen eine frei erzeugte iterative Theorie existiert, und Elgot et al. [EBT78] zeigten dann, dass freie iterative Theorien durch die *rationalen* Σ-Bäume gegeben sind. Letztere sind solche Σ-Bäume, die nur endlich viele verschiedene Unterbäume besitzen; diese Beschreibung geht auf Ginali [Gi79] zurück. Der ursprüngliche Beweis von Elgot et al. mit den Methoden der allgemeinen Algebra ist sehr kompliziert und nimmt über 100 Seiten in den Artikeln [El75, BE74, EBT78] ein. Deshalb war es ein großer Fortschritt, als Nelson [Ne83] und Tiuryn [Ti80] *iterative Algebren* für eine Signatur einführten, um damit einen einfacheren Zugang zu Elgots iterativen Theorien zu erhalten. Nelson bewies, dass für jede Generatormenge X die rationalen Σ-Bäume über X die frei erzeugte iterative Algebra über X formen und dass darüber hinaus frei erzeugte iterative Algebren die freie iterative Theorie von Elgot ergeben.

Analog zu iterativen Theorien gibt es noch die *vollständig* iterativen Theorien, in denen eindeutige Lösungen auch für unendliche rekursive Gleichungssysteme postuliert werden. Eine frei erzeugte vollständig iterative Theorie über einer Signatur Σ ist die Theorie aller (endlicher und unendlicher) Σ-Bäume, siehe [EBT78].

(Vollständig) iterative Theorien erlauben es a priori nicht, rekursive Funktionsdefinitionen wie die Definition der Fakultätsfunktion zu lösen. Dies ist das Thema der algebraischen Semantik, siehe beispielsweise [Gu81]. Wie wir noch sehen werden, spielen auch hier Σ-Bäume eine überaus wichtige Rolle.

Doch zunächst wenden wir uns nun dem ersten Teil der Ergebnisse der Dissertation zu.

2 Vollständig Iterative Algebren

Der Artikel [Mi05a] erweitert und verallgemeinert die klassischen Ergebnisse von Elgot et al. und Nelson. Dies wird erreicht durch Benutzung von Methoden aus der Theorie der *Koalgebren*. Koalgebren sind in den letzten Jahren nach und nach zu einem unverzichtbaren Werkzeug des praktizierenden theoretischen Informatikers geworden. Sie erlauben es, viele verschiedene Typen von Systemen in uniformer Weise auf einer abstrakten Ebene zu studieren. So sind beispielsweise die üblichen sequentiellen deterministischen Automaten, die aus der Prozesstheorie bekannten markierten Transitionssysteme, probabilistische Automaten, aber auch infinitäre Datenstrukturen wie Streams, (un)endliche Bäume oder Graphen als Koalgebren beschreibbar; siehe [JR97, Ru00, Gu99, Ad05] für einführende Texte.

Unsere Ergebisse beruhen auf der Beobachtung, dass für eine Signatur Σ die Σ-Bäume die terminale Koalgebra des zu Σ gehörigen Polynomialfunktors auf Mengen bilden. Dies erlaubt es, die klassischen Ergebnisse mit den konzeptuellen Methoden der Kategorientheorie zu beweisen.

Genauer gesagt beginnt man statt mit einer Signatur mit einem Endofunktor H der Kategorie der Mengen (oder einer allgemeinen Kategorie, die nur sehr schwache Voraussetzungen erfüllen muss).

Eine (Ko-)Algebra für einen Endofunktor ist dann eine Menge A mit einer Abbildung von HA nach A (bzw. von A nach HA). In [Mi05a] wird das Konzept der vollständig iterativen Algebren eingeführt. Dies sind Algebren A eines Endofunktors mit der Eigenschaft, dass alle rekursiven Gleichungen eines gewissen sehr einfachen Typs (diesen nennen wir *flach*) eine eindeutige Lösung in A besitzen. Für eine Signatur Σ sind die Algebren $T_\Sigma X$ aller Σ-Bäume über der Generatormenge X vollständig iterativ. Weitere wichtige Beispiele sind Algebren mit einem vollständigen metrischen Raum als Trägermenge. Wir haben ja bereits den goldenen Schnitt und die Cantor-Menge als Lösung rekursiver Gleichungen gesehen. Die entsprechenden vollständig iterativen Algebren werden von dem Intervall $[1, 2]$ bzw. von dem Raum der nicht leeren abgeschlossenen Teilmengen von $[0, 1]$ mit der Hausdorff'schen Metrik getragen. Ein grundlegendes Ergebnis der Theorie der vollständig iterativen Algebren ist der

Satz 2.1. *[Mi05a] Eine frei erzeugte vollständig iterative Algebra über X ist genau dasselbe wie eine terminale Koalgebra des Funktors $H(-) + X$.*

Dieser Satz bringt erstmalig algebraische und koalgebraische Methoden unter ein Dach, und er ist damit von großer Bedeutung für Anwendungen. Terminale Koalgebren dienen oft als semantische Bereiche für Systemverhalten, welches typischerweise auch rekursiv definiert wird: endliche und unendliche Σ-Bäume, Streams oder formale Sprachen, die von Automaten akzeptiert werden, formen terminale Koalgebren. Die Charakterisierung terminaler Koalgebren als freie vollständig iterative Algebren eröffnet damit die Möglichkeit, die koalgebraischen Methoden in die algebraische Semantik zu transportieren.

Deshalb arbeiten wir im Folgenden meist mit Endofunktoren H, die *genug terminale Koalgebren* besitzen, das heißt, dass für jede Menge X eine terminale Koalgebra TX des Funktors $H(-) + X$ existiert. Dies ist eine sehr schwache Voraussetzung, und es existieren in der Literatur gute Kriterien, die dies in fast allen praktisch relevanten Fällen sicherstellen.

Satz 2.2. *[Ac03, Mi05a] Es sei H ein Endofunktor mit genug terminalen Koalgebren TX. Dann ist T die Objektzuweisung einer Monade, und diese Monade ist die freie von H erzeugte vollständig iterative Monade.*

Dieses Ergebnis ist auf der einen Seite ganz analog zu Nelsons Ergebnis über iterative Algebren, denn der obige Satz sagt aus, dass freie vollständig iterative Algebren eine freie vollständig iterative Theorie ergeben. Auf der anderen Seite ist dieses Ergebnis viel allgemeiner und darüber hinaus liefern die verwendeten Beweismittel ein konzeptionell viel klareres Argument. Für die kategorientheoretische Semantik rekursiver Funktionsdefinitionen in Abschnitt 5 unten ist dieser Satz eine der unabdingbaren mathematischen Grundlagen. In dem Spezialfall, der sich durch die von Signaturen erzeugten Polynomialfunktoren ergibt, liefert die Freiheit der Monade T auch die sogenannte *Substitution zweiter Ordnung*. Diese ist notwendig, um überhaupt zu formulieren, was eine Lösung einer rekursiven Funktionsdefinition sein soll. Anscheinend ist dies auch das erste Mal, dass die Substitution zweiter Ordnung durch die Kategorientheorie in konzeptueller Weise verstanden wurde.

3 Iterative Algebren

Die gemeinsamen Arbeiten [AMV03, AMV06a] mit Adámek und Velebil beschäftigen sich mit iterativen Algebren und iterativen Theorien. Hierbei interessiert man sich bekanntlich für eindeutige Lösungen *endlicher* Systeme gegenseitig rekursiver Gleichungen. Als eine geeignete mathematische Umgebung für diese Studien haben sich die lokal endlich präsentierbaren Kategorien von Gabriel und Ulmer [GU71] herausgestellt. Beispiele hiervon sind Mengen, halbgeordnete Mengen, Graphen, endliche Varietäten von Algebren usw. In dieser Umgebung führen wir iterative Algebren eines *finitären* Endofunktors ein. Es zeigt sich, dass alle vollständig iterativen Algebren auch iterativ sind. Hingegen formen rationale Bäume für eine Signatur Σ eine iterative Algebra, die nicht vollständig iterativ ist.

Satz 3.1. *[AMV06a] Für einen finitären Endofunktor H lässt sich eine frei erzeugte iterative Algebra RX auf einem Objekt X als Kolimes aller Koalgebren des Endofunktors $H(-) + X$, die eine endliche Trägermenge haben, konstruieren.*

Die Existenz freier iterativer Algebren folgt aus allgemeinen Ergebnissen der Kategorientheorie. Die in dem obigen Ergebnis auch enthaltene Konstruktion ist hingegen nicht trivial und überaus wichtig. Alle folgenden Ergebnisse über Iterativität hängen von ihr ab. Darüberhinaus erlaubt die Konstruktion, konkrete Beschreibungen der freien iterativen Algebren zu gewinnen: rationale Bäume im Falle von Polynomialfunktoren, reguläre Sprachen im Falle des Funktors, der Automaten beschreibt, oder periodische Streams. Das Hauptergebnis von [AMV03, AMV06a] ist der folgende

Satz 3.2. *Für jeden finitären Endofunktor H einer lokal endlich präsentierbaren Kategorie ist die Monade der frei erzeugten iterativen Algebren die von H frei erzeugte iterative Monade.*

In dem Spezialfall, der durch Polynomialfunktoren für Signaturen gegeben wird, ist dies genau das klassische Ergebnis von Nelson. Unser Beweis ist viel kürzer als Elgots ursprünglicher und er ist auch konzeptionell viel einfacher als irgendeiner der vorhergehenden Beweise.

Nach diesen mathematischen Grundlagen kommen wir nun zum zweiten Teil der Ergebnisse der Dissertation, die die Semantik rekursiver Funktionsdefinitionen betreffen.

4 Elgot-Algebren

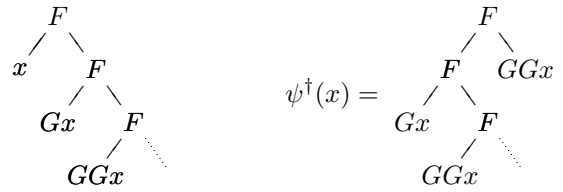
Das klassische Gebiet der algebraischen Semantik, siehe zum Beispiel [Gu81], beschäftigt sich mit der Lösung von so genannten rekursiven Programmschemas, bei denen es sich um Systeme gegenseitig rekursiver formaler Funktionsdefinitionen handelt. Zum Beispiel werden in dem System

$$\varphi(x) \approx F(x, \varphi(Gx)) \qquad \psi(x) \approx F(\varphi(Gx), GGx) \tag{4.1}$$

die Operationen φ und ψ rekursiv aus den gegebenen Operationen F und G definiert. Zur Vereinfachung werden hier immer die Daten von der eigentlichen Rekursion getrennt. Deshalb unterscheidet man auch zwischen uninterpretierter und interpretierter Semantik eines

rekursiven Programmschemas. Die uninterpretierte Semantik verleiht jedem Programmschema eine universelle Bedeutung, die völlig unabhängig von den Daten ist, auf denen die gegebenen Operationen arbeiten. Man betrachtet daher das Programmschema als rein syntaktisches Konstrukt, und die uninterpretierte Semantik liefert für jedes neu definierte Operationssymbol einen Baum, der sich durch schrittweises „Abrollen“ der rekursiven Definition ergibt. Das obige rekursive Programmschema (4.1) hat als uninterpretierte Lösung die unendlichen Bäume

$$\varphi^\dagger(x) = \quad \psi^\dagger(x) = \tag{4.2}$$



Bei der interpretierten Semantik ist hingegen zusätzlich zu dem Programmschema noch eine geeignete „Datenalgebra“ gegeben, die allen gegebenen Operationssymbolen, welche man zur Signatur Σ zusammenfasst, eine Bedeutung mitgibt. So ergibt das Programm (1.1) das Programmschema

$$f(n) \approx \mathsf{ifzero}(n, \mathsf{one}, f(\mathsf{pred}(n)) * n)) \,, \tag{4.3}$$

wobei one, pred und $*$ als die üblichen Operationen auf natürlichen Zahlen intrepretiert werden und $\mathsf{ifzero}(k, n, m)$ als die Funktion, die n zurückgibt, falls $k = 0$ gilt und m sonst. Diese vier Operationen formen Σ. Im Allgemeinen soll ein Programmschema also neue Operationen auf der gegebenen Datenalgebra definieren. Bevor man sich also überhaupt mit der Semantik von (interpretierten) rekursiven Programmschemas befassen kann, ergibt sich somit die Frage:

Welche Σ-Algebren sind geeignet für Semantik?

Mit anderne Worten, in welchen Σ-Algebren lassen sich rekursive Programmschemas lösen?

Wie wir bereits erwähnten, sind in der Literatur verschiedene Antworten vorgeschlagen worden. In der klassichen Theorie arbeitet man mit stetigen Algebren, die von vollständigen Halbordnungen (CPOs) getragen werden, siehe [Gu81]. Man kann aber auch mit Algebren auf vollständigen metrischen Räumen arbeiten. Möchte man indes zusätzliche Struktur vermeidem, bieten sich vollständig iterative Algebren an. Diese umfassen die vollständig metrisierten, allerdings nicht die stetigen Algebren. Aus diesem Grunde haben Bloom und Ésik [BÉ93] Iterationsalgebren vorgeschlagen, deren Ziel es ist, alle Gleichungseigenschaften der stetigen Algebren axiomatisch zu erfassen.

Analysiert man nun alle diese Typen von Algebren, stellt man fest, dass es eine entscheidende Gemeinsamkeit gibt, die die Lösung von rekursiven Programmschemas ermöglicht. Diese Algebren erlauben die *Auswertung* von beliebigen Σ-Bäumen. Das heißt man arbeitet immer mit einer Algebra A, die eine kanonische Auswertungsfunktion $T_\Sigma A \to A$ besitzt, die jedem endlichen oder unendlichen Σ-Baum (aus gegebenen Operationen und Parametern aus A) seinen Wert in A zuordnet. Unsere Untersuchung in [AMV06b] zeigt, dass

die Auswertungsfunktion kompatibel mit der syntaktischen Substitution von Σ-Bäumen für Parameter (in den Blättern) sein muss. Mathematisch hat diese Eigenschaft eine präzise Formulierung: wir interessieren uns für die Eilenberg-Moore-Algebren der Monade T_Σ, vgl. Satz 2.2. Uns ist es gelungen, eine sehr einfache Charakterisierung dieser Algebren zu erhalten.

Anstatt mit Mengen, Signaturen und Bäumen arbeiten wir wieder mit allgemeinen Kategorien, Endofunktoren und terminalen Koalgebren. Für einen Endofunktor führen wir *vollständige Elgot-Algebren* ein. Sie sind dadurch definiert, dass man für jedes rekursive Gleichungssystem einer gewissen sehr einfachen Art (die bereits genannten *flachen* Gleichungen) eine Lösung auswählt, und diese Auswahl muss zwei einfachen und intuitiv klaren Axiomen genügen. Alle oben diskutierten Typen von Algebren sind Beispiele von vollständigen Elgot-Algebren. Unser Begriff stellt damit so etwas wie die Essenz dessen dar, was eine geeignete Datenalgebra für Semantik rekursiver Programmschemas ist.

Satz 4.1. *[AMV06b] Es sei H ein Endofunktor mit genug terminalen Koalgebren. Dann ist die Kategorie der vollständigen Elgot-Algebren isomorph zur Kategorie der Eilenberg-Moore-Algebren für die freie vollständig iterative Monade T über H (vgl. Satz 2.2).*

Als Variation haben wir auch (nicht vollständige) Elgot-Algebren eingeführt, in denen eine Auswahl von Lösungen *endlicher* flacher Gleichungssysteme gefordert wird, die wieder denselben Axiomen genügt.

Satz 4.2. *[AMV06b] Es sei H ein finitärer Endofunktor auf einer lokal endlich präsentierbaren Kategorie. Dann ist die Kategorie der Elgot-Algebren isomorph zur Kategorie der Eilenberg-Moore Algebren für die freie iterative Monade R über H (vgl. Satz 3.2).*

5 Rekursive Programmschemas

In dem gemeinsamen Artikel [MM06] mit Moss haben wir uns schließlich der Semantik rekursiver Programmschemas gewidmet. Wir haben damit begonnen, die klassische Theorie der algebraischen Semantik mit unseren kategorientheoretischen Methoden aufzuarbeiten. Wie schon zuvor ersetzen wir Mengen, Signaturen und Bäume durch Kategorien, Endofunktoren und terminale Koalgebren. In dieser Umgebung formulieren wir den (geeignet verallgemeinerten) Begriff des rekursiven Programmschemas und seiner uninterpretierten und interpretierten Lösungen. Unsere Hauptergebnisse liefern für jedes *geschützte*[1] rekursive Programmschema folgende Aussagen:

(a) Eine eindeutige uninterpretierte Lösung existiert.
(b) In jeder vollständigen Elgot-Algebra existiert eine kanonische interpretierte Lösung.
(c) In jeder vollständig iterativen Algebra existiert eine eindeutige interpretierte Lösung.
(d) Uninterpretierte Lösungen aus (a) und interpretierte aus (b) bzw. (c) sind konsistent.

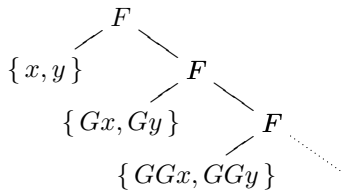
Diese Ergebnisse verallgemeinern sämtliche vorherigen Arbeiten zu diesem Thema. Dies ermöglicht es uns beispielsweise, Semantik für rekursive Funktionsdefinitionen anzugeben, die mit klassischen Methoden nicht behandelt werden können. Dies betrifft zum Bei-

[1] Geschütztheit (engl.: guardedness) ist eine einfache syntaktische Bedingung, die durch eine ähnliche Bedingung (die Greibach-Normalform) aus der klassischen Theorie inspiriert ist.

spiel Operationen, die bestimmte Gleichungen erfüllen sollen. Als konkretes Beispiel betrachten wir das uninterpretierte Programmschema

$$\varphi(x, y) \approx F(x, y, \varphi(Gx, Gy)),$$

wobei wir festlegen, dass die gegebene Operation F in jeder möglichen Datenalgebra die Gleichung $F(x, y, z) = F(y, x, z)$ erfüllen muss. Dann definiert φ eine kommutative Operation, also mit $\varphi(x, y) = \varphi(y, x)$. Diese Festlegung für F und die Forderung, dass φ kommutativ ist, können in unserer Theorie fest in das Programmschema eingebaut werden, so dass jede (interpretierte) Lösung immer kommutativ ist. Diese Tatsache wird übrigens bereits durch die uninterpretierte Lösung des obigen Programmschemas reflektiert. Man erhält den Baum



wobei für jeden mit F markierten Knoten die Reihenfolge der ersten beiden Kinder nicht unterscheidbar ist; dies machen wir hier mit Mengenklammern deutlich.

Weiterhin kann man aus unserem Ergebnissen über interpretierte Lösungen die klassische denotationelle Semantik, die stetige Algebren benutzt, zurückgewinnen. Man arbeitet hierbei wieder mit Polynomialfunktoren auf Mengen, die durch Signaturen gegeben werden. Damit erhält man dann zum Beispiel die Fakultätsfunktion als interpretierte Lösung des Programmschemas (4.3).

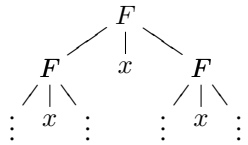
Die Verwendung von vollständigen metrischen Räumen liefert neue Anwendungen, die bisher gar nicht im Zusammenhang mit Arbeiten über die Semantik rekursiver Funktionsdefinitionen im Bereich der theoretischen Informatik erschienen. Als Beispiel betrachte die erste Gleichung in dem rekursiven Programmschema (4.1) zusammen mit der Datenalgebra, die durch das Interval $[0, 1]$ mit den Operationen $F(x, y) = \frac{x+y}{4}$ und $G(x) = \frac{1}{2}\sin(x)$ gegeben ist. Dann ist laut unseren Ergebnissen die interpretierte Lösung eine eindeutig bestimmte kontraktive Funktion $f : [0, 1] \to [0, 1]$, die folgende Gleichung erfüllt:

$$f(x) = \frac{1}{4}\left(x + f\left(\frac{1}{2}\sin x\right)\right).$$

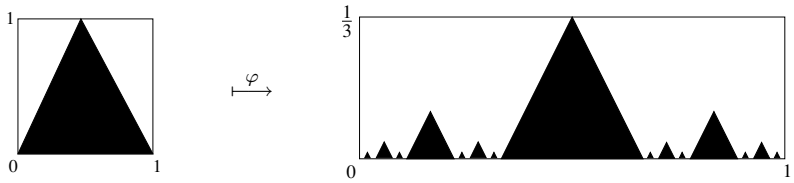
Abschließend nennen wir noch ein Beispiel aus dem Bereich der Fraktale. Man betrachte das rekursive Programmschema

$$\varphi(x) \approx F(\varphi(x), x, \varphi(x))$$

mit der uninterpretierten Lösung



Als Datenalgebra wählen wir den vollständigen metrischen Raum der nicht leeren abgeschlossenen Mengen des Einheitsquadrates $[0,1] \times [0,1]$ mit einer geigneten Operation F. Die eindeutige interpretierte Lösung ist eine kontraktive Funktion φ, die eine abgeschlossene Teilmenge c des Einheitsquadrates auf ein Fraktal $\varphi(c)$ abbildet, das aus drei Teilen wie folgt aufgebaut ist: der mittlere Teil ist eine Kopie von c, die um $\frac{1}{3}$ skaliert wurde und der rechte und linke Teil sind je eine Kopie der ganzen Menge $\varphi(c)$, die ebenfalls jeweils mit dem Faktor $\frac{1}{3}$ skaliert wurden. Als Beispiel betrachte man die folgende Abbildung:



6 Zusammenfassung

Die hier präsentierten Ergebnisse der Dissertation [Mi05b] tragen wesentlich zu einem neuen koalgebraischen Verständnis der Semantik von Rekursion bei, das in den letzten Jahren von verschiedenen Forschern entwickelt wurde. Die Semantik rekursiver Programmschemas ist ein zentrales Thema in der theoretischen Informatik. Unsere Ergebnisse zeigen, wie man diese Semantik auf eine konzeptuell klare und allgemeine Weise mit koalgebraischen Methoden behandeln kann.

Unser Zugang mit Hilfe von Methoden der Theorie der Koalgebren und der Kategorientheorie ergibt dabei zunächt einmal Verallgemeinerungen bekannter mathematischer Ergebnisse. Dies führt dann aber auch zu neuen Werkzeugen für die Semantik, und man bekommt auch grundlegende neue Einsichten in die Mechanismen, die der Semantik von Rekursion zugrunde liegen. Unsere Anwendungsbeispiele zeigen, dass wir eine einheitliche Theorie für die Lösung einer großen Klasse rekursiver Gleichungen entwickelt haben, die sowohl die klassische algebraische Semantik als auch neue Anwendungen umfasst.

Literatur

[Ac03] Peter Aczel, Jiří Adámek, Stefan Milius and Jiří Velebil, Infinite Trees and Completely Iterative Theories: A Coalgebraic View, *Theoret. Comput. Sci.* 300 (2003), 1–45.

[Ad05] Jiří Adámek, Introduction to Coalgebra, *Theory Appl. Categ.* 14 (2005), 157–199.

[AMV03] Jiří Adámek, Stefan Milius and Jiří Velebil, Free Iterative Theories: a coalgebraic view, *Math. Structures Comput. Sci.* 13 (2003), no. 2, 259–320.

[AMV06a] Jiří Adámek, Stefan Milius and Jiří Velebil, Iterative Algebras at Work, akzeptiert zur Veröffentlichung in *Math. Structures Comput. Sci.*, verfügbar unter `http://www.stefan-milius.eu`, Kurzfassung erschienen in *Electron. Notes Theor. Comput. Sci.* 106 (2004), 3–24.

[AMV06b] Jiří Adámek, Stefan Milius and Jiří Velebil, Elgot Algebras, *eingereicht* (2006), verfügbar unter `http://www.stefan-milius.eu`, Zusammenfassung erscheint demnächst in *Electron. Notes Theor. Comput. Sci.*

[BE74] Stephen L. Bloom and Calvin C. Elgot, The Existence and Construction of Free Iterative Theories, *J. Comput. System Sci.* 12 (1974), 305–318.

[BÉ93] Stephen L. Bloom and Zoltán Ésik, *Iteration Theories: The equational logic of iterative processes,* EATCS Monographs on Theoretical Computer Science, Berlin: Springer-Verlag (1993).

[El75] Calvin C. Elgot, Monadic Computation and Iterative Algebraic Theories, in: *Logic Colloquium '73* (eds: H. E. Rose and J. C. Shepherdson), North-Holland Publishers, Amsterdam, 1975.

[Co83] Bruno Courcelle, Fundamental properties of infinite trees, *Theoret. Comput. Sci.* 25 (1983), Nr. 2, 95–169.

[EBT78] Calvin C. Elgot, Stephen L. Bloom and Ralph Tindell, On the Algebraic Structure of Rooted Trees, *J. Comput. System Sci.* 16 (1978), 361–399.

[GU71] Peter Gabriel and Friedrich Ulmer, *Lokal präsentierbare Kategorien*, Lecture N. Math. 221, Springer-Verlag, Berlin 1971.

[Gi79] Susanna Ginali, Regular trees and the free iterative theory, *J. Comput. System Sci.* 18 (1979), 228–242.

[Gu81] Irène Guessarian, *Algebraic Semantics*. Lecture Notes in Comput. Sci. 99, Springer, 1981.

[Gu99] Heinz-Peter Gumm, Elements of the General Theory of Coalgebras, LUATCS'99, Rand Africaans University, Johannesburg, South Africa, 1999.

[JR97] Bart Jacobs and Jan J. M. M. Rutten, A Tutorial on (Co)Algebras and (Co)Induction, *Bull. Eur. Assoc. Theor. Comput. Sci. EATCS* 62 (1997), 222–259.

[La63] F. William Lawvere, *Functorial Semantics of Algebraic Theories*, PhD thesis, Columbia University, 1963.

[Mi02] Stefan Milius, On Iteratable Endofunctors, *Electron. Notes Theor. Comput. Sci.* 69 (2002), 18 S.

[Mi05a] Stefan Milius, Completely Iterative Algebras and Completely Iterative Monads, *Inform. and Comput.* 196 (2005), 1–41.

[Mi05b] Stefan Milius, Coalgebras, Monads and Semantics, Dissertation, Technische Universität Braunschweig, 2005.

[MM06] Stefan Milius und Lawrence S. Moss, The Category Theoretic Solution of Recursive Program Schemes, akzeptiert zur Veröffentlichung in *Theoret. Comput. Sci.*, Zusammenfassung erschien in *Lecture Notes in Comput. Sci.* 3629 (2005), 293–312.

[Ne83] Evelyn Nelson, Iterative Algebras, *Theoret. Comput. Sci.* 25 (1983), 67–94.

[Ru00] Jan J. M. M. Rutten, Universal coalgebra, a theory of systems, *Theoret. Comput. Sci.* 249 (2000), Nr. 1, 3–80.

[Ti80] Jerzy Tiuryn, Unique Fixed Points vs. Least Fixed Points, *Theoret. Comput. Sci.* 12 (1980), 229–254.

Stefan Milius wurde am 3. Juni 1975 in Magdeburg geboren. Dort legte er 1994 sein Abitur am Werner-von-Siemens-Gymnasium (bis 1992 war dies die Spezialschule mathematisch-naturwissenschaftlich-technischer Richtung) ab. Nach dem Zivildienst bei den evangelischen Stiftungen Neuerkerode folgte von 1995 bis 2000 das Studium der Informatik an der Technischen Universität Braunschweig, das er mit dem Diplom abschloss. Für seine Diplomarbeit über „Relationen in Kategorien“ erhielt er den Preis für hervorragende Leistungen im Studium der TU Braunschweig. Zwischendurch studierte er von 1999 bis 2000 an der York University in Toronto ON, Kanada, reine Mathematik, worin er einen Master of Arts ablegte. Seit 2000 ist er wissenschaftlicher Mitarbeiter am Institut für Theoretische Informatik der TU Braunschweig, wo er im Jahr 2005 promovierte. Von seinen 16 in namhaften Fachzeitschriften bzw. Tagungsbänden veröffentlichten Artikeln und den 11 teilweise bereits eingereichten Manuskripten bilden 7 der ersteren Artikel die Grundlage seiner kumulativen Promotion. Auch hat er sich durch mehr als 25 Vorträge über seine Forschungsarbeit bei internationalen Konferenzen einen Namen gemacht.