Semantik rekursiver Funktionsdefinitionen mit Striktheitsinformation

Klaus Indermark
Lehrstuhl für Informatik II
RWTH Aachen

17. Dezember 1998

Zusammenfassung

Die Eindeutigkeit rekursiver Funktionsdefinitionen erfordert eine Festlegung des Verhaltens bei nicht-definierten Funktionsargumenten. Unter Berücksichtigung dieser Zusatzinformation entwickeln wir eine algebraische Darstellung von Fixpunkt- und Reduktionssemantik und beweisen ihre Äquivalenz. Wir stellen ferner eine deterministische Auswertungsstrategie vor, welche die bekannten call-by-name und call-by-value Strategien verallgemeinert.

1 Einleitung

Rekursive Definitionen können sowohl als Gleichungssysteme wie auch als Termersetzungssysteme aufgefaßt werden und spielen daher als ausführbare Spezifikationen eine zentrale Rolle in der Informatik. Dies gilt besonders für die rekursive Definition von Funktionen in funktionalen Programmiersprachen. Dabei erfordert eine vollständige Spezifikation die Hinzunahme geeigneter Striktheitsinformationen für Funktionsargumente. Wir steuern diese Informationen durch sogenannte Striktheitsindizes und erreichen so eine uniforme Behandlung verschiedener Auswertungsstrategien. Im Mittelpunkt dieser Arbeit steht ein algebraischer Beweis für die Äquivalenz zwischen denotationaler Fixpunktsemantik und operationaler Reduktionssemantik von rekursiven Funktionsdefinitionen mit Striktheitsinformation. Er vereinheitlicht und vereinfacht entsprechende Beweise für Definitionen mit call-by-name bzw. call-by-value Semantik.

Zur Motivation unseres Vorgehens betrachten wir das folgende Beispiel einer rekursiven Funktionsdefinition über der Menge $\mathbb{N}$ der natürlichen Zahlen:

$$\begin{array}{lcl} F(x) & = & G(H(x), H(x+3)) \\ G(x,y) & = & \mathsf{if}\ x \leq 3\ \mathsf{then}\ x\ \mathsf{else}\ y \\ H(x) & = & \mathsf{if}\ x = 3\ \mathsf{then}\ 0\ \mathsf{else}\ H(x+1) \end{array}$$

1.1 Operationale Semantik

Zunächst interpretieren wir die Gleichungen als Termersetzungsregeln. Der nichtdeterministische Charakter der dadurch bestimmten Reduktionsrelation erfordert für eine Implementierung die Auswahl einer geeigneten Auswertungsreihenfolge. Wir wählen drei Auswertungsstrategien und zeigen ihr unterschiedliches Verhalten bei der Berechnung von $F(1)$.

1.1.1 Call-by-value Auswertung

Alle Funktionsargumente werden vor Ausführung des Funktionsaufrufs ausgewertet. Dies ergibt für die Berechnung von $F(1)$ einen unendlichen Reduktionsprozeß:

$$\begin{array}{rcl} F(1) & \Longrightarrow & G(H(1), H(1+3)) \\ & \stackrel{*}{\Longrightarrow} & G(H(2), H(1+3)) \\ & \stackrel{*}{\Longrightarrow} & G(H(3), H(1+3)) \\ & \stackrel{*}{\Longrightarrow} & G(0, H(4)) \\ & \stackrel{*}{\Longrightarrow} & G(0, H(5)) \\ & \vdots & \end{array}$$

Wir schließen daraus, daß $F(1)$ nicht definiert ist.

1.1.2 Call-by-name Auswertung

Funktionsaufrufe werden zuerst ausgeführt, so daß Argumente nur bei Bedarf ausgewertet werden:

$$\begin{array}{rcl} F(1) & \Longrightarrow & G(H(1), H(1+3)) \\ & \stackrel{*}{\Longrightarrow} & \mathsf{if}\ H(1) \leq 3\ \mathsf{then}\ H(1)\ \mathsf{else}\ H(1+3) \\ & \stackrel{*}{\Longrightarrow} & \mathsf{if}\ H(1+1) \leq 3\ \mathsf{then}\ H(1)\ \mathsf{else}\ H(1+3) \\ & \stackrel{*}{\Longrightarrow} & \mathsf{if}\ H((1+1)+1) \leq 3\ \mathsf{then}\ H(1)\ \mathsf{else}\ H(1+3) \\ & \stackrel{*}{\Longrightarrow} & H(1) \\ & \stackrel{*}{\Longrightarrow} & H(1+1) \\ & \stackrel{*}{\Longrightarrow} & H((1+1)+1) \\ & \stackrel{*}{\Longrightarrow} & 0 \end{array}$$

In diesem Fall ergibt sich $F(1) = 0$.

1.1.3 Gemischte Auswertung

Die mangelnde Effizienz der call-by-name Auswertung ist offensichtlich: $H(1)$ wird zweimal ausgewertet wegen des doppelten Vorkommens von x im definierenden Term von G. Da aber das erste Argument von G in jedem Fall benötigt wird, kann es auch vor dem Aufruf von G berechnet und dadurch eine wiederholte Berechnung vermieden werden.

$$\begin{aligned} F(1) &\Longrightarrow G(H(1), H(1+3)) \\ &\overset{*}{\Longrightarrow} G(H(1+1), H(1+3)) \\ &\overset{*}{\Longrightarrow} G(H((1+1)+1), H(1+3)) \\ &\overset{*}{\Longrightarrow} G(0, H(1+3)) \\ &\overset{*}{\Longrightarrow} 0 \end{aligned}$$

Diese gemischte Strategie, bei der nur die Berechnung des zweiten Arguments von G verzögert wird, erlaubt eine kürzere Berechnung als die call-by-name Auswertung, ohne die Termination der Auswertung zu gefährden.

1.2 Denotationale Semantik

Wir wollen nun diese operationalen Unterschiede auf einem abstrakteren denotationalen Niveau modellieren. Dazu betrachten wir die rekursive Definition als Gleichungssystem und stellen geeignete Lösungsstrategien vor. Die zweite Gleichung

$$G(x,y) \;=\; \mathsf{if}\ x \leq 3\ \mathsf{then}\ x\ \mathsf{else}\ y$$

scheint genau eine Lösung zu besitzen, nämlich

$$g : \mathbb{N}^2 \to \mathbb{N} \quad \text{mit} \quad g(a,b) = \begin{cases} a & \text{falls} \quad 0 \leq a \leq 3 \\ b & \text{sonst} \end{cases} .$$

Demgegenüber hat die dritte Gleichung

$$H(x) \;=\; \mathsf{if}\ x = 3\ \mathsf{then}\ 0\ \mathsf{else}\ H(x+1)$$

mehrere Lösungen, und zwar für jedes $k \in \mathbb{N}$

$$h_k : \mathbb{N} \to \mathbb{N} \quad \text{mit} \quad h_k(a) = \begin{cases} 0 & \text{falls} \quad 0 \leq a \leq 3 \\ k & \text{sonst} \end{cases}$$

und außerdem die partielle Funktion

$$h : \mathbb{N} \dashrightarrow \mathbb{N} \quad \text{mit} \quad h(a) = \begin{cases} 0 & \text{falls} \quad 0 \leq a \leq 3 \\ \text{nicht definiert} & \text{sonst} \end{cases} .$$

Offensichtlich ist nur die partielle Funktion h ein vernünftiger Lösungskandidat, da sich die übrigen Lösungen nicht aus der Gleichung berechnen lassen: jedes Argument $a \in \mathbb{N}$ mit $a \geq 3$ ergibt eine unendliche Berechnung. Zur Lösung der noch verbleibenden ersten Gleichung

$$F(x) \;=\; G(H(x), H(x+3))$$

braucht man lediglich die Lösungen g und h entsprechend zu komponieren und erhält

$$f : \mathbb{N} \dashrightarrow \mathbb{N} \quad \text{mit} \quad f(a) = g(h(a), h(a+3)).$$

Hier ergibt sich nun der entscheidende Spielraum, weil die Komposition partieller Funktionen unterschiedlich definiert werden kann. Und mit diesen Unterschieden lassen sich die oben vorgestellten operationalen Varianten auf einer denotationalen Ebene genau modellieren. Technisch gesehen, wählen wir ein neues Element $\perp$ zur Darstellung eines nicht-definierten Werts. Eine partielle Funktion $\varphi : \mathbb{N}^n \dashrightarrow \mathbb{N}$ kann dann als totale Funktion $\overline{\varphi} : \mathbb{N}^n \to \mathbb{N}_\perp$ mit $\mathbb{N}_\perp := \mathbb{N} \cup \{\perp\}$ aufgefaßt werden. Um solche Funktionen zu komponieren, müssen geeignete Erweiterungen $\widehat{\varphi} : {\mathbb{N}_\perp}^n \to \mathbb{N}_\perp$ konstruiert werden, die das Verhalten bei nicht-definierten Argumenten beschreiben.

1.2.1 Strikte Erweiterung

Wir fordern zunächst, daß ein nicht-definiertes Argument direkt einen nicht-definierten Funktionswert impliziert. In diesem Fall erfolgt keine Auswertung des definierenden Terms der Funktion. In unserem Beispiel erhalten wir

$$\begin{aligned} f_s(1) &= g_s(h_s(1), h_s(4)) \\ &= g_s(0, \perp) \\ &= \perp \end{aligned}$$

wegen der Striktheit von $g_s : {\mathbb{N}_\perp}^2 \to \mathbb{N}_\perp$ in beiden Argumenten. Strikte Erweiterung erweist sich als semantisch äquivalent zur call-by-value Auswertung.

1.2.2 Nicht-strikte Erweiterung

Nicht-definierte Argumente werden ebenso behandelt wie definierte und in den Funktionsterm zur Auswertung kopiert. Möglicherweise erhält man auch bei dieser Lösungsstrategie ein nicht-definiertes Ergebnis. Es kann aber sein, daß der Funktionsterm nicht alle Argumentvariablen enthält oder bei Verzweigung nicht-definierte Argumente überspringt und so einen definierten Funktionswert liefert. Für unser Beispiel folgt in diesem Fall

$$\begin{aligned} f_n(1) &= g_n(h_n(1), h_n(4)) \\ &= g_n(0, \perp) \\ &= 0\ , \end{aligned}$$

weil $g_n(0, \bot) = (\mathsf{if}\ 0 \leq 3\ \mathsf{then}\ 0\ \mathsf{else}\ \bot) = 0$. Diese Erweiterung entspricht der call-by-name Auswertung.

1.2.3 Gemischte Erweiterung

Wie bei den Auswertungsstrategien müssen auch denotational nicht alle Argumente gleich behandelt werden. Wir können strikte und nicht-strikte Erweiterung mischen und für jedes Funktionsargument festlegen, ob es strikt behandelt wird oder zur Auswertung an den Funktionsterm übergeben wird. In unserem Beispiel können wir eine Lösungsfunktion

$$g_m : \mathbb{N}_\bot^{\,2} \longrightarrow \mathbb{N}_\bot$$

für die zweite Gleichung so konstruieren, daß das erste Argument strikt behandelt wird, während das zweite an den Funktionsterm übergeben wird. Man beachte, daß in diesem Beispiel $g_m = g_n$, obwohl sich die Erweiterungsmethoden unterscheiden. Diese Verhältnisse entsprechen genau der gemischten Auswertung.

Die Diskussion unseres Beispiels hat gezeigt, daß die Semantik einer rekursiven Funktionsdefinition erst durch Striktheitsinformationen für alle Funktionsargumente eindeutig festgelegt wird. Neben der Gleichbehandlung aller Argumente in strikter bzw. nicht-strikter Weise ist auch eine gemischte Methode möglich, bei welcher die Funktionsargumente unterschiedlich behandelt werden. Wir hatten zwar die gemischte Auswertung als Optimierung der call-by-name Auswertung eingeführt, wollen jedoch im folgenden die Striktheitsinformation der Funktionsargumente frei vorgeben. Der Einfachheit halber vereinbaren wir, daß die strikten Argumente einer Funktion den nicht-strikten vorausgehen. Durch geeignetes Vertauschen der Funktionsargumente auf beiden Seiten einer rekursiven Funktionsdefinition läßt sich dies stets erreichen. Die Striktheitsinformation einer n-stelligen Funktionsvariablen kann deshalb durch einen Index σ mit $0 \leq \sigma \leq n$ angegeben werden. Auf diese Weise gewinnt man eine vollständige Spezifikation von Funktionen mit einer eindeutig bestimmten Fixpunktsemantik und einer dazu äquivalenten Reduktionssemantik. Dieser Ansatz verallgemeinert und uniformisiert die bisherigen call-by-name und call-by-value Techniken.

1.3 Historische Entwicklung

In der Literatur finden sich zahlreiche Arbeiten zu der hier behandelten Thematik. Bereits Kleene [5] hat in seinem ersten Rekursionstheorem einen Zusammenhang zwischen operationaler Semantik und Fixpunktsemantik formuliert. Er beschränkte sich dabei auf strikte arithmetische Funktionen und benutzte eine andere, der call-by-value Auswertung entsprechende Berechnungsmethode. Die praktische Bedeutung rekursiver Funktionsdefinitionen wurde dann durch ihre Verwendung in Programmiersprachen offenkundig. Als frühes Beispiel sei die Sprache LISP erwähnt, deren theoretische Grundlagen McCarthy in [8] dargelegt hat. Die denotationale Methode, welche die Bedeutung eines Programms durch Induktion über seine syntaktische Struktur und nicht durch Berechnungen festlegt, geht auf Strachey [10] zurück. Scott [9] erkannte die Bedeutung topologischer Hilfsmittel und verwendete insbesondere stetige Funktionen auf vollständigen Halbordnungen. Die algebraische Natur des denotationalen Ansatzes wurde wohl am deutlichsten von Goguen, Thatcher, Wagner und Wright in [4] beschrieben. Den Zusammenhang zwischen Fixpunktsemantik und Auswertungsstrategien haben Manna, Cadiou und Vuillemin [6, 1, 7] ausführlich untersucht. Sie ließen jedoch nur den call-by-name Fixpunkt zu und sahen im Vergleich dazu die call-by-value Auswertung als inkorrekt an. Daß sich auch für diese eine Fixpunktsemantik angeben läßt, hat de Bakker in [3] gezeigt. Er wählte den auch hier vertretenen allgemeinen Ansatz der Spezifikation strikter und nicht-strikter Funktionsargumente, geht aber bei der Festlegung der operationalen Semantik einen induktiven, ergebnisorientierten Weg, der heute auch als natürliche Semantik bezeichnet wird und der denotationalen Methode näher steht. Winskel führt in seinem Buch über Semantik [11] die entsprechenden Äquivalenzbeweise noch einmal explizit für die Spezialfälle strikter call-by-value und nicht-strikter call-by-name Semantik durch. Im Unterschied zu diesen Arbeiten ist die hier gewählte operationale Semantik eine Einzelschritt-Semantik, die an einer Stackimplementierung orientiert ist und den Nachweis der Compilerkorrektheit unterstützen soll. Auf dieser Basis hat Chitil in [2] entsprechende Äquivalenzresultate für Konstruktor-basierte Gleichungssysteme erzielt.

1.4 Gliederung der Arbeit

Die beiden folgenden Abschnitte enthalten die algebraischen und ordnungstheoretischen Grundlagen. Sie dienen der terminologischen Präzisierung und bereiten die Äquivalenzbeweise vor. Im Hauptabschnitt über rekursive Funktionsdefinitionen werden unter Berücksichtigung der Striktheitsinformation drei Semantiken vorgestellt und ihre Äquivalenz bewiesen: eine Fixpunktsemantik, eine nichtdeterministische Reduktionssemantik und eine deterministische Auswertungsstrategie.

2 Algebraische Grundlagen

Die algebraische Sichtweise von Programmen erlaubt es, diese als Elemente einer frei erzeugten Algebra zu verstehen und so ihre syntaktische Struktur unabhängig von einer konkreten Darstellung festzulegen. Damit läßt sich die Semantik ohne Verwendung von Berechnungen durch Homomorphismen, also durch Induktion über die syntaktische Struktur beschreiben, so daß auch die Äquivalenzbeweise transparent mit Hilfe struktureller Induktion geführt werden können. Nach Darstellung der algebraischen Grundbegriffe stellen wir das Vertauschungslemma für Substitutionen vor, welches ein wichtiges Bindeglied für den Zusammenhang von operationaler und denotationaler Semantik bildet. Die abschließend eingeführten Termfunktionen dienen als technisches Hilfsmittel für die Semantik rechter Gleichungsseiten.

Da in rekursiven Funktionsdefinitionen Daten verschiedenen Typs wie etwa Wahrheitswerte und Zahlen auftreten, gehen wir von mehreren Basismengen aus. Der Einfachheit halber nehmen wir diese als disjunkt an und fassen sie zu einer sortierten Menge zusammen.

Definition (Sortierte Menge)
Sei A eine Menge und $\mathsf{sort} : A \to S$ eine Abbildung von A in eine nicht-leere Menge S von *Sorten*. A heißt dann *S-sortiert*, und für $s \in S$ bezeichnet $A^s := \mathsf{sort}^{-1}(s)$ die Teilmenge von Elementen der Sorte s. Allgemeiner bezeichnet A^w für beliebiges $w \in S^*$ ein cartesisches Produkt, welches definiert ist durch

$$\begin{array}{rrcl} & A^\varepsilon & := & \{()\} \\ \text{und} & A^{s_1 \ldots s_n} & := & A^{s_1} \times \ldots \times A^{s_n} \quad \text{für} \quad n > 0. \end{array}$$

Eine Abbildung $f : A \to B$ zwischen S-sortierten Mengen heißt *sortentreu*, falls $f(A^s) \subseteq B^s$ für alle $s \in S$. Eine solche Abbildung f induziert für jedes $w \in S^*$ die kanonische Fortsetzung $f : A^w \to B^w$ mit

$$\begin{array}{rrcl} & f() & := & () \\ \text{und} & f(a_1, \ldots, a_n) & := & (f(a_1), \ldots, f(a_n)) \end{array} \qquad \triangleleft$$

Wir setzen im folgenden voraus, daß alle verwendeten Abbildungen zwischen S-sortierten Mengen sortentreu sind, ohne dies jeweils explizit zu erwähnen.

Als syntaktische Basis rekursiver Funktionsdefinitionen wählen wir neben einer Menge von Sorten als Grundtypen eine geeignet sortierte Menge von Funktionssymbolen.

Definition (Signatur)
Sei S eine Menge von *Grundtypen*, $D(S) := S^* \times S$ die Menge der *Funktionstypen* von S und $\mathcal{F}$ eine $D(S)$-sortierte Menge von *Funktionssymbolen* mit der Teilmenge $\mathcal{C} := \bigcup_{s \in S} \mathcal{F}^{(\varepsilon, s)}$ der *Konstantensymbole* . Dann heißt $\Sigma = \langle S, \mathcal{F} \rangle$ eine *Signatur*. $\triangleleft$

Wir erklären die Semantik einer Signatur durch eine algebraische Struktur, indem wir Grundtypen als Mengen und Funktionssymbole als Funktionen über diesen Mengen interpretieren.

Definition (Funktion über A)
Sei A eine S-sortierte Menge und $\tau = (w, s) \in D(S)$ ein Funktionstyp. Dann heißt eine Abbildung $f : A^w \to A^s$ eine *Funktion über* A vom Typ τ. Wir bezeichnen mit $\mathbf{F}(A)$ die Menge der Funktionen über A und mit $\mathbf{F}^\tau(A)$ die Teilmenge der Funktionen vom Typ τ. $\lhd$

Definition (Σ-Algebra)
Ist $\Sigma = \langle S, \mathcal{F} \rangle$ eine Signatur, A eine S-sortierte Menge und $\alpha : \mathcal{F} \to \mathbf{F}(A)$ eine Abbildung, so heißt $\mathfrak{A} = \langle A; \alpha \rangle$ eine Σ-*Algebra*. Für $f \in \mathcal{F}$ wird $\alpha(f)$ eine *Grundfunktion* von $\mathfrak{A}$ genannt und auch durch $f_{\mathfrak{A}}$ bezeichnet. $\lhd$

Beachte, daß sowohl $\mathcal{F}$ als auch $\mathbf{F}(A)$ $D(S)$-sortiert sind und daher α vereinbarungsgemäß sortentreu ist. Also gilt für alle $\tau \in D(S)$: $\alpha(\mathcal{F}^\tau) \subseteq \mathbf{F}^\tau(A)$. Insbesondere wird ein Konstantensymbol $c \in \mathcal{F}^{(\varepsilon,s)}$ als konstante Grundfunktion $c_{\mathfrak{A}} : A^\varepsilon \to A^s$ mit $c_{\mathfrak{A}}() \in A^s$ interpretiert.

Ausgehend von einer Signatur konstruieren wir weitere syntaktische Objekte als Terme von Funktionssymbolen und Variablen. Diese lassen sich semantisch als abgeleitete Funktionen interpretieren. Dabei stellen wir Terme als Elemente einer frei erzeugten Algebra und ihre Semantik durch Homomorphismen dar.

Definition (Σ-Termalgebra)
Sei $\Sigma = \langle S, \mathcal{F} \rangle$ eine Signatur und X eine S-sortierte Menge von Variablen. Dann ist die Σ-*Termalgebra* über X

$$\mathfrak{T}_\Sigma(X) = \langle T_\Sigma(X); \alpha_T \rangle$$

wie folgt definiert:

- $T_\Sigma(X)$ ist die kleinste S-sortierte Menge von Termen, welche alle Variablen enthält und abgeschlossen ist unter freier Applikation der Funktionssymbole, d. h.,
  - $X \subseteq T_\Sigma(X)$ und
  - für alle $(w, s) \in D(S)$ und $f \in \mathcal{F}^{(w,s)}$ ist mit $(t_1, \dots, t_n) \in T_\Sigma(X)^w$ auch $f t_1 \dots t_n \in T_\Sigma(X)^s$.
- α_T interpretiert jedes $f \in \mathcal{F}^{(w,s)}$ als Konstruktorfunktion über Termen
  - $\alpha_T(f) : T_\Sigma(X)^w \to T_\Sigma(X)^s \quad \text{mit} \quad \alpha_T(f)(t_1, \dots, t_n) := f t_1 \dots t_n.$ $\lhd$

Diese Definition schließt für $w = \varepsilon$ den Spezialfall der Konstantensymbole ein. Es folgt für $c \in \mathcal{C}$, daß $c \in T_\Sigma(X)$ und $\alpha_T(c)() = c$.

Eine Termalgebra repräsentiert eine besondere algebraische Struktur auf eine bestimmte Weise, nämlich durch Zeichenreihen in Präfixnotation ohne Klammern und Kommata. Um ihre strukturellen Eigenschaften zu beschreiben, führen wir Homomorphismen ein. Diese strukturerhaltenden Abbildungen spielen eine zentrale Rolle bei der algebraischen Behandlung von Syntax und Semantik.

Definition (Homomorphismus)
Seien $\mathfrak{A} = \langle A; \alpha \rangle$ und $\mathfrak{B} = \langle B; \beta \rangle$ Σ-Algebren. Eine Abbildung $h : A \to B$ heißt ein *Homomorphismus*, bezeichnet durch $h : \mathfrak{A} \to \mathfrak{B}$, falls

$$h(f_{\mathfrak{A}}(\bar{a})) = f_{\mathfrak{B}}(h(\bar{a}))$$

für alle $(w, s) \in D(S)$, $f \in \mathcal{F}^{(w,s)}$ und $\bar{a} \in A^w$. $\lhd$

Offensichtlich gilt, daß die identische Abbildung id_A ein Homomorphismus von $\mathfrak{A}$ in sich ist, und daß Homomorphismen unter Komposition abgeschlossen sind.

Die wesentliche Eigenschaft einer Termalgebra besteht in ihrer freien Erzeugung. Das heißt, daß jedes Element in eindeutiger Weise durch Anwendung der Grundfunktionen aus einer „erzeugenden" Teilmenge gewonnen werden kann. Die folgende äquivalente Definition ist vorteilhafter für Beweise.

Definition (Frei erzeugte Algebra)
Sei $\mathfrak{A} = \langle A; \alpha \rangle$ eine Σ-Algebra und $X \subseteq A$. Dann heißt $\mathfrak{A}$ von X *frei erzeugt*, wenn jede Abbildung $\chi : X \to B$ in eine beliebige Σ-Algebra $\mathfrak{B} = \langle B; \beta \rangle$ eindeutig zu einem Homomorphismus

$$\overline{\chi} : \mathfrak{A} \to \mathfrak{B}$$

fortgesetzt werden kann. $\lhd$

Termalgebren bestätigen die Existenz frei erzeugter Algebren:

Satz (Existenz frei erzeugter Algebren)
Die Σ-Termalgebra $\mathfrak{T}_\Sigma(X)$ ist frei von X erzeugt.

Der Beweis beruht auf der eindeutigen Zerlegung von Termen in Teilterme und wird übergangen. Stattdessen zeigen wir das folgende Resultat über die Eindeutigkeit frei erzeugter Algebren. Es liefert die formale Basis für den Begriff der abstrakten Syntax.

Satz (Eindeutigkeit frei erzeugter Algebren)
Seien $\mathfrak{A} = \langle A; \alpha \rangle$ und $\mathfrak{B} = \langle B; \beta \rangle$ von X frei erzeugte Σ-Algebren. Dann sind $\mathfrak{A}$ und $\mathfrak{B}$ isomorph, d. h., es existiert ein bijektiver Homomorphismus $h : \mathfrak{A} \to \mathfrak{B}$.

Beweis: Da $X \subseteq A$ und $X \subseteq B$, kann die Identität id_X auf zweifache Weise als Inklusion aufgefaßt werden, nämlich als $\mathsf{in}_{X,A} : X \to A$ und als $\mathsf{in}_{X,B} : X \to B$. Wegen der freien Erzeugung lassen sich diese eindeutig zu Homomorphismen $\overline{\mathsf{in}_{X,A}} : \mathfrak{B} \to \mathfrak{A}$ und $\overline{\mathsf{in}_{X,B}} : \mathfrak{A} \to \mathfrak{B}$ fortsetzen. Dann ist die Komposition $\overline{\mathsf{in}_{X,A}} \circ \overline{\mathsf{in}_{X,B}} : \mathfrak{A} \to \mathfrak{A}$ ein Homomorphismus auf A, der mit dem identischen Homomorphismus auf A übereinstimmt, da beide die Inklusion $\mathsf{in}_{X,A}$ fortsetzen. Vertauscht man die Rollen von A und B, so folgt, daß $\overline{\mathsf{in}_{X,B}} \circ \overline{\mathsf{in}_{X,A}} : \mathfrak{B} \to \mathfrak{B}$ den identischen Homomorphismus auf B darstellt. Daher muß $\overline{\mathsf{in}_{X,B}}$ injektiv und surjektiv sein und liefert somit den gewünschten Isomorphismus. □

Abstrakte Syntax

Da die Semantik einer Programmiersprache, ebenso wie ihre Übersetzung in Maschinencode, nur von ihren Struktureigenschaften abhängt, können wir Programme syntaktisch als Elemente einer frei erzeugten Algebra ansehen. Bei dieser algebraischen Sicht ist nur noch der Aufbau eines Programms aus Komponenten von Bedeutung. Die konkrete Darstellung kann wegen der Isomorphie frei erzeugter Algebren beliebig gewählt werden.

Als Konsequenz dieser Auffassung ergibt sich die Möglichkeit, Programmeigenschaften durch *strukturelle Induktion* zu beweisen: für $M \subseteq T_\Sigma(X)$ gilt $M = T_\Sigma(X)$ genau dann, wenn $X \subseteq M$ und M unter allen Grundfunktionen $\alpha_T(f)$ abgeschlossen ist.

Algebraische Semantik

Die eindeutige homomorphe Fortsetzung $\overline{\chi} : \mathfrak{T}_\Sigma(X) \to \mathfrak{A}$ einer Abbildung $\chi : X \to A$ gestattet eine sehr einfache Definition der Programmsemantik. Sie ist kompositionell und erfordert keinen Berechnungsbegriff.

Definition
Sei $t \in T_\Sigma(X)$ und $\chi : X \to A$ eine Abbildung in eine Σ-Algebra $\mathfrak{A} = \langle A; \alpha \rangle$. Dann heißt

$$[\![t]\!]_{(\mathfrak{A},\chi)} := \overline{\chi}(t) \in A$$

die *algebraische Semantik* von t bzgl. der *Interpretation* $\mathfrak{A}$ und der *Belegung* χ. ◁

Beachte, daß wegen der Homomorphie von $\overline{\chi}$ folgende Gleichungen gelten. Sie beschreiben die Kompositionalität der algebraischen Semantik und werden in Induktionsbeweisen häufig benutzt.

$$\begin{aligned} [\![x]\!]_{(\mathfrak{A},\chi)} &= \chi(x) \quad \text{für alle} \quad x \in X \\ [\![ft_1 \dots t_n]\!]_{(\mathfrak{A},\chi)} &= f_{\mathfrak{A}}([\![t_1]\!]_{(\mathfrak{A},\chi)}, \dots, [\![t_n]\!]_{(\mathfrak{A},\chi)}) \quad \text{für alle} \quad ft_1 \dots t_n \in \mathfrak{T}_\Sigma(X) \end{aligned}$$

Abkürzungen: Wenn $(t_1, \dots, t_n) \in T_\Sigma(X)^w$, so bezeichnen wir dieses n-Tupel kurz durch t^w und verwenden dieses Kürzel auch in Termen wie ft^w, wenngleich dies wegen Klammern und Kommata nicht genau unserer Termnotation entspricht. Die abstrakt-syntaktische Vorstellung erlaubt jedoch die Identifikation von $f(t_1, \ldots, t_n)$ mit $ft_1 \ldots t_n$. Daher gilt für alle $ft^w \in T_\Sigma(X)$

$$[\![ft^w]\!]_{(\mathfrak{A},\chi)} = f_\mathfrak{A}([\![t^w]\!]_{(\mathfrak{A},\chi)}).$$

Es gibt zwei Spezialfälle der algebraischen Semantik $[\![t]\!]_{(\mathfrak{A},\chi)}$, die besondere Beachtung verdienen: χ ist eine Inklusion und läßt somit ihre Argumente unverändert, oder $\mathfrak{A}$ ist ebenfalls eine Termalgebra, so daß die Funktionssymbole erhalten bleiben. Die entsprechenden Homomorphismen heißen Evaluation und Substitution.

- *Evaluation:* Wenn $X \subseteq A$ und $\chi = \mathsf{in}_{X,A}$, so schreiben wir $[\![t]\!]_\mathfrak{A}$ statt $[\![t]\!]_{(\mathfrak{A},\mathsf{in}_{X,A})}$. Falls $X = \emptyset$, so heißt $[\![t]\!]_\mathfrak{A}$ *initiale Semantik* von t bzgl. $\mathfrak{A}$.
- *Substitution:* Wenn $\chi = \mathsf{sub} : X \to T_\Sigma(Y)$, so verwenden wir die Postfixnotation $t\mathsf{sub}$ statt $[\![t]\!]_{(\mathfrak{A},\mathsf{sub})}$. Es gilt:

$$\begin{array}{rcll} x\mathsf{sub} & = & \mathsf{sub}(x) \quad \text{für alle} & x \in X \\ \text{und} \quad (ft_1 \ldots t_n)\mathsf{sub} & = & f(t_1\mathsf{sub}) \ldots (t_n\mathsf{sub}) & \\ \text{bzw.} \quad (ft^w)\mathsf{sub} & = & f(t^w\mathsf{sub}) \quad \text{für alle} & ft^w = ft_1 \ldots t_n \in T_\Sigma(X). \end{array}$$

Das folgende Lemma beschreibt eine gewisse Vertauschbarkeit bei der Komposition von Evaluation und Substitution.

Vertauschungslemma
Sei $\overline{\mathsf{sub}} : \mathfrak{T}_\Sigma(X) \to \mathfrak{T}_\Sigma(A)$ eine Substitution und $\overline{\mathsf{in}_{A,B}} : \mathfrak{T}_\Sigma(A) \to \mathfrak{B}$ eine Evaluation. Dann gilt

$$\overline{\mathsf{in}_{A,B}} \circ \overline{\mathsf{sub}} = \overline{\overline{\mathsf{in}_{A,B}} \circ \mathsf{sub}}.$$

Beweis: Beide Homomorphismen stimmen auf X überein und müssen daher gleich sein. $\square$

Termfunktionen

Mit Hilfe von Termen lassen sich aus Grundfunktionen weitere Funktionen ableiten. Dazu wählen wir ein S-sortiertes *Standardalphabet von Argumentvariablen*

$$\mathbb{X} := \{x_i^s \mid i \in \mathbb{N}, s \in S\}.$$

Jedes $w = s_1 \ldots s_n \in S^*$ bestimmt die folgende Teilmenge

$$\mathbb{X}_w := \{x_1^{s_1}, \ldots, x_n^{s_n}\},$$

welche von $\mathbb{X}^w = \mathbb{X}^{s_1} \times \ldots \times \mathbb{X}^{s_n}$ zu unterscheiden ist. Mit $x^w := (x_1^{s_1}, \ldots, x_n^{s_n})$ abstrahieren wir von $t \in T_\Sigma(\mathbb{X}_w)^s$ die *explizite Funktionsdefinition*

$$\lambda x^w. t,$$

welche bzgl. einer Σ-Algebra $\mathfrak{A}$ die *Termfunktion*

$$[\![\lambda x^w.\, t]\!]_{\mathfrak{A}} : A^w \to A^s$$

induziert: ein Argumentvektor $a^w = (a_1, \ldots, a_n) \in A^w$ bestimmt die Belegung

$$[x^w/a^w] : \mathbb{X}_w \to A \quad \text{mit} \quad x_i^{s_i} \longmapsto a_i \quad \text{für} \quad i = 1, \ldots, n,$$

so daß die Termfunktion durch

$$[\![\lambda x^w.\, t]\!]_{\mathfrak{A}}(a^w) := [\![t]\!]_{(\mathfrak{A}, [x^w/a^w])}$$

festgelegt werden kann.

Wir werden diese algebraische Technik der Definition von Funktionen durch Terme insbesondere bei der Fixpunktsemantik rekursiver Funktionsdefinitionen einsetzen.

Abschließend formulieren wir das Vertauschungslemma für den Spezialfall endlicher Substitutionen, wie sie bei Termfunktionen auftreten.

Vertauschungslemma (für endliche Substitutionen)
Für $\overline{[x^w/u^w]} : \mathfrak{T}_\Sigma(\mathbb{X}_w) \to \mathfrak{T}_\Sigma(A)$, $\overline{\mathsf{in}_{A,B}} : \mathfrak{T}_\Sigma(A) \to \mathfrak{B}$ *und* $t \in T_\Sigma(\mathbb{X}_w)$ *gilt*

$$[\![t[x^w/u^w]]\!]_{\mathfrak{B}} = [\![t]\!]_{(\mathfrak{B}, [x^w/[\![u^w]\!]_{\mathfrak{B}}])}.$$

Auf dieser Vertauschbarkeit von Substitution und Evaluation beruht die Äquivalenz von Fixpunktsemantik und Reduktionssemantik.

3 Ordnungstheoretische Grundlagen

Während die Semantik expliziter Funktionsdefinitionen rein algebraisch mit Hilfe von Homomorphismen definiert werden konnte, so erfordern rekursive Definitionen die zusätzliche Verwendung ordnungstheoretischer Methoden. Diese gestatten ebenfalls eine berechnungsunabhängige Festlegung der Semantik, nämlich durch kleinste Fixpunkte stetiger Transformationen auf vollständigen Halbordnungen. Zu diesem Zweck erweitern wir die Grundmengen zu flachen Halbordnungen und ersetzen partielle durch stetige Funktionen. Dabei werden insbesondere Striktheitseigenschaften beschrieben, die ein denotationales Analogon zu verschiedenen Auswertungsstrategien bilden. Für die technische Realisierung erweitern wir Termfunktionen über flachen Halbordnungen und gewinnen durch Abstraktion von zusätzlich in den Termen auftretenden Funktionsvariablen die bei der Fixpunktsemantik eingesetzten Termfunktionale.

Definition (Halbordnung)
Sei A eine nicht-leere Menge und $\leq\ \subseteq A^2$. Ist $\leq$ reflexiv, transitiv und antisymmetrisch, so heißt $\mathfrak{A} = \langle A; \leq \rangle$ eine *Halbordnung.* $\lhd$

Die einfachsten hier benutzten Halbordnungen sind flache Erweiterungen S-sortierter Mengen. Dazu wählen wir für jede Sorte $s \in S$ einen neuen Wert $\perp^s$, welcher das Nicht-Definiertsein der Sorte s symbolisiert.

Definition (Flache Erweiterung)
Sei A eine S-sortierte Menge. Dann definieren wir ihre *flache Erweiterung*

$$\langle A_\perp; \leq \rangle \qquad \text{durch}$$

- $A_\perp := \bigcup_{s\in S} {A_\perp}^s$ mit ${A_\perp}^s := A^s \cup \{\perp^s\}$ und
- $a \leq b$ gdw. $\{a, b\} \subseteq {A_\perp}^s$ und $(a = \perp^s$ oder $a = b)$ für ein $s \in S$. $\lhd$

Offensichtlich ist $A_\perp$ S-sortiert und ebenso wie alle ${A_\perp}^s$ durch $\leq$ halbgeordnet. Weitere Halbordnungen entstehen aus diesen durch Konstruktion von Produkt- und Funktionenräumen.

Definition und Lemma (Produkt- und Funktionenraum)
Sind $\mathfrak{A}_1 = \langle A_1; \leq_1 \rangle$ und $\mathfrak{A}_2 = \langle A_2; \leq_2 \rangle$ Halbordnungen, so auch

- der *Produktraum* $\mathfrak{A}_1 \times \mathfrak{A}_2 := \langle A_1 \times A_2; \leq \rangle$
  mit $(a_1, a_2) \leq (b_1, b_2)$ gdw. $(a_1 \leq_1 b_1$ und $a_2 \leq_2 b_2)$ und
- der *Funktionenraum* $[\mathfrak{A}_1 \to \mathfrak{A}_2] := \langle \{f \mid f : A_1 \to A_2\}; \leq \rangle$
  mit $f \leq g$ gdw. $f(a) \leq_2 g(a)$ für alle $a \in A_1$. $\lhd$

Damit erhalten wir aus den flachen Halbordnungen $\langle {A_\perp}^s; \leq \rangle$ für $w = s_1 \dots s_n \in S^*$ und $s \in S$

- den Produktraum ${A_\perp}^w = {A_\perp}^{s_1} \times \dots \times {A_\perp}^{s_n}$ und
- den Funktionenraum $\mathbf{F}^{(w,s)}(A_\perp) := \{f \mid f : {A_\perp}^w \to {A_\perp}^s\}$

mit den obigen Halbordnungsrelationen. Beachte, daß ${A_\perp}^\varepsilon = \{()\}$ als triviale Halbordnung mit einem Element aufgefaßt werden kann.

Da Funktionen über $A_\perp$ bei der Semantik rekursiver Funktionsdefinitionen eine zentrale Rolle spielen, beschreiben wir zunächst einige ihrer Eigenschaften, besonders das Verhalten auf $\perp$-Argumenten.

Definition (Strikte Funktion)
Sei $f \in \mathbf{F}^{(w,s)}(A_\perp)$ mit $w = s_1 \dots s_n \in S^+$. f heißt *strikt*, falls für alle $\overline{a} = (a_1, \dots, a_n) \in {A_\perp}^w$ und alle $i \in \{1, \dots, n\}$ das Argument $a_i = \perp^{s_i}$ den Wert $f(\overline{a}) = \perp^s$ impliziert. $\lhd$

Jede Funktion $f \in \mathbf{F}^{(w,s)}(A)$ mit $w \neq \varepsilon$ besitzt genau eine strikte Erweiterung $f_\perp \in \mathbf{F}^{(w,s)}(A_\perp)$. Es werden aber auch nicht-strikte Funktionen wie z.B. die Verzweigung benötigt. Als charakteristische Eigenschaft aller hier auftretenden Funktionen stellen wir fest, daß sie die „Informationshalbordnung" erhalten, also monoton sind.

Definition (Monotone Abbildung)
Seien $\mathfrak{A}_1 = \langle A_1; \leq_1 \rangle$ und $\mathfrak{A}_2 = \langle A_2; \leq_2 \rangle$ Halbordnungen. Dann heißt eine Abbildung $f : A_1 \to A_2$ *monoton*, wenn für alle $a, b \in A_1$ gilt: $a \leq_1 b$ impliziert $f(a) \leq_2 f(b)$. $\lhd$

Es folgt unmittelbar, daß eine strikte Funktion ebenso wie die Komposition monotoner Abbildungen monoton ist. Wegen der Beschränkung auf monotone Funktionen vereinbaren wir für jedes $(w, s) \in D(S)$ den Funktionenraum

$$\mathbf{mF}^{(w,s)}(A_\perp) := \{f \mid f : {A_\perp}^w \to {A_\perp}^s,\ f \text{ monoton}\},$$

der als Teilmenge von $\mathbf{F}^{(w,s)}(A_\perp)$ halbgeordnet ist. Falls $w = \varepsilon$, so gilt

$$\mathbf{mF}^{(\varepsilon,s)}(A_\perp) = \mathbf{F}^{(\varepsilon,s)}(A_\perp).$$

Die Vereinigung dieser Funktionenräume ist $D(S)$-sortiert:

$$\mathbf{mF}(A_\perp) := \bigcup_{\tau \in D(S)} \mathbf{mF}^\tau(A_\perp),$$

so daß wir auch hier die Kurzschreibweise für cartesische Produkte verwenden können:

$$\mathbf{mF}^\rho(A_\perp) = \mathbf{mF}^{\tau_1}(A_\perp) \times \dots \times \mathbf{mF}^{\tau_r}(A_\perp), \quad \text{falls} \quad \rho = \tau_1 \dots \tau_r \in D(S)^+.$$

Wir sind nun in der Lage, das Verhalten von Funktionen über A für nicht-definierte Argumente durch geeignete Erweiterung zu monotonen Funktionen über $A_\perp$ zu beschreiben. Der Einfachheit halber wählen wir für alle Grundfunktionen mit Ausnahme der Verzweigung die strikte Erweiterung. Zu diesem Zweck gehen wir im folgenden von Signaturen aus, welche eine boolesche Sorte und Verzweigungssymbole besitzen, und interpretieren diese entsprechend.

Definition (Signatur und Algebra mit Verzweigung)
Eine Signatur $\Sigma = \langle S, \mathcal{F} \rangle$ heißt *Signatur mit Verzweigung*, falls sie eine spezielle Sorte $\mathsf{bool} \in S$ und für jedes $s \in S$ ein Verzweigungssymbol $\mathsf{cond}_s \in \mathcal{F}^{(\mathsf{bool}\ ss,s)}$ besitzt. Somit zerfällt $\mathcal{F}$ in $\mathcal{F}_{\mathsf{cond}} := \{\mathsf{cond}_s \mid s \in S\}$ und $\mathcal{F}_{\mathsf{base}} := \mathcal{F} \setminus \mathcal{F}_{\mathsf{cond}}$.

Eine Σ-Algebra $\mathfrak{A} = \langle A; \alpha \rangle$ heißt Σ-*Algebra mit Verzweigung*, falls Σ eine Signatur mit Verzweigung ist und außerdem bool und cond_s wie folgt interpretiert werden:

- $A^{\mathsf{bool}} := \mathbb{B} = \{\mathsf{true}, \mathsf{false}\}$ und
- $\alpha(\mathsf{cond}_s) : \mathbb{B} \times A^s \times A^s \to A^s$ mit
  $(\mathsf{true}, a, b) \longmapsto a$ und $(\mathsf{false}, a, b) \longmapsto b$ für alle $s \in S$ und $a, b \in A^s$. $\lhd$

Wir erweitern eine solche Σ-Algebra mit Verzweigung durch Hinzunahme von $\perp$-Elementen und geeignete Interpretation der Funktionssymbole durch monotone Funktionen. Dabei lassen wir neben strikten Funktionen und nicht-strikten Verzweigungen auch beliebige monotone Funktionen zu. Der Grund ist technischer Natur: bei der formalen Beschreibung der Semantik rekursiver Funktionsdefinitionen ergänzen wir die Signatur Σ um neue Funktionssymbole, welche beliebig monoton interpretiert werden.

Definition (Strikte und monotone Algebren)
Sei $\mathfrak{A} = \langle A; \alpha \rangle$ eine Σ-Algebra mit Verzweigung. Wir definieren ihre *strikte Erweiterung* $\mathfrak{A}_\perp = \langle A_\perp; \alpha_\perp \rangle$ durch folgende Erweiterung der Grundfunktionen:

- $\alpha_\perp(\mathsf{cond}_s)(\perp^{\mathsf{bool}}, a, b) = \perp^s$
  $\alpha_\perp(\mathsf{cond}_s)(\mathsf{true}, a, b) = a$ (auch für $b = \perp^s$)
  $\alpha_\perp(\mathsf{cond}_s)(\mathsf{false}, a, b) = b$ (auch für $a = \perp^s$) für alle $s \in S$,
- $\alpha_\perp(f) := \alpha(f)_\perp$ für alle $f \in \mathcal{F}_{\mathsf{base}} \setminus \mathcal{C}$ und
  $\alpha_\perp(c) := \alpha(c)$ für alle $c \in \mathcal{C}$.

Allgemeiner heißt eine Σ-Algebra $\mathfrak{A}_m = \langle A_\perp; \alpha_m \rangle$ *monoton*, falls Σ eine Signatur mit Verzweigung ist und außerdem gilt:

- $A^{\mathsf{bool}} = \mathbb{B}$,
- $\alpha_m(\mathsf{cond}_s) = \alpha_\perp(\mathsf{cond}_s)$ für alle $s \in S$ und
- $\alpha_m(f)$ ist monoton für alle $f \in \mathcal{F}_{\mathsf{base}}$. $\lhd$

Man prüft leicht nach, daß $\mathfrak{A}_\perp$ eine monotone Σ-Algebra ist.

Erweiterte Termfunktionen

Nach der Erweiterung der Grundfunktionen wenden wir uns jetzt den Funktionen zu, die durch Terme definiert werden. Dabei soll ein nicht-definiertes Argument nicht notwendig ein nicht-definiertes Ergebnis erzwingen, also strikt behandelt werden; stattdessen soll es auch möglich sein, daß die Termsemantik dieses Argument benutzt und somit eventuell einen definierten Wert liefert, weil die Semantik von diesem Argument nicht abhängt. Wir legen deshalb in einer expliziten Funktionsdefinition fest, ob ein Argument strikt behandelt wird oder nicht. Der Einfachheit halber sollen die Argumente mit strikter Auswertung den Argumenten mit Termauswertung vorausgehen, so daß die erforderliche Striktheitsinformation durch einen Index σ zwischen 0 und der Anzahl der Argumente bestimmt ist.

Definition (Striktheitsinformation)
Sei $\Sigma = \langle S, \mathcal{F} \rangle$ eine Signatur mit Verzweigung, $(w, s) \in D(S)$, $t \in T_\Sigma(\mathbb{X}_w)^s$ und $0 \leq \sigma \leq |w|$. Dann heißt

$$\lambda^\sigma x^w.\, t$$

eine explizite Funktionsdefinition mit *Striktheitsinformation*. Bezüglich einer monotonen Σ-Algebra $\mathfrak{A}_m$ bestimmt sie die *erweiterte Termfunktion*

$$[\![\lambda^\sigma x^w.\, t]\!]_{\mathfrak{A}_m} : {A_\perp}^w \to {A_\perp}^s,$$

wobei für $a^w = (a_1, \ldots, a_n) \in {A_\perp}^w$

$$[\![\lambda^\sigma x^w.\, t]\!]_{\mathfrak{A}_m}(a^w) := \begin{cases} [\![t]\!]_{(\mathfrak{A}_m, [x^w/a^w])} & \text{falls} \quad a_1, \ldots, a_\sigma \in A \\ \perp^s & \text{sonst.} \end{cases}$$

$\lhd$

Man weist unschwer durch Induktion über t nach, daß die erweiterten Termfunktionen monoton sind.

Termfunktionale

Auf den rechten Seiten rekursiver Funktionsgleichungen treten neben Grundfunktionssymbolen auch Funktionsvariablen auf. Wir ordnen solchen Termen durch Abstraktion von diesen Variablen Termfunktionale zu, welche als Abbildungen von Funktionenräumen die Grundlage der Fixpunktsemantik bilden. Für die formale Definition gehen wir aus von einem $D(S)$-sortierten *Standardalphabet von Funktionsvariablen*

$$\mathbb{F} := \{F_j^\tau \mid j \in \mathbb{N}, \tau \in D(S)\}.$$

Jedes $\rho = \tau_1 \ldots \tau_r \in D(S)^+$ bestimmt die Teilmenge

$$\mathbb{F}_\rho := \{F_1^{\tau_1}, \ldots, F_r^{\tau_r}\}$$

von Funktionsvariablen einer rekursiven Funktionsdefinition. Zur Beschreibung ihrer rechten Gleichungsseiten bilden wir die erweiterte Signatur

$$\Sigma[\mathbb{F}_\rho] := \langle S, \mathcal{F} \cup \mathbb{F}_\rho \rangle.$$

Dann läßt sich eine rechte Gleichungsseite als Term $t \in T_{\Sigma[\mathbb{F}_\rho]}(\mathbb{X}_w)^s$ mit $(w, s) \in D(S)$ angeben. Dieser bestimmt für $F^\rho := (F_1^{\tau_1}, \ldots, F_r^{\tau_r})$ und $0 \leq \sigma \leq |w|$ die *funktionale Termdefinition*

$$\lambda F^\rho.\, \lambda^\sigma x^w.\, t\ .$$

Sie liefert für eine Σ-Algebra $\mathfrak{A}$ mit Verzweigung das *Termfunktional*

$$[\![\lambda F^{\rho}.\, \lambda^{\sigma} x^{w}.\, t]\!]_{\mathfrak{A}_\perp} : \mathbf{mF}^{\rho}(A_\perp) \to \mathbf{mF}^{(w,s)}(A_\perp),$$

welches wie folgt erklärt ist. Für ein Argument $g^{\rho} = (g_1, \dots, g_r) \in \mathbf{mF}^{\rho}(A_\perp)$ erweitern wir $\mathfrak{A}_\perp$ zu der $\Sigma[\mathbb{F}_\rho]$-Algebra $\mathfrak{A}_\perp[g^\rho] := \langle A_\perp; \alpha_{g^\rho}\rangle$ mit $\alpha_{g^\rho}(F_j^{\tau_j}) := g_j$ für $j = 1, \dots, r$ und setzen

$$[\![\lambda F^{\rho}.\, \lambda^{\sigma} x^{w}.\, t]\!]_{\mathfrak{A}_\perp}(g^\rho) := [\![\lambda^{\sigma} x^{w}.\, t]\!]_{\mathfrak{A}_\perp[g^\rho]}$$

in Analogie zur Definition der Termfunktionen. Man verifiziert leicht, daß die Ergebnisfunktion ebenso wie das Termfunktional selbst monoton sind.

Vollständige Halbordnungen, stetige Funktionen

Termfunktionale besitzen neben der Monotonie noch eine weitere, für die Konstruktion von Fixpunkten wichtige Eigenschaft: sie erhalten gewisse Limites. Wir formalisieren dies durch stetige Funktionen auf vollständigen Halbordnungen.

Definition (Vollständige Halbordnung)
Sei $\mathfrak{A} = \langle A; \leq\rangle$ eine Halbordnung, $T \subseteq A$ und $a \in A$.

- T heißt *gerichtet*, falls $T \neq \emptyset$ und $\forall a, b \in T\ \exists c \in T : \{a, b\} \leq c$.
- a heißt *obere Schranke* von T, falls $T \leq a$.
- a heißt *kleinstes Element* von T, falls $a \leq T$ und $a \in T$.
- Besitzt $\{a \mid T \leq a\}$ ein kleinstes Element, so heißt dies *kleinste obere Schranke* von T und wird bezeichnet durch $\bigsqcup T$ oder $\bigsqcup_{t \in T} t$.
- $\mathfrak{A}$ heißt eine *vollständige* Halbordnung, falls gilt:
  - $\mathfrak{A}$ besitzt ein kleinstes Element $\perp_{\mathfrak{A}} \in A$.
  - Jede gerichtete Teilmenge $T \subseteq A$ hat eine kleinste obere Schranke $\bigsqcup T \in A$. $\lhd$

Man sieht leicht, daß die flachen Erweiterungen $\langle {A_\perp}^s; \leq\rangle$ vollständig sind und daß diese Eigenschaft bei der Konstruktion von Produkt- und Funktionenräumen erhalten bleibt. Dies gilt auch bei der Beschränkung auf monotone Funktionen. Somit ist $\mathbf{mF}^{\tau}(A_\perp)$ für jedes $\tau \in D(S)$ vollständig. Die kleinsten oberen Schranken lassen sich dabei wie folgt komponentenweise konstruieren.
Für vollständige Halbordnungen $\mathfrak{A}_1 = \langle A_1; \leq_1\rangle$ und $\mathfrak{A}_2 = \langle A_2; \leq_2\rangle$ und gerichtete Mengen $T \subseteq A_1 \times A_2$ und $D \subseteq \{f \mid f : A_1 \to A_2\}$ gilt:

- $\bigsqcup T = (\bigsqcup \mathsf{proj}_1(T), \bigsqcup \mathsf{proj}_2(T))$ und
- $(\bigsqcup D)(a) = \bigsqcup_{f \in D} f(a)$.

Definition (Stetigkeit)
Seien $\mathfrak{A}_1 = \langle A_1; \leq_1 \rangle$ und $\mathfrak{A}_2 = \langle A_2; \leq_2 \rangle$ vollständige Halbordnungen und $f : A_1 \to A_2$ eine Abbildung. Dann heißt f *stetig*, falls

- f monoton ist und
- $f(\bigsqcup T) = \bigsqcup f(T)$ für jede gerichtete Menge $T \subseteq A_1$. $\lhd$

Man sieht leicht, daß jedes $f \in \mathbf{mF}^{\tau}(A_\perp)$ stetig ist. Etwas aufwendiger ist der Stetigkeitsnachweis für Termfunktionale, den wir aus Platzgründen hier übergehen.

Lemma (Stetigkeit von Termfunktionalen)
Jedes Termfunktional

$$[\![\lambda F^{\rho}.\, \lambda^{\sigma} x^{w}.\, t]\!]_{\mathfrak{A}_\perp} : \mathbf{mF}^{\rho}(A_\perp) \to \mathbf{mF}^{(w,s)}(A_\perp)$$

ist stetig.

Zum Abschluß dieser ordnungstheoretischen Vorbereitungen zitieren wir den bekannten Fixpunktsatz für stetige Transformationen.

Fixpunktsatz
Sei $\mathfrak{A} = \langle A; \leq \rangle$ vollständig und $f : A \to A$ stetig. Dann existiert

$$\mathsf{fix}(f) := \bigsqcup_{k \in \mathbb{N}} f^k(\perp)$$

und ist der kleinste Fixpunkt *von f, also das kleinste $a \in A$ mit $a = f(a)$.*

4 Rekursive Funktionsdefinitionen

Mit diesen algebraischen und ordnungstheoretischen Grundlagen haben wir einen geeigneten Rahmen geschaffen, um Syntax und Semantik rekursiver Funktionsdefinitionen unter besonderer Berücksichtigung von Striktheitsinformationen festzulegen. Nach der Definition der Fixpunktsemantik konstruieren wir eine nichtdeterministische Reduktionssemantik sowie eine deterministische Auswertungsstrategie. Der Nachweis ihrer Äquivalenz, besonders der Vollständigkeit der beiden Reduktionssemantiken, bildet den Kern dieses Abschnitts.

Definition (Syntax)
Sei $\Sigma = \langle S, \mathcal{F} \rangle$ eine Signatur mit Verzweigung. Dann heißt

$$R := (F_j^{\tau_j} = \lambda^{\sigma_j} x^{w_j}.\, t_j \mid 1 \leq j \leq r)$$

eine *rekursive Funktionsdefinition* über Σ, wenn folgende Bedingungen für $j = 1, \ldots, r$ erfüllt sind:

- $\tau_j = (w_j, s_j)$,
- $0 \leq \sigma_j \leq |w_j|$ und
- $t_j \in T_{\Sigma[\mathbb{F}_\rho]}(\mathbb{X}_{w_j})^{s_j}$ mit $\rho = \tau_1 \ldots \tau_r$. $\lhd$

Da wir F_1 als definierende Funktionsvariable ansehen, nehmen wir ihren Typ als Typ von R: $\mathsf{sort}(R) := \tau_1$. Somit ist die Menge $\mathbf{Rfd}_\Sigma$ der rekursiven Funktionsdefinitionen über Σ $D(S)$-sortiert:

$$\mathbf{Rfd}_\Sigma := \bigcup_{\tau \in D(S)} \mathbf{Rfd}_\Sigma^\tau.$$

Definition (Fixpunktsemantik)
Sei $R \in \mathbf{Rfd}_\Sigma$ und $\mathfrak{A} = \langle A; \alpha \rangle$ eine Σ-Algebra mit Verzweigung. Wir definieren die *Fixpunktsemantik* von R bzgl. $\mathfrak{A}$ durch

$$\mathsf{Fp}[\![R]\!]_\mathfrak{A} := \mathsf{proj}_1(\mathsf{fix}(\Phi_{(R,\mathfrak{A})})) : {A_\perp}^{w_1} \to {A_\perp}^{s_1}$$

unter Verwendung der durch R und $\mathfrak{A}$ bestimmten Transformation

$$\Phi_{(R,\mathfrak{A})} : \mathbf{mF}^\rho(A_\perp) \to \mathbf{mF}^\rho(A_\perp)$$

mit $\quad \Phi_{(R,\mathfrak{A})}(g^\rho) := ([\![\lambda F^\rho.\, \lambda^{\sigma_1} x^{w_1}.\, t_1]\!]_{\mathfrak{A}_\perp}(g^\rho), \ldots, [\![\lambda F^\rho.\, \lambda^{\sigma_r} x^{w_r}.\, t_r]\!]_{\mathfrak{A}_\perp}(g^\rho)).$ $\lhd$

Die Transformation $\Phi_{(R,\mathfrak{A})}$ beschreibt also die Ersetzung der Funktionsvariablen von $F^\rho = (F_1^{\tau_1}, \ldots, F_r^{\tau_r})$ durch die Argumentfunktionen von $g^\rho = (g_1, \ldots, g_r)$ mit anschließender Auswertung. Daher können die Gleichungslösungen von R bzgl. $\mathfrak{A}$ als Fixpunkte von $\Phi_{(R,\mathfrak{A})}$ aufgefaßt werden. Die Existenz des kleinsten Fixpunkts ergibt sich aus der Stetigkeit der Termfunktionale.

Reduktionssemantik

Nach der denotationalen Interpretation von rekursiven Funktionsdefinitionen als Gleichungen wollen wir nun einen operationalen Standpunkt einnehmen und die Gleichungen als Termersetzungsregeln auffassen. Sie erlauben die Berechnung eines Funktionswerts aus gegebenen Argumenten durch schrittweise Reduktion.

Wir ordnen einer rekursiven Funktionsdefinition $R \in \mathbf{Rfd}_\Sigma$ bzgl. einer Σ-Algebra $\mathfrak{A}$ mit Verzweigung die Menge

$$\mathbf{T}(R, A) := T_{\Sigma[\mathbb{F}_\rho]}(A)$$

ihrer *Berechnungsterme* und die folgenden *Berechnungsregeln* zu:

- **Grundreduktion:** $f a^w \longrightarrow f_{\mathfrak{A}}(a^w)$
  für alle $f \in \mathcal{F}^{(w,s)}_{\mathsf{base}}$, $(w,s) \in D(S)$ und $a^w \in A^w$

- **Verzweigungsreduktion:** $\mathsf{cond}_s\ \mathsf{true}\ u_1 u_2 \longrightarrow u_1$
  $\mathsf{cond}_s\ \mathsf{false}\ u_1 u_2 \longrightarrow u_2$
  für alle $s \in S$ und $u_1, u_2 \in \mathbf{T}(R,A)^s$

- **Funktionsreduktion:** $F_j^{\tau_j} u^{w_j} \longrightarrow t_j[x^{w_j}/u^{w_j}]$
  für alle $j = 1, \ldots, r$ und $u^{w_j} \in \mathbf{T}(R,A)^{w_j}$
  mit der Striktheitseigenschaft $u_1, \ldots, u_{\sigma_j} \in A$

Als Spezialfall der Grundreduktion erhalten wir für $w = \varepsilon$ die Konstantenreduktion: $c \longrightarrow c_{\mathfrak{A}}()$. Eine Funktionsreduktion kann erst dann durchgeführt werden, wenn die durch den zugehörigen Striktheitsindex bestimmten Argumente als Elemente aus A vorliegen.

Ein Berechnungsterm kann mehrere reduzierbare Teilterme enthalten, so daß eine sequentielle Implementierung eine geeignete Auswahl erfordert. Wir werden zeigen, daß der am weitesten links stehende reduzierbare Teilterm ein passender Kandidat ist. Zunächst wollen wir jedoch die Auswahl offen lassen, ja sogar parallele Reduktionsschritte zulassen, um den Vollständigkeitsbeweis möglichst transparent zu gestalten.

Wir definieren daher die *Reduktionsrelation*

$$\Longrightarrow \quad \subseteq \quad \mathbf{T}(R,A)^2$$

durch folgende strukturelle Induktion:

- Für jede Berechnungsregel $u \longrightarrow v$ gilt $u \Longrightarrow v$.
- Für jeden Berechnungsterm $u \in \mathbf{T}(R,A)$ gilt $u \Longrightarrow u$.
- Für jedes $(w,s) \in D(S)$, $\varphi \in (\mathcal{F} \cup \mathbb{F}_\rho)^{(w,s)}$ und $u^w, v^w \in \mathbf{T}(R,A)$ gilt $\varphi u^w \Longrightarrow \varphi v^w$, falls $u_i \Longrightarrow v_i$ für $i = 1, \ldots, |w|$.

Das folgende Lemma bestätigt, daß wir beliebige Teilterme eines Berechnungsterms simultan reduzieren können.

Lemma (Reduktion von Teiltermen)
Sei $w \in S^*$, $t \in T_{\Sigma[\mathbb{F}_\rho]}(\mathbb{X}_w)$ und $u^w, v^w \in \mathbf{T}(R,A)^w$. Dann gilt mit $u_i \Longrightarrow v_i$ für $i = 1, \ldots, |w|$ auch

$$t[x^w/u^w] \Longrightarrow t[x^w/v^w].$$

Beweis durch strukturelle Induktion über t. Beachte dazu insbesondere die Homomorphieeigenschaft der Substitution:

$$(\varphi t_1 \dots t_n)[x^w/u^w] = \varphi(t_1[x^w/u^w]) \dots (t_n[x^w/u^w]) \qquad \square$$

Trotz ihres Nichtdeterminismus induziert die Reduktionsrelation eine geeignete Semantik von Berechnungstermen und damit von rekursiven Funktionsdefinitionen. Denn für ein $u \in \mathbf{T}(R, A)$ gibt es höchstens ein $a \in A$ mit $u \stackrel{*}{\Longrightarrow} a$. Zum Nachweis dieser speziellen Konfluenzeigenschaft übertragen wir die Fixpunktsemantik auf Berechnungsterme und zeigen ihre Invarianz unter Reduktion.

Die folgenden Abkürzungen für den Fixpunkt der Transformation $\Phi_{(R,\mathfrak{A})}$ und seine Approximationen vereinfachen die Darstellung.

Für $\Phi_{(R,\mathfrak{A})} : \mathbf{mF}^\rho(A_\perp) \to \mathbf{mF}^\rho(A_\perp)$ setzen wir

$$\pi^\rho = (\pi_1^{\tau_1}, \dots, \pi_r^{\tau_r}) := \mathsf{fix}(\Phi_{(R,\mathfrak{A})})$$

und schreiben auch kürzer ohne Typinformation $\overline{\pi} = (\pi_1, \dots, \pi_r)$. Sei

$$\perp^\rho = (\perp_1^{\tau_1}, \dots, \perp_r^{\tau_r}) := \perp_{\mathbf{mF}^\rho(A_\perp)},$$

wo $\perp^{(w,s)} : {A_\perp}^w \to {A_\perp}^s$ gegeben ist durch $\perp^{(w,s)}(a^w) := \perp^s$ für alle $a^w \in {A_\perp}^w$. Dann bezeichnen wir die Approximationen von $\overline{\pi}$ durch

$$\overline{\pi}^{(k)} = (\pi_1^{(k)}, \dots, \pi_r^{(k)}) := \Phi_{(R,\mathfrak{A})}^k(\perp^\rho) \quad \text{für jedes} \quad k \in \mathbb{N}.$$

Der Fixpunkt $\overline{\pi}$ und seine Approximationen $\overline{\pi}^{(k)}$ bestimmen die $\Sigma[\mathbb{F}_\rho]$-Algebren

$$\mathfrak{A}_\perp[\overline{\pi}] \quad \text{bzw.} \quad \mathfrak{A}_\perp[\overline{\pi}^{(k)}].$$

Damit erklären wir die Fixpunktsemantik eines Berechnungsterms $u \in \mathbf{T}(R, A)$ durch

$$\mathsf{Fp}[\![u]\!]_{(R,\mathfrak{A})} := [\![u]\!]_{\mathfrak{A}_\perp[\overline{\pi}]}.$$

Sie steht in folgender Beziehung zur Fixpunktsemantik rekursiver Gleichungssysteme:

$$\mathsf{Fp}[\![R]\!]_{\mathfrak{A}}(a^{w_1}) = \mathsf{Fp}[\![F_1 a^{w_1}]\!]_{(R,\mathfrak{A})} \quad \text{für jedes } a^{w_1} \in A^{w_1}.$$

Lemma (Invarianz der Fixpunktsemantik unter Reduktion)
Für alle $u, v \in \mathbf{T}(R, A)$ gilt:

$$\textit{aus} \quad u \Longrightarrow v \quad \textit{folgt} \quad \mathsf{Fp}[\![u]\!]_{(R,\mathfrak{A})} = \mathsf{Fp}[\![v]\!]_{(R,\mathfrak{A})}.$$

Beweis durch Induktion über den Aufbau der Reduktionsrelation $\Longrightarrow$:

(1) Grundreduktion: Für $fa^w \longrightarrow f_{\mathfrak{A}}(a^w)$ folgt direkt

$$[\![fa^w]\!]_{\mathfrak{A}_\perp[\overline{\pi}]} = f_{\mathfrak{A}}(a^w) = [\![f_{\mathfrak{A}}(a^w)]\!]_{\mathfrak{A}_\perp[\overline{\pi}]}.$$

(2) Verzweigungsreduktion: Auch für $\mathsf{cond}_s\ \mathsf{true}\ u_1u_2 \longrightarrow u_1$ verifiziert man unmittelbar, daß

$$\begin{array}{rcl} [\![\mathsf{cond}_s\ \mathsf{true}\ u_1u_2]\!]_{\mathfrak{A}_\perp[\overline{\pi}]} & = & \alpha_\perp(\mathsf{cond}_s)(\mathsf{true}, [\![u_1]\!]_{\mathfrak{A}_\perp[\overline{\pi}]}, [\![u_2]\!]_{\mathfrak{A}_\perp[\overline{\pi}]}) \\ & = & [\![u_1]\!]_{\mathfrak{A}_\perp[\overline{\pi}]}, \end{array}$$

und entsprechend im false-Fall.

(3) Funktionsreduktion: Sei $F_j\overline{u} \longrightarrow t_j[\overline{x}/\overline{u}]$ eine Berechnungsregel zur Funktionsreduktion. Dann muß die Striktheitsbedingung $u_1, \dots, u_{\sigma_j} \in A$ erfüllt sein. Die Behauptung ergibt sich wie folgt:

$$\begin{array}{rcll} [\![F_j\overline{u}]\!]_{\mathfrak{A}_\perp[\overline{\pi}]} & = & \pi_j([\![\overline{u}]\!]_{\mathfrak{A}_\perp[\overline{\pi}]}) & \\ & = & \mathsf{proj}_j(\Phi_{(R,\mathfrak{A})}(\overline{\pi}))([\![\overline{u}]\!]_{\mathfrak{A}_\perp[\overline{\pi}]}) & (\overline{\pi} \text{ ist Fixpunkt von } \Phi_{(R,\mathfrak{A})}) \\ & = & [\![\lambda\overline{F}.\,\lambda^{\sigma_j}\overline{x}.\,t_j]\!]_{\mathfrak{A}_\perp}(\overline{\pi})([\![\overline{u}]\!]_{\mathfrak{A}_\perp[\overline{\pi}]}) & \\ & = & [\![\lambda^{\sigma_j}\overline{x}.\,t_j]\!]_{\mathfrak{A}_\perp[\overline{\pi}]}([\![\overline{u}]\!]_{\mathfrak{A}_\perp[\overline{\pi}]}) & \\ & = & [\![t_j]\!]_{(\mathfrak{A}_\perp[\overline{\pi}],[\overline{x}/[\![\overline{u}]\!]_{\mathfrak{A}_\perp[\overline{\pi}]}])} & (u_1, \dots, u_{\sigma_j} \in A) \\ & = & [\![t_j[\overline{x}/\overline{u}]]\!]_{\mathfrak{A}_\perp[\overline{\pi}]}. & \text{(Vertauschungslemma)} \end{array}$$

(4) Induktion: Sei $\varphi u^w \Longrightarrow \varphi v^w$ aufgrund von $u_i \Longrightarrow v_i$ für $i = 1, \dots, |w|$. Mit der Induktionsvoraussetzung, daß $[\![u_i]\!]_{\mathfrak{A}_\perp[\overline{\pi}]} = [\![v_i]\!]_{\mathfrak{A}_\perp[\overline{\pi}]}$ für alle i, folgt:

$$\begin{array}{rcl} [\![\varphi u^w]\!]_{\mathfrak{A}_\perp[\overline{\pi}]} & = & \varphi_{\mathfrak{A}_\perp[\overline{\pi}]}([\![u^w]\!]_{\mathfrak{A}_\perp[\overline{\pi}]}) \\ & = & \varphi_{\mathfrak{A}_\perp[\overline{\pi}]}([\![v^w]\!]_{\mathfrak{A}_\perp[\overline{\pi}]}) \\ & = & [\![\varphi v^w]\!]_{\mathfrak{A}_\perp[\overline{\pi}]} \end{array}$$

und damit die Behauptung. □

Aus der Invarianz der Fixpunktsemantik bei Reduktionsschritten ergibt sich die für die Definition der Reduktionssemantik erforderliche spezielle Konfluenzeigenschaft und ebenso ihre Korrektheit gegenüber der Fixpunktsemantik.

Definition und Lemma (Reduktionssemantik)
Für $u \in \mathbf{T}(R, A)^s$ und $b_1, b_2 \in A$ gilt:

$$\text{aus} \quad u \stackrel{*}{\Longrightarrow} b_1 \quad \text{und} \quad u \stackrel{*}{\Longrightarrow} b_2 \quad \text{folgt} \quad b_1 = b_2.$$

Wir definieren daher die *Reduktionssemantik* von u durch

$$\mathsf{Rd}[\![u]\!]_{(R,\mathfrak{A})} := b \quad \text{gdw} \quad u \stackrel{*}{\Longrightarrow} b \text{ mit } b \in A.$$

Sie ist nicht definiert, falls kein solches $b \in A$ existiert.

Für $R \in \mathbf{Rfd}_\Sigma^{(w,s)}$ erklären wir die *Reduktionssemantik* als die partielle Abbildung

$$\mathsf{Rd}[\![R]\!]_{\mathfrak{A}} : A^w \dashrightarrow A^s$$

mit $$\mathsf{Rd}[\![R]\!]_{\mathfrak{A}}(a^w) = b \quad \text{gdw} \quad \mathsf{Rd}[\![F_1 a^w]\!]_{(R,\mathfrak{A})} = b. \qquad \lhd$$

Korollar (Korrektheit der Reduktionssemantik)
Sei $R \in \mathbf{Rfd}_\Sigma^{(w,s)}$, $\mathfrak{A} = \langle A; \alpha \rangle$ eine Σ-Algebra mit Verzweigung, $\overline{a} \in A^w$ und $b \in A^s$. Dann gilt:

$$\textit{aus} \quad \mathsf{Rd}[\![R]\!]_{\mathfrak{A}}(\overline{a}) = b \quad \textit{folgt} \quad \mathsf{Fp}[\![R]\!]_{\mathfrak{A}}(\overline{a}) = b.$$

Beweis: Aus $\mathsf{Rd}[\![R]\!]_{\mathfrak{A}}(\overline{a}) = b$ folgt $F_1\overline{a} \overset{*}{\Longrightarrow} b$. Somit gilt $\mathsf{Fp}[\![R]\!]_{\mathfrak{A}}(\overline{a}) = \pi_1(\overline{a}) = \mathsf{Fp}[\![F_1\overline{a}]\!]_{(R,\mathfrak{A})} = b$ wegen der Invarianz der Fixpunktsemantik bei Reduktionen. □

Zum Nachweis der Vollständigkeit der Reduktionssemantik, also der Umkehrung des obigen Korollars, zeigen wir zunächst, daß für die Fixpunktsemantik eines Berechnungsterms eine endliche Approximation $\overline{\pi}^{(k)}$ des Fixpunkts $\overline{\pi}$ ausreicht.

Lemma (Approximationseigenschaft)
Für jeden Berechnungsterm $u \in \mathbf{T}(R, A)$ gibt es ein $k \in \mathbb{N}$, so daß gilt:

$$[\![u]\!]_{\mathfrak{A}_\perp[\overline{\pi}]} \; = \; [\![u]\!]_{\mathfrak{A}_\perp[\overline{\pi}^{(k)}]}.$$

Falls außerdem $[\![u]\!]_{\mathfrak{A}_\perp[\overline{\pi}]} = \perp^s$, so gilt dies mit $k = 0$.

Beweis: Aus der Fixpunkteigenschaft von $\overline{\pi}$ folgt $\overline{\pi} = \bigsqcup_{k\in\mathbb{N}} \overline{\pi}^{(k)}$. Wie bei der Stetigkeit der Termfunktionale kann man auch hier die Vertauschbarkeit der Grenzwertbildung zeigen:

$$[\![u]\!]_{\mathfrak{A}_\perp[\bigsqcup_{k\in\mathbb{N}} \overline{\pi}^{(k)}]} = \bigsqcup_{k\in\mathbb{N}} [\![u]\!]_{\mathfrak{A}_\perp[\overline{\pi}^{(k)}]}.$$

Da für alle k $[\![u]\!]_{\mathfrak{A}_\perp[\overline{\pi}^{(k)}]} \leq [\![u]\!]_{\mathfrak{A}_\perp[\overline{\pi}^{(k+1)}]}$ in $A_\perp$, folgt die Behauptung. □

Nun können wir die Vollständigkeit der Reduktionssemantik durch Induktion über die Struktur von Berechnungstermen beweisen.

Satz (Vollständigkeit der Reduktionssemantik)
Sei $u \in \mathbf{T}(R, A)$, $a \in A$ und $k \in \mathbb{N}$. Dann gilt:

$$(*) \qquad \textit{aus} \quad [\![u]\!]_{\mathfrak{A}_\perp[\overline{\pi}^{(k)}]} = a \quad \textit{folgt} \quad u \overset{*}{\Longrightarrow} a.$$

Beweis durch Induktion über $k \in \mathbb{N}$:
(1) $k = 0$: Wir zeigen die Behauptung $(*)$ für $k = 0$ durch Induktion über die Struktur von $u \in T_{\Sigma[\mathbb{F}_\rho]}(A)$.

(1.1) $u = a \in A$: Offensichtlich gilt $(*)$ wegen $a \stackrel{0}{\Longrightarrow} a$.

(1.2) $u = fu_1 \dots u_n$ mit $f \in \mathcal{F}_{\mathsf{base}}$ und $(*)$ gelte mit $k = 0$ für $u_1, \dots, u_n$. Dann folgt:

$$\begin{aligned} a &= [\![fu_1 \dots u_n]\!]_{\mathfrak{A}_\perp[\overline{\pi}^{(0)}]} \\ &= f_{\mathfrak{A}_\perp}([\![u_1]\!]_{\mathfrak{A}_\perp[\overline{\pi}^{(0)}]}, \dots, [\![u_n]\!]_{\mathfrak{A}_\perp[\overline{\pi}^{(0)}]}). \end{aligned}$$

$f_{\mathfrak{A}_\perp}$ ist strikt. Daher existiert für $i = 1, \dots, n$ $a_i \in A$ mit $[\![u_i]\!]_{\mathfrak{A}_\perp[\overline{\pi}^{(0)}]} = a_i$, so daß nach Induktionsvoraussetzung $u_i \stackrel{*}{\Longrightarrow} a_i$. Diese Reduktionen lassen sich kombinieren zu $fu_1 \dots u_n \stackrel{*}{\Longrightarrow} fa_1 \dots a_n \Longrightarrow a$.

(1.3) $u = \mathsf{cond}\, u_0u_1u_2$ und $(*)$ gelte mit $k = 0$ für u_0, u_1, u_2. Dann folgt aus $[\![u]\!]_{\mathfrak{A}_\perp[\overline{\pi}^{(0)}]} = a$, daß

$$\begin{array}{lll} \text{entweder} & [\![u_0]\!]_{\mathfrak{A}_\perp[\overline{\pi}^{(0)}]} = \mathsf{true} & \text{und} \quad a = [\![u_1]\!]_{\mathfrak{A}_\perp[\overline{\pi}^{(0)}]} \\ \text{oder} & [\![u_0]\!]_{\mathfrak{A}_\perp[\overline{\pi}^{(0)}]} = \mathsf{false} & \text{und} \quad a = [\![u_2]\!]_{\mathfrak{A}_\perp[\overline{\pi}^{(0)}]}. \end{array}$$

In beiden Fällen gibt es nach Induktionsvoraussetzung Reduktionsfolgen, die sich komponieren lassen zu $\mathsf{cond}\, u_0u_1u_2 \stackrel{*}{\Longrightarrow} a$.

(1.4) $u = F_j u^w$: In diesem letzten Fall gilt $[\![u]\!]_{\mathfrak{A}_\perp[\overline{\pi}^{(0)}]} = \pi_j^{(0)}([\![u^w]\!]_{\mathfrak{A}_\perp[\overline{\pi}^{(0)}]}) = \perp^{s_j}$, so daß gar kein entsprechendes $a \in A$ existieren kann. $(*)$ gilt daher trivialerweise.

Nach struktureller Induktion gilt somit $(*)$ für $k = 0$.

(2) Induktion von k auf $k+1$: Nach Induktionsvoraussetzung gelte $(*)$ mit $k \in \mathbb{N}$ für alle $u \in \mathbf{T}(R, A)$ und $a \in A$. Den Nachweis von $(*)$ für $k+1$ führen wir wiederum durch Induktion über die Struktur von $u \in T_{\Sigma[\mathbb{F}_\rho]}(A)$.
In den Fällen (2.1) $u = a$, (2.2) $u = fu_1 \dots u_n$ und (2.3) $u = \mathsf{cond}\, u_0u_1u_2$ schließt man genauso wie in (1.1) - (1.3), weil der Approximationsindex k hier ohne Einfluß ist. Damit bleibt als Beweiskern der Fall

(2.4) $u = F_j u^w$ mit $u^w = (u_1, \dots, u_n)$. Nach struktureller Induktionsvoraussetzung gelte $(*)$ mit $k+1$ für $u_1, \dots, u_n$. Dann folgt für u:

$$\begin{aligned} a &= [\![F_j u^w]\!]_{\mathfrak{A}_\perp[\overline{\pi}^{(k+1)}]} \\ &= \pi_j^{(k+1)}([\![u^w]\!]_{\mathfrak{A}_\perp[\overline{\pi}^{(k+1)}]}) \\ &= \mathsf{proj}_j(\Phi_{(R,\mathfrak{A})}(\overline{\pi}^{(k)}))([\![u^w]\!]_{\mathfrak{A}_\perp[\overline{\pi}^{(k+1)}]}) \\ &= [\![\lambda^{\sigma_j} x^w.\, t_j]\!]_{\mathfrak{A}_\perp[\overline{\pi}^{(k)}]}([\![u^w]\!]_{\mathfrak{A}_\perp[\overline{\pi}^{(k+1)}]}) \\ &= [\![t_j]\!]_{(\mathfrak{A}_\perp[\overline{\pi}^{(k)}],[x^w/[\![u^w]\!]_{\mathfrak{A}_\perp[\overline{\pi}^{(k+1)}]}])} \end{aligned}$$

Dabei erzwingt der Striktheitsindex σ_j wegen des Resultats $a \in A$, daß $a_i := [\![u_i]\!]_{\mathfrak{A}_\perp[\overline{\pi}^{(k+1)}]} \in A$ für $i = 1, \dots, \sigma_j$. Für $i > \sigma_j$ ist es möglich, daß $a_i = \perp^{s_i}$. Um den

Approximationsindex von $k+1$ auf k zu drücken und die Induktionsvoraussetzung für k anwenden zu können, wählen wir für alle $i = 1, \ldots, n$ die Berechnungsterme

$$v_i := \begin{cases} a_i \text{ falls } a_i \in A \\ u_i \text{ falls } a_i = \perp^{s_i} \end{cases} .$$

Es folgt $[\![v_i]\!]_{\mathfrak{A}_\perp[\overline{\pi}^{(k)}]} = [\![u_i]\!]_{\mathfrak{A}_\perp[\overline{\pi}^{(k+1)}]}$, da insbesondere wegen der Monotonie der Termsemantik

$$[\![u_i]\!]_{\mathfrak{A}_\perp[\overline{\pi}^{(k)}]} \leq [\![u_i]\!]_{\mathfrak{A}_\perp[\overline{\pi}^{(k+1)}]}.$$

Damit ergibt sich

$$\begin{aligned} a &= [\![t_j]\!]_{(\mathfrak{A}_\perp[\overline{\pi}^{(k)}],[x^w/[\![v^w]\!]_{\mathfrak{A}_\perp[\overline{\pi}^{(k)}]}])} \\ &= [\![t_j[x^w/v^w]]\!]_{\mathfrak{A}_\perp[\overline{\pi}^{(k)}]} \end{aligned}$$

nach dem Vertauschungslemma, so daß nach Induktionsvoraussetzung für k folgt:

$$t_j[x^w/v^w] \stackrel{*}{\Longrightarrow} a.$$

Da v^w der Striktheitsbedingung für die Funktionsreduktion von F_j genügt, gilt außerdem

$$F_j v^w \Longrightarrow t_j[x^w/v^w].$$

Schließlich folgt aus der Induktionsvoraussetzung für die Terme u_i mit $k+1$ im Falle $v_i = a_i = [\![u_i]\!]_{\mathfrak{A}_\perp[\overline{\pi}^{(k+1)}]}$, daß

$$u_i \stackrel{*}{\Longrightarrow} v_i.$$

In den übrigen Fällen gilt dies aber ohnehin wegen $u_i = v_i$, so daß $F_j u^w \stackrel{*}{\Longrightarrow} F_j v^w$. Durch Komposition erhalten wir die gewünschte Reduktion

$$F_j u^w \stackrel{*}{\Longrightarrow} a.$$

Dies beschließt die strukturelle Induktion über u und damit auch die Induktion über k. $\square$

Korollar (Äquivalenz von Reduktions- und Fixpunktsemantik)
Sei $R \in \mathbf{Rfd}_\Sigma^{(w,s)}$ und $\mathfrak{A} = \langle A; \alpha \rangle$ eine Σ-Algebra mit Verzweigung. Dann gilt für alle $\overline{a} \in A^w$ und $b \in A^s$

$$\mathsf{Rd}[\![R]\!]_{\mathfrak{A}}(\overline{a}) = b \quad gdw \quad \mathsf{Fp}[\![R]\!]_{\mathfrak{A}}(\overline{a}) = b.$$

Beweis: Da die Korrektheit der Reduktionssemantik bereits gezeigt worden ist, verbleibt noch der Vollständigkeitsnachweis. Sei also $\mathsf{Fp}[\![R]\!]_{\mathfrak{A}}(\overline{a}) = b$. Dann folgt

$$b = \pi_1(\overline{a}) = [\![F_1\overline{a}]\!]_{\mathfrak{A}_\perp[\overline{\pi}]} = [\![F_1\overline{a}]\!]_{\mathfrak{A}_\perp[\overline{\pi}^{(k)}]}$$

für ein passendes $k \in \mathbb{N}$. Der Vollständigkeitssatz impliziert $F_1\overline{a} \stackrel{*}{\Longrightarrow} b$, also $\mathsf{Rd}[\![F_1\overline{a}]\!]_{(R,\mathfrak{A})} = b$ und daher $\mathsf{Rd}[\![R]\!]_{\mathfrak{A}}(\overline{a}) = b$. □

l-Reduktionssemantik

Als Basis einer sequentiellen Implementierung ist die Reduktionssemantik wegen ihres Nichtdeterminismus noch unzureichend. Wir stellen deshalb zum Abschluß eine deterministische Auswertungsstrategie vor, die ebenfalls die gewünschte Vollständigkeitseigenschaft besitzt. Wie schon erwähnt, erweist sich der am weitesten links stehende reduzierbare Teilterm eines Berechnungsterms als geeignete Wahl für den nächsten Reduktionsschritt.

Definition (l-Reduktion)
Die *l-Reduktionsrelation*

$$\underset{l}{\Longrightarrow} \;\subseteq\; \mathbf{T}(R,A)^2$$

ist durch folgende strukturelle Induktion definiert:

- Für jede Berechnungsregel $u \longrightarrow v$ gilt $u \underset{l}{\Longrightarrow} v$.
- Für $fa_1 \dots a_{i-1}u_i \dots u_n \in \mathbf{T}(R,A)$
  mit $f \in \mathcal{F}_{\mathsf{base}}$, $1 \le i \le n$, $a_1, \dots, a_{i-1} \in A$ und $u_i \notin A$ gilt:
  wenn $u_i \underset{l}{\Longrightarrow} v_i$, so auch $fa_1 \dots a_{i-1}u_i \dots u_n \underset{l}{\Longrightarrow} fa_1 \dots a_{i-1}v_iu_{i+1} \dots u_n$.
- Für $\mathsf{cond}_s uu_1u_2 \in \mathbf{T}(R,A)$ mit $u \notin A$ gilt:
  wenn $u \underset{l}{\Longrightarrow} v$, so auch $\mathsf{cond}_s uu_1u_2 \underset{l}{\Longrightarrow} \mathsf{cond}_s vu_1u_2$.
- Für $F_ja_1 \dots a_{i-1}u_i \dots u_n \in \mathbf{T}(R,A)$
  mit $1 \le i \le \sigma_j$, $a_1, \dots, a_{i-1} \in A$ und $u_i \notin A$ gilt:
  wenn $u_i \underset{l}{\Longrightarrow} v_i$, so auch $F_ja_1 \dots a_{i-1}u_i \dots u_n \underset{l}{\Longrightarrow} F_ja_1 \dots a_{i-1}v_iu_{i+1} \dots u_n$.

◁

Lemma (Determinismus der l-Reduktion)
Für jedes $u \in \mathbf{T}(R,A)$ gilt:

- *Wenn $u \notin A$, so gibt es genau ein $v \in \mathbf{T}(R,A)$ mit $u \underset{l}{\Longrightarrow} v$.*
- *Wenn $u \in A$, so gibt es kein solches v.*

Beweis durch Induktion über die Struktur von u. □

Durch Beschränkung auf l-Reduktionsschritte erhalten wir die l-Reduktionssemantik.

Definition (l-Reduktionssemantik)
Für $u \in \mathbf{T}(R, A)$ definieren wir die *l-Reduktionssemantik* durch

$$\mathsf{lRd}[\![u]\!]_{(R,\mathfrak{A})} := b \quad \text{gdw} \quad u \underset{l}{\overset{*}{\Longrightarrow}} b \;\text{ mit }\; b \in A$$

und übertragen sie auf $R \in \mathbf{Rfd}_\Sigma^{(w,s)}$ als die partielle Abbildung

$$\mathsf{lRd}[\![R]\!]_\mathfrak{A} : A^w \dashrightarrow A^s$$

mit $$\mathsf{lRd}[\![R]\!]_\mathfrak{A}(a^w) = b \quad \text{gdw} \quad \mathsf{lRd}[\![F_1 a^w]\!]_{(R,\mathfrak{A})} = b. \qquad \triangleleft$$

Satz (Vollständigkeit der l-Reduktionssemantik)
Sei $R \in \mathbf{Rfd}_\Sigma^{(w,s)}$ und $\mathfrak{A} = \langle A; \alpha\rangle$ eine Σ-Algebra mit Verzweigung. Dann gilt:

$$\mathsf{lRd}[\![R]\!]_\mathfrak{A} = \mathsf{Rd}[\![R]\!]_\mathfrak{A}.$$

Beweis: Offensichtlich gilt mit $F_1\overline{a} \underset{l}{\overset{*}{\Longrightarrow}} b$ auch $F_1\overline{a} \overset{*}{\Longrightarrow} b$. Für die Umkehrung, also die Vollständigkeit der l-Reduktionssemantik, zeigen wir durch Induktion über $m \in \mathbb{N}$, daß für alle $u \in \mathbf{T}(R, A)$ und $a \in A$ gilt:

$$\text{aus} \quad u \overset{m}{\Longrightarrow} a \quad \text{folgt} \quad u \underset{l}{\overset{*}{\Longrightarrow}} a.$$

(1) $m = 0$: Aus $u \overset{0}{\Longrightarrow} a$ folgt $u = a$ und somit $u \underset{l}{\overset{*}{\Longrightarrow}} a$.

(2) Induktion von m auf $m' := m + 1$: Nach Induktionsvoraussetzung gelte mit $u \overset{m}{\Longrightarrow} a$ auch $u \underset{l}{\overset{*}{\Longrightarrow}} a$. Nun sei $u \overset{m'}{\Longrightarrow} a$. Wir unterscheiden vier Fälle für u.

(2.1) $u = a$: Wie in (1) folgt direkt $u \underset{l}{\overset{*}{\Longrightarrow}} a$.

(2.2) $u = fu_1 \dots u_n$ mit $f \in \mathcal{F}_{\mathsf{base}}$: Dann gilt $fu_1 \dots u_n \overset{m}{\Longrightarrow} fa_1 \dots a_n \Longrightarrow f_\mathfrak{A}(a_1, \dots, a_n) = a$. Nach Definition der Reduktionsrelation basiert dies auf der simultanen Durchführung von Reduktionen $u_i \overset{m}{\Longrightarrow} a_i$, so daß nach Induktionsvoraussetzung $u_i \underset{l}{\overset{*}{\Longrightarrow}} a_i$ für $i = 1, \dots, n$ gilt. Nach Definition der l-Reduktion lassen sich diese sequentiell kombinieren zu

$$fu_1 \dots u_n \underset{l}{\overset{*}{\Longrightarrow}} fa_1u_2 \dots u_n \underset{l}{\overset{*}{\Longrightarrow}} fa_1 \dots a_n \underset{l}{\Longrightarrow} a.$$

(2.3) $u = \mathsf{cond}\ u_0u_1u_2$: Hier gilt entweder $u_0 \overset{m}{\Longrightarrow} \mathsf{true}$ und $u_1 \overset{m}{\Longrightarrow} a$ oder $u_0 \overset{m}{\Longrightarrow} \mathsf{false}$ und $u_2 \overset{m}{\Longrightarrow} a$. Nach Induktionsvoraussetzung erhalten wir im ersten Fall $u_0 \underset{l}{\overset{*}{\Longrightarrow}} \mathsf{true}$ und $u_1 \underset{l}{\overset{*}{\Longrightarrow}} a$, so daß $\mathsf{cond}\ u_0u_1u_2 \underset{l}{\overset{*}{\Longrightarrow}} \mathsf{cond}\ \mathsf{true}\ u_1u_2 \underset{l}{\Longrightarrow} u_1 \underset{l}{\overset{*}{\Longrightarrow}} a$. Im false-Fall schließt man analog.

(2.4) $u = F_ju^w$ mit $u^w = (u_1, \dots, u_n)$: Dann muß für $u \overset{m'}{\Longrightarrow} a$ eine Reduktionsfolge $F_ju^w \overset{p}{\Longrightarrow} F_jv^w \Longrightarrow t_j[x^w/v^w] \overset{q}{\Longrightarrow} a$ existieren, so daß $p + q = m$,

$v^w = (v_1, \dots, v_n) \in \mathbf{T}(R, A)^w$ und $v_1, \dots, v_{\sigma_j} \in A$. Folglich gilt $u_i \overset{p}{\Longrightarrow} v_i$ für $i = 1, \dots, n$ und somit nach Induktionsvoraussetzung $u_i \overset{*}{\underset{l}{\Longrightarrow}} v_i$ für $i = 1, \dots, \sigma_j$. Diese Reduktionen lassen sich kombinieren zu $F_j u^w \overset{*}{\underset{l}{\Longrightarrow}} F_j v_1 \dots v_{\sigma_j} u_{\sigma_j+1} \dots u_n \underset{l}{\Longrightarrow} t_j[x^w/(v_1, \dots v_{\sigma_j}, u_{\sigma_j+1}, \dots, u_n)] =: t'$. Nach dem Lemma über die Reduktion von Teiltermen folgt weiter $t' \overset{p}{\Longrightarrow} t_j[x^w/v^w] \overset{q}{\Longrightarrow} a$, so daß sich wiederum nach Induktionsvoraussetzung wegen $p + q = m$ $\ t' \overset{*}{\underset{l}{\Longrightarrow}} a$ ergibt, und damit auch die Behauptung. $\square$

Beachte, daß der Beweis wesentlich auf der simultanen Reduktionsmöglichkeit von Teiltermen beruht. Im Fall (2.4) kann die Reduktion der Funktionsargumente $u_{\sigma_j+1}, \dots, u_n$ nach Ausführung der Funktionsreduktion erfolgen, ohne daß sich die Anzahl der Reduktionsschritte erhöht. Damit kann eine Induktion über die Länge der Reduktionsfolge durchgeführt werden.

Korollar
Sei $R \in \mathbf{Rfd}_\Sigma^{(w,s)}$ und $\mathfrak{A} = \langle A; \alpha \rangle$ eine Σ-Algebra mit Verzweigung. Dann gilt für alle $\overline{a} \in A^w$ und $b \in A^s$

$$\boxed{\mathsf{lRd}[\![R]\!]_{\mathfrak{A}}(\overline{a}) = b \quad gdw \quad \mathsf{Rd}[\![R]\!]_{\mathfrak{A}}(\overline{a}) = b \quad gdw \quad \mathsf{Fp}[\![R]\!]_{\mathfrak{A}}(\overline{a}) = b.}$$

Schlußbemerkungen

Wir haben gezeigt, daß eine algebraische Sicht rekursiver Funktionsdefinitionen einen geeigneten formalen Rahmen für die Definition denotationaler und operationaler Semantiken sowie den Nachweis ihrer Äquivalenz bietet. Die Striktheitsinformation der Funktionsargumente wurde nicht nur zur Festlegung einer Auswertungsstragie verwendet, sondern auch bei der Fixpunktsemantik konsequent berücksichtigt. Daß dies ebenso bei der nichtdeterministischen Reduktionssemantik möglich ist, verdanken wir einem Hinweis von Olaf Chitil. Diese Semantik erlaubt die simultane Reduktion von Teiltermen und trägt wesentlich zur Vereinfachung der Äquivalenzbeweise bei. Für die operationale Ebene haben wir eine sequentielle Einzelschrittsemantik gewählt, die als Vorstufe einer Stackimplementierung gesehen werden kann. In einer weiteren Arbeit wird die Übersetzung rekursiver Funktionsdefinitionen in Stackcode vorgestellt und ihre Korrektheit nachgewiesen. So wollen wir mit diesen Ergebnissen zu einer algebraischen Theorie der Compilerkorrektheit beitragen.

An dieser Stelle möchte ich Thomas Noll für seine unermüdliche Diskussionsbereitschaft und die sorgfältige Durchsicht dieser Arbeit danken, die er mit zahlreichen Verbesserungsvorschlägen unterstützt hat.

Literatur

[1] J.M. Cadiou. *Recursive Definitions of Partial Functions and Their Computation.* PhD thesis, Stanford University, 1972.

[2] O. Chitil. The ς–semantics: A comprehensive semantics for functional programs. *Fundamenta Informaticae*, 31:253–294, 1997.

[3] J.W. de Bakker. Least fixed points revisited. *Theor. Computer Science*, 2:155–181, 1976.

[4] J.A. Goguen, J.W. Thatcher, E.G. Wagner, and J.B. Wright. Initial algebra semantics and continuous algebras. *Journal ACM*, 24:68–95, 1977.

[5] S.C. Kleene. *Introduction to Metamathematics.* North–Holland, Amsterdam, 1952.

[6] Z. Manna. *Mathematical Theory of Computation.* McGraw–Hill, New York, 1974.

[7] Z. Manna and J. Vuillemin. Fixpoint approach to the theory of computation. *Comm. ACM*, 15:528–536, 1972.

[8] J. McCarthy. Recursive functions of symbolic expressions and their computation by machine, part I. *Comm. ACM*, 3:184–195, 1960.

[9] D. Scott. Outline of a mathematical theory of computation. In *Proc. of the 4th Annual Princeton Conference on Information Sciences and Systems*, pages 169–176, Princeton, 1970.

[10] D. Scott and C. Strachey. Towards a mathematical semantics for computer languages. In J. Fox, editor, *Proc. Symp. Computers and Automata*, pages 19–46, 1971.

[11] G. Winskel. *The Formal Semantics of Programming Languages.* The MIT Press, Cambridge, Mass., 1993.