Karlsruher Institut für Technologie

Lehrstuhl für Programmierparadigmen
Sebastian Ullrich sebastian.ullrich@kit.edu
Max Wagner maximilian.wagner@kit.edu

Semantik von Programmiersprachen – SS 2019

http://pp.ipd.kit.edu/lehre/SS2019/semantik

Lösungen zu Blatt 10: Fixpunkttheorie Besprechung: 01.06.2019

1. Welche der folgenden Aussagen sind richtig, welche falsch? (H)

(a) Für $f(\sigma) = \sigma[\mathtt{i} \mapsto 1]$ und $g(\sigma) = \sigma[\mathtt{i} \mapsto 2]$ gilt $f \sqsubseteq g$.

(b) Jede Teilmenge von $\mathbb{R}_0^+ = \{\, x \in \mathbb{R} \mid x \geq 0 \,\}$ hat bezüglich der normalen Ordnung $\leq$ auf den reellen Zahlen $\mathbb{R}$ ein kleinstes Element.

(c) Jede Teilmenge einer total geordneten Menge ist eine Kette.

(d) In einer ccpo $(D, \sqsubseteq)$ hat jede Menge $M \subseteq D$ eine obere Schranke.

(e) Die Menge $\mathfrak{P}^{fin}(\mathbb{N})$ der endlichen Teilmengen von $\mathbb{N}$ ist mit der Teilmengenrelation $\subseteq$ als Ordnung eine ccpo.

(f) Jede ccpo $(D, \sqsubseteq)$ hat ein kleinstes Element.

(g) Das abgeschlossene Intervall $[0, 1] \subseteq \mathbb{R}$ ist eine ccpo mit $\leq$ als Ordnung.

(h) IF $(p,\ f,\ g)$ ist strikt in f.

(i) Wenn $f \circ g$ kettenstetig ist, dann sind auch f und g kettenstetig.

Lösung:

(1a) Falsch. Die Approximationsordnung $\sqsubseteq$ vergleicht nicht die Größe der Werte, die einer Variablen zugewiesen werden, sondern die Definiertheit der Funktionen selbst.

(1b) Falsch. Beispielsweise hat $\{x \in \mathbb{R} \mid x > 0\}$ kein kleinstes Element, da 0 nicht enthalten ist. In $\mathbb{R}_0^+$ hat zwar jede Menge M eine kleinste untere Schranke, aber diese muss nicht in M enthalten sein.

Nebenbemerkung: Nimmt man das Auswahlaxiom der Mengenlehre als gültig an, gibt es auf $\mathbb{R}_0^+$ eine (von $<$ verschiedene) Ordnung, bezüglich der jede Teilmenge ein kleinstes Element hat.

(1c) Richtig. Eine Kette ist nichts anderes als eine total geordnete Menge. Teilmengen total geordneter Mengen bleiben total geordnet.

(1d) Falsch. Sei $D = \Sigma \rightharpoonup \Sigma$ und $\sqsubseteq$ die Approximationsordnung. Dann gibt es keine obere Schranke für $\{\, f,\ id \,\}$ mit $f(\sigma) = \sigma[\mathtt{x} \mapsto 1]$.

(1e) Falsch. Betrache die Menge $M = \{\, \{\, 0, \ldots, n \,\} \mid n \in \mathbb{N} \,\} \subseteq \mathfrak{P}^{fin}(\mathbb{N})$ Dann ist M eine Kette bezüglich $\subseteq$. Die kleinste obere Schranke von M ist $\mathbb{N}$, aber $\mathbb{N} \notin \mathfrak{P}^{fin}(\mathbb{N})$.

(1f) Richtig. Die leere Menge $\emptyset$ ist immer eine Kette in $(D, \sqsubseteq)$. $\bigsqcup \emptyset$ ist dann das kleinste Element von D:

Sei $d \in D$ beliebig. Dann ist d eine obere Schranke von $\emptyset$. Da $\bigsqcup \emptyset$ die kleinste obere Schranke von $\emptyset$ ist, gilt $\bigsqcup \emptyset \sqsubseteq d$.

(1g) Richtig. $\mathbb{R}$ ist total geordnet, damit auch das Intervall, somit ist jede Teilmenge eine Kette. Sei also $M \subseteq [0, 1]$. Definiere

$$\bigsqcup M = \begin{cases} 0 & \text{falls } M = \emptyset \\ \sup(M) & \text{sonst} \end{cases}$$

wobei $\sup(M)$ die kleinste obere Schranke in $\mathbb{R}$ liefert (die für alle nicht-leeren, beschränkten Mengen M in $\mathbb{R}$ existiert). Da M durch 0 und 1 beschränkt ist, muss $0 \leq \sup(M) \leq 1$ gelten. Somit $\bigsqcup M \in [0, 1]$.

Anmerkung: Wenn das Intervall nicht abgeschlossen wäre, wäre es keine ccpo, weil entweder das kleinste Element (links offen) oder das größte Element (rechts offen) fehlt.

(1h) Falsch. Das `if`-Konstrukt ist in allen (vernünftigen) Programmiersprachen nicht strikt – sonst müsste es immer sowohl `then`- als auch den `else`-Zweig auswerten. Thm. 109 zeigt zwar, dass $\text{IF}\,(p,\ f,\ g)$ kettenstetig ist. Der Beweis funktioniert aber nicht, wenn man auch die leere Kette $\emptyset$ für Y erlaubt, falls $p \neq \lambda\sigma.\ \mathbf{tt}$ gilt. Hier ein Gegenbeispiel mit $p(\sigma) = \mathbf{tt}$ und $g = id$.

$$\text{IF}\left(\lambda\sigma.\ \mathbf{ff},\ \bigsqcup\emptyset,\ id\right) = id \not\sqsubseteq \bot = \bigsqcup\emptyset = \bigsqcup\{\,\text{IF}\,(\lambda\sigma.\ \mathbf{ff},\ f,\ id) \mid f \in \emptyset\,\}$$

(1i) Falsch. Sei z.B. $g = \bot$, f beliebig nicht kettenstetig. Dann ist $f \circ g = \bot$ und $\bot$ ist kettenstetig.

2. Monotonie und Fixpunkte (H)

Finden Sie eine Halbordnung $(D, \sqsubseteq)$ mit kleinstem Element $\bot$ und eine monotone Funktion $f : D \to D$, die mehrere Fixpunkte besitzt, aber keinen kleinsten.

Lösung: Das geforderte D kann nicht endlich sein, da alle endlichen Halbordnungen $(D, \sqsubseteq)$ mit kleinstem Element kettenvollständig und alle monotonen Funktionen auf endlichen Halbordnungen $(D, \sqsubseteq)$ automatisch kettenstetig sind.

Eine Lösung:
$D = \mathbb{N} \cup \{\,A,\ B\,\}$
$m \sqsubseteq n = m \leq n$, $m \sqsubseteq A$, $m \sqsubseteq B$, $A \sqsubseteq A$, $B \sqsubseteq B$ für alle $n, m \in \mathbb{N}$
$f(n) = n + 1$ für $n \in \mathbb{N}$, $f(A) = A$, $f(B) = B$.

Dann hat f nur die Fixpunkte A und B, aber weder $A \sqsubseteq B$ noch $B \sqsubseteq A$.

3. Exkurs: monadische Semantik (H)

In der funktionalen Programmierung kennt man das Abstraktionskonzept der *Monade*, die unter anderem durch die folgenden zwei Operationen charakterisiert werden kann:

$$\begin{aligned} &\mathit{pure} :: a \to m\ a \\ &(>\!\!=\!\!>) :: (a \to m\ b) \to (b \to m\ c) \to (a \to m\ c) \end{aligned}$$

Statt auf $\Sigma \rightharpoonup \Sigma$ können wir $\mathcal{D}\,[\![\_]\!]$ auf dem allgemeineren Typ $\Sigma \to m\ \Sigma$ definieren, angenommen $m\ \Sigma$ ist eine ccpo[1].

$$\begin{aligned} \mathcal{D}_m\,[\![\texttt{skip}]\!] &= \mathit{pure} \\ \mathcal{D}_m\,[\![x\ \texttt{:=}\ a]\!] &= \mathit{pure} \circ \lambda\sigma.\ \sigma[x \mapsto \mathcal{A}\,[\![a]\!]\,\sigma] \\ \mathcal{D}_m\,[\![c_1\texttt{;}\ c_2]\!] &= \mathcal{D}_m\,[\![c_1]\!] >\!\!=\!\!> \mathcal{D}_m\,[\![c_2]\!] \\ \mathcal{D}_m\,[\![\texttt{if (}b\texttt{) then}\ c_1\ \texttt{else}\ c_2]\!] &= \text{IF}\,(\mathcal{B}\,[\![b]\!],\ \mathcal{D}_m\,[\![c_1]\!],\ \mathcal{D}_m\,[\![c_2]\!]) \\ \mathcal{D}_m\,[\![\texttt{while (}b\texttt{) do}\ c]\!] &= \text{FIX}\,(\lambda f.\ \text{IF}\,(\mathcal{B}\,[\![b]\!],\ \mathcal{D}_m\,[\![c]\!] >\!\!=\!\!> f,\ \mathit{pure})) \end{aligned}$$

$\mathcal{D}_{Maybe}\,[\![\_]\!]$ entspricht damit der alten Semantik. Fallen Ihnen andere Monaden ein, die sinnvolle Semantiken ergeben?

[1] Man überlege sich, dass $\Sigma \to m\ \Sigma$ dann auch eine ccpo sein muss

Lösung: Ein paar Vorschläge:

Monade	Ordnung	modelliert
Maybe	Approximationsordnung	Nichttermination
~~Identity~~	Hat kein kleinstes Element!	
List	$\subseteq$	Nichtdeterminismus

Viele Erweiterungen lassen sich als Monad-Transformer modellieren und sind damit beliebig kombinierbar! Als Ordnung kann die der Basis-Monade *Maybe* benutzt werden.

Transformer	modelliert
ExceptT	Exceptions
ListT	Nichtdeterminismus
ConT	Continuations
ReaderT	Eingaben
WriterT	Ausgaben

4. `repeat c until b`-Schleife (Ü)

In einer früheren Aufgabe haben wir schon die operationale Semantik einer `repeat`-Schleife betrachtet. Com wird dazu um das Syntaxkonstrukt `repeat` c `until` b erweitert und die operationale Big-Step-Semantik durch die Regeln

$$\textsc{RepeatTT:}\ \frac{\langle c,\ \sigma\rangle \Downarrow \sigma' \qquad \mathcal{B}[\![b]\!]\,\sigma' = \mathbf{tt}}{\langle \texttt{repeat}\ c\ \texttt{until}\ b,\ \sigma\rangle \Downarrow \sigma'}$$

$$\textsc{RepeatFF:}\ \frac{\langle c,\ \sigma\rangle \Downarrow \sigma' \qquad \mathcal{B}[\![b]\!]\,\sigma' = \mathbf{ff} \qquad \langle \texttt{repeat}\ c\ \texttt{until}\ b,\ \sigma'\rangle \Downarrow \sigma''}{\langle \texttt{repeat}\ c\ \texttt{until}\ b,\ \sigma\rangle \Downarrow \sigma''}$$

(a) Leiten Sie daraus die Rekursionsgleichung für $\mathcal{D}[\![\texttt{repeat}\ c\ \texttt{until}\ b]\!]$ her.

(b) Erweitern Sie die Definition von $\mathcal{D}[\![\cdot]\!]$ um `repeat` c `until` b.

(c) Prüfen Sie, ob die Semantik mit Ihrer Erweiterung weiterhin wohldefiniert und kompositional ist.

(d) Zeigen oder widerlegen Sie: $\mathcal{D}[\![\texttt{repeat}\ c\ \texttt{until}\ b]\!] = \mathcal{D}[\![c\texttt{; while (not } b\texttt{) do } c]\!]$

Lösung:

(4a) Analog zur `while`-Schleife erhält man:

$$\mathcal{D}[\![\texttt{repeat}\ c\ \texttt{until}\ b]\!]\,\sigma = \begin{cases} \mathcal{D}[\![c]\!]\,\sigma & \text{falls } \mathcal{B}[\![b]\!]\,(\mathcal{D}[\![c]\!]\,\sigma) = \mathbf{tt} \\ \mathcal{D}[\![\texttt{repeat}\ c\ \texttt{until}\ b]\!]\,(\mathcal{D}[\![c]\!]\,\sigma) & \text{falls } \mathcal{B}[\![b]\!]\,(\mathcal{D}[\![c]\!]\,\sigma) = \mathbf{ff} \end{cases}$$

Problematisch an dieser Gleichung ist, dass $\mathcal{D}[\![c]\!]\,\sigma$ ggf. gar nicht definiert ist, somit die Fallunterscheidung mit $\mathcal{B}[\![b]\!]$ ebenfalls nicht.

Alternativer Lösungsvorschlag:
In dieser Lösung wird das o.g. Problem umgangen, da die Funktionskomposition $\circ$ selbst dann definiert ist, wenn $\mathcal{D}[\![c]\!]\,\sigma$ undefiniert ist.

$$\begin{aligned}\mathcal{D}[\![\texttt{repeat}\ c\ \texttt{until}\ b]\!] &= \left(\lambda\sigma.\ \begin{cases}\sigma & \text{falls } \mathcal{B}[\![b]\!]\,\sigma = \mathbf{tt} \\ \mathcal{D}[\![\texttt{repeat}\ c\ \texttt{until}\ b]\!]\,(\sigma) & \text{falls } \mathcal{B}[\![b]\!]\,\sigma = \mathbf{ff}\end{cases}\right) \circ \mathcal{D}[\![c]\!] \\ &= \mathrm{IF}\,(\mathcal{B}[\![b]\!],\ id,\ \mathcal{D}[\![\texttt{repeat}\ c\ \texttt{until}\ b]\!]) \circ \mathcal{D}[\![c]\!]\end{aligned}$$

(4b) Analog zur `while`-Schleife nimmt man den Fixpunkt des zur Rekursionsgleichung zugehörigen Funktionals:

$$\mathcal{D}[\![\texttt{repeat } c \texttt{ until } b]\!] = \mathrm{FIX}\,(\lambda f.\ \mathrm{IF}\,(\mathcal{B}[\![b]\!],\ id,\ f) \circ \mathcal{D}[\![c]\!])$$

Je nach Lösungsweg von Teilaufgabe a) muss man hier zuerst das $\mathrm{IF}\,(\cdot,\ \cdot,\ \cdot)$-Funktional auf partielle Prädikate erweitern.

(4c) Für Wohldefiniertheit ist zu zeigen, dass das neue Funktional unter dem Fixpunkt immer kettenstetig ist, wie Thm. 109 für das Funktional von `while`.

Beweis. Zu zeigen ist, dass das Funktional

$$F = (\lambda f.\ \mathrm{IF}\,(\mathcal{B}[\![b]\!],\ id,\ f) \circ \mathcal{D}[\![c]\!])$$

monoton und kettenstetig ist.

Dazu schreibt man zuerst das Funktional wieder in einfachere Bestandteile um:

$$F = (\lambda f.\ f \circ \mathcal{D}[\![c]\!]) \circ (\lambda f.\ \mathrm{IF}\,(\mathcal{B}[\![b]\!],\ id,\ f))$$

Nach Bsp. 102 wissen wir, dass $\mathrm{IF}\,(p,\ f,\ g)$ monoton in g ist. Außerdem ist nach Thm. 109 $(\lambda f.\ f \circ g)$ monoton. Mit Lemma 103 ist daher F monoton.

Analog zu Thm. 109 lässt sich zeigen, dass $\mathrm{IF}\,(p,\ g,\ f)$ kettenstetig in f ist. Außerdem ist nach Thm. 109 $(\lambda f.\ f \circ g)$ kettenstetig. Mit Lemma 107 ist daher F auch kettenstetig. □

Die erweiterte Definition ist weiterhin kompositional, da nur die Semantiken der Teile verwendet werden. Eine nicht-kompositionale Definition wäre z.B. über die `while`-Schleife:

$$\mathcal{D}[\![\texttt{repeat } c \texttt{ until } b]\!] = \mathcal{D}[\![\texttt{while (not } b\texttt{) do } c]\!] \circ \mathcal{D}[\![c]\!]$$

(4d) Zuerst benötigen wir ein einfaches Hilfslemma über das $\mathrm{IF}\,(\cdot,\ \cdot,\ \cdot)$-Funktional:

$$\mathrm{IF}\,(\mathcal{B}[\![\texttt{not } b]\!],\ f,\ g) = \mathrm{IF}\,(\mathcal{B}[\![b]\!],\ g,\ f)$$

Beweis.

$$\begin{aligned}\mathrm{IF}\,(\mathcal{B}[\![\texttt{not } b]\!],\ f,\ g)\,\sigma &= \begin{cases} f(\sigma) & \text{falls } \mathcal{B}[\![\texttt{not } b]\!]\,\sigma = \mathbf{tt} \\ g(\sigma) & \text{falls } \mathcal{B}[\![\texttt{not } b]\!]\,\sigma = \mathbf{ff} \end{cases} = \begin{cases} f(\sigma) & \text{falls } \mathcal{B}[\![b]\!]\,\sigma = \mathbf{ff} \\ g(\sigma) & \text{falls } \mathcal{B}[\![b]\!]\,\sigma = \mathbf{tt} \end{cases} \\ &= \mathrm{IF}\,(\mathcal{B}[\![b]\!],\ f,\ g)\,\sigma\end{aligned}$$ □

Einsetzen der Definitionen ergibt:

$$\begin{aligned}\mathcal{D}[\![c\texttt{; while (not } b\texttt{) do } c]\!] &= \mathrm{FIX}\,(\lambda f.\ \mathrm{IF}\,(\mathcal{B}[\![\texttt{not } b]\!],\ f \circ \mathcal{D}[\![c]\!],\ id)) \circ \mathcal{D}[\![c]\!] \\ &= \mathrm{FIX}\,(\lambda f.\ \mathrm{IF}\,(\mathcal{B}[\![b]\!],\ id,\ f \circ \mathcal{D}[\![c]\!])) \circ \mathcal{D}[\![c]\!] \\ &=: \mathrm{FIX}\,(W) \circ \mathcal{D}[\![c]\!] \\ \mathcal{D}[\![\texttt{repeat } c \texttt{ until } b]\!] &= \mathrm{FIX}\,(\lambda f.\ \mathrm{IF}\,(\mathcal{B}[\![b]\!],\ id,\ f) \circ \mathcal{D}[\![c]\!]) \\ &=: \mathrm{FIX}\,(R)\end{aligned}$$

Wir beweisen die Gleichheit der beiden Denotationen, in dem wir beweisen, dass folgende beiden Ungleichungen gelten:

$$\mathrm{FIX}\,(R) \sqsubseteq \mathrm{FIX}\,(W) \circ \mathcal{D}[\![c]\!] \qquad \mathrm{FIX}\,(R) \sqsupseteq \mathrm{FIX}\,(W) \circ \mathcal{D}[\![c]\!]$$

Die Gleichheit folgt dann aus der Antisymmetrie unserer Ordnung.

Beweis. Da $\mathrm{FIX}\,(\cdot)$ uns bereits den *kleinsten* Fixpunkt liefert, genügt es, zu zeigen, dass $\mathrm{FIX}\,(W) \circ \mathcal{D}\,[\![c]\!]$ ein Fixpunkt von R ist:

$$\begin{aligned} R(\mathrm{FIX}\,(W) \circ \mathcal{D}\,[\![c]\!]) &= \mathrm{IF}\,(\mathcal{B}\,[\![b]\!],\ id,\ \mathrm{FIX}\,(W) \circ \mathcal{D}\,[\![c]\!]) \circ \mathcal{D}\,[\![c]\!] \\ &= W(\mathrm{FIX}\,(W)) \circ \mathcal{D}\,[\![c]\!] \\ &= \mathrm{FIX}\,(W) \circ \mathcal{D}\,[\![c]\!] \end{aligned}$$

Außerdem gilt

$$\mathrm{FIX}\,(R) = R(\mathrm{FIX}\,(R)) = \mathrm{IF}\,(\mathcal{B}\,[\![b]\!],\ id,\ \mathrm{FIX}\,(R)) \circ \mathcal{D}\,[\![c]\!]$$

Damit lässt sich die zweite Ungleichung so umschreiben:

$$\mathrm{FIX}\,(W) \circ \mathcal{D}\,[\![c]\!] \sqsubseteq \mathrm{IF}\,(\mathcal{B}\,[\![b]\!],\ id,\ \mathrm{FIX}\,(R)) \circ \mathcal{D}\,[\![c]\!]$$

Aufgrund der Monotonie von $\circ$ (s.o.) genügt es wiederum,

$$\mathrm{FIX}\,(W) \sqsubseteq \mathrm{IF}\,(\mathcal{B}\,[\![b]\!],\ id,\ \mathrm{FIX}\,(R))$$

zu zeigen. Es gilt:

$$\begin{aligned} W(\mathrm{IF}\,(\mathcal{B}\,[\![b]\!],\ id,\ \mathrm{FIX}\,(R))) &= \mathrm{IF}\,(\mathcal{B}\,[\![b]\!],\ id,\ \mathrm{IF}\,(\mathcal{B}\,[\![b]\!],\ id,\ \mathrm{FIX}\,(R)) \circ \mathcal{D}\,[\![c]\!]) \\ &= \mathrm{IF}\,(\mathcal{B}\,[\![b]\!],\ id,\ R(\mathrm{FIX}\,(R))) \\ &= \mathrm{IF}\,(\mathcal{B}\,[\![b]\!],\ id,\ \mathrm{FIX}\,(R)) \end{aligned}$$

Damit ist $\mathrm{IF}\,(\mathcal{B}\,[\![b]\!],\ id,\ \mathrm{FIX}\,(R))$ ein Fixpunkt von W und es gilt auch die zweite Ungleichung. □