

Übungen zur Semantik von Programmiersprachen

Aufgabe 1 (H) (*Funktionsräume*)

Abgabetermin: Montag, 14. 1. 2002

Seien (C, \sqsubseteq_C) und (D, \sqsubseteq_D) cpos. Der Funktionenraum $[C \rightarrow D]$ besteht aus der Menge der stetigen Funktionen von C nach D . Sei weiterhin die Ordnung \sqsubseteq auf diesem Funktionenraum wie folgt definiert:

$$f \sqsubseteq g \iff \forall c \in C. f(c) \sqsubseteq_D g(c)$$

Die kleinste obere Schranke einer ω -Kette $(f_i)_{i \in \mathbb{N}}$ ergibt sich damit punktweise, d.h. zu $f_0 \sqsubseteq f_1 \sqsubseteq f_2 \sqsubseteq f_3 \dots$ ist das Supremum $\bigsqcup_i f_i$ wie folgt definiert:

$$\bigsqcup_i f_i = \lambda c. \bigsqcup_i (f_i(c))$$

Zeigen Sie daß $\bigsqcup_i f_i$ wohldefiniert ist (existiert die rechte Seite?), und daß es sich dabei tatsächlich um das Supremum handelt. Zeigen Sie weiterhin, daß das Supremum wieder ein Element von $[C \rightarrow D]$ ist, und damit $([C \rightarrow D], \sqsubseteq)$ eine cpo. Sie können dabei Aufgabe 5 von Blatt 10 benutzen.

Aufgabe 2 (Ü) (*Stetigkeit von Δ*)

Sei die Funktion $\Delta : 2^\Sigma \rightarrow 2^\Sigma$ zur Terminierungsmenge einer Schleife **while** b **do** s definiert wie in der Vorlesung:

$$\Delta(M) = \{\sigma \mid \mathcal{B}[b]\sigma = \# \Rightarrow (\sigma \in \mathcal{T}[s] \wedge \mathcal{S}[s]\sigma \subseteq M)\}$$

Zeigen Sie mit Hilfe der Aussage

$$\sigma \in \mathcal{T}[s] \Rightarrow |\mathcal{S}[s]\sigma| < \infty$$

die Stetigkeit von Δ .

Aufgabe 3 (Ü) (*Sprünge in WHILE*)

Die Sprache WHILE wird um Markendeklarationen und Sprünge erweitert. Die Menge **Stmt** der Anweisungen ist definiert durch:

$$s ::= \text{skip} \mid X := a \mid s_0 ; s_1 \mid \text{if } b \text{ then } s_0 \text{ else } s_1 \mid \text{while } b \text{ do } s \mid l : s \mid \text{goto } l$$

Dabei stammt l aus einer Menge **Lab** von Marken.

Für **Stmt** definieren wir eine denotationelle *Continuation*-Semantik. Sei die Menge **Cont** = $\Sigma \rightarrow \Sigma$ definiert wie in der Vorlesung.

Jeder Marke l , die in einem Programm definiert wird, wird eine *Continuation* zugeordnet, die das Programmverhalten ab dem definierenden Auftreten " $l : \dots$ " von l beschreibt. Alle solchen Zuordnungen sind in der Menge

$$\mathbf{Lenv} = \mathbf{Lab} \rightarrow \mathbf{Cont}$$

von Markenumgebungen enthalten.

- (a) Geben Sie eine Funktion

$$\mathcal{C} : \mathbf{Stmt} \rightarrow \mathbf{Lenv} \rightarrow \mathbf{Cont} \rightarrow \mathbf{Cont}$$

an mit folgender Bedeutung: Ist s eine Anweisung in einem Programm, e die durch das Programm festgelegte Markenumgebung und k die *Continuation* nach Ausführung von s , so gibt $\mathcal{C}[[s]]e k$ das Verhalten von s im Kontext von e und k wieder.

- (b) Geben Sie eine Funktion

$$\mathcal{J} : \mathbf{Stmt} \rightarrow \mathbf{Lenv} \rightarrow \mathbf{Cont} \rightarrow \mathbf{Lenv}$$

an mit folgender Bedeutung: Ist s eine Anweisung, e eine Markenumgebung und k die *Continuation* nach Ausführung von s , so erweitert $\mathcal{J}[[s]]e k$ die Markenumgebung e um alle in s definierten Marken. Ist eine Marke mehrmals in einem Programm definiert, so zählt das textuell letzte Auftreten. Stützen Sie die Definition von \mathcal{J} auf die Funktion \mathcal{C} aus Teilaufgabe (a).

- (c) Geben Sie eine Funktion

$$\mathcal{P} : \mathbf{Stmt} \rightarrow \mathbf{Cont}$$

an, die eine Anweisung als Programm interpretiert, d.h.: einer Anweisung s wird ihr Zustandsübergangsverhalten unter der Markenumgebung zugeordnet, die in s definiert wird. Stützen Sie die Definition von \mathcal{P} auf die Funktionen \mathcal{C} und \mathcal{J} aus den Teilaufgaben (a) und (b). Beschreiben Sie die durch s definierte Markenumgebung durch den kleinsten Fixpunktoperator.

Aufgabe 4 (P) (*Denotationelle Semantik von Prozeduren*)

Abgabe: Montag, 14. 1. 2002, an kleing@in.tum.de

In der Vorlesung wurde die Sprache WHILE und ihre denotationelle Semantik um Prozeduren erweitert. Diese Erweiterung soll nun analog zur Aufgabe 5, Blatt 9, in Gofer programmiert werden.

In der Datei

```
/usr/proj/semantik/prog/gofer/proc/proc.gs
```

wird Ihnen ein Rahmen für das Gofer-Programm zur Verfügung gestellt. Vervollständigen Sie diesen und testen Sie Ihre Implementierung.