Probeklausur
Semantik von Programmiersprachen
SS98

Diese Probeklausur soll als Beispiel dafür dienen, welche Aufgaben in der realen Klausur vorkommen könnten.
Sie ist in 4 Abschnitte unterteilt:

- Posets und cpos
- IMP bezogene Aufgaben
- Hoare-Kalkül
- Funktionale Sprachen

1 Posets und cpos

Aufgabe I.1 Sei $P = \{a, b\}^*$ die Menge der endlichen Worte über dem Alphabet $\{a, b\}$, geordnet durch $v \sqsubseteq w$ gdw. $\exists x \in P : vx = w$ (Präfixordnung)

1. Beweisen Sie, daß P eine Poset ist.
2. Ist P eine cpo? (Beweis oder Gegenbeispiel)
3. Ist die Konkatenation: $\cdot : P \times P \to P$ monoton?
4. Die Funktion $\mathit{tail} : P \to P$ bildet das leere Wort ϵ auf ϵ ab und entfernt das 1. Zeichen von nichtleeren Worten. Ist *tail* monoton? (Beweis oder Gegenbeispiel)

Aufgabe I.2 Sei $(P, \sqsubseteq)$ eine cpo und $f : P \to P$ stetig.
Zeigen Sie, daß $\{x \in P \mid f(x) \sqsubseteq x\}$ zulässig ist.

Aufgabe I.3 Seien P und Q zwei cpos mit kleinsten Elementen $\perp_P$ und $\perp_Q$ und seien $f : P \to P$ und $g : Q \to Q$ stetig, sowie $h : P \to Q$ stetig mit $h \circ f = g \circ h$ und $h(\perp_P) = \perp_Q$.
Zeigen Sie: $h(\mathit{lfp}\ f) = \mathit{lfp}\ g$.

Aufgabe I.4 Seien P und Q zwei cpos und sei $f : P \to Q$ surjektiv mit der Eigenschaft $x \sqsubseteq_P y \iff f(x) \sqsubseteq_Q f(y)$.

Zeigen Sie: f ist stetig.

2 IMP and friends

Aufgabe II.1 Die Sprache IMP sei um eine neue Anweisung der Form

$$\texttt{do}\ C\ \texttt{while}\ B$$

erweitert, wobei C eine Anweisung und B eine Bedingung ist.
Dies stellt eine Schleifenanweisung dar, welche auf jeden Fall zuerst C ausführt, dann die Bedingung B testet. Ist diese `false`, endet die Schleife, ist sie `true` beginnt ein neuer Durchlauf.

Geben Sie die operationale Ergebnissemantik dieser Anweisung an.

Aufgabe II.2 Die Sprache IMP soll um eine Anweisung `break` erweitert werden. Wenn diese im Rumpf einer Schleife ausgeführt wird, wird die Schleife sofort verlassen. Tritt es nicht im Rumpf einer Schleife auf, wird das Programm verlassen.

Um diese erweiterte Sprache *denotationell* beschreiben zu können, muß der Typ der semantischen Funktionen geändert werden zu $[\![C]\!] : \Sigma \to (\Sigma \times \mathtt{T})_\perp$ für Anweisungen C, wobei $\mathtt{T}$ die Menge der Wahrheitswerte ist. Bei $[\![C]\!]\ \sigma = (\sigma', b)$ ist $b = \mathtt{true}$ gdw. C durch Ausführung eines `break` endete, das innerhalb von C nicht in einer Schleife liegt, so daß eine Schleife, die C als Rumpf enthält, beendet werden muß.

Durch den geändertern Typ müssen alle semantischen Regeln geändert werden. Ihre Aufgabe besteht darin, dies für `skip`, `break`, $C_1 ; C_2$ und `while` B `do` C zu tun.

3 Hoare Kalkül

Aufgabe III.1 Zeigen Sie mit Hilfe der Hoare-Regeln, daß für alle Anweisungen C gilt:

$$\vdash \{\mathtt{true}\} C \{\mathtt{true}\}$$

Aufgabe III.2 Welche der folgenden Hoare-Tripel sind gültig ($\models$)? Begründen Sie Ihre Antwort.

1. $\{P\}\ X\mathtt{:=}1\ \{P[1/X]\}$
2. $\{\mathtt{true}\}\ X := Y\ \{X = Y\}\quad (X, Y \in \mathit{Loc})$
3. $\{\mathtt{true}\}\ \mathtt{while}\ X > 0\ \mathtt{do}\ X\mathtt{:=}X + 1\ \{X < 0\}$

Aufgabe III.3 Zeigen Sie: Für alle Anweisungen C und Prädikate P und Q:

$$\mathit{wp}(C \mid P \wedge Q) \cong \mathit{wp}(C \mid P) \wedge \mathit{wp}(C \mid Q)$$

Gilt $\mathit{wp}(C \mid \mathtt{true}) \cong \mathtt{true}$? (Begründen Sie Ihre Antwort!)

4 Funktionale Sprachen

Aufgabe IV.1 In der funktionalen Sprache 1. Ordnung sei eine Funktion $f : i \to i$ wie folgt definiert:

```
f n = if n = 0 then 0 else if n>0 then f(0-n) else f(n+1)
```

1. Geben Sie an, wie `f(1)` operational zu einem Wert reduziert wird (call-by-value und call-by-name).
2. Betrachten Sie die Fixpunktiteration $\varphi_0 \sqsubseteq \varphi_1 \sqsubseteq \cdots$ für die Programmsemantik $[\![P]\!]$. Bestimmen Sie $\varphi_0(f), \varphi_1(f), \varphi_2(f)$, geben Sie eine allgemeine Formel für $\varphi_k(f)$ an und bestimmen Sie

$[\![P]\!](f)$. Gibt es dabei einen Unterschied zwischen call-by-name und call-by-value?

Aufgabe IV.2 Erweitern Sie die funktionale Sprache 1. Ordnung mit call-by-value Auswertung um Ausdrücke der Form

$$E_1 \ \texttt{orelse} \ E_2 \text{ mit Typisierung } \frac{\Gamma \vdash E_1 : b \quad \Gamma \vdash E_2 : b}{\Gamma \vdash E_1 \ \texttt{orelse} \ E_2 : b}$$

Dabei soll E_1 zuerst ausgewertet werden. Wenn dies `true` ergibt, ist dies der Wert des Ausdruckes. Andernfalls liefert die Auswertung von E_2 den Wert des Ausdruckes. Geben Sie eine geeignete operationelle und denotationelle Semantik für die neuen Ausdrücke an und ergänzen Sie den Beweis von

$$E \to E' \text{ bzgl. } P \Rightarrow [\![E]\!][\![P]\!]\rho = [\![E']\!][\![P]\!]\rho$$

aus der Vorlesung um den Fall der neuen Ausdrücke.