PHILIPPS-UNIVERSITÄT MARBURG
Fachbereich Mathematik und Informatik
Prof. Dr. R. Loogen
Dipl.-Inform. Jost Berthold

D-35032 Marburg
Hans Meerwein Straße
Lahnberge
8. Juli 2003

Übungen zur „Semantik von Programmiersprachen“, SS 2003

Nr. 9, Besprechung der mündlichen Aufgaben: 11. Juli in der Übung, Abgabe der Hausaufgaben: 15. Juli in der Vorlesung

A. Mündliche Aufgaben

9.1 Gegeben sei die folgende CCS-Modellierung eines Schokoriegelautomaten. Der Automat akzeptiert 50 Cent- und 1 EUR-Münzen und bietet entsprechend für 1 EUR große und für 50 Cents kleine Schokoriegel an. Es werden 5 Aktionen des Automaten unterschieden:

$\underline{\text{50 Cents}}$, $\underline{\text{1 EUR}}$:	erhalte Münze im Wert von 50 Cents bzw. 1 Euro
$\underline{\text{Klein}}$, $\underline{\text{Groß}}$:	Auswahltaste „kleiner“ bzw. „großer“ Riegel wird gedrückt
$\overline{\text{Ausgabe}}$:	Schokoriegel ausgeben

Eine einfache Definition des Verkaufsautomaten V ist etwa:

$$V = \underline{\text{50 Cents}}.\underline{\text{Klein}}.\overline{\text{Ausgabe}}.V + \underline{\text{1 EUR}}.\underline{\text{Groß}}.\overline{\text{Ausgabe}}.V$$

(a) Modifizieren Sie die Spezifikation von V so, dass man nach Einwurf von 50 Cents entweder einen kleinen Schokoriegel anfordern oder weitere 50 Cents zum Kauf eines großen Schokoriegels einwerfen kann.

(b) Modifizieren Sie die Spezifikation von V so, dass man für 1 EUR zwei kleine oder einen großen Schokoriegel kaufen kann.

(c) Entwerfen Sie einen benutzerfreundlichen Automaten U, der jederzeit eine der obigen Aktionen durchführen kann, mit den folgenden Ausnahmen:

- U macht weder Gewinn noch Verlust.
- U kann ein Guthaben nur bis höchstens 2 EUR speichern.
- Das Ausgabefach von U kann maximal einen Schokoriegel enthalten.

9.2 Erweitern Sie die in der Vorlesung vorgestellte abstrakte Zustandsmaschine (ASM) CLOCK so, dass eine Stoppuhr modelliert wird. Die Uhr werde über die externen Ereignisse *StartStop* und *Reset* bedient, die als nullstellige beobachtete Boolesche Konstanten modelliert werden. Weiterhin kann eine beobachtete Boolesche Konstante *ClockTick* angenommen werden, die anzeigt, wann das kleinste messbare Zeitintervall (Uhrtick) vorüber ist.

B. Hausaufgaben

Die Abgabe der Hausaufgaben ist in Zweiergruppen erlaubt.

9.3 Eine Prozesssprache habe die folgende Syntax: 5 Punkte

$$p ::= 0 \mid a \mid p;p \mid p+p \mid p \times p \mid P \mid \mathrm{rec}(P=p)$$

Dabei sei $a \in \Sigma$ eine Aktion und $P \in \mathit{Var}$ sei ein Prozessbezeichner, der in rekursiv definierten Prozessen $\mathrm{rec}(P = p)$ verwendet wird. Prozesse führen Sequenzen von Aktionen aus. Zwischen geschlossenen Prozesstermen p und endlichen Sequenzen $s \in \Sigma^*$ wird die Ausführungsrelation $\to$ definiert:

$$0 \to \varepsilon \qquad a \to a \qquad \frac{p \to s \quad q \to t}{p;q \to st} \qquad \frac{p \to s}{p+q \to s} \qquad \frac{q \to s}{p+q \to s}$$

$$\frac{p \to s \quad q \to s}{p \times q \to s} \qquad \frac{p[P \mapsto \mathrm{rec}(P=p)] \to s}{\mathrm{rec}(P=p) \to s}$$

Die Notation $p[P \mapsto q]$ bezeichne den Term, der durch Ersetzen aller freien Vorkommen von P in p durch q entsteht.

(a) Definieren Sie induktiv eine denotationelle Semantik $\mathcal{T}[\![.]\!]$, die Prozesstermen p unter Bezug auf eine Umgebung $\varrho : \mathit{Var} \to \wp(\Sigma^*)$ als Semantik eine Teilmenge $T \subseteq \Sigma^*$ zuordnet, für die gilt: $p \to s \Rightarrow s \in T$. Beachten Sie, dass $(\wp(\Sigma^*), \subseteq)$ eine vollständige Halbordnung bildet.

(b) Geben Sie einen geschlossenen Prozessterm an, der als denotationelle Semantik die Menge $\{a,b\}^*$ hat.

9.4 Entwerfen Sie eine abstrakte Zustandsmaschine (ASM), die für jede der folgenden Operationen auf doppelt verketteten Listen eine Regel enthält. 7 Punkte

Die doppelt verketteten Listen sollen Elemente aus dem Bereich *Value* enthalten. Gegeben sei eine Menge *Node* von Knoten und eine spezielle Konstante *null*, die nicht in *Node* enthalten ist. Die einstelligen dynamischen Funktionen $\mathit{prev}, \mathit{next} : \mathit{Node} \to \mathit{Node}$ und $\mathit{cont} : \mathit{Node} \to \mathit{Value}$ liefern zu einem Knoten den vorherigen Knoten, den nächsten Knoten und den Inhalt des Knotens. Eine doppelt verkettete Liste L werde dargestellt durch die Knoten $\mathit{head}(L)$ und $\mathit{tail}(L)$, d.h. durch den ersten und den letzten Knoten von L.

(a) Implementieren Sie die folgenden Operationen:
- $\mathit{createList}(L)$: erzeuge eine leere doppelt verkettete Liste L
- $\mathit{append}(L, V)$: hänge am Ende von L einen neuen Knoten mit Inhalt V an
- $\mathit{insert}(L, V, i)$: füge vor dem i-ten Element von L einen neuen Knoten mit Inhalt V an
- $\mathit{delete}(L, i)$: entferne den i-ten Knoten von L
- $\mathit{update}(L, i, V)$: aktualisiere den Inhalt des i-ten Knotens von L mit V

(b) Zeigen Sie, dass die folgenden Eigenschaften erfüllt sind:
  i. Falls für einen Knoten N gilt: $\mathit{next}(N) = N'$, so gilt für N': $\mathit{prev}(N') = N$.
  ii. Eine Liste L ist genau dann leer, wenn $\mathit{next}(\mathit{head}(L)) = \mathit{tail}(L)$.
  iii. $\mathit{delete}(\mathit{insert}(L, V, i), i) = L$.