Prof. Dr. R. Loogen, M. Dieterle
Fachbereich Mathematik und Informatik
Hans-Meerwein-Straße
D-35032 Marburg

25. Mai 2010

Philipps Universität Marburg

Übungen zu „Semantik von Programmiersprachen“, Sommer 2010

Nr. 7, Abgabe der Aufgaben: 08. Juni 2010 vor der Vorlesung

Hinweis: Dieses Übungsblatt hat wegen Fronleichnam (Donnerstag, 3.6.) eine Bearbeitungszeit von 2 Wochen.

Aufgaben

7.1 Sei $(D, \sqsubseteq)$ eine kettenvollständige Halbordnung mit kleinstem Element $\perp_D := \bigsqcup \emptyset$. Die Menge der *Listen über* D, $L(D)$, ist induktiv definiert durch: **8 Punkte**

- $[\,] \in L(D)$ (die *leere Liste*) und
- mit $d \in D \cup L(D)$ und $l \in L(D)$ ist $(d : l) \in L(D)$.

Sei $\perp$ ein neues Symbol. Wir erweitern mit der folgenden Definition $\sqsubseteq$ induktiv auf $L(D)_\perp := L(D) \cup \{\perp\}$:

- für alle $l \in L(D)_\perp$ sei $\perp \sqsubseteq l$ und $l \sqsubseteq l$, sowie
- $(d : l) \sqsubseteq (d' : l')$ genau dann, wenn $d \sqsubseteq d'$ und $l \sqsubseteq l'$.

(a) Zeigen Sie, dass $(L(D)_\perp, \sqsubseteq)$ eine kettenvollständige Halbordnung mit kleinstem Element $\perp$ ist. / 3

(b) Weisen Sie nach: Ersetzt man in der induktiven Erweiterung der Relation $\sqsubseteq$ das Symbol $\perp$ durch $[\,]$ oder $(\perp_D : [\,])$, so sind die entsprechenden Halbordnungen nicht kettenvollständig. / 2

(c) Zur Manipulation von Listen verwenden wir die Funktionen / 3
tail : $L(D)_\perp \to L(D)_\perp$ und *cons* : $L(D)_\perp \times L(D)_\perp \to L(D)_\perp$ mit

$$tail(l) = \begin{cases} l' & \text{falls } l = (d : l') \\ \perp & \text{sonst.} \end{cases} \qquad cons(d, l) = \begin{cases} (d : l) & \text{falls } d, l \in L(D) \\ \perp & \text{sonst.} \end{cases}$$

für alle $d, l \in L(D)_\perp$.
Beweisen Sie, dass *tail* und *cons* in $(L(D)_\perp, \sqsubseteq)$ stetig sind.

7.2 Der Bereich **Sign** der Vorlesung enthält als verwertbare Informationen nur POS, NEG und ZERO. Man kann den Bereich auch um die jeweiligen Verneinungen NON-NEG, NON-POS und NON-ZERO erweitern. **2 Punkte**

(a) Wie sollten die neuen Werte in die Halbordnung auf **Sign** eingefügt werden? Skizzieren Sie die Halbordnung in einem Diagramm und begründen Sie Ihre Lösung.

(b) Erweitern Sie die abstrakten Operationen $-^s$ und $*^s$ für die Vorzeicheninterpretation $\mathcal{A}^s$ arithmetischer Ausdrücke in geeigneter Weise.

Bitte wenden!

7.3 **Statische Analyse: "Constant Propagation"** 8 Punkte

In der folgenden Aufgabe wird analog zur Vorzeichenanalyse der Vorlesung eine statische Analyse entwickelt, die feststellt, ob der Wert eines arithmetischen Ausdrucks in einem Programm konstant ist.

Als Grundbereich dient dabei die Menge **Const** $= N \cup \{\text{ANY}, \text{NONE}\}$ mit einer Halbordnung, für die gilt:

$$\text{NONE} \sqsubseteq p \sqsubseteq \text{ANY}\ \forall p \in \textbf{Const}$$

Unterschiedliche Werte aus **N** seien mittels $\sqsubseteq$ unvergleichbar.
Als Zustandsmenge wird $\Pi^c = \{\pi : \textbf{Var} \rightarrow \textbf{Const}\}$ benutzt.

Ergibt die Analyse einen Wert $n \in N$, so bedeute dies, dass der interpretierte Ausdruck stets exakt den Wert n liefert. ANY und NONE werden analog zur Vorzeichenanalyse als "nicht konstant" und "undefiniert" interpretiert.

(a) Verdeutlichen Sie die Anordnung der Elemente von **Const** in einem Diagramm.

(b) Definieren Sie die nötigen Abstraktionsfunktionen

$$\begin{aligned}
\mathcal{A}^c &: \textbf{AExp} \rightarrow \Pi^c \rightarrow \textbf{Const} \\
\mathcal{B}^c &: \textbf{BExp} \rightarrow \Pi^c \rightarrow \textbf{T}^s \\
\mathcal{C}^c &: \textbf{Cmd} \rightarrow \Pi^c \rightarrow \Pi^c
\end{aligned}$$

(c) Wenden Sie Ihre Analyse auf die folgende Anweisung an:

$$X := 2; Y := 10*X;\ \textbf{if}\ Z < 0\ \textbf{then}\ (X := (1-Y)*X; Z := X+2*Y)\ \textbf{else}\ Z := X$$