PHILIPPS-UNIVERSITÄT MARBURG
Fachbereich Mathematik und Informatik
Prof. Dr. R. Loogen
Dipl.Inf. J.Berthold

D-35032 Marburg
Hans Meerwein Straße

1. Dezember 2004

Übungen zu „Semantik von Programmiersprachen“, WS 2004/05

Nr. 6, Besprechung mündlicher Aufgaben: 6. Dezember 2004 in der Übung, Abgabe der Hausaufgaben: 8. Dezember 2004 vor der Vorlesung

---

Mündliche Aufgaben

6.1 (a) Zeigen Sie mit Hilfe der statischen Vorzeichenanalyse aus der Vorlesung, dass die Anweisung

$$Z := 0;\ \textbf{while } Y \leq X \textbf{ do } (Z := Z + 1; X := X - Y)$$

nie in einen Zustand mit negativem Z führt.

(b) Welche Aussagen kann die Analyse über das Vorzeichen von X machen, wenn vor der Anweisung X und Y positive Werte hatten?

6.2 Dijkstras „guarded commands“ besitzen im wesentlichen die Gestalt

$$\begin{array}{ll} \textbf{do} & b_1 \to c_1 \\ & b_2 \to c_2 \\ \textbf{od} & \end{array}$$

wobei $b_1, b_2 \in \textbf{BExp}$ und $c_1, c_2 \in \textbf{Com}$. Sie verallgemeinern die **while**-Schleife:

> Solange ein Test b_i wahr ist, wird die entsprechende Anweisung c_i ausgeführt. Die Erfülltheit beider Tests führt zu einer nichtdeterministischen Auswahl der Anweisung. Die Berechnung terminiert, wenn kein Test mehr wahr ist.

(a) Welche Funktion auf den natürlichen Zahlen wird laut dieser informellen Semantikbeschreibung durch die folgende Anweisung berechnet?

$$c = \begin{array}{ll} \textbf{do} & X > Y \to X := X - Y \\ & Y > X \to Y := Y - X \\ \textbf{od} & \end{array}$$

(b) Für $i \in \{1, 2\}$ seien $b_1, b_2 \in \textbf{BExp}$ sich gegenseitig ausschließende Tests und $c_1, c_2 \in \textbf{Com}$.
Wie kann man die Semantik von

$$\begin{array}{ll} \textbf{do} & b_1 \to c_1 \\ & b_2 \to c_2 \\ \textbf{od} & \end{array}$$

als kleinsten Fixpunkt einer stetigen Abbildung $\Phi : (\Sigma \dashrightarrow \Sigma) \longrightarrow (\Sigma \dashrightarrow \Sigma)$ gewinnen?

(c) Berechnen Sie die Semantik der Anweisung c aus Teil (a) unter Verwendung von Φ nach dem Fixpunktsatz von Knaster und Tarski.

Bitte wenden!

Schriftliche Aufgaben

6.3 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. $\boxed{\text{4 Punkte}}$

(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.

(c) Geben Sie die Abbildung $\mathcal{B}^s[\![X * X > 0]\!]$ für den erweiterten Bereich an.

6.4 **Statische Analyse: "Constant Propagation"** $\boxed{\text{8 Punkte}}$

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:

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