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

D-35032 Marburg
Hans Meerwein Straße

12.Januar 2005

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

**Nr. 10, Besprechung mündlicher Aufgaben: 17.Januar 2005 in der Übung,
Abgabe der Hausaufgaben: 19.Januar 2005 vor der Vorlesung**

Mündliche Aufgaben

10.1 **Rekursive Fakultät**

Gegeben sei die nebenstehende Deklaration einer Prozedur zur Fakultätsberechnung. In der Startumgebung $\rho \in \mathbf{Env}$ sei $\rho(X) = v_x$ und $\rho(Y) = v_Y$.
Bestimmen Sie die durch die Deklarationssemantik bestimmte Umgebung.

```
proc p is
 begin  var Z;
        Z := X;
        if X = 1 then skip
        else (  X := X - 1;
                call p;
                Y := Y * Z  )
 end
```

10.2 **Static Scope / Dynamic Scope**

Betrachten Sie im folgenden Programm mit Blöcken und Prozeduren den Effekt unterschiedlicher Umgebungskonzepte (Sichtbarkeit von Deklarationen):

```
begin var X;
      proc p is X := X * X;
      proc q is call p;
      X := 0;
      begin var X;
            proc p is X := X - 1;
            X := 5;
            call q;
            Y := X
      end
end
```

Bestimmen Sie den Wert von Y nach Ablauf des Programms

- im Falle statischer Sichtbarkeit für Variablen und Prozeduren,
- im Falle dynamischer Sichtbarkeit für Variablen und Prozeduren,
- falls für Variablen dynamische Sichtbarkeit, aber statische Sichtbarkeit für Prozeduren gilt.

Bitte wenden!

Schriftliche Aufgaben

10.3 Zeigen Sie, dass eine **while**-Schleife durch eine (endrekursive) Prozedur simuliert werden kann, indem Sie nachweisen, dass die Anweisung **while** b **do** c (mit $b \in$ **BExp** und $c \in$ **Cmd**) zu der folgenden Anweisung äquivalent ist. **6 Punkte**

$$
\begin{array}{l}
\textbf{begin} \\
\quad \textbf{proc}\ p\ \ \textbf{is}\ \ (\textbf{if}\ b\ \textbf{then}\ (c;\ \ \textbf{call}\ p)\ \textbf{else skip}); \\
\quad \textbf{call}\ p \\
\textbf{end}
\end{array}
$$

10.4 Die Syntax von **IMP** werde so modifiziert, dass jede Prozedur zwei Wertparameter erhält: **6 Punkte**

$$
\begin{array}{rcl}
\mathbf{PDec} \ni d & ::= & \textbf{proc}\ p(X_1, X_2)\ \textbf{is}\ c; d \mid \varepsilon \\
\mathbf{Com} \ni c & ::= & \ldots \mid \textbf{call}\ p(a_1, a_2)
\end{array}
$$

Die Bedeutung einer Prozedur hängt nicht mehr nur vom Speicherzustand, sondern auch von den Werten der Parameter ab. Die Definition der Umgebungen wird deshalb wie folgt verallgemeinert:

$$
\begin{array}{rl}
\mathbf{Env} := \{\rho \mid & \rho : (\mathbf{Var} \cup \mathbf{PVar}) \to \mathbf{Loc} \cup (\mathbf{N} \times \mathbf{N}) \to (\Sigma \dashrightarrow \Sigma) \\
& \text{mit } \rho(\mathbf{Var}) \subseteq \mathbf{Loc} \text{ und } \rho(\mathbf{PVar}) \subseteq (\mathbf{N} \times \mathbf{N}) \to (\Sigma \dashrightarrow \Sigma)\}.
\end{array}
$$

(a) Passen Sie das semantische Funktional $\mathcal{C}[\![.]\!]$ für Anweisungen an.

(b) Geben Sie eine entsprechend modifizierte Version des Funktionals $\mathcal{D}[\![.]\!]$ für die Deklaration nichtrekursiver sowie rekursiver Prozeduren an.