

Übungen zu „Semantik von Programmiersprachen“, Sommer 2010

Nr. 10, Abgabe der Aufgaben: 29. Juni 2010 vor der Vorlesung

Aufgaben

10.1 Parametrisierte Prozeduren

6 Punkte

Die Syntax von WHILE werde so modifiziert, dass jede Prozedur n Wertparameter erhält:

$$\begin{aligned} \mathbf{PDec} \ni d &::= \mathbf{proc} \ p(X_1, \dots, X_n) \ \mathbf{is} \ c; d \mid \varepsilon \\ \mathbf{Cmd} \ni c &::= \dots \mid \mathbf{call} \ p(a_1, \dots, a_n) \end{aligned}$$

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

$$\begin{aligned} \mathbf{Env} &::= \{ \rho \mid \rho : (\mathbf{Var} \cup \mathbf{PVar}) \rightarrow \mathbf{Loc} \cup (\mathbf{Num}^n \rightarrow (\Sigma \rightarrow \Sigma)) \\ &\quad \text{mit } \rho(\mathbf{Var}) \subseteq \mathbf{Loc} \text{ und } \rho(\mathbf{PVar}) \subseteq (\mathbf{Num}^n \rightarrow (\Sigma \rightarrow \Sigma)) \}. \end{aligned}$$

- Passen Sie das semantische Funktional $\mathcal{C}[\cdot]$ für Anweisungen an.
- Geben Sie eine entsprechend modifizierte Version des Funktionals $\mathcal{D}[\cdot]$ für die Deklaration nichtrekursiver sowie rekursiver Prozeduren an.

10.2 Static Scope / Dynamic Scope

4 Punkte

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 anhand der Gesamtschrittsemantik den Wert von Y nach Ablauf des Programms

- im Falle statischer Sichtbarkeit für Variablen und Prozeduren,
- im Falle dynamischer Sichtbarkeit für Variablen und Prozeduren.