Technische Universität München
Institut für Informatik
Prof. Tobias Nipkow, Ph.D.
Gerwin Klein

WS 2001/2002
Übungsblatt 5
12. 11. 2001

Übungen zur Semantik von Programmiersprachen

Aufgabe 1 (H) (*Prozeduren mit Parameter*) Abgabetermin: Montag, 19. 11. 2001

Die Sprache WHILE soll um Prozeduren mit einem Parameter erweitert werden. Die Syntax für Prozeduren ändert sich damit auf

$$\textbf{proc}\ p(x) = s \quad \text{und} \quad \textbf{call}\ p(x)$$

Hierbei sei x wie immer vom Typ Integer. Es sollen im folgenden statische Sichtbarkeitsregeln verwendet werden.

(a) Modifizieren Sie die Big-Step-Semantik so, dass der Parameter mit *call-by-value* übergeben wird. Im Programmfragment $x := 1;\ \textbf{call}\ p(x)$ soll x also auch nach dem Aufruf den Wert 1 behalten.

(b) Modifizieren Sie die Big-Step-Semanitk so, dass der Parameter mit *call-by-reference* übergeben wird. Im Programmfragment $x := 1;\ \textbf{call}\ p(x)$ sollen Änderungen des Wertes von x in p auch nach dem Aufruf von p sichtbar bleiben.

(c) Geben Sie für beide Formen jeweils ein Beispiel-Programm mit Ableitungsbaum an, das die Anwendung der neuen Regeln zeigt. Ihre Beispiel-Programme sollen den Unterschied zwischen *call-by-value* und *call-by-reference* deutlich machen.

Aufgabe 2 (Ü) (*Prozeduren als formale Parameter*)

In Aufgabe 1 war der Parameter bei Prozeduraufrufen vom Typ Integer. In dieser Aufgabe sollen auch Prozeduren als formale Parameter zugelassen werden.

Wir erweitern die Syntax von WHILE dazu weiter um

$$\textbf{proc}\ p(q) = s \quad \text{und den Aufruf} \quad \textbf{call}\ p(q)$$

(a) Erweitern Sie die Big-Step-Semantik um Prozeduren als formale Parameter.

(b) Geben Sie ein Beispiel-Programm an, das die Anwendung der neuen Regeln demonstriert.

Aufgabe 3 (Ü) *(Compiler für arithmetische Ausdrücke)*

Es soll die Korrektheit des Compilers für arithmetische Ausdrücke der Sprache WHILE aus Aufgabe 4 gezeigt werden.

(a) Formulieren Sie dazu die Korrektheitsaussage formal.

(b) Beweisen Sie Ihre Aussage.

Aufgabe 4 **(P)** (*Compiler für arithmetische Ausdrücke*)
Abgabe: Montag, 19. 11. 2001, 12 Uhr an kleing@in.tum.de

(a) Für die arithmetischen Ausdrücke der Sprache WHILE sollen ein Codegenerator und eine Maschine, die den übersetzten Code abarbeitet, in der funktionalen Programmiersprache Gofer implementiert werden.

Der Codegenerator nimmt einen Ausdruck und übersetzt ihn in eine Liste von Instruktionen. Dabei stehen folgende Instruktionen zur Verfügung:

i) `Const c`: Lege die Konstante c auf den Keller.

ii) `Load a`: Lade den Inhalt der Adresse a auf den Keller.

iii) `Apply f`: Wende die binäre Funktion f auf die beiden obersten Kellerelemente an und ersetze sie durch das Ergebnis.

Die Maschine, auf der die übersetzten Ausdrücke abgearbeitet werden sollen, nimmt eine List von Instruktionen, einen (anfangs leeren) Keller und die Variablenbelegung als Argumente, und liefert einen Resultatkeller als Ergebnis. Am Ende der Abarbeitung soll auf dem Keller nur noch das Ergebnis der Berechnung auf dem Keller liegen.

In der Datei

`/usr/proj/semantik/prog/gofer/code-gen/codegen.gs`

wird Ihnen ein Rahmen für das Gofer-Programm zur Verfügung gestellt.

(b) Überprüfen Sie anhand der Beispiele in

`/usr/proj/semantik/prog/gofer/code-gen/dialog.txt`

ob die Maschine das korrekte Ergebnis liefert. Geben Sie Protokolle der Beispiel-Läufe ab.