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

D-35032 Marburg
Hans Meerwein Straße

26.Januar 2006

11. Übung zu „Grundlagen des Compilerbaus“, WS 2005/06

Abgabe schriftlicher Aufgaben: **Do, 2.Februar 2006 (vor der Vorlesung)**
Besprechung mündlicher Aufg.: **ab 30.Januar 2006 in der Übung**

Mündliche Aufgaben

11.1 Aktivierungsblöcke in PSP

(a) Gegeben sei der folgende *MP*-Zustand

$$(5, 4, 8 : 3 : 10 : 7 : 4 : 3 : 27 : 9 : 8 : 3 : 54 : 1 : ...) \in ZR$$

Ermitteln Sie den Zustand nach Ausführung der nächsten vier Befehle des folgenden *MP*-Codes:

```
     ⋮
 5: LOAD(1,1)
 6: ADD
 7: RET
 8: LOAD(0,2)
 9: STORE(2,1)
10: CALL (84,1,3);
11: RET
     ⋮
```

(b) Welche der folgenden Prozedurkeller können das Ergebnis der Ausführung eines übersetzten PSP-Programms sein?

i. `15:4:9:-3:5:5:6:3:2:1:5:4:12:3:4:4:3:25:0:0:0:0:12`
ii. `15:4:9:-3:5:10:4:3:2:1:5:4:12:3:4:4:3:25:0:0:0:0:12`
iii. `15:4:9:-3:5:14:4:3:2:1:5:4:12:3:4:4:3:25:0:0:0:0:12`

11.2 PSP-Erweiterung: Funktionsaufrufe

Häufig bieten prozedurale Sprachen neben (parameterlosen) Prozeduren auch die Möglichkeit, Funktionsprozeduren zu verwenden, deren Resultat als Ausdruck behandelt wird. Eine entsprechende Erweiterung der Sprache PSP könnte wie folgt aussehen:

$$\begin{aligned} \Delta &::= \Delta_C \Delta_V \Delta_p \Delta_F \\ \Delta_F &::= \varepsilon \mid \textbf{fun } I_1; B_1; \ldots I_n; B_n; \\ E &::= \ldots \mid I() \\ \Gamma &::= \ldots \mid \textbf{return } E \end{aligned}$$

Funktionsprozeduren werden wie parameterlose Prozeduren, aber mit **fun** statt **proc** deklariert. Ihr Aufruf ist innerhalb von Ausdrücken erlaubt. Die neue Anweisung **return** beendet den Aufruf und liefert gleichzeitig das Ergebnis der Funktion zurück.

Geben Sie entsprechende Erweiterungen der Übersetzungsfunktionen up, dt, ct und et an.

Bitte wenden!

Schriftliche Aufgaben

11.3 MP-Implementierung

6 Punkte

(a) Implementieren Sie die Maschine MP in Haskell, indem Sie die MA-Implementierung aus *Machine.hs* in geeigneter Weise abändern.

(b) Schreiben Sie eine Funktion zur Konvertierung von MA-Code in MP-Code.

(c) Testen Sie Ihre Implementierung mit MA- und (von Hand erstellten) MP-Programmen.

11.4 PSP-Übersetzung

6 Punkte

Gegeben sei das nebenstehende Programm P in der Sprache PSP:

(a) Berechnen Sie **trans**(P).

(b) Führen Sie das übersetzte Programm mit Belegung $X = 7$ und $Y = 3$ aus.

```
in/out X,Y;
var Z;
proc F;
  if (X > Z) then
      X := X - Z;
      Z := Z * 2;
      F();
    else
      Y := Y - 1;
  Y := Y + 1;
Y := 0;
Z := 1;
F();
```