

10. Übung zu „Grundlagen der funktionalen Programmierung“,
Abgabe: 11.Januar 2007 vor der Vorlesung WS06/07

Aufgaben

10.1 Implementierung der Stackmaschine für rekursive Programme

12 Punkte

Auf der Vorlesungsseite finden Sie eine (unvollständige) Implementierung der Stackmaschine aus der Vorlesung. Sprungziele werden hier stets (auch im Maschinenbefehl) als `Int`-Listen codiert, weil sonst ein eigener Datentyp für jede Adressform nötig wäre. Die Maschine kennt außerdem einen zusätzlichen Befehl `Noop` (no operation), welcher nur den Befehlszähler ändert:

$$\mathcal{C}[\![Noop]\!]_{\mathcal{A}}(m, d, h) = (m + 1, d, h)$$

Diese Befehle werden jeweils eingefügt, wenn ein Ergebnis der Übersetzung Sprungziel sein könnte.

- (a) Ergänzen Sie das Programm um eine Funktion, welche die Sprungadressen in lineare Form übersetzt. Entfernen Sie gleichzeitig alle `Noop` aus dem Programm. / 4
- (b) Finden und beheben Sie (mindestens :-)) 2 Fehler in der Implementierung. / 2
- (c) Erweitern Sie die Stackmaschine gemäß der Implementierung rekursiver Programme mit nicht-strikter Semantik und schreiben Sie eine entsprechende Übersetzungsfunktion. / 6

Die Maschine sollte möglichst in der Lage sein, sowohl ein strikt übersetztes, als auch ein nicht-strikt übersetztes Programm abzuarbeiten.

Zusatzaufgabe:

10.2 Übersetzung in Stack-Code

8 Punkte

Gegeben sei das rekursive Programm $\tilde{R}_{\mathbb{Z}_{\perp}}$ mit

$$\begin{aligned} \tilde{R} : F_1(x, y) &= \text{if } y = 0 \text{ then } F_2(x - 1, y) \text{ else } F_2(F_1(x + 1, y), y - 1) \\ F_2(x, y) &= \text{if } x = 0 \text{ then } 5 \text{ else } F_2(F_1(x, y - 1), y - 1) \end{aligned}$$

und der üblichen Interpretation der Operationssymbole.

- (a) Berechnen Sie die nicht-strikte und die strikte Fixpunktsemantik von $\tilde{R}_{\mathbb{Z}_{\perp}}$.
- (b) Übersetzen Sie das Programm (in beiden Varianten) in Stack-Code.



**Frohe Weihnachten
und alles Gute in 2007**

