

2. Übung „Übersetzerbau“

Abgabe am 2. Mai vor der Vorlesung

Aufgabe 5

8 Punkte

In der Vorlesung wurde die Programmiersprache Simple und Haskell-Datenstrukturen zur abstrakten Repräsentation von Simple-Programmen vorgestellt.

Definieren Sie einen Interpreter für Simple-Programme, der alle Ausgaben des Simple-Programms in einer Liste von Zahlen aufammelt, also eine Funktion

```
simpleInt :: Stm -> [Int]
```

Wir definieren hierzu eine Variablenbelegung als Liste von Tupeln von Bezeichnern und Werten:

```
type Bindings = [(Id,Int)]
```

Definieren Sie zunächst eine Funktion

```
getBind :: Id -> Bindings -> Int
```

welche die Bindung einer Variablen bestimmt. Definieren Sie außerdem eine Funktion

```
updateBind :: Id -> Int -> Bindings -> Bindings
```

zum Aktualisieren von Variablenbindungen.

Unter Verwendung von Variablenbelegungen können Sie dann eine Hilfsfunktion

```
stmCalc :: Bindings -> Stm -> (Bindings,[Int])
```

definieren, welche die Variablenbelegungen berechnet und gleichzeitig die Ausgabe aufbaut. Der Zugriff auf eine nicht belegte Variable soll einen Laufzeitfehler ergeben. Überlegen Sie auch, wie Sie eine entsprechende Funktion für Ausdrücke definieren können. Beachten Sie hierbei, dass innerhalb von Ausdrücken auch wieder Anweisungen und damit Ausgaben möglich sind.

Aufgabe 6

3 Punkte

Sei PEVAL ein partieller Auswerter. Zeigen Sie, dass

$$cc := I \text{ PEVAL}(\text{PEVAL}, \text{PEVAL})$$

ein Compiler-Compiler ist, d.h. einen Interpreter in einen Compiler übersetzt.

Aufgabe 7

4 Punkte

Ein Crosscompiler wird verwendet um einen Compiler für eine Sprache Q von einer Rechnerarchitektur M_1 auf eine andere Rechnerarchitektur M_2 zu portieren, auf der es noch keinen Compiler für Q gibt. In der Vorlesung wurde die Darstellung von Compilern als T-Diagramme eingeführt. Wir wollen mit dieser Darstellung den Vorgang des Crosscompilierens beschreiben. Gehen Sie davon aus, dass der Compiler für M_1 in ein Frontend $c_f : Q \rightarrow Z$ und ein Backend $c_b : Z \rightarrow M_1$ aufgeteilt ist, welche beide in Q geschrieben sind.

Für die Crosscompilierung muss nur das Backend neu implementiert werden. Geben Sie das T-Diagramm für das neue Backend c'_b an und veranschaulichen Sie den Vorgang des Crosscompilierens mit Hilfe der T-Diagramme.

Aufgabe 8

5 Punkte

In der Übung wurden die Funktion `foldr` und `foldl` definiert. Veranschaulichen Sie sich noch einmal das Verhalten und den Typ beider Funktionen.

- a) Definieren Sie die Fakultätsfunktion unter Verwendung der Funktion `foldr` `foldl` (Einzeiler!)
- b) Welche Funktionen werden durch folgende Ausdrücke definiert (umgangssprachlich)?
 - `foldr (:) []`
 - `foldl (:) []`
 - `foldl (-) 1`
 - `foldr (-) 1`
- c) Definieren Sie die Funktion `length` unter Verwendung der `fold`-Funktionen. Begründen Sie, weshalb Sie `foldr` bzw. `foldl` verwenden. Ist es auch möglich die duale `fold`-Funktion zu verwenden?
- d) Definieren Sie die Funktion `reverse` unter Verwendung der `fold`-Funktionen. Ist eine der beiden Lösungen effizienter?