

## Übungen zu „Grundlagen des Compilerbau“, Winter 2011/12

Nr. 10, Abgabe der Aufgaben: 24. Januar 2012 vor der Vorlesung

---

### Aufgaben

#### 10.1 PSA-Übersetzung

9 Punkte
----------

Die auf Blatt 6 beschriebene While-Sprache ist der Sprache PSA sehr ähnlich. Die Ausgabe eines While-Parsers kann als abstrakter Syntaxbaum eines PSA-Programms betrachtet werden.

- (a) Definieren Sie passend zu den Übersetzungsfunktionen der Vorlesung eine Funktion zur Bestimmung der Codelänge, gehen sie aber von der While-Sprache anstelle der Sprache PSA aus:

$$cl : Expr \cup Stmt \rightarrow \mathbb{N}$$

Die Definition kann direkt in Haskell erfolgen bzw. mit Papier und Stift (wer nicht programmieren will).

Erläutern Sie, wo und wie diese Funktion für die Übersetzung von While-Programmen in MA-Code zu verwenden ist.

- (b) Schreiben Sie einen Codegenerator in Haskell, welcher den abstrakten Syntaxbaum eines While Programms (Ausgabe des Parsers) in MA-Code überführt. Auf der VL-Seite finden Sie einen While-Scanner und Parser, der die Eingabe für die Zwischencodegenerierung liefert.

Auf der Webseite zur Vorlesung finden Sie eine Implementierung der MA-Maschine, mit der Sie den generierten MA-Code ausführen können. Testen Sie, ob Ihr Codegenerator korrekten Code erzeugt.

#### 10.2 Jumping Code

3 Punkte
----------

Für zusammengesetzte Boolesche Ausdrücke kann anstelle der Maschinenoperationen (AND, OR) sog. *Jumping Code* generiert werden, so dass unnötige Teile gar nicht erst ausgewertet werden (siehe Skript S. 92).

Gegeben sei die Symboltabelle  $st = st_{\emptyset}[I/(\text{var}, 1), J/(\text{var}, 2)]$  sowie die PSA-Anweisung

```
while (I > 0 and (J > I or J < 0)) do
  I := J - I;
  J := 1 - J
```

Übersetzen Sie diese PSA-Anweisung mit und ohne Verwendung von Jumping Code. Vergleichen Sie die Codelänge und die Zahl der ausgeführten Befehle für die Schleife bei verschiedenen Variablenbelegungen.