Prof. Dr. R. Loogen, M. Dieterle
Fachbereich Mathematik und Informatik
Hans-Meerwein-Straße
D-35032 Marburg

13. Januar 2010

Philipps Universität Marburg

Übungen zu „Grundlagen des Compilerbau“, Winter 2009/10

Nr. 9, Abgabe der Aufgaben: 27. Januar 2010 vor der Vorlesung

---

Hinweise:

- Dieses Blatt hat eine Bearbeitungszeit von 2 Wochen.
- Am Freitag, dem 22. Januar fällt das Tutorium aus.

Aufgaben

9.1 **Zirkularitätstest** — 4 Punkte

Geben Sie für die folgende Grammatik $syn(x)$ und $inh(x)$ für alle Symbole x an und untersuchen Sie mit Hilfe des in der Vorlesung vorgestellten Verfahrens, ob die Grammatik zirkulär ist:

$$
\begin{array}{lcll}
G: S & \rightarrow & AaB & s.0 = s_2.3,\ i_1.1 = s_2.3 \cdot s.2,\ i_2.1 = s_1.3 \cdot s_2.1,\ i_1.3 = 1,\ i_2.3 = s_1.1 \\
A & \rightarrow & Ba & s_1.0 = 2 \cdot s_2.1,\ s_2.0 = s_1.1,\ i_1.1 = i_1.0,\ i_2.1 = i_2.0 + s.2 \\
A & \rightarrow & a & s_1.0 = 1,\ s_2.0 = s.1 \\
B & \rightarrow & b & s_1.0 = i_1.0,\ s_2.0 = s.1 \\
B & \rightarrow & c & s_1.0 = s.1,\ s_2.0 = i_2.0
\end{array}
$$

9.2 **PSA-Übersetzung** — 10 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.

Bitte wenden!

9.3 **Jumping Code** 4 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.