Albert-Ludwigs-Universität Freiburg
Einführung in die Informatik
Wintersemester 2012/13

Prof. Dr. Wolfram Burgard
Dipl. Ing. Jörg Röwekämper
Dipl. Inf. Henrik Kretzschmar
Dipl. Inf. Markus Kuderer

Übungsblatt 11

Abgabe bis Freitag, 25.01.2012, 12:00 Uhr

Hinweis: Lösungen immer per Email an den zuständigen Tutor schicken. Die Emailadressen sind auf der Homepage zur Vorlesung gelistet. Verwenden Sie für alle Programmieraufgaben die Struktur des Beispielprojekts [1]. Kompilieren Sie Ihre Projekte mit Hilfe des `ant`-Buildsystems.

Aufgabe 11.1

In der Vorlesung wurde ein Applet vorgestellt, das eine Turing-Maschine simuliert[2]. Laden Sie das Applet herunter und entpacken Sie die Dateien. Mit Hilfe des Befehls

```
appletviewer -J-Djava.security.policy=tmpolicy TuringMachineHtml.html
```

starten Sie das Applet.

Nun können Sie eine Turing-Transitionstabelle `*.tm` laden (siehe `README`), in der pro Zeile eine Transition angegeben ist (Sie finden ein Beispielprogramm für das Addieren von Binärzahlen in dem entpackten Ordner).

Erstellen Sie ein Turing-Programm, das die Eingabe $\in \{1, 0\}$ kopiert, z.B. bei der Eingabe $*101*$ wird $*101 * 101*$ ausgeben.

Schreiben Sie die Transitionstabelle Ihrer Turing-Maschine in die Datei `copy.tm` und testen Sie das Programm mit Hilfe des Applets.

[1] http://ais/teaching/ws12/info/material/MyProject.zip
[2] http://ais.informatik.uni-freiburg.de/turing-applet/turing-file-new.tar

Aufgabe 11.2

Die Syntax von WHILE-Programmen mit den Variablen a, b, und c kann durch folgende Grammatik beschrieben werden:

$$G = (\{\langle\text{Anweisung}\rangle, \langle\text{Konstante}\rangle, \langle\text{Variable}\rangle\}, T, P, \langle\text{Anweisung}\rangle\ ).$$

Die Terminalsymbole T bestehen aus

$$T = \{\texttt{while}, \texttt{do}, \texttt{end}\ , +, -, :=, \neq, ;, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, a, b, c\}.$$

Die Produktionen P sind wie folgt gegeben:

$$\begin{aligned}
P = \{ & \\
& \langle\text{Anweisung}\rangle \rightarrow \langle\text{Anweisung}\rangle; \langle\text{Anweisung}\rangle, && (1)\\
& \langle\text{Anweisung}\rangle \rightarrow \langle\text{Variable}\rangle := \langle\text{Konstante}\rangle, && (2)\\
& \langle\text{Anweisung}\rangle \rightarrow \langle\text{Variable}\rangle := \langle\ \text{Variable}\rangle + \langle\text{Konstante}\rangle, && (3)\\
& \langle\text{Anweisung}\rangle \rightarrow \langle\text{Variable}\rangle := \langle\ \text{Variable}\rangle - \langle\text{Konstante}\rangle, && (4)\\
& \langle\text{Anweisung}\rangle \rightarrow \texttt{while}\ \langle\text{Variable}\rangle \neq 0\ \texttt{do}\ \langle\text{Anweisung}\rangle\ \ \texttt{end}, && (5)\\
& \langle\text{Variable}\rangle \rightarrow a|b|c, && (6)\\
& \langle\text{Konstante}\rangle \rightarrow \langle\text{Konstante}\rangle\langle\text{Konstante}\rangle, && (7)\\
& \langle\text{Konstante}\rangle \rightarrow 0|1|2|3|4|5|6|7|8|9 && (8)\\
\} &
\end{aligned}$$

Überprüfen Sie, ob folgende Ausdrücke Teil der Sprache $L(G)$ sind. Wenn ja, geben Sie die Produktionen an, die zum Erstellen der Ausdrücke notwendig sind. Die Formatierung (Zeilenumbrüche und Einrückung) dient dabei nur zur besseren Lesbarkeit und ist nicht Teil der Grammatik.

1. ```
   a := 11;
   b := 0;
   while a ≠ 0 do
     a := a - 2;
     b := b + 1
   end
   ```

2. ```
   c := 1;
   b := a + b;
   while a  ≠ 5 do
     b := b - 2;
     a := a + 1;
   end
   ```