

2. Februar 2004

Übungen zu „Grundlagen des Compilerbaus“, WS 2003/04

Nr. 12, Abgabe und Besprechung: 9. Februar 2004 in der Übung

Hinweise: Dies ist das letzte Übungsblatt mit schriftlichen Aufgaben. *Eine* weitere Übung wird nur noch mündliche Aufgaben enthalten.

Die **Klausur** findet am **19.Februar 2004, 13:30 - 15:30 Uhr in HS I, LE** statt.

Mündliche Aufgaben

12.1 Zirkulartätstest

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

$$\begin{array}{lll} G : S & \rightarrow & AaB \quad s_0 = s_2.3, i_1.1 = s_2.3 * s_2, i_2.1 = s_1.3 * s_2.1, i_1.3 = 1, i_2.3 = s_1.1 \\ A & \rightarrow & Ba \quad s_1.0 = 2 * s_2.1, s_2.0 = s_1.1, i_1.1 = i_1.0, i_2.1 = i_2.0 + s_2 \\ A & \rightarrow & a \quad s_1.0 = 1, s_2.0 = s_1 \\ B & \rightarrow & b \quad s_1.0 = i_1.0, s_2.0 = s_1 \\ B & \rightarrow & c \quad s_1.0 = s_1, s_2.0 = i_2.0 \end{array}$$

12.2 Zwischencode für ein PSA-Programm

- (a) Ermitteln Sie die Übersetzung in Maschinencode ($\text{trans}(P)$) für das nebenstehende PSA-Programm P.
- (b) Berechnen Sie die Konfigurationsfolge der abstrakten Maschine, die bei Ausführung des resultierenden MA-Programms und dem Startzustand $(1, \varepsilon, h_D)$ mit $h_D(i) = 0 \forall i \in \mathbb{N}$ durchlaufen wird.

```
const X = 4;
var Y;
Y := 1;
while (Y * Y) < X do
    Y := Y + 1
```

Schriftliche Aufgaben

12.3 L-Attributgrammatiken

7 Punkte

Gegeben sei die folgende L-Attributgrammatik, wobei der Wertebereich aller Attribute die ganzen Zahlen seien:

$$\begin{array}{lll} G : S & \rightarrow & AB \quad s_0 = s_2, i_2 = s_1, i_1 = 0 \\ A & \rightarrow & aA \quad i_2 = i_0 + 1, s_0 = s_2 \\ & & | \varepsilon \quad s_0 = i_0 \\ B & \rightarrow & bB \quad i_2 = i_0 - 1, s_0 = s_2 \\ & & | \varepsilon \quad s_0 = i_0 \end{array}$$

- (a) Der in der Vorlesung definierte TDA für Attributgrammatiken berechnet das Attribut s des Startsymbols und ermittelt dabei alle benötigten Attribute im Ableitungsbaum. Berechnen Sie s für die Analyse des Worts *aabbb* und geben Sie alle dabei durchlaufenen Konfigurationen an. / 4

Die Berechnung von Attributen kann auch mittels der von Martin Jourdan entwickelten rekursiven Berechnung [Jou84] durchgeführt werden: Jedem synthetischen Attribut eines Nonterminals wird eine (rekursive) Prozedur zugeordnet, deren Argumente die inheriten Attribute und die Nachfolgerknoten sind. Die Abhängigkeit von den synthetischen Attributen der Nachfolger im Baum ergibt die Rekursion.

```
Haskell Code
-- Datentypen für Syntaxbaum
data S = S A B
    deriving Show
data A = Node Char A
    | Eps
    deriving Show
-- für B gleiche Darstellung
type B = A
```

- (b) Programmieren Sie für die gegebene Grammatik rekursive Prozeduren, welche aus einem gegebenen Syntaxbaum das Attribut s der Wurzel berechnen. Geben Sie ein allgemeines Schema an, um solche Prozeduren automatisch zu erzeugen. / 3

12.4 Verkürzte Auswertung mit Jumping Code

5 Punkte

Anstelle der Auswertung aller Bestandteile eines Booleschen Ausdrucks in PSA mittels

$$bt : BExpr \times Tab \times Adr \longrightarrow Code$$

kann auch eine verkürzte Auswertung implementiert werden, bei der die Auswertung abgebrochen wird, wenn das Ergebnis feststeht; z.B. falls das erste Argument eines Ausdrucks mit **OR** den Wert **true** ergibt. Dies wird in zahlreichen Programmiersprachen eingesetzt und über sog. *Jumping Code* realisiert.

- (a) Definieren Sie eine rekursive Code-Erzeugungsfunktion

$$bt_{jmp} : BExpr \times Tab \times Adr^3 \longrightarrow Code,$$

welche als Adressenparameter zusätzlich Sprungziele für die Fälle **true** und **false** erhält und nicht benötigten Auswertungscode überspringt.

- (b) Modifizieren Sie die Funktion $ct : Cmd \times Tab \times Adr \longrightarrow Code$ der Vorlesung, so dass sie für Anweisungen mit **if** und **while** die verkürzte Auswertung benutzt.

[Jou84] Martin Jourdan. Strongly non-circular attribute grammars and their recursive evaluation. *ACM SIGPLAN Notices*, 19(6):81–93, June 1984.