PHILIPPS-UNIVERSITÄT MARBURG
Fachbereich Mathematik und Informatik
Prof. Dr. R. Loogen
J.Berthold

D-35032 Marburg
Hans Meerwein Straße

12.Januar 2006

9. Übung zu „Grundlagen des Compilerbaus“, WS 2005/06

Abgabe schriftlicher Aufgaben: Do, 19.Januar 2006 (vor der Vorlesung)
Besprechung mündlicher Aufg.: ab 16.Januar 2006 in der Übung

Mündliche Aufgaben

9.1 **Zirkularitätstest**
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:

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

9.2 **Semantik von PSA-Programmen**
Gegeben sei das nebenstehende PSA-Programm $\Delta\Gamma$.

Bestimmen Sie die Semantik $\mathcal{M}[\![\Delta\Gamma]\!]$ dieses Programms, aufgefasst als Abbildung von Zuständen $\mathcal{M}[\![\Delta\Gamma]\!] : \sigma \mapsto \sigma'$

PSA
```
Δ : const z=3
    var x,y;
Γ : y := 2;
    while y != 0
          y := y - 1;
          x := z * x
```

Schriftliche Aufgaben

9.3 **L-Attributgrammatiken** 7 Punkte
Gegeben sei die folgende L-Attributgrammatik, wobei der Wertebereich aller Attribute die ganzen Zahlen seien:

$$\begin{array}{rcll} G: S & \to & AB & s.0 = s.2,\ i.2 = s.1,\ i.1 = 0 \\ A & \to & aA & i.2 = i.0 + 1,\ s.0 = s.2 \\ & & \mid \varepsilon & s.0 = i.0 \\ B & \to & bB & i.2 = i.0 - 1,\ s.0 = s.2 \\ & & \mid \varepsilon & s.0 = i.0 \end{array}$$

(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

Bitte wenden!

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) Funktion 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
```
-- Syntaxbaum
data S = S   A B
     deriving Show
data A = N A
       | E
     deriving Show
-- für B gleiche Darstellung
type B = A

-- Berechnung
s_inS :: S -> Int
s_inA :: A -> Int -> Int
s_inB :: B -> Int -> Int
```

(b) Programmieren Sie für die gegebene Grammatik rekursive Funktionen, welche das synthetische Attribut s des jeweiligen Nonterminals berechnen. / 3

9.4 PSA- und MA-Programme

5 Punkte

Betrachten Sie die nebenstehende Codesequenz für die abstrakte Maschine MA:

(a) Geben Sie die bei Ausführung dieses Codes durchlaufenen Sequenzen von Maschinenzuständen mit anfänglichem Hauptspeicher H_i an.

$$H_1(n) = \begin{cases} 16 & : n = 1 \\ 24 & : n = 2 \\ \perp & : \text{sonst} \end{cases} \quad , \quad H_2(n) = \begin{cases} 14 & : n = 1 \\ -7 & : n = 2 \\ \perp & : \text{sonst} \end{cases}$$

Unterteilen Sie den Code in geeignete Abschnitte und kürzen Sie gleichartige Schritte ab.

(b) Formulieren Sie ein äquivalentes Programm in der Sprache PSA.

Welche Randbedingungen müssen für eine erfolgreiche Berechnung gelten, was wird berechnet?

MA-Code
```
0   LOAD 1
1   LOAD 2
2   NEQ
3   JPFALSE 18
4   LOAD 1
5   LOAD 2
6   LESS
7   JPFALSE 13
8   LOAD 2
9   LOAD 1
10  SUB
11  STORE 2
12  JMP 17
13  LOAD 1
14  LOAD 2
15  SUB
16  STORE 1
17  JMP 0
18  NOOP
```

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