

Vorlesung “Compilerbau” SS 2010

10. Übungsblatt
Abgabe: 30.Juni 2010

Aufgabe 1 (6 Punkte)

Betrachten sie die Grammatik $G = (V_N, V_T, P, S)$:

- $V_N = \{S, A\}$
- $V_T = \{a, b, c, d\}$
- $P = \{S \rightarrow Aa \mid bAc \mid dc \mid bAa, A \rightarrow d\}$

Erstellen sie die LALR(1)-Parsertafel !

Aufgabe 2 (6 Punkte)

Sei $\varphi(a) = 5$, $\varphi(b) = 6$ und $\varphi(c) = 7$. Bestimmen Sie

- $code(a = ((a + b) == c)) \varphi$
- $code_R(a + (a + (a + b))) \varphi$
- $code_R(((a + a) + a) + b) \varphi$
- Wieviele Stackzellen werden bei der Ausführung der im Fall b) und der im Fall c) erzeugten Befehlsfolgen belegt?

Aufgabe 3 (8 Punkte)

- Überlegen Sie, wie die Übersetzung „realistischer“ `switch`-Anweisungen aussähe, d. h. lassen Sie zu, dass die `case`-Komponenten in beliebiger Reihenfolge auftreten können, dass mehrere Selektoren die gleiche Anweisungsfolge auswählen können und dass nicht alle Selektoren des Intervalls $[0, k - 1]$ auftreten müssen.
- Ein `break` bewirkt einen Sprung an das Ende der nächsten umfassenden Schleife bzw. der nächsten umfassenden `switch`-Anweisung. Wie muss man die Übersetzungsschemata abändern, damit `breaks` an beliebiger Stelle richtig übersetzt werden?

Aufgabe 4 (6 Punkte)

Übersetzen Sie die folgenden Definitionen in C-Maschinen-Code.

```
int i;
struct list {
    int info;
    struct list *next;
} *l, *t;

int ithElement(struct list *l, int i){
    if (i == 1)
        return l->info;
    else
        return ithElement(l->next, i - 1);
}
```