

Einführung in die Praktische Informatik
6. Übungsblatt WS 99/00
9. Dezember 1999

Aufgabe 1 (Auflösung von Mehrdeutigkeit durch Prioritäten)

Gegeben sei folgende mehrdeutige Grammatik $G_n = (N, T_n, S, P_n)$ für arithmetische Ausdrücke mit n binären Operatoren:

$$N = \{S, E\}, \quad T_n = \{\text{id}, (,), \theta_1, \dots, \theta_n, \$\},$$

mit den Produktionen

$$S \rightarrow E\$, \quad E \rightarrow E\theta_1E | E\theta_2E | \dots | E\theta_nE | (E) | \text{id}$$

Es gelte: θ_i hat höhere Priorität als $\theta_j \Leftrightarrow i > j$.

- Berechnen Sie $dchar(G_n)$. Stellen Sie die Anzahl der Zustände als Funktion von n dar.
- Stellen Sie die $SLR(1)$ -Action-Tabelle auf. Die sich ergebenden Konflikte sollen wie folgt aufgelöst werden:

Aktionen mit θ_i haben Vorrang vor Aktionen mit θ_j , wenn $i > j$. Bei gleichem Operator soll vorrangig reduziert werden, d.h. die Operatoren sollen linksassoziativ sein.

Parsen Sie die drei Worte $\text{id}\theta_i\text{id}\theta_i\text{id}\$$ (i beliebig) und $\text{id}\theta_i\text{id}\theta_j\text{id}\$$ für $i < j$ und $i > j$.

Aufgabe 2 (lex, yacc, Fehlerstrategie)

Erzeugen Sie mit `lex` einen Scanner und mit `yacc` einen Parser für die Grammatik

```
program    →  statlist
statlist  →  statlist statement | statement
statement →  id := expression ; | if expression then statlist fi ; | beep ;
expression → expression + id | id
```

Integrieren Sie mittels des `error` Nonterminals eine Fehlererkennungsstrategie mit Wiederaufsetzen und testen Sie diese mit den folgenden syntaktisch falschen Programmen:

- `id:=id id:=id;` c) `id:=fi;` e) `if;` g) `fi;`
- `id:=id id+id;` d) `id then beep; fi;` f) `then;`