

Vorlesung “Compilerbau” SS 2010

2. Übungsblatt Abgabe: 5. Mai 2010

Aufgabe 1 (4 Punkte)

Gegeben ist die kontextfreie Grammatik $S \rightarrow S S + \mid S S * \mid a$

- Zeigen Sie, wie der String $aa+a*$ von dieser Grammatik erzeugt wird und erstellen Sie einen Parse-Baum für diesen String.
- Welche Sprache wird von dieser Grammatik generiert? Begründen Sie Ihre Antwort.

Aufgabe 2 (4 Punkte)

In der Grammatik für die Programmiersprache C wird der Aufzählungstyp mit den folgenden Regeln spezifiziert:

```
enum-specifier:  
  enum identifizieropt { enumerator-list }  
  enum identifizier  
enumerator-list:  
  enumerator  
  enumerator-list , enumerator  
enumerator:  
  identifizier  
  identifizier = constant-expression
```

Die Schreibweise ist folgendermaßen zu verstehen: Die *kursiv* geschriebenen Worte stellen Nichtterminale dar, die in **Schreibmaschinenschrift** geschriebenen Worte sind Terminalsymbole. Weiterhin stellt jeder übergeordnete Begriff gefolgt von einem Doppelpunkt die linke Seite einer Produktion und die untergeordneten Zeilen Alternativen der rechten Seite dar. Ein optionales Symbol ist mit dem Index “opt” markiert.

Schreiben Sie diese Regeln in der erweiterten Backus-Naur-Form (EBNF) auf und zeichnen Sie das dazugehörige Syntaxdiagramm.

Aufgabe 3 (4 Punkte)

Gegeben sei die Grammatik G mit:

```
 $S \rightarrow aA \mid a$   
 $A \rightarrow aA \mid a \mid bB$   
 $B \rightarrow aB \mid bA \mid b$   
(S=Startzustand, {S, A, B}=Nichtterminalsymbole, {a, b}=Terminalsymbole)
```

Was ist die Sprache ? Begründen sie ihre Antwort.

Aufgabe 4 (6 Punkte)

Konstruieren Sie links-lineare und rechts-lineare Grammatiken für die folgenden Sprachen

- $(0|1)^*00(0|1)^*$
- $0^*(1(0|1))^*$
- $((((0|1|10)^*11)^*00))^*$
- $0^*10^*10^*10^*$