

Vorlesung “Compilerbau” WS 2011/2012

2. Übungsblatt

Abgabe: 9. November 2011

Aufgabe 1 (4 Punkte)

Überführen Sie die folgende Grammatik G in eine äquivalente Grammatik G' in Chomsky-Normalform.

$$S \rightarrow (S)S|\varepsilon$$

Wenden Sie den Algorithmus für das Wortproblem aus der Vorlesung auf G' an und demonstrieren Sie dessen Ablauf mit den Eingabewörtern $(())()$ und $(())(())$.

Erweitern Sie den Algorithmus derart, dass die Ableitung des Eingabewortes ausgegeben werden kann.

Aufgabe 2 (4 Punkte)

Entfernen Sie aus der folgenden Grammatik mit Startsymbol A alle direkten und indirekten Linksrekursionen:

- $A \rightarrow BC|a$
- $B \rightarrow CD|Db$
- $C \rightarrow c$
- $D \rightarrow Aa$

Aufgabe 3 (5 Punkte)

Erstellen Sie eine attributierte Grammatik, die den Wert einer römischen Zahl ausrechnet.

Aufgabe 4 (4 Punkte)

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

```
enum-specifier:  
  enum identifieropt { enumerator-list }  
  enum identifier  
enumerator-list:  
  enumerator  
  enumerator-list , enumerator  
enumerator:  
  identifier  
  identifier = 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.