

Übungsblatt 1

Aufgabe 1: Betrachte die Kontext-freie Grammatik

- (1) $S \rightarrow NP VP$
- (2) $NP \rightarrow d n$
- (3) $VP \rightarrow v$
- (4) $VP \rightarrow VP NP$

Die großgeschriebenen Symbole sind nichtterminale, die anderen terminale Symbole.

Bestimmen Sie die Left-most Derivation, die Right-most Derivation, und eine beliebige andere Derivation des Strings $d n v d n$ für diese Grammatik.

Aufgabe 2: Betrachte Grammatik G mit den Produktionen

- $S \rightarrow A S$
- $S \rightarrow B S$
- $S \rightarrow \varepsilon$
- $A \rightarrow a$
- $B \rightarrow a$

Wieviele Analysen (= Parse-Bäume) erhalten wir für die folgenden Strings: $\varepsilon, a, aa, aaa, a^n$

Bestimmen Sie eine rechtslineare Grammatik (die nur Regeln der Form $A \rightarrow a$ und $A \rightarrow a B$ enthält) und eine linkslineare Grammatik (die nur Regeln der Form $A \rightarrow a$ und $A \rightarrow B a$ enthält) die jeweils die gleiche Sprache wie G erzeugt (aber mit anderen Analysen).

Zeichnen Sie den Parse-Wald für den String $a a a a$ und beide Grammatiken.

Aufgabe 3: Betrachte Grammatik $G = \langle \{S\}, \{0, 1\}, P, S \rangle$ mit der folgenden Menge an Regeln:

- $S \rightarrow S 0 S$
- $S \rightarrow 1$

Ist G eindeutig (unambig/ nicht ambig)? Ist $L(G)$ unambig? Erklären Sie Ihre Antwort.

Bemerkung: Eine Grammatik ist eindeutig, wenn kein String verschiedene Analysen erhält. Eine Sprache ist eindeutig, wenn es eine eindeutige Grammatik gibt die die Sprache erzeugt.

Aufgabe 4: Zeigen Sie anhand eines Beispiels, dass Grammatiken, welche sowohl rechtslineare als auch linkslineare Regeln enthalten, nicht-reguläre Sprachen erzeugen können.

Tipp: Die Sprache $\{a^n b^n | n \in \mathcal{N}\} = \{ab, aabb, aaabbb, \dots\}$ ist eine nicht-reguläre Sprache. Erstellen Sie eine Grammatik mit rechtslinearen und linkslinearen Regeln, welche diese Sprache erzeugt.

Aufgabe 5: Gegeben sei die Grammatik

- (1) $S \rightarrow NP VP$
- (2) $NP \rightarrow a n$
- (3) $VP \rightarrow v$
- (4) $VP \rightarrow VP NP$

Welche Folge von Konfigurationen durchläuft ein Top-Down-Erkennen, der Linksableitungen erzeugt und durch Backtracking die Produktionen von oben nach unten der Reihe nach durchprobiert, bei der Eingabe “*a n v a n*”.

Welche Konfigurationen (mit Backtracking) durchläuft ein Bottom-Up-Erkennen, der Rechtsableitungen erzeugt, bei derselben Grammatik und Eingabe, wenn er Reduktionen vor Shift-Operationen probiert und die Regeln von oben nach unten probiert?

Aufgabe 6: Welches Wort entspricht dem Linksparse (Linkskontrollwort) *1 2 4 4 3 2 2* bei der obigen Grammatik? Wie lautet der Rechtsparse für dasselbe Wort?

Aufgabe 7: Zeigen Sie, daß die folgende Grammatik

- $S \rightarrow A a A b \mid B b B a$
- $A \rightarrow \varepsilon$
- $B \rightarrow \varepsilon$

die LL(1)-Eigenschaft besitzt, indem Sie die LL(1) Steuertabelle generieren.

Aufgabe 8: Konstruieren Sie die LL(1)-Steuertabelle für die Grammatik

- $S \rightarrow NP VP$
- $VP \rightarrow v NP$
- $NP \rightarrow d N1$
- $NP \rightarrow N1$
- $N1 \rightarrow a N1$
- $N1 \rightarrow n$
- $N1 \rightarrow \varepsilon$

Aufgabe 9:

Wieviele verschiedene Wörter generiert eine LL(0)-Grammatik mindestens/höchstens?

Beachten Sie, dass ein LL(0)-Parser ganz ohne Vorschauzeichen entscheiden muss, welche Regel er anwendet.