Automatentheorie und formale Sprachen
kontextfreie Sprachen

Dozentin: Wiebke Petersen

24.6.2009

Typ 2-Sprachen: kontextfreie Grammatiken und Sprachen

Definition

Eine Grammatik (N, T, S, P) *heißt* kontextfrei*, wenn alle Regeln/Produktionen die folgende Form haben:*

$$A \rightarrow \alpha, \text{ mit } A \in N \textit{ und } \alpha \in (T \cup N)^*.$$

Eine durch eine kontextfreie Grammatik erzeugte Sprache heißt kontextfrei.

Typ 2-Sprachen: kontextfreie Grammatiken und Sprachen

Definition

Eine Grammatik (N, T, S, P) heißt kontextfrei*, wenn alle Regeln/Produktionen die folgende Form haben:*

$$A \rightarrow \alpha, \text{ mit } A \in N \textit{ und } \alpha \in (T \cup N)^*.$$

Eine durch eine kontextfreie Grammatik erzeugte Sprache heißt kontextfrei.

Theorem

Die Menge der kontextfreien Sprachen bildet eine echte Obermenge der Menge der regulären Sprachen.

Typ 2-Sprachen: kontextfreie Grammatiken und Sprachen

Definition

Eine Grammatik (N, T, S, P) heißt kontextfrei, wenn alle Regeln/Produktionen die folgende Form haben:

$$A \to \alpha, \text{ mit } A \in N \textit{ und } \alpha \in (T \cup N)^*.$$

Eine durch eine kontextfreie Grammatik erzeugte Sprache heißt kontextfrei.

Theorem

Die Menge der kontextfreien Sprachen bildet eine echte Obermenge der Menge der regulären Sprachen.

Proof: Aus der Definition rechts-linearer Grammatiken folgt, dass jede reguläre Sprache kontextfrei ist.

$L(a^n b^n)$ ist kontextfrei, aber nicht regulär ($S \to aSb, S \to \epsilon$).

Examples of context-free languages

- $L_1 = \{ww^R : w \in \{a,b\}^*\}$
- $L_2 = \{a^i b^j : i \geq j\}$
- $L_3 = \{w \in \{a,b\}^* : \text{more } a's \text{ than } b's\}$
- $L_4 = \{w \in \{a,b\}^* : \text{number of } a's \text{ equals number of } b's\}$

Examples of context-free languages

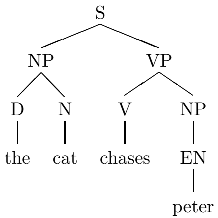
- $L_1 = \{ww^R : w \in \{a,b\}^*\}$
- $L_2 = \{a^i b^j : i \geq j\}$
- $L_3 = \{w \in \{a,b\}^* : \text{more } a's \text{ than } b's\}$
- $L_4 = \{w \in \{a,b\}^* : \text{number of } a's \text{ equals number of } b's\}$

$$\left\{\begin{array}{lllllllll} S & \rightarrow & aB & A & \rightarrow & a & B & \rightarrow & b \\ S & \rightarrow & bA & A & \rightarrow & aS & B & \rightarrow & bS \\ S & \rightarrow & \epsilon & A & \rightarrow & bAA & B & \rightarrow & aBB \end{array}\right\}$$

Derivation tree

$G_1 = \langle\{\mathsf{S,NP,VP,N,V,D,N,EN}\}, \{\mathsf{the,\ cat,\ peter,\ chases}\}, \mathsf{S}, P\rangle$

$$P = \left\{ \begin{array}{rcl rcl rcl} \mathsf{S} & \rightarrow & \mathsf{NP\ VP} & \mathsf{VP} & \rightarrow & \mathsf{V\ NP} & \mathsf{NP} & \rightarrow & \mathsf{D\ N} \\ \mathsf{NP} & \rightarrow & \mathsf{EN} & \mathsf{D} & \rightarrow & \mathsf{the} & \mathsf{N} & \rightarrow & \mathsf{cat} \\ \mathsf{EN} & \rightarrow & \mathsf{peter} & \mathsf{V} & \rightarrow & \mathsf{chases} & & & \end{array} \right\}$$



One derivation determines one derivation tree, but
the same derivation tree can result from different derivations.

Ambiguous grammars and ambiguous languages

Definition

Given a context-free grammar G: A derivation which always replaces the left furthest nonterminal symbol is called left-derivation

Left-derivations and derivation trees determine each other!

Ambiguous grammars and ambiguous languages

Definition

Given a context-free grammar G: A derivation which always replaces the left furthest nonterminal symbol is called left-derivation

Left-derivations and derivation trees determine each other!

Definition

A context-free grammar G is ambiguous iff there exists a $w \in L(G)$ with more than one left-derivation, $S \longrightarrow^* w$.

Ambiguous grammars and ambiguous languages

Definition

Given a context-free grammar G: A derivation which always replaces the left furthest nonterminal symbol is called left-derivation

Left-derivations and derivation trees determine each other!

Definition

A context-free grammar G is ambiguous iff there exists a $w \in L(G)$ with more than one left-derivation, $S \longrightarrow^* w$.

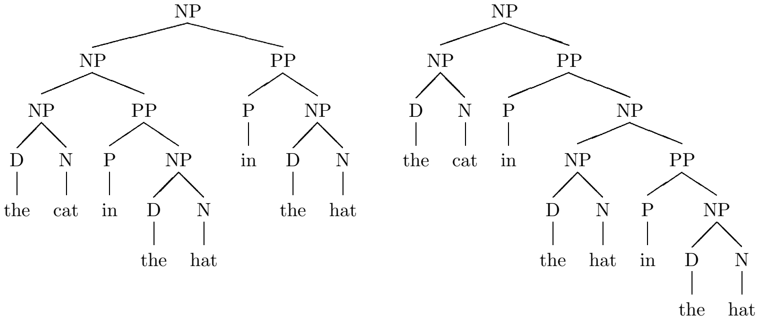
Definition

A context-free language L is inherently ambiguous iff each context-free grammar G with $L(G) = L$ is ambiguous.

Example of an ambiguous grammar

$G = (N, T, \mathrm{NP}, P)$ with $N = \{\mathrm{D}, \mathrm{N}, \mathrm{P}, \mathrm{NP}, \mathrm{PP}\}$, $T = \{\mathrm{the}, \mathrm{cat}, \mathrm{hat}, \mathrm{in}\}$,

$$P = \left\{ \begin{array}{lllllllll} \mathrm{NP} & \rightarrow & \mathrm{D\ N} & \mathrm{D} & \rightarrow & \mathrm{the} & \mathrm{N} & \rightarrow & \mathrm{hat} \\ \mathrm{NP} & \rightarrow & \mathrm{NP\ PP} & \mathrm{N} & \rightarrow & \mathrm{cat} & \mathrm{P} & \rightarrow & \mathrm{in} \\ \mathrm{PP} & \rightarrow & \mathrm{P\ NP} & & & & & & \end{array} \right\}$$



Chomsky Normal Form

Definition

A grammar is in Chomsky Normal Form (CNF) if all production rules are of the form

1. $A \longrightarrow a$
2. $A \longrightarrow BC$

   with $A, B, C \in T$ *and* $a \in \Sigma$ *(and if necessary* $S \longrightarrow \epsilon$ *in which case* S *may not occur in any right-hand side of a rule).*

Theorem

Each context-free language is generated by a grammar in CNF.

Each context-free language is generated by a grammar in CNF

4 steps

0. Adapt the grammar such that no rules of the type $A \longrightarrow \epsilon$ occur (if $S \longrightarrow \epsilon$, S may not occur in any right-hand-side of a rule).
1. Adapt the grammar such that terminals only occur in rules of type $A \longrightarrow a$.
2. Eliminate $A \longrightarrow B$ rules.
3. Eliminate $A \longrightarrow B_1 B_2 \dots B_n$ $(n > 2)$ rules.

Each context-free language is generated by a grammar in CNF

4 steps

0. Adapt the grammar such that no rules of the type $A \rightarrow \epsilon$ occur (if $S \rightarrow \epsilon$, S may not occur in any right-hand-side of a rule).
1. Adapt the grammar such that terminals only occur in rules of type $A \rightarrow a$.
2. Eliminate $A \rightarrow B$ rules.
3. Eliminate $A \rightarrow B_1 B_2 \dots B_n$ $(n > 2)$ rules.

example

$G = (\{S\}, \{a, b\}, S, P)$ with $P = \{S \rightarrow aSb, S \rightarrow ab\}$

1. Adapt the grammar such that terminals only occur in rules of type $A \rightarrow a$.
   $P' = \{S \rightarrow X_a S X_b, S \rightarrow X_a X_b, X_a \rightarrow a, X_b \rightarrow b\}$
2. Eliminate $A \rightarrow B$ rules.
   $P'' = P'$
3. Eliminate $A \rightarrow B_1 B_2 \dots B_n$ $(n > 2)$ rules.
   $P''' = \{S \rightarrow X_a Z, Z \rightarrow S X_b, S \rightarrow X_a X_b, X_a \rightarrow a, X_b \rightarrow b\}$

Hausaufgaben

Exercise 1

1. *Geben sie eine kontextfreie Grammatik zu den folgenden Sprachen an:*
   - $L_1 = \{ww^R : w \in \{a,b\}^*\}$
   - $L_2 = \{a^i b^j : i \geq j\}$
   - $L_3 = \{w \in \{a,b\}^* : w \text{ enthält mehr } a\text{'s als } b\text{'s}\}$
2. - *Bringen sie die folgende Grammatik in die Chomsky-Normalenform:*
     $G = (N, T, NP, P)$ *mit* $N = \{S, NP, VP, N, V, N, EN\}$, $T = \{the, cat, dog, peter, mary\ chases\}$, *und*

     $$P = \left\{ \begin{array}{rclrclrcl} S & \rightarrow & NP\ VP & VP & \rightarrow & V\ NP & NP & \rightarrow & the\ N \\ NP & \rightarrow & EN & N & \rightarrow & cat & N & \rightarrow & dog \\ EN & \rightarrow & peter & EN & \rightarrow & mary & V & \rightarrow & chases \end{array} \right\}$$
   - *Bringen sie die Grammatik zur Sprache L_4 von Folie 3 in Chomsky-Normalenform.*

Gruppenarbeit

Gruppe 1: Führen sie das in Klabunde beschriebene Verfahren zur Konstruktion eines deterministischen endlichen Automaten aus einem nichtdeterministischen Automaten an einem kleinen Beispiel im Detail durch.

Gruppe 2: Beschreiben sie an 2-3 Beispielen, wie man effizient einen deterministischen Automaten aus einem nichtdeterministischen konstruiert.

Gruppe 3: Beschreiben sie an 2-3 Beispielen, wie man ϵ-Übergänge eliminiert.

Gruppe 4: Beschreiben sie an Beispielen, wie man endliche Automaten zur Konkatenation zweier regulärer Sprachen und zur Vereinigung und zur Schnittmenge zweier Sprachen bildet, und wie man endliche Automaten zum Kleenschen Stern einer regulären Sprache und zum Komplement einer regulären Sprache konstruiert.