



Grundlegende Parsingalgorithmen

Einführung und formale Sprachen

Kurt Eberle

k.eberle@lingenio.de

(Viele Folien, Teile von Folien, Materialien von **Helmut Schmid's**
Parsing-Kurs WS14 Tübingen, u.a.)

30. Juli, 2018



Überblick

Organisatorisches

Einführung

Formale Sprachen

Das Wortproblem

Grammatik

Grammatik

Erkenner und Parser

Chomsky Hierarchie

Unterscheidungskriterien

Natürliche Sprache



Ziele

- Organisatorisches
- Motivation
- Plan
- Formale Sprachen



Überblick

Organisatorisches

Einführung

Formale Sprachen

Das Wortproblem

Grammatik

Grammatik

Erkenner und Parser

Chomsky Hierarchie

Unterscheidungskriterien

Natürliche Sprache



Zeiten

- ▶ 9.15-10.45
- ▶ 11.00-12.30
- ▶ 13.00-14.30



Übungen

- ▶ Übungsblätter
- ▶ Besprechung (teilweise) im Unterricht (letzte Stunde)



Vorbedingungen

- ▶ Einführung in CL
- ▶ Kenntnisse in Programmierung nicht wirklich notwendig
- ▶ Eher theoretisch und 'Block & Bleistift'
- ▶ Aber: Algorithmen soweit angegeben, dass sie leicht implementiert werden können



Kriterien für einen Schein

- ▶ Klausur
- ▶ 17. August?



Kurshomepage

- ▶ URL: `http://www.cl.uni-heidelberg.de/courses/ss18/parsing/`
mehr Material, etwas ausführlicher:
- ▶ `http://www.sfs.uni-tuebingen.de/~keberle/Parsing/ParsingHP.html`



Überblick

Organisatorisches

Einführung

Formale Sprachen

Das Wortproblem

Grammatik

Grammatik

Erkenner und Parser

Chomsky Hierarchie

Unterscheidungskriterien

Natürliche Sprache



Was bedeutet 'Parsing'?

- ▶ von Lateinisch 'pars' = 'der Teil'
- ▶ 'parsen' = 'in Teile zerlegen' = 'analysieren'
- Strings/strukturierte Informationen/Sätze in Teile zerlegen!



Motivation

Parsing

- ▶ Warum?



Motivation

Parsing

- ▶ Warum?
- ▶ Finde Informationsatome und beziehe diese aufeinander
- Um eine Maschine zu steuern (Programmiersprache)
- Um den Inhalt kategorial richtig zu erfassen (Lexika, etc.)
- Um den Inhalt zu 'verstehen' (Natürliche Sprache)



Parsing

Problem

- ▶ Mehrdeutigkeit!
Deshalb: unterscheide Anwendung bezogen auf
- ▶ *Formale Sprachen*: nicht mehrdeutig und
- ▶ *Natürliche Sprachen*: mehrdeutig
= viel schwieriger



Beispiel: Natürliche Sprache: mehrere Lesarten

! : weil er zwei Tage alte Zeitungen las.

Syntactic analysis no. 1. Evaluation = 0.1209 ...

	top weil782344(1,7)	subconj
	subj(n) er206261(2)	noun(pron(pers3), [nom, sg, m, kda!X4], nwh)
	nadj zwei(3,u)	noun(num, [[!X7!p1]], f, o!X8], X9)
	anum tag689060(4)	noun(cn, [[!nom!p1], [gen!p1], [acc!p1], m, o!inum], X6)
	nadj alt23110(5)	adj(nMnaFNSnaP, X5)
	obj(n) zeitung807082(6)	noun(cn, [acc, pl, f, na!aj], nwh)
	sccomp les437118(7,2,6,u)	verb(fin([pers3!sg], past, X1:dc1:nwh))

because he read two days old newspapers.

Syntactic analysis no. 3. Evaluation = 0.1309 ...

	top weil782344(1,7)	subconj
	subj(n) er206261(2)	noun(pron(pers3), [nom, sg, m, kda!X4], nwh)
	nadj zwei(3,u)	noun(num, [[!X5!p1]], f, o!X6], X7)
	vadv tag689060(4)	noun(cn, [acc, pl, m, o!inum], nwh)
	nadj alt23110(5)	adj(nMnaFNSnaP, X10)
	obj(n) zeitung807082(6)	noun(cn, [acc, pl, f, na!aj], nwh)
	sccomp les437118(7,2,6,u)	verb(fin([pers3!sg], past, X1:dc1:nwh))

because he read old newspapers for two days.



Ambiguität

'Explosion'

- ▶ Ambiguitäten können sich multiplizieren!



Syntaktische Ambiguität

Syntactic analysis no. 5. Evaluation = 3.43209 ...

—	subj(n)	er206261(1)	noun(pron(pers3), [nom, sg, m, kda iX3], nwh)
■	top	konn395620(2,1,u)	verb(fin([pers3:sg], past, ind:dc1:nwh))
—	nadj	zwei(3,u)	noun(num, [[iX17:pl]], f, o iX18], X19)
—	vadv	tag689060(4)	noun(cn, [acc, pl, m, o inum], nwh)
—	adjpre	zu814914(5)	qual(pre, X16)
—	vadv	spät649940(6)	adv(X15, adv)
—	sep(com)	,	separator
—	vsubconj	weil782344(8,14)	subconj
—	subj(n)	er206261(9)	noun(pron(pers3), [nom, sg, m, kda iX7], nwh)
—	nadj	zwei(10,u)	noun(num, [[iX10:pl]], f, o iX11], X12)
—	anum	tag689060(11)	noun(cn, [[iX10:pl], [gen:pl], f, acc:pl]], m, o inum], X9)
—	nadj	alt23110(12)	adj(nMnaFNSnaP, X8)
—	obj(n)	zeitung807082(13)	noun(cn, [acc, pl, f, na iaj], nwh)
—	sccomp	les437118(14,9,13,u)	verb(fin([pers3:sg], past, X4:dc1:nwh))

he was late two days because he read two days old newspapers.

Syntactic analysis no. 7. Evaluation = 3.43309 ...

—	subj(n)	er206261(1)	noun(pron(pers3), [nom, sg, m, kda iX3], nwh)
■	top	konn395620(2,1,u)	verb(fin([pers3:sg], past, ind:dc1:nwh))
—	nadj	zwei(3,u)	noun(num, [[iX16:pl]], f, o iX17], X18)
—	vadv	tag689060(4)	noun(cn, [acc, pl, m, o inum], nwh)
—	adjpre	zu814914(5)	qual(pre, X15)
—	vadv	spät649940(6)	adv(X14, adv)
—	sep(com)	,	separator
—	vsubconj	weil782344(8,14)	subconj
—	subj(n)	er206261(9)	noun(pron(pers3), [nom, sg, m, kda iX7], nwh)
—	nadj	zwei(10,u)	noun(num, [[iX8:pl]], f, o iX9], X10)
—	vadv	tag689060(11)	noun(cn, [acc, pl, m, o inum], nwh)
—	nadj	alt23110(12)	adj(nMnaFNSnaP, X13)
—	obj(n)	zeitung807082(13)	noun(cn, [acc, pl, f, na iaj], nwh)
—	sccomp	les437118(14,9,13,u)	verb(fin([pers3:sg], past, X4:dc1:nwh))

he was late two days because he read old newspapers for two days.



Natürliche Sprache

Ambiguität

- ▶ morphologische Ebene
- ▶ syntaktische Ebene
- ▶ semantische Ebene
- ▶ pragmatische Ebene



Natürliche Sprache

Suchraum ist ein Problem

- ▶ Versuche den effizientesten Algorithmus zu finden!



Natürliche Sprache

Der Grammatiktyp ist wichtig

Kann ich eine Sprache mit einer

- ▶ 'regulären' / 'Kontext-freien' / 'Kontext-sensitiven'
Grammatik beschreiben?

→ unterschiedliche Problemklassen für das *Erkennungsproblem*



Programm

Mo 30.07		
09.15	Einführung & Formale Sprachen	
11.00	Top-Down & Bottom-Up Parsing	
13.00	Tabellen-gesteuertes Parsing: LL Parsing	
Di 31.07		
09.15	Tabellen-gesteuertes Parsing: LR Parsing	
11.00	xLR Erkennen	
13.00	Übungen	
Mi 01.08.		
09.15	Tomita-Parser	
11.00	Chart Parser: CYK	
13.00	Übungen	
Do 02.08		
09.15	Chart Parser: Earley Parser	
11.00	Chart Parser: Left-Corner Parsing	
13.00	Übungen	
Fr 03.08		
09.15	Grammatik-Transformationen, DCGs, Feature Constraints	
11.00	Dependency Grammar Parsing	
13.00	Statistisches Parsing	



Literatur I

- ▶ Folien auf <http://www.cl.uni-heidelberg.de/courses/ss18/parsing/> und ebenfalls dort:
- ▶ Skript von Helmut Schmid
- ▶ Nederhof/Satta: *Tabular Parsing*
<http://arxiv.org/pdf/cs/0404009.pdf>
- ▶ Andreas Kunert: *LR(k)-Analyse für Pragmatiker*
<https://amor.cms.hu-berlin.de/~kunert/papers/lr-analyse/lr.pdf>



Literatur II

- ▶ Grune/Jacobs: *Parsing Techniques: A Practical Guide*
http://dickgrune.com/Books/PTAPG_1st_Edition/BookBody.pdf
Informatik-orientiert
- ▶ Aho/Sethi/Ullman: *Compilers: Principles, Techniques, and Tools*
LL- and LR Parsing, Informatik-orientiert
- ▶ Jurafsky/Martin: *Speech and Language Processing*
(Draft auf der Webseite) *Linguistik-orientiert*
- ▶ Aho/Ullman: *The Theory of Parsing, Translation, and Compiling.*
Vol. 1 (Parsing)
Grundlagen, Kontext-freie Analyse (ohne Tomita Parser)
- ▶ Hopcroft/Ullman: *Introduction to Automata Theory, Languages and Computation*
Zu formalen Sprachen



Überblick

Organisatorisches

Einführung

Formale Sprachen

Das Wortproblem

Grammatik

Grammatik

Erkenner und Parser

Chomsky Hierarchie

Unterscheidungskriterien

Natürliche Sprache



Parsing und Formale Sprachen: Grundbegriffe

Was wird geparst?

Strings eines Alphabets

- ▶ Alphabet: set of symbols $\Sigma = \{a, b, c\}$
- ▶ $\Sigma^n, \Sigma^*, \Sigma^+$
- ▶ String: $w \in \Sigma^*$ *cbaac, baca*
- ▶ Concatenation of strings ww'
- ▶ Length of a string $|w|$
- ▶ Empty string ε
- ▶ (True) prefix/suffix
- ▶ Notation: $a^n b^m$



Das Wortproblem

Aufgabe

- ▶ Gegeben eine formale Grammatik und ein Eingabestring:
- ▶ Entscheide, ob der String eine Ableitung aus der Grammatik hat
- ▶ Folgt er aus dem Startsymbol der Grammatik, ist er ein Satz der zur Grammatik gehörenden Sprache
- ▶ Eine Ableitung ist eine Analyse
- ▶ Analysen können als *Parse-Bäume* oder *Regelsequenzen* dargestellt werden
- ▶ Es ist möglich, dass ein Satz mehrere Analysen hat



Grammatik

- ▶ Grammar: $G = (V, \Sigma, P, S)$
 - V : finite set of *non-terminal symbols*
 - Σ : finite set of *terminal symbols* ($V \cap \Sigma = \emptyset$)
 - P : finite set of *productions* (grammar rules)
 - S : *start symbol* of the grammar ($S \in V$)
- ▶ Productions $p \in P$ have the form $\alpha \rightarrow \beta$, where $\alpha \in (V \cup \Sigma)^* V (V \cup \Sigma)^*$ and $\beta \in (V \cup \Sigma)^*$
- ▶ Notational convention:
 $a, a_i \in \Sigma$; $A, B, C \in V$; $w, r \in \Sigma^*$; $\alpha, \beta \in (V \cup \Sigma)^*$



Sprache einer Grammatik

Die aus G erzeugte Sprache: $L(G)$

- ▶ Derivability:
 - ▶ v is immediately derivable from u ($u \Rightarrow v$), iff
 $u = \gamma\alpha\delta, v = \gamma\beta\delta$ and $\alpha \rightarrow \beta \in P$
 - ▶ v is in n steps derivable from u ($u \xRightarrow{n} v$), iff
 $\exists u_0 \dots u_n : u = u_0 \Rightarrow u_1 \Rightarrow \dots \Rightarrow u_n = v$
 - ▶ v is (in any number of steps) derivable from u ($u \xRightarrow{*} v$), iff $u \xRightarrow{n} v$
 for some $n \geq 0$.
- ▶ Sentential form:

$$\{s \in (V \cup \Sigma)^* \mid S \xRightarrow{*} s\}$$
- ▶ Language generated by G :

$$L(G) = \{w \in \Sigma^* \mid S \xRightarrow{*} w\}$$
- ▶ Equivalence of grammars:

$$G \equiv G' \iff L(G) = L(G')$$



Einige Ableitungen I

Derivations

$G_{e_2}: S \rightarrow ABC, A \rightarrow aA, A \rightarrow a, B \rightarrow bB, B \rightarrow b, C \rightarrow cC, C \rightarrow c$

Left-most

S

ABC

aABC

aaBC

aabBC

aabbC

aabbc

Right-most

S

ABC

ABc

AbBc

Abbc

aAbbc

aabbc



Überblick

Organisatorisches

Einführung

Formale Sprachen

Das Wortproblem

Grammatik

Grammatik

Erkenner und Parser

Chomsky Hierarchie

Unterscheidungskriterien

Natürliche Sprache



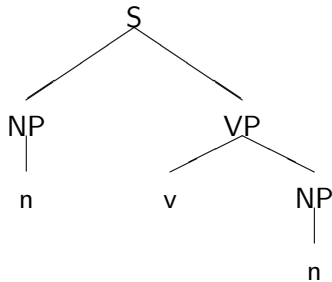
Erkenner und Parser

- ▶ Recognizer: checks whether $w \in L(G)$
- ▶ Parser: additionally prints the parse(s)
- ▶ Parse:
 - Sequence of rules used in a derivation
(i.e.: history of derivation)
 - N.B. not necessarily displayed graphically
- ▶ Left-most/right-most parse
- ▶ ambiguous/unambiguous grammar
(ambiguous: more than one (left-most/right-most) parse)
- ▶ ambiguous/unambiguous language (language L is unambiguous iff exists unambiguous G with $L = L(G)$)



Parse-Baum

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





Parse und Parse-Baum

- ▶ Parse = sequence of derivations (+ reference to rule) vs.
- ▶ Parse tree (without sequential information)



Überblick

Organisatorisches

Einführung

Formale Sprachen

Das Wortproblem

Grammatik

Grammatik

Erkenner und Parser

Chomsky Hierarchie

Unterscheidungskriterien

Natürliche Sprache



Formale Sprachen

Chomsky Hierarchie

- ▶ Type-0 grammars:

$\alpha \rightarrow \beta$ (unrestricted)

- ▶ Type-1 grammars (context sensitive):

$\alpha \rightarrow \beta$ with $|\alpha| \leq |\beta|$ (exception: $S \rightarrow \epsilon$)

- ▶ **Type-2** grammars (context free):

$A \rightarrow \alpha$

- ▶ Type-3 grammars (regular):

$A \rightarrow wB$ or $A \rightarrow w$ (right linear) and

$A \rightarrow Bw$ or $A \rightarrow w$ (left linear) respectively

where $w \in \Sigma^*$ (= extended RG vs strict RG: $w = a, A \rightarrow \epsilon$)

⇒ General phrase structure, Context sensitive, context free, regular languages



Reguläre Grammatiken und Reguläre Ausdrücke

Reguläre Ausdrücke

Given an alphabet of symbols Σ , the following are all and only the *regular expressions* over the alphabet Σ ($\cup \{ \emptyset, \epsilon, |, [,], * \}$):

\emptyset	empty set
ϵ	the empty string (sometimes: '0' or \square)
a	for all $a \in \Sigma$
$[\alpha \mid \beta]$	Union (for α, β reg. expr.) ($\alpha \cup \beta$)
$[\alpha \beta]$	Concatenation (for α, β reg. expr.)
$[\alpha^*]$	Kleene star (for α reg. expr.)



Reguläre Ausdrücke

Sind $L(A)$, $L(B)$, $L(C)$ regulär, so sind:
 $L(A) \& L(B)$, $L(A) - L(B)$, $\neg L(A)$ regulär.

Deshalb können die folgenden Operatoren dazugenommen werden:

$\alpha \& \beta$ intersection
 $\alpha - \beta$ relative complement
 $\sim \alpha$ complement



Reguläre Sprache

Definition

- ▶ durch regulären Ausdruck



Reguläre Sprache

Definition

- ▶ durch regulären Ausdruck
- ▶ durch reguläre Grammatik
(\rightarrow syntaktische Definitionen)



Reguläre Sprache

Definition

- ▶ durch regulären Ausdruck
- ▶ durch reguläre Grammatik
(→syntaktische Definitionen)
- ▶ durch FSA
(→prozedurale Definition)



Finite-state Automata

Definition (FSA)

A finite-state automaton is a quintuple $(\Sigma, Q, i, F, \Delta)$ where

- ▶ Σ is a finite set called the alphabet,



Finite-state Automata

Definition (FSA)

A finite-state automaton is a quintuple $(\Sigma, Q, i, F, \Delta)$ where

- ▶ Σ is a finite set called the alphabet,
- ▶ Q is a finite set of states,



Finite-state Automata

Definition (FSA)

A finite-state automaton is a quintuple $(\Sigma, Q, i, F, \Delta)$ where

- ▶ Σ is a finite set called the alphabet,
- ▶ Q is a finite set of states,
- ▶ $i \in Q$ is the initial state,



Finite-state Automata

Definition (FSA)

A finite-state automaton is a quintuple $(\Sigma, Q, i, F, \Delta)$ where

- ▶ Σ is a finite set called the alphabet,
- ▶ Q is a finite set of states,
- ▶ $i \in Q$ is the initial state,
- ▶ $F \subseteq Q$ the set of final states, and



Finite-state Automata

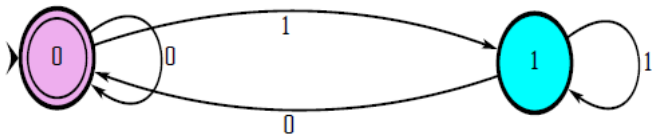
Definition (FSA)

A finite-state automaton is a quintuple $(\Sigma, Q, i, F, \Delta)$ where

- ▶ Σ is a finite set called the alphabet,
- ▶ Q is a finite set of states,
- ▶ $i \in Q$ is the initial state,
- ▶ $F \subseteq Q$ the set of final states, and
- ▶ $\Delta \subseteq Q \times (\Sigma \cup \epsilon) \times Q$ is the set of edges (the transition relation).



FSA - Example





Chomsky Hierarchy - Examples

Regular Languages

$\langle G, \Sigma, S, R \rangle$

$\Sigma = \{\text{un, be, lehr, bar, keit}\},$

$\Phi = \{S, A1, A2, A3, A4\},$

$S = S,$

$R = \{ S \rightarrow \text{un } S,$

$S \rightarrow \text{lehr } A2,$

$S \rightarrow \text{be } A1,$

$A1 \rightarrow \text{lehr } A2,$

$A2 \rightarrow \text{bar } A3,$

$A3 \rightarrow \text{keit } A4,$

$A3 \rightarrow \epsilon,$

$A4 \rightarrow \epsilon\}.$

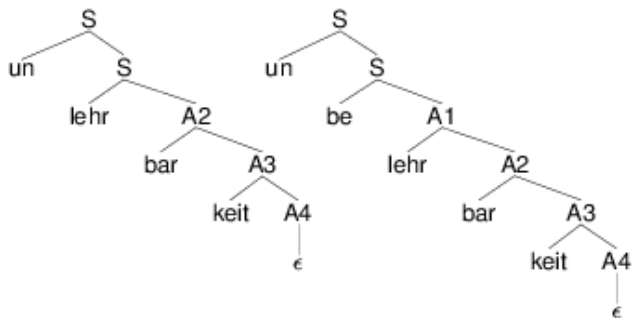
$L(G) = \{ \text{unbelehrbarkeit, lehrbarkeit, unlehrbar, lehrbar, ..} \}$

$\text{unbarkeit} \notin L(G)$



Chomsky Hierarchie - Beispiele

Rechtslinearer Baum





Chomsky Hierarchie - Beispiele

Kontext-freie Sprachen

$$L = \{a^n b a^n \mid n \geq 0\}$$

is not regular!

$\langle G, \Sigma, S, R \rangle$

$$\Sigma = \{a, b\},$$

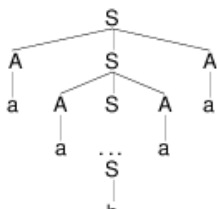
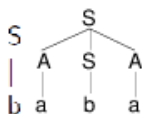
$$\Phi = \{S, A\},$$

$$S = S,$$

$$R = \{ S \rightarrow A S A,$$

$$S \rightarrow b,$$

$$A \rightarrow a\}$$





Chomsky Hierarchie - Beispiele

Kontext-sensitive Sprachen

$L = \{a^n b^n c^n \mid n \geq 1\}$ ist nicht Kontext-frei!

$\langle G, \Sigma, S, R \rangle$

mit

$$R = \{ \begin{array}{ll} S \rightarrow a S B C, & a B \rightarrow a b, \\ S \rightarrow a B C, & b B \rightarrow b b, \\ C B \rightarrow B C, & b C \rightarrow b c, \\ & c C \rightarrow c c \end{array} \}$$

$a^3 b^3 c^3 = aaabbbccc \in L(G)$ gdw. $S \xrightarrow{*} aaabbbccc$

S	
$\Rightarrow a S B C$	$\Rightarrow a a a \mathbf{b} B B C C C$
$\Rightarrow a a S B C B C$	$\Rightarrow a a a b \mathbf{b} B C C C$
$\Rightarrow a a a \mathbf{B} \underline{C} \mathbf{B} C \mathbf{B} C$	$\Rightarrow a a a b b \mathbf{b} C C C$
$\Rightarrow a a a \mathbf{B} \mathbf{B} \underline{C} \underline{C} \mathbf{B} C$	$\Rightarrow a a a b b b \mathbf{c} C C$
$\Rightarrow a a a \mathbf{B} \mathbf{B} \underline{C} \mathbf{B} C C$	$\Rightarrow a a a b b b c \mathbf{c} C$
$\Rightarrow a a a \mathbf{B} \mathbf{B} \mathbf{B} C C C$	$\Rightarrow a a a b b b c c \mathbf{c}$



Chomsky Hierarchie - Beispiele

Sprachen zu allgemeinen Phrasen-Struktur Grammatiken

$$L = \{a^{2^n} \mid n \geq 1\}$$

ist nicht Kontext-sensitiv!

$$G = \langle \{S, A, B, C, D, E\}, \{a\}, R, S \rangle$$

$$\begin{array}{llll} S & \rightarrow & ACaB. & CB \rightarrow E. & aE & \rightarrow & Ea. \\ Ca & \rightarrow & aaC. & aD & \rightarrow & Da. & AE & \rightarrow & \epsilon. \\ CB & \rightarrow & DB. & AD & \rightarrow & AC. \end{array}$$

$$a^{2^2} = aaaa \in L(G) \text{ gdw. } S \xrightarrow{*} aaaa$$

$$\begin{array}{llll} \underline{S} & \Rightarrow & \underline{ACaB} & Aaaaa\underline{CB} & \Rightarrow & Aaaaa\underline{E} \\ \underline{ACaB} & \Rightarrow & \underline{AaaCB} & Aaaaa\underline{E} & \Rightarrow & Aaaa\underline{Ea} \\ \underline{AaaCB} & \Rightarrow & \underline{AaaDB} & Aaaa\underline{Ea} & \Rightarrow & Aaa\underline{Eaa} \\ \underline{AaaDB} & \Rightarrow & \underline{AaDaB} & Aaa\underline{Eaa} & \Rightarrow & Aa\underline{Eaaa} \\ \underline{AaDaB} & \Rightarrow & \underline{ADaaB} & Aa\underline{Eaaa} & \Rightarrow & A\underline{Eaaaa} \\ \underline{ADaaB} & \Rightarrow & \underline{ACaaB} & \underline{AEaaaa} & \Rightarrow & aaaa \\ \underline{ACaaB} & \Rightarrow & \underline{AaaCaB} & & & \end{array}$$



Eigenschaften von Grammatiken

- ▶ ε rule: $A \rightarrow \varepsilon$
- ▶ ε free: $\neg \exists A \in V : A \rightarrow \varepsilon$ (Exception: $S \rightarrow \varepsilon$)
- ▶ cycle free: $\neg \exists A \in V : A \overset{+}{\Rightarrow} A$.
- ▶ chain rules: $A \rightarrow B$
- ▶ unreachable non-terminals: $\neg S \overset{*}{\Rightarrow} \alpha A \beta$
- ▶ unproductive non-terminals: $\neg \exists w \in \Sigma^* : A \overset{*}{\Rightarrow} w$
- ▶ left-recursive/right-recursive non-terminals:
 $A \overset{+}{\Rightarrow} A\alpha$ bzw. $A \overset{+}{\Rightarrow} \alpha A$.
- ▶ left-recursive/right-recursive Grammars



Eigenschaften von Kontext-freien Grammatiken

- ▶ Chomsky normal form:

$$\forall p \in P : p = A \rightarrow BC \text{ or } p = A \rightarrow a$$

(Exception: $S \rightarrow \varepsilon$)

- ▶ Greibach normal form:

$$\forall p \in P : p = A \rightarrow a\alpha$$

where: $a \in \Sigma$ and $\alpha \in V^*$

(Exception: $S \rightarrow \varepsilon$)



Eigenschaften von Kontext-freien Grammatiken

It can be shown that for all context-free grammars G there is a context-free grammar G' with:

- ▶ $L(G) = L(G')$;
- ▶ G' is ε -free;
- ▶ G' does not contain chain rules;
- ▶ G' is cycle free;
- ▶ G' does not contain unreachable non-terminals;
- ▶ G' does not contain unproductive symbols.



Überblick

Organisatorisches

Einführung

Formale Sprachen

Das Wortproblem

Grammatik

Grammatik

Erkenner und Parser

Chomsky Hierarchie

Unterscheidungskriterien

Natürliche Sprache



Sprachen - Unterscheidungskriterien

Pumping Lemma

- ▶ Notwendige Bedingung für reguläre Sprachen

$$\begin{aligned}
 \forall L \in \mathcal{L}_3. \exists n \in \mathbb{N}. \forall z \in L. |z| \geq n \implies \exists u, v, w. & \quad z = u \cdot v \cdot w \wedge \\
 & \quad |uv| \leq n \wedge \\
 & \quad |v| > 0 \wedge \\
 & \quad \forall i \in \mathbb{N}_0. u \cdot v^i \cdot w \in L
 \end{aligned}$$



Sprachen - Unterscheidungskriterien

Pumping Lemma

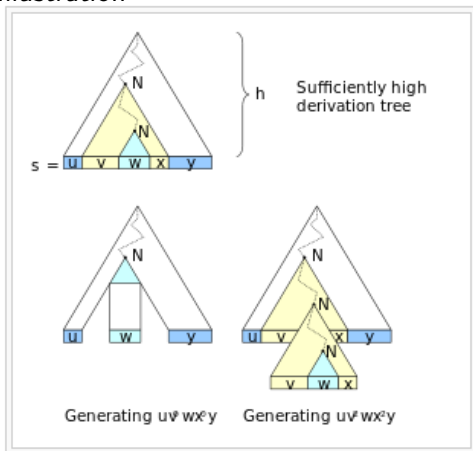
- ▶ Entsprechend für Kontext-freie Sprachen

$\forall L \subseteq \Sigma^*. (\text{context_free}(L) \Rightarrow \exists p \geq 1. \forall s \in L. (|s| \geq p \Rightarrow \exists \underline{u, v, w, x, y} \in \Sigma^*. (s = \underline{uvwx}y \wedge |\underline{vwx}| \leq p \wedge |\underline{vx}| \geq 1 \wedge \forall n \geq 0. \underline{uv^nwx^ny} \in L)))$

Sprachen - Unterscheidungskriterien

Pumping Lemma

Illustration





Überblick

Organisatorisches

Einführung

Formale Sprachen

Das Wortproblem

Grammatik

Grammatik

Erkenner und Parser

Chomsky Hierarchie

Unterscheidungskriterien

Natürliche Sprache



In welche Klasse fallen die natürlichen Sprachen?

Unterscheidung

- ▶ syntaktisch korrekt
- ▶ semantisch korrekt
- ▶ pragmatisch korrekt



In welche Klasse fallen die natürlichen Sprachen?

Beispiele

- ▶ der Hund jagt die Katze
- ▶ jagt der Hund die Katze
- ▶ (*)jagt der die Hund Katze
- ▶ (*) der Hund jagst die Katze
- ▶ (*) (?) Colorless green ideas sleep furiously (Chomsky 57)
- ▶ (*) (?) The king of France is bald (Russell)



In welche Klasse fallen die natürlichen Sprachen?

Sascha Brawer:

<http://www.brawer.ch/prolog/sprachenhierarchie.pdf>

Sind natürliche Sprachen regulär?

Endliche Automaten als Grammatikformalismus?

- ◆ Das Englische besitzt Strukturen der Art ux^ny^nw :

A white male hired another white male

A white male whom a white male hired hired another white male

*A white male whom a white male whom a white male hired hired
hired another white male*

*A white male whom a white male whom a white male whom a white male hired
hired hired hired another white male*

- ◆ u = "A white male"
- ◆ x = "whom a white male"
- ◆ v = ""
- ◆ y = "hired"



In welche Klasse fallen die natürlichen Sprachen?

Annahme

- ▶ Kontext-freier Kern (/Syntax)
- ▶ Aber



Offene Frage

Zürich-Deutsch

(1) ...mer em Hans es huus hälfed aastriiche
...we Hans-DAT the house-ACC helped paint
'... we helped Hans paint the house.'

(2) ...mer de Hans es huus lönd aastriiche
...we Hans-ACC the house-ACC let paint
'... we let Hans paint the house'

(3) ...*mer em Hans es huus lönd aastriiche
...we Hans-DAT the house-ACC let paint
'... we let Hans paint the house.'



In welche Klasse fallen die natürlichen Sprachen?

Zürichdeutsch ist nicht kontextfrei

Wird Zürichdeutsch mit der regulären Sprache

De Jan säit, dass mer Akk.-Obj.* Dat.-Obj.*
es Huus händ wele Akk.-V* Dat.-V* aastriiche.

geschnitten, ergibt sich die Sprache

De Jan säit, dass mer Akk.-Obj.ⁿ Dat.-Obj.^m
es Huus händ wele Akk.-Vⁿ Dat.-V^m aastriiche.

Es ist beweisbar, ...

- ◆ dass diese Sprache nicht kontextfrei ist.
- ◆ Also kann die ursprüngliche Sprache, Zürichdeutsch, auch nicht kontextfrei sein.