The formal Complexity of Natural Languages

Wiebke Petersen

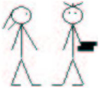
Heinrich-Heine-Universität Düsseldorf
Institute of Language and Information
Computational Linguistics
www.phil-fak.uni-duesseldorf.de/~petersen/

Riga, 2006

Formal complexity of natural languages

- Latvian, German, English, Chinese, ...
- Prolog, Pascal, ...
- Esperanto, Volapük, Interlingua, ...
- proposition logic, predicate logic
- ...

Formal complexity of natural languages

- Latvian, German, English, Chinese, . . .
- vague, ambiguous,
- ambiguities
  - lexical ambiguities (call me tomorrow - the call of the beast)
  - structural ambiguities:
    - the woman sees ⌜ the man ⌝ with the binoculars 
    - the woman sees ⌜ the man with the binoculars ⌝ 
- only experts: humans
- natural languages develop

Formal complexity of natural languages

- difficult to learn as first / second language
- complex phonology / morphology / syntax / . . .
- difficult to parse

Formal complexity of natural languages

- computational complexity
- structural complexity

- Natural languages are modeled as abstract symbol systems with construction rules.
- Questions about the grammaticality of natural sentences corresponds to questions about the the syntactic correctness of programs or about the well-formedness of logic expressions.

How complex are English sentences?

1. Anne sees Peter
2. Anne sees Peter in the garden with the binoculars
3. Anne who dances sees Peter whom she met yesterday in the garden with the binoculars
4. Anne sees Peter and Hans and Sabine and Joachim and Elfriede and Johanna and Maria and Jochen and Thomas and Andrea

The length of a sentence influences the processing complexity, but it is not a sign of structural complexity.!

Grammar Theories vs. Natural Language Theory

Grammar Theories

- explain language data
- are language specific (Latvian, German, . . .)

Natural Language Theory

- a theory about the structure of symbol strings
- not language specific
- allows statements about the mechanisms for generating and recognizing sets of symbol strings

Formal Languages

- Formal languages are sets of words (NL: sets of sentences) which are strings of symbols (NL: words). Everything in the set is a “grammatical word”, everything else isn’t.
- Structured formal languages can be generated by a grammar, i.e. a finite set of production rules.

Formal languages

Definition

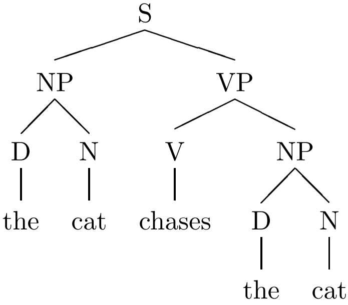
- **alphabet** Σ: nonempty, finite set of **symbols**
- **word**: a finite string $x_1 \dots x_n$ of symbols
- **empty word** ϵ: the word of length 0
- Σ^* is the set of all words over Σ
- **formal language** L: a set of words over an alphabet Σ, i.e. $L \subseteq \Sigma^*$

Formal Grammar

- A formal grammar is a **generating device** which can generate (and analyze) strings/words.
- The set of all strings generated by a grammar is a formal language (= generated language).
- Grammars are finite rule systems.

S	$\rightarrow$	NP VP	VP	$\rightarrow$	V NP	NP	$\rightarrow$	D N
D	$\rightarrow$	the	N	$\rightarrow$	cat	V	$\rightarrow$	chases

S	→	NP VP	VP	→	V NP	NP	→	D N
D	→	the	N	→	cat	V	→	chases



Formal grammar

Definition

A **formal grammar** is a 4-tupel $G = (N, T, S, P)$ with

- an alphabet of terminals T (also denoted Σ),
- an alphabet of nonterminals N with $N \cap T = \emptyset$,
- a start symbol $S \in N$,
- a finite set of rules/productions $P \subseteq \{\alpha_i \rightarrow \beta_i \mid \alpha_i, \beta_i \in (N \cup T)^* \text{ and } \alpha_i \notin T^*\}$.

Context-free language

Definition

A grammar (N, T, S, P) is ***context-free*** *if all production rules are of the form:*

$$A \rightarrow \alpha, \text{ with } A \in N \text{ and } \alpha \in (T \cup N)^*.$$

A language generated by a context-free grammar is said to be context-free.

Proposition

The set of context-free languages is a strict superset of the set of regular languages.

Proof: Each regular language is per definition context-free.
$L(a^n b^n)$ is context-free but not regular ($S \rightarrow aSb$, $S \rightarrow \epsilon$).

Examples of context-free languages

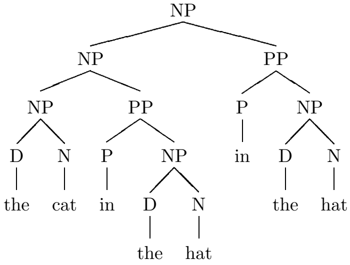
- $L_1 = \{ww^R : w \in \{a,b\}^*\}$
- $L_2 = \{a^i b^j : i \geq j\}$
- $L_3 = \{w \in \{a,b\}^* : \text{ more a's than b's}\}$
- $L_4 = \{w \in \{a,b\}^* : \text{ number of a's equals number of b's}\}$

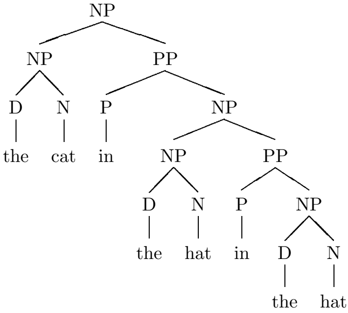
$$\left\{ \begin{array}{lllllllll} S & \rightarrow & aB & A & \rightarrow & a & B & \rightarrow & b \\ S & \rightarrow & bA & A & \rightarrow & aS & B & \rightarrow & bS \\ & & & A & \rightarrow & bAA & B & \rightarrow & aBB \end{array} \right\}$$

Example of an ambiguous grammar

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

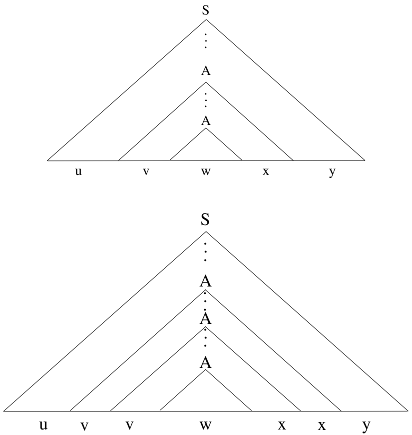
$$P = \left\{ \begin{array}{lllllllll} \mathrm{NP} & \rightarrow & \mathrm{D\ N} & \mathrm{D} & \rightarrow & \mathrm{the} & \mathrm{N} & \rightarrow & \mathrm{hat} \\ \mathrm{NP} & \rightarrow & \mathrm{NP\ PP} & \mathrm{N} & \rightarrow & \mathrm{cat} & \mathrm{P} & \rightarrow & \mathrm{in} \\ \mathrm{PP}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





A grammar is ambiguous if there exists a generated string with two derivation trees!

Pumping lemma: proof sketch



$|vwx| \leq p$, $vx \neq \epsilon$ and $uv^iwx^iy \in L$ for any $i \geq 0$.

Existence of non context-free languages

- $L_1 = \{a^n b^n c^n\}$
- $L_2 = \{a^n b^m c^n d^m\}$
- $L_1 = \{ww : w \in \{a, b\}^*\}$

Closure properties of context-free languages

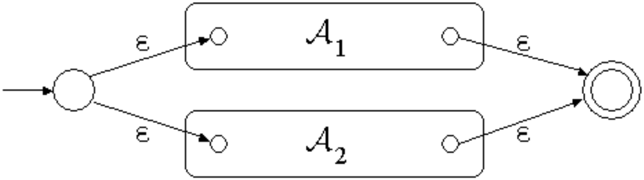
	Type3	Type2	Type1	Type0
union	+	+	+	+
intersection	+	-	+	+
complement	+	-	+	-
concatenation	+	+	+	+
Kleene’s star	+	+	+	+
intersection with a regular language	+	+	+	+

Context-free languages are closed under union

If $G_1 = (N_1, T_1, S_1, P_1)$ and $G_2 = (N_2, T_2, S_2, P_2)$ are two grammars,
then the set of productions of the grammar which generates $L(G_1) \cup L(G_2)$ is

$$P_1 \cup P_2 \cup \{S \to S_1, S \to S_2\}.$$

Remember (union for finite-state automata):



Pushdown automaton

- A push-down automaton is a finite state automaton enriched with an unrestricted stack.
- The stack is accessed: first-in-last-out.
- A separate stack alphabet is needed.
- In one transition step one can:
  - read an input symbol
  - remove one stack symbol from the stack (pop up)
  - push one word over the stack alphabet onto the stack (push down)
  - change to a new state

Acceptance through an PDA

A word is accepted by an PDA iff in the end:

- the word is totally read
- the stack is empty
- the PDA is in a final state

Transition

top symbol of the stack
(is removed)
- pop up -

new state

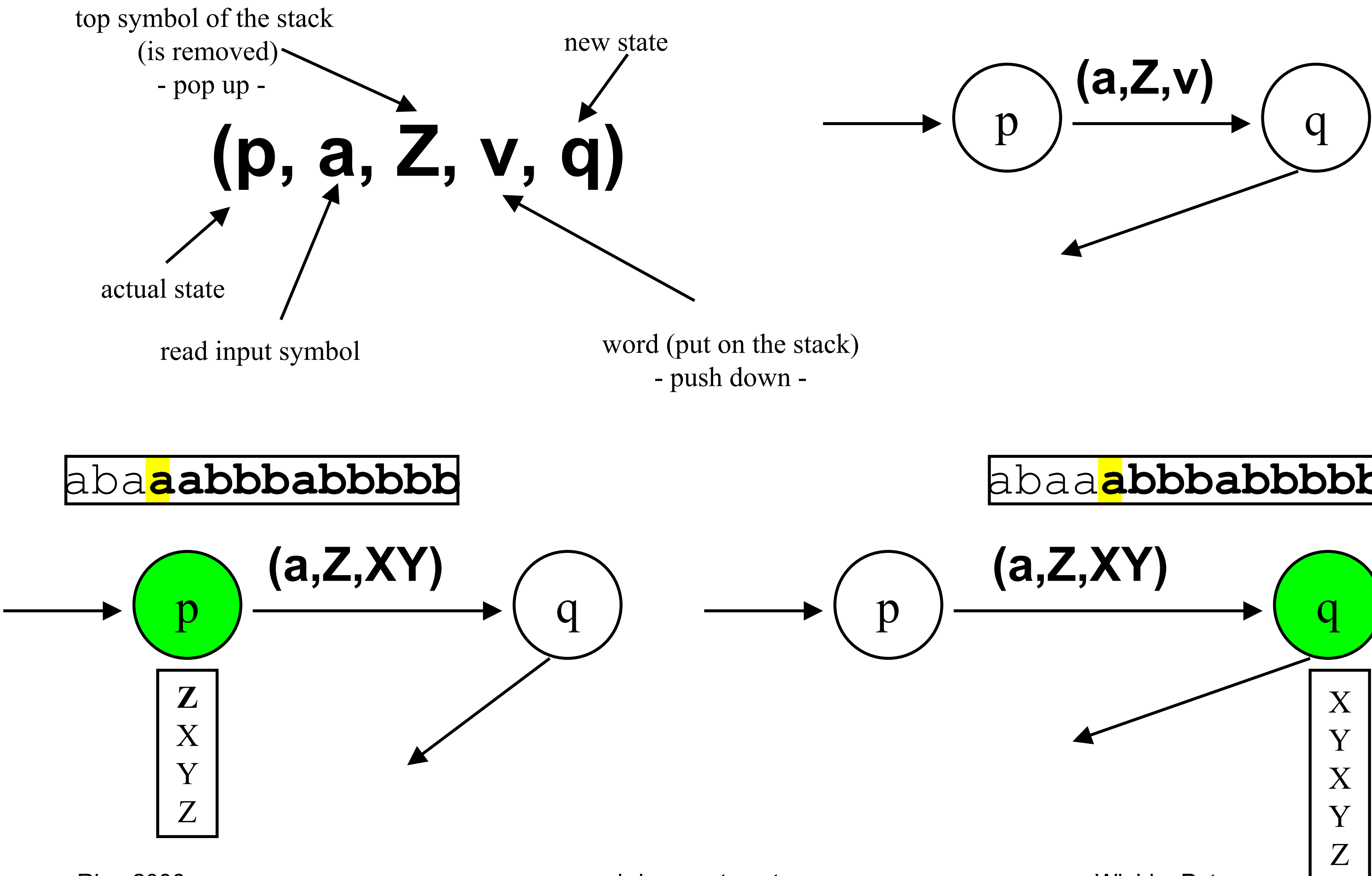
$$(p,\ a,\ Z,\ v,\ q)$$

actual state

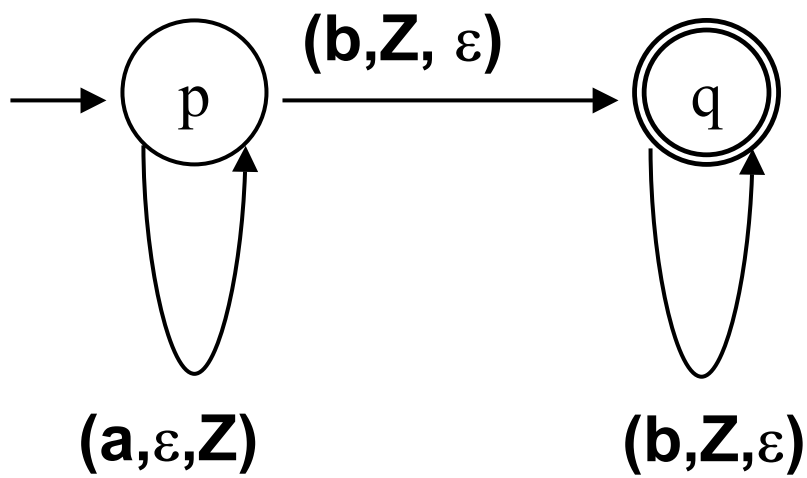
read input symbol

word (put on the stack)
- push down -

Transition


top symbol of the stack
(is removed)
- pop up -
new state
(p, a, Z, v, q)
actual state
read input symbol
word (put on the stack)
- push down -
(a,Z,v)
p
q
abaaabbbabbbbb
(a,Z,XY)
p
q
Z
X
Y
Z
abaaabbbabbbbb
(a,Z,XY)
p
q
X
Y
X
Y
Z

example PDA



this PDA accepts the language $L(a^n b^n)$

Chomsky-hierarchy

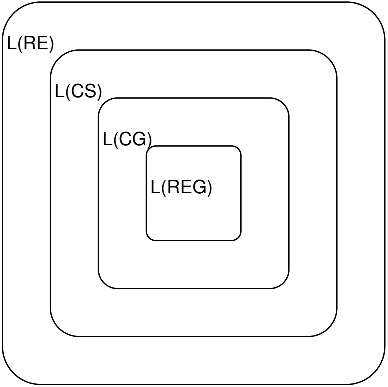
- The Chomsky-hierarchy is a hierarchy over the conditions on the rule structures of formal grammars.
- Linguists benefit from the rule-focussed definition of the Chomsky-hierarchy.

Chomsky-hierarchy (1956)

regular languages	Type 3, REG	$A \rightarrow bA$	$a^* b^*$
context-free languages	Type 2, CF	$A \rightarrow \beta$	$a^n b^n$, $w^R w$
context-sensitive languages	Type 1, CS	$\alpha A \nu \rightarrow \alpha \beta \nu$	$a^n b^n c^n$, ww
recursively enumerable languages	Type 0, RE	$\alpha \rightarrow \beta$	

Main theorem

$$L(REG) \subset L(CG) \subset L(CS) \subset L(RE)$$



decision problems

Given: grammars $G = (N, \Sigma, S, P)$, $G' = (N', \Sigma, S', P')$, and a word $w \in \Sigma^*$

word problem Is w derivable from G ?

emptiness problem Does G generate a nonempty language?

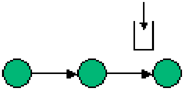
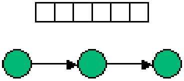
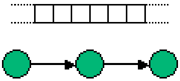
equivalence problem Do G and G' generate the same language ($L(G) = L(G')$)?

Results for the decision problems

	Type3	Type2	Type1	Type0
word problem	D	D	D	U
emptiness problem	D	D	U	U
equivalence problem	D	U	U	U

D: decidable; U: undecidable

Chomsky-hierarchy (1956)

Type 3: REG	finite state automaton		WP: linear
Type 2: CF	pushdown-automaton		WP: cubic
Type 1: CS	linearly restricted automaton		WP: exponential
Type 0: RE	Turing machine		WP: not decidable

Which is the class of natural languages?

Why is the formal complexity of natural languages interesting?

- It gives information about the general structure of natural language
- It allows to draw conclusions about the adequacy of grammar formalisms
- It determines a lower bound for the computational complexity of natural language processing tasks
- It allows to draw conclusions about human language processing

Which idealizations about NL are necessary?

1. The family of natural languages exists:
   - all natural languages are structurally similar
   - all natural languages have a similar generative capacity
2. Language = set of strings over an alphabet:
   - native speakers have full competence
   - consistent grammaticality judgements
3. $NL \subset RE$
   - each natural language is describable by a formal grammar (a finite rule system)
4. Each NL consists of an *infinite* set of strings

About the idealizations

The family of natural languages exists:

- all natural languages are structurally similar
- all natural languages have a similar generative capacity

Arguments:

- all NLs serve for the same tasks
- children can learn each NL as their native language (within a similar period of time)

$\Rightarrow$ No evidence for a principal structural difference

About the idealizations (cont.)

Language = sets of strings over an alphabet:

- native speakers have full competence
- consistent grammaticality judgements

Arguments:

- all mistakes are due to performance not to competence
- Mathews (1979) counter examples:
  - The canoe floated down the river sank.
  - The editor authors the newspaper hired liked laughed.
  - The man (that was) thrown down the stairs died.
  - The editor (whom) the authors the newspaper hired liked laughed.

About the idealizations (cont.)

$NL \subset RE$:

- each natural language is describable by a formal grammar (a finite rule system)

Arguments:

Rogers (1967)

- Laws of nature are universal
- Church's thesis is universal
- human oracle + Church's thesis $\Rightarrow$ NL is RE

About the idealizations (cont.)

Each NL consists of an *infinite* set of strings

Arguments:

- Recursion in NL:
  - john likes peter
  - john likes peter and mary
  - john likes peter and mary and sue
  - john likes peter and mary and sue and otto and ...
- (Donaudampfschiffskapitänsmützenschirm ...)

Are natural languages regular?

Chomsky (1957):

- “English is not a regular language”
- context-free languages: “I do not know whether or not English is itself literally outside the range of such analysis”

Are natural languages regular?

- a woman hired another woman
- a woman whom another woman hired hired another woman
- a woman whom another woman whom another woman hired hired hired another woman
- a woman whom another woman whom another woman whom another woman hired hired hired hired another woman
- ...
-
- a woman whom (another woman)^n (hired)^n hired another woman ($n > 0$)

Natural languages are not regular

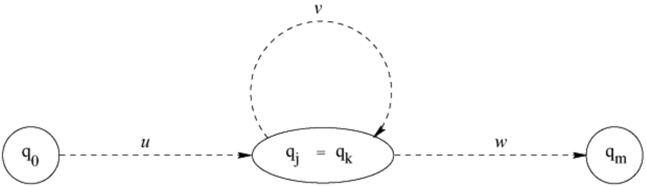
- Let $x =$ "another woman", $y =$ "hired", $w =$ "a woman", and $v =$ "hired another woman".
- wx^*y^*v is a regular language
- $\textsc{English} \cap wx^*y^*v = wx^ny^nv$.
- if $\textsc{English}$ is regular, then wx^ny^nv has to be regular, too (REG is closed under intersection)
- contradiction to the pumping lemma

Pumping lemma for regular languages (cont.)

Lemma (Pumping-Lemma)

If L is an infinite regular language over Σ*, then there exists words* $u, v, w \in \Sigma^*$ *such that* $v \neq \epsilon$ *and* $uv^i w \in L$ *for any* $i \geq 0$.

proof sketch:



Kornai (1985): NL are regular

Self-embedding (nested) structures in NL are not iterative!

> This is the woman whom the man whom the girl whom the boy whom the teacher whom the doctor admired met called chased liked

NL $\subseteq$ CF? wrong arguments

- inadmissible induction
  - no known CFG describes Englisch adequately, thus no adequate description with CFG's exists

An Introduction to the Principles of Transformational Syntax (Akmajian & Heny, 1975):

(description of auxiliary-initial interrogatives) "Since **there seems to be no way** of using such PS rules to represent an obviously signigicant generalization about one language, namely, English, we can be sure that phrase structure grammars cannot possibly represent all the significant aspects of language structure."

NL $\subseteq$ CF? wrong arguments

- "context-freeness" intuitively understood

- the girl sees the dog

  the girls see the dog
- the girl who climbed the tree which was planted last year when it rained so much sees the dog

  the girls who climbed the tree which was planted last year when it rained so much see the dog

NL $\subseteq$ CF? wrong arguments

Transformational grammar (Grinder & Elgin, 1973):

- the defining characteristic of a context-free rule is that the symbol to be rewritten **is to be rewritten without reference to the context** in which it occurs ... Thus by definition, one cannot write a context-free rule that will expand the symbol **V** into *kiss* in the context of being immediately preceded by the sequence *the girls* and that will expand the symbol **V** into *kisses* in the context of being immediately preceded by the sequence *the girl*.

NL $\subseteq$ CF? wrong arguments

A realistic transformational grammar (Bresnan, 1987):

- "in many cases the number of a verb agrees with that of a noun phrase at some **distance** from it ... this type of syntactic dependency can extend as memory or patience permits ...

  the distant type of agreement ... **cannot be** adequately **described** even **by context-sensitive** phrase-structure rules, for **the possible context is not correctly describable as a finite string of phrases**."

Gazdar & Pullum (1982 & 1985)

- thesis: all published arguments for the non-context-freeness of NL are not compelling
  1. folklore
  2. wrong data
  3. formal mistakes
- 30 years of fruitless search for a non-context-free language
- human seem able to parse sentences in linear time

Are natural languages context-free?

embedding of subordinate clauses in Swiss-German

- mer d'chind em Hans es huus lönd hälfe aastriiche
  we the childs-ACC the Hans-DAT the house-ACC let help paint
  $NP_1\ NP_2\ NP_3\ VP_1\ VP_2\ VP_3$ "cross serial dependencies"
- *mer d'chind de Hans es huus lönd hälfe aastriiche
  we the children-ACC Hans-CC the house-ACC let help paint

embedding of subordinate clauses in German

- er die Kinder dem Hans das Haus streichen helfen ließ
  he the children the Hans the house paint help let
  $NP_1\ NP_2\ NP_3\ VP_3\ VP_2\ VP_1$ "nested dependencies"

NL $\not\subset$ CF: Proof Shieber 1985

Homomorphism:	f("laa") = c	f("es huus haend wele") = x
f("d'chind") = a	f("hälfe") = d	f("Jan säit das mer") = w
f("em Hans") = b	f("aastriiche") = y	f(s) = z otherwise

- f(Swiss-German) $\cap$ $wa^*b^*xc^*d^*y = wa^mb^nxc^md^ny$
- $wa^mb^nxc^md^ny$ is not context-free ($\rightarrow$pumping lemma)
- $wa^*b^*xc^*d^*y$ is regular
- context-free languages are closed unter
  - homomorphisms
  - intersection with regular languages
- ➢ Swiss-German is note context-free

potential attack points of the proof

- **wrong data**
  - grammaticality judgements
- **case is not a syntactic phenomenon**
  - case is determined by semantics (unterstützen/helfen)
- **the length of the sentences is restricted**
  - Shieber: "Down this path lies tyranny. Acceptance of this argument opens the way to proofs of natural languages as regular, nay, finite. The linguist proposing this counterargument to salvage the context-freeness of natural language may have won the battle, but has certainly lost the war.

mildly context-sensitive languages (MCSL)

mildly context-sensitive languages = subset of the context-sensitive languages

- restricted grow:
  there is a k such that for all $w \in L$ there is a $w' \in L$ with $|w'| \leq |w| + k$
- word problem is decidable in polynomial time
- a MCSL contains the following non-context-free languages:
  - $L_1 = \{a^n b^n c^n \mid n \geq 0\}$ (multiple agreement),
  - $L_2 = \{a^n b^m c^n d^m \mid m,n \geq 0\}$ (crossed dependencies),
  - $L_3 = \{ww \mid w \in \{a,b\}^*\}$ (duplication)

RL $\subset$ CFL $\subset$ MCSL $\subset$ CSL $\subset$ RE

L	CFL	MCSL	CSL
$a^n b^n$	√	√	√
$a^n b^n c^n$, ww	–	√	√
a^{2^n}	–	–	√

Thesis: natural languages are mildly context-sensitive

restricted formalisms

first approach: extend CFG's

– transformation grammar: CFG + transformations
– HPSG: CFG-basis + feature structures

not restricted!

second approach: replace CFG's

– Tree Adjoining Grammar (TAG)
  tree rewriting instead of ***string rewriting***

Conclusion

- finite automaton are very useful in practical applications:
  - Phonologie
  - Morphologie
  - ...
- human parse very fast => low complexity class
- learnability of NL has to be explained

weak and strong generative capacity

- The **weak generative capacity** of a linguistic formalism is the ability to generate all grammatical sentences of a language.
- The **strong generative capaity** of a linguistic formalism is the ability to assign to all grammatical sentences *their structure*
- CFG's ????

Literature

Beesley & Karttunen (2003) *Finite State Morphology*. CSLI.

Hopcroft, Motwani & Ullman (2001) *Introduction to Automata and Language Theory*. Addison-Wesley, 2nd edition.

Partee, ter Meulen & Wall (1990) *Mathematical Methods in Linguistics*. Kluwer Academic Publishers.

Sipser (2005) *Introduction to the Theory of Computation*. Thomson Course Technology, 2nd edition.

Sudkamp (1996) *Languages and Machines: An Introduction to the Theory of Computer Science*. Addison Wesley, 2nd edition.

Literatur (1)

- **Bach**, Emmon und **Marsh**, William 1987: *An Elementary Proof of the Peters-Ritchie Theorem*. In Savitch et al. 1987, 41-55.
- **Chomsky**, Noam 1956: *Three models for the description of language.* In IRE Transactions on Information Theory 2(3), 113-124.
- **Chomsky**, Noam 1965: *Aspects of a Theory of Syntax*. MIT Press, Cambridge, Massachusetts.
- **Gazdar**, Gerald und **Pullum**, Geoffrey K. 1987[1985]: *Computationally Relevant Properties of Natural Languages and Their Grammars.* In Savitch et al. 1987, 41-55. (1985 erschienen in Technical Report CSLI-85-24.)
- **Joshi**, Aravind K. and **Schabes**, Yves 1997: *Tree-Adjoining Grammars*. In Handbook of Formal Languages, G. Rozenberg and A. Salomaa (Hrg.), Vol. 3, Springer, Berlin, New York, 1997, 69 - 124.
- **Kornai**, Andras 1985: *Natural languages and the Chomsky hierarchy.* In M. King (Hrg.): Proceedings of the 2nd European Conference of the Association for Computational Linguistics (1985),1-7.

Literatur (2)

- **Matthews**, Robert J. 1979: *Are the Grammatical Sentences of a Language a Recursive Set?* In Synthese 40, 209--224.
- **Pullum**, Geoffrey K. und **Gazdar**, Gerald 1987 [1982]: *Natural Languages and Context-Free Languages.* In Savitch et al. 1987, 138-183. (1982 erschienen in Linguistics and Philosophy, 4:471--504.)
- **Rogers**, Hartley 1967: *Theory of Recursive Functions and Effective Computability*. McGraw-Hill Book Company, New York, 1967.
- **Savitch** et. al. 1987: *The Formal Complexity of Natural Language.* Reidel, Dordrecht.
- **Shieber**, Stuart M. 1987 [1985]. *Evidence against the context-freeness of natural language*. In Savitch et al. 1987, 320-335. (1985 erschienen in Linguistics and Philosophy, 8:333--343.)