Mathematics for linguists

Gerhard Jäger
gerhard.jaeger@uni-tuebingen.de

Uni Tübingen, WS 2009/2010

December 1, 2009

Context-free languages and pushdown automata

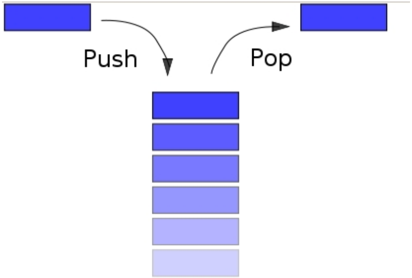
- **Context-free grammars (type-2 grammars):** All rules have the form

$$A \to \gamma$$

  where A is a non-terminal symbol and γ is a string consisting of non-terminal and terminal symbols.
- **Context-free languages:** Languages that are generated by a type-2 grammar.
- Every regular language is context-free.
- Examples for context-free languages (that are not regular):
  - $\{a^n b^n \mid n \geq 0\}$
  - $\{a^n b^{2n} \mid n \geq 0\}$
  - $\{\vec{w}\vec{w}^R \mid \vec{w} \in \{a, b\}^*\}$ (palindrome language)

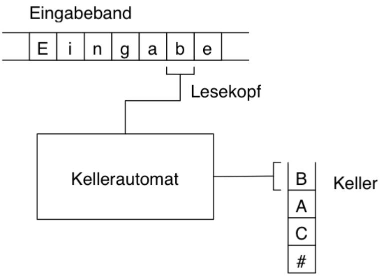
Context-free languages and pushdown automata

- **Pushdown automaton:** finite automaton with a *stack*
- **Stack:**
  - orders symbol in a linear sequence
  - manipulation according to the principle *last in—first out*



Context-free languages and pushdown automata

- in initial state the stack is empty
- in each transition: remove at most one item from the stack and add a finite number of items
- an input string is accepted if
  - after processing the string, the automaton is in a final state, and
  - the stack is empty.



Context-free languages and pushdown automata

- example for a pushdown automaton that recognizes the language $\{a^n b^n \mid n \geq 0\}$:

states:	K	$=$	$\{q_0, q_1\}$
input alphabet:	Σ	$=$	$\{a, b\}$
stack alphabet:	Γ	$=$	$\{A\}$
initial state:	q_0		
final states:	F	$=$	$\{q_0, q_1\}$
state transitions:	Δ	$=$	$\left\{ \begin{array}{l} (q_0, a, \epsilon) \rightarrow (q_0, A) \\ (q_0, b, A) \rightarrow (q_1, \epsilon) \\ (q_1, b, A) \rightarrow (q_1, \epsilon) \end{array} \right\}$

Context-free languages and pushdown automata

Theorem

Every pushdown automaton accepts a context-free language, and every context-free language is accepted by a pushdown automaton.

Pumping lemma for context-free languages

- If a string $\vec{x}$ is generated by a context-free grammar G, there is a "syntax tree" for $\vec{x}$ that only uses rules from G.
- There are finitely many rules in G. Let r be the number of rules in G.
- Every rule from G has a certain number of symbols on its right-hand side. Let s be the maximal number of symbols on the right-hand side of a rule.

Pumping lemma for context-free languages

- Suppose
  - $\vec{x}$ is generated by G,
  - T is the syntax tree for $\vec{x}$, and
  - there is no non-terminal symbol that dominates itself in T.

  Then it holds that:
  - there are at most s^r branches in T.
  - Hence there are at most $r \cdot s^r$ many rule applications in the derivation of $\vec{x}$. ["$\cdot$" means multiplication in $\mathbb{N}$ here.]
  - In every rule application, at most s terminal symbols are generated.
  - Hence the length of $\vec{x}$ is at most $s \cdot r \cdot s^r$.

Pumping lemma for context-free languages

If $L(G)$ is infinite, it contains strings that are longer than $s \cdot r \cdot s^r$.
The corresponding syntax tree then contains at least one non-terminal symbol that dominates itself. To be more precise: there are two nodes α und β that are labeled with the same non-terminal symbol, such that β is dominated by α.
This leads to the following results:

Theorem (Pumping lemma for context-free languages)

Let L be an infinite context-free language. Then there is a number n such that all words $\vec{x} \in L$ can be decomposed into $\vec{x} = \vec{u} \cdot \vec{v} \cdot \vec{w} \cdot \vec{y} \cdot \vec{z}$, such that

- $l(\vec{v}) + l(\vec{y}) > 0$,
- $l(\vec{v}) + l(\vec{w}) + l(\vec{y}) \leq n$, *und*
- *for all* $i \in \mathbb{N} : \vec{u} \cdot \vec{v}^i \cdot \vec{w} \cdot \vec{y}^i \cdot \vec{z} \in L$.

NL and the Chomsky hierarchy

The respectively argument

- Bar-Hillel and Shamir (1960):
  - English contains copy language
  - cannot be context-free
- Consider the sentence

  John, Mary, David, ... are a widower, a widow, a widower, ..., respectively.
- Claim: the sentence is only grammatical under the condition that if the nth name is male (female) then the nth phrase after the copula is *a widower (a widow)*

NL and the Chomsky hierarchy

- suppose the claim is true
- intersect English with regular language

$$L_1 = (\textit{Paul}|\textit{Paula})^+ \textit{are}(\textit{a widower}|\textit{a widow})^+ \textit{respectively}$$

English $\cap L_1 = L_2$

- homomorphism $L_2 \rightsquigarrow L_3$:

$$\begin{aligned} \textit{John, David, Paul, ...} &\mapsto a \\ \textit{Mary, Paula, Betty, ...} &\mapsto b \\ \textit{a widower} &\mapsto a \\ \textit{a widow} &\mapsto b \\ \textit{are, respectively} &\mapsto \epsilon \end{aligned}$$

NL and the Chomsky hierarchy

- result: copy language L_3

$$\{\vec{w}\vec{w}|\vec{w} \in (a|b)^+\}$$

- copy language is not context-free due to pumping lemma (exercise: why is this so?)
- hence L_2 is not context-free
- hence English is not context-free

NL and the Chomsky hierarchy

Counterargument

- crossing dependencies triggered by *respectively* are semantic rather than syntactic
- compare above example to

(Here are John, Mary and David.) They are a widower, a widow and a widower, respectively.

Cross-serial dependencies in Dutch

- Huybregts (1976):
  - Dutch has copy language like structures
  - thus Dutch is not context-free

(1) dat Jan Marie Pieter Arabisch laat zien schrijven
THAT JAN MARIE PIETER ARABIC LET SEE WRITE
'that Jan let Marie see Pieter write Arabic'

NL and the Chomsky hierarchy

Counterargument

- crossing dependencies only concern argument linking, i.e. semantics
- Dutch has no case distinctions
- as far as plain string are concerned, the relevant fragment of Dutch has the structure

$$NP^n V^n$$

which is context-free

Are natural languages context-free?

- definitiv argument (Huybregts 1984, Shieber 1985): Swiss German is not context-free
- crucial insight:
  - Context-free grammars can describe arbitrarily deeply **nested** dependencies.
  - Context-free grammars cannot describe arbitrarily long **crossing** dependencies.
  - In natural languages, we do find — marginally, but still — crossing dependencies.

Are natural languages context-free?

- Type-1 grammars ("context-sensitive grammars") are, in the general case, too "powerful" for linguistic purposes
- **mildly context-sensitive grammars:** family of grammar formalisms that are only slightly more powerful than type-2 grammars, but are able to express crossing dependencies
- most important representatives:
  - *Tree Adjoining Grammars/TAG*
  - *Combinatory Categorial Grammar/CCG)*