

Theory of Computer Science

C6. Context-free Languages: Closure & Decidability

Gabriele Röger

University of Basel

April 3, 2019

Theory of Computer Science

April 3, 2019 — C6. Context-free Languages: Closure & Decidability

C6.1 Pumping Lemma

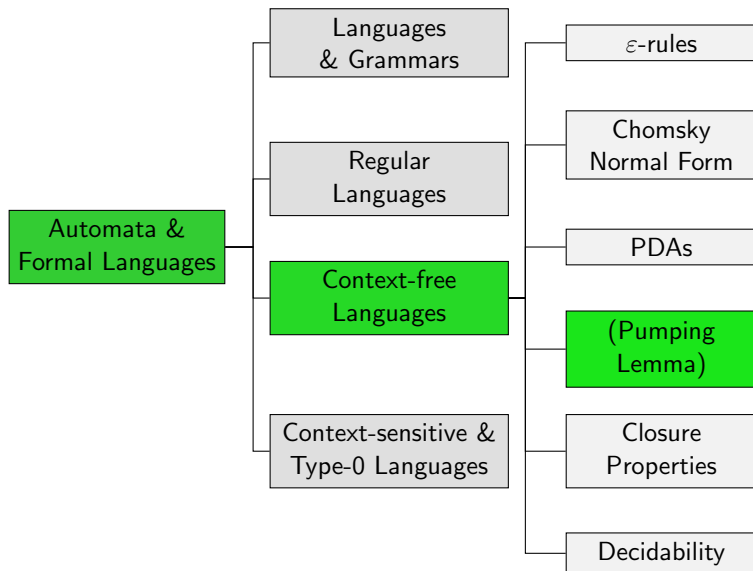
C6.2 Closure Properties

C6.3 Decidability

C6.4 Summary

C6.1 Pumping Lemma

Overview



Pumping Lemma for Context-free Languages



We used the pumping lemma from chapter C3 to show that a language is not regular. Is there a similar lemma for **context-free** languages?

Yes!

Picture courtesy of [imagerymajestic](#) / [FreeDigitalPhotos.net](#)

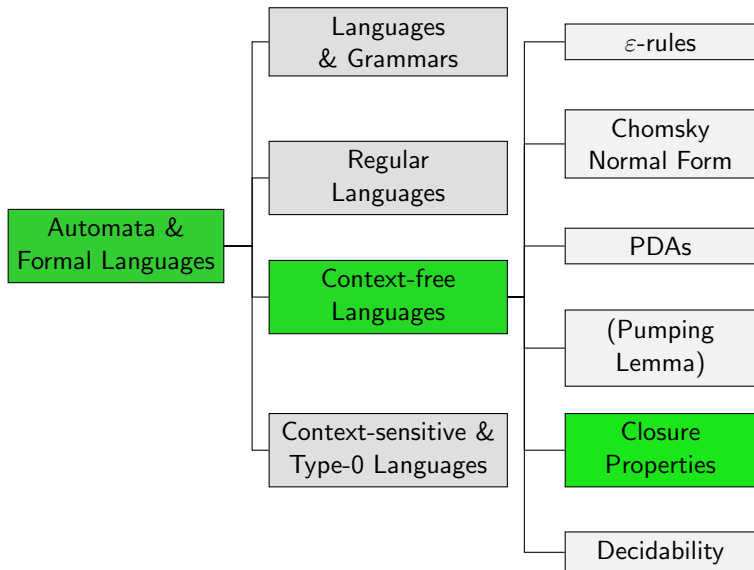
Pumping Lemma for Context-free Languages

Pumping lemma for context-free languages:

- ▶ It is possible to prove a variant of the **pumping lemma** for context-free languages.
- ▶ Pumping is more complex than for regular languages:
 - ▶ word is decomposed into the form $uvwx$
with $|vx| \geq 1$, $|vwx| \leq n$
 - ▶ pumped words have the form uv^iwx^iy
- ▶ This allows us to prove that certain languages are **not context-free**.
- ▶ **example:** $\{a^n b^n c^n \mid n \geq 1\}$ is not context-free
(we will later use this without proof)

C6.2 Closure Properties

Overview



Closure under Union, Product, Star

Theorem

The context-free languages are closed under:

- ▶ *union*
- ▶ *product*
- ▶ *star*

Closure under Union, Product, Star: Proof

Proof.

Closed under union:

Let $G_1 = \langle \Sigma_1, V_1, P_1, S_1 \rangle$ and $G_2 = \langle \Sigma_2, V_2, P_2, S_2 \rangle$
be context-free grammars. W.l.o.g., $V_1 \cap V_2 = \emptyset$.

Then $\langle \Sigma_1 \cup \Sigma_2, V_1 \cup V_2 \cup \{S\}, P_1 \cup P_2 \cup \{S \rightarrow S_1, S \rightarrow S_2\}, S \rangle$
(where $S \notin V_1 \cup V_2$) is a context-free grammar for $\mathcal{L}(G_1) \cup \mathcal{L}(G_2)$
(possibly requires rewriting ε -rules). ...

Closure under Union, Product, Star: Proof

Proof (continued).

Closed under product:

Let $G_1 = \langle \Sigma_1, V_1, P_1, S_1 \rangle$ and $G_2 = \langle \Sigma_2, V_2, P_2, S_2 \rangle$
be context-free grammars. W.l.o.g., $V_1 \cap V_2 = \emptyset$.

Then $\langle \Sigma, V_1 \cup V_2 \cup \{S\}, P_1 \cup P_2 \cup \{S \rightarrow S_1 S_2\}, S \rangle$
(where $S \notin V_1 \cup V_2$) is a context-free grammar for $\mathcal{L}(G_1)\mathcal{L}(G_2)$
(possibly requires rewriting ε -rules). ...

Closure under Union, Product, Star: Proof

Proof (continued).

Closed under star:

Let $G = \langle \Sigma, V, P, S \rangle$ be a context-free grammar where w.l.o.g. S never occurs on the right-hand side of a rule.

Then $G = \langle \Sigma, V \cup \{S'\}, P', S' \rangle$ with $S' \notin V$ and $P' = (P \cup \{S' \rightarrow \varepsilon, S' \rightarrow S, S' \rightarrow SS'\}) \setminus \{S \rightarrow \varepsilon\}$ is a context-free grammar for $\mathcal{L}(G)^*$ after rewriting ε -rules. □

No Closure under Intersection or Complement

Theorem

The context-free languages are not closed under:

- ▶ *intersection*
- ▶ *complement*

No Closure under Intersection or Complement: Proof

Proof.

Not closed under intersection:

The languages $L_1 = \{a^i b^j c^j \mid i, j \geq 1\}$
and $L_2 = \{a^i b^j c^i \mid i, j \geq 1\}$ are context-free.

- ▶ For example, $G_1 = \langle \{a, b, c\}, \{S, A, X\}, P, S \rangle$ with $P = \{S \rightarrow AX, A \rightarrow a, A \rightarrow aA, X \rightarrow bc, X \rightarrow bXc\}$ is a context-free grammar for L_1 .
- ▶ For example, $G_2 = \langle \{a, b, c\}, \{S, B\}, P, S \rangle$ with $P = \{S \rightarrow aSc, S \rightarrow B, B \rightarrow b, B \rightarrow bB\}$ is a context-free grammar for L_2 .

Their intersection is $L_1 \cap L_2 = \{a^n b^n c^n \mid n \geq 1\}$.

We have remarked before that this language is not context-free.

...

No Closure under Intersection or Complement: Proof

Proof (continued).

Not closed under complement:

By contradiction: assume they were closed under complement.

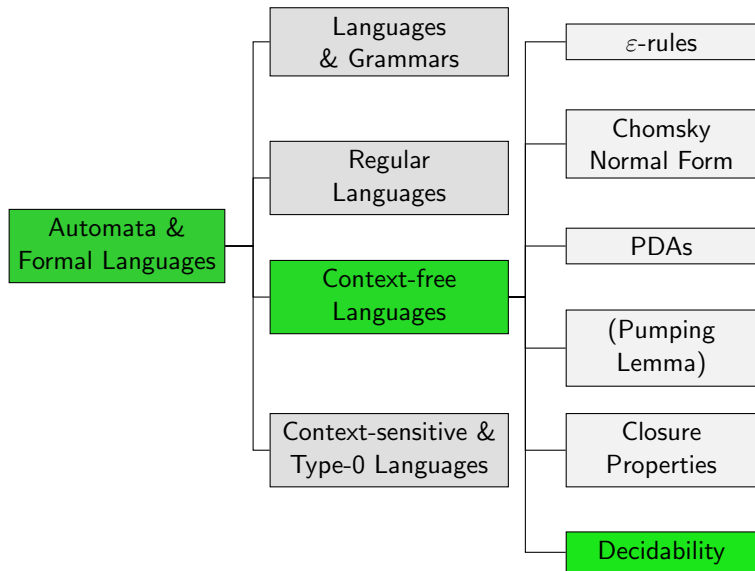
Then they would also be closed under intersection because they are closed under union and

$$L_1 \cap L_2 = \overline{\overline{L_1} \cup \overline{L_2}}.$$

This is a contradiction because we showed that they are not closed under intersection. □

C6.3 Decidability

Overview



Word Problem

Definition (Word Problem for Context-free Languages)

The **word problem** P_{\in} for context-free languages is:

Given: context-free grammar G with alphabet Σ
and word $w \in \Sigma^*$

Question: Is $w \in \mathcal{L}(G)$?

Decidability: Word Problem

Theorem

The word problem P_{\in} for context-free languages is *decidable*.

Proof.

If $w = \varepsilon$, then $w \in \mathcal{L}(G)$ iff $S \rightarrow \varepsilon$ with start variable S is a rule of G .

Since for all other rules $w_l \rightarrow w_r$ of G we have $|w_l| \leq |w_r|$, the intermediate results when deriving a non-empty word never get shorter.

So it is possible to systematically consider all (finitely many) derivations of words up to length $|w|$ and test whether they derive the word w . □

Note: This is a terribly inefficient algorithm.

Emptiness Problem

Definition (Emptiness Problem for Context-free Languages)

The **emptiness problem** P_{\emptyset} for context-free languages is:

Given: context-free grammar G

Question: Is $\mathcal{L}(G) = \emptyset$?

Decidability: Emptiness Problem

Theorem

*The emptiness problem for context-free languages is **decidable**.*

Proof.

Given a grammar G , determine all variables in G that allow deriving words that only consist of terminal symbols:

- ▶ First mark all variables A for which a rule $A \rightarrow w$ exists such that w only consists of terminal symbols.
- ▶ Then mark all variables A for which a rule $A \rightarrow w$ exists such that all nonterminal systems in w are already marked.
- ▶ Repeat this process until no further markings are possible.

$\mathcal{L}(G)$ is empty iff the start variable is unmarked at the end of this process. □

Finiteness Problem

Definition (Finiteness Problem for Context-free Languages)

The **finiteness problem** P_∞ for context-free languages is:

Given: context-free grammar G

Question: Is $|\mathcal{L}(G)| < \infty$?

Decidability: Finiteness Problem

Theorem

*The finiteness problem for context-free languages is **decidable**.*

We omit the proof. A possible proof uses the pumping lemma for context-free languages.

Proof sketch:

- ▶ We can compute certain bounds $l, u \in \mathbb{N}_0$ for a given context-free grammar G such that $\mathcal{L}(G)$ is infinite iff there exists $w \in \mathcal{L}(G)$ with $l \leq |w| \leq u$.
- ▶ Hence we can decide finiteness by testing all (finitely many) such words by using an algorithm for the word problem.

Intersection Problem

Definition (Intersection Problem for Context-free Languages)

The **intersection problem** P_{\cap} for context-free languages is:

Given: context-free grammars G and G'

Question: Is $\mathcal{L}(G) \cap \mathcal{L}(G') = \emptyset$?

Equivalence Problem

Definition (Equivalence Problem for Context-free Languages)

The **equivalence problem** $P_{=}$ for context-free languages is:

Given: context-free grammars G and G'

Question: Is $\mathcal{L}(G) = \mathcal{L}(G')$?

Undecidability: Equivalence and Intersection Problem

Theorem

*The equivalence problem for context-free languages and the intersection problem for context-free languages are **not decidable**.*

We cannot show this with the means currently available, but we will get back to this in Part D (computability theory).

C6.4 Summary

Summary

- ▶ The context-free languages are **closed** under **union**, **product** and **star**.
- ▶ The context-free languages are **not closed** under **intersection** or **complement**.
- ▶ The **word** problem, **emptiness** problem and **finiteness** problem for the class of context-free languages are **decidable**.
- ▶ The **equivalence** problem and **intersection problem** for the class of context-free languages are **not decidable**.

Further Topics on Context-free Languages and PDAs

- ▶ With the **CYK-algorithm** (Cocke, Younger and Kasami) it is possible to decide $w \in \mathcal{L}(G)$ in time $O(|w|^3)$ for a grammar in Chomsky normal form and a word w .
- ▶ **Deterministic push-down automata** have the restriction $|\delta(q, a, A)| + |\delta(q, \varepsilon, A)| \leq 1$ for all $q \in Q, a \in \Sigma, A \in \Gamma$. They accept with **end states** rather than empty stack.
- ▶ The languages accepted by deterministic PDAs are called **deterministic context-free languages**. They form a strict superset of the regular languages and a strict subset of the context-free languages.