Theory of Computer Science

B10. Context-free Languages: Closure & Decidability

Gabriele Röger

University of Basel

April 3, 2024

Theory of Computer Science
April 3, 2024 — B10. Context-free Languages: Closure & Decidability

B10.1 Pumping Lemma

Pumping Lemma for Context-free Languages

We used the pumping lemma from chapter B7 to show that a language is not regular. Is there a similar lemma for context-free languages?

Yes!

Picture courtesy of imagerymajestic / FreeDigitalPhotos.net

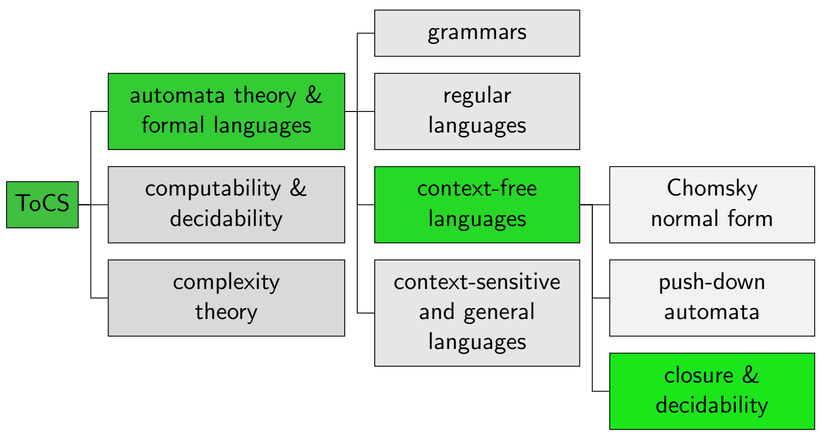
Pumping Lemma for Context-free Languages

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 $uvwxy$ with $|vx| \geq 1$, $|vwx| \leq p$
  - pumped words have the form uv^iwx^iy
- This allows us to prove that certain languages are not context-free.
- example: $\{a^nb^nc^n \mid n \geq 1\}$ is not context-free (we will later use this without proof)

B10.2 Closure Properties

Content of the Course



Closure under Union, Concatenation, Star

Theorem

The context-free languages are closed under:

- *union*
- *concatenation*
- *star*

Closure under Union, Concatenation, Star: Proof

Proof.

Closed under union:

Let $G_1 = \langle V_1, \Sigma_1, R_1, S_1 \rangle$ and $G_2 = \langle V_2, \Sigma_2, R_2, S_2 \rangle$ be context-free grammars. W.l.o.g., $V_1 \cap V_2 = \emptyset$.

Then $\langle V_1 \cup V_2 \cup \{S\}, \Sigma_1 \cup \Sigma_2, R_1 \cup R_2 \cup \{S \to S_1, S \to S_2\}, S \rangle$ (where $S \notin V_1 \cup V_2$) is a context-free grammar for $\mathcal{L}(G_1) \cup \mathcal{L}(G_2)$.

...

Closure under Union, Concatenation, Star: Proof

Proof (continued).

Closed under concatenation:

Let $G_1 = \langle V_1, \Sigma_1, R_1, S_1 \rangle$ and $G_2 = \langle V_2, \Sigma_2, R_2, S_2 \rangle$ be context-free grammars. W.l.o.g., $V_1 \cap V_2 = \emptyset$.

Then $\langle V_1 \cup V_2 \cup \{S\}, \Sigma_1 \cup \Sigma_2, R_1 \cup R_2 \cup \{S \to S_1 S_2\}, S \rangle$ (where $S \notin V_1 \cup V_2$) is a context-free grammar for $\mathcal{L}(G_1)\mathcal{L}(G_2)$.

...

Closure under Union, Concatenation, Star: Proof

Proof (continued).

Closed under star:

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

Then $G' = \langle V \cup \{S'\}, \Sigma, R', S' \rangle$ with $S' \notin V$ and $R' = R \cup \{S' \to \varepsilon, S' \to S, S' \to SS'\}$ is a context-free grammar for $\mathcal{L}(G)^*$. □

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

B10.3 Decidability

Word Problem

Definition (Word Problem for Context-free Languages)
The word problem $\mathrm{P}_\in$ 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 $\mathrm{P}_\in$ for context-free languages is decidable.

Proof.

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

Since for all other rules $w_l \to w_r$ 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 $\mathrm{P}_\emptyset$ 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 $A \to w$ exists such that w only consists of terminal symbols or $w = \varepsilon$.
- Then mark all variables A for which a rule $A \to w$ 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 $\mathrm{P}_\cap$ 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 $\mathrm{P}_=$ 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 C (computability theory).

B10.4 Summary

Summary

- The context-free languages are closed under union, concatenation 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 $q \in Q, a \in \Sigma, A \in \Gamma$.
- The languages recognized 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.