Theory of Computer Science

C5. Context-free Languages: Normal Forms, Closure, Decidability

Malte Helmert

University of Basel

April 3, 2017

Agenda for this Chapter

- repetition of context-free grammars
- ε-rules
- normal form for context-free grammars
- closure properties
- decidability

Context-free Grammars and ε-Rules

Repetition: Context-free Grammars

Definition (Context-free Grammar)

A context-free grammar is a 4-tuple $\langle \Sigma, V, P, S \rangle$ with

1. Σ finite alphabet of terminal symbols,
2. V finite set of variables (with $V \cap \Sigma = \emptyset$),
3. $P \subseteq (V \times (V \cup \Sigma)^+) \cup \{\langle S, \varepsilon \rangle\}$ finite set of rules,
4. If $S \to \varepsilon \in P$, then all other rules in $V \times ((V \setminus \{S\}) \cup \Sigma)^+$.
5. $S \in V$ start variable.

Repetition: Context-free Grammars

Definition (Context-free Grammar)

A context-free grammar is a 4-tuple $\langle \Sigma, V, P, S \rangle$ with

1. Σ finite alphabet of terminal symbols,
2. V finite set of variables (with $V \cap \Sigma = \emptyset$),
3. $P \subseteq (V \times (V \cup \Sigma)^+) \cup \{\langle S, \varepsilon \rangle\}$ finite set of rules,
4. If $S \to \varepsilon \in P$, then all other rules in $V \times ((V \setminus \{S\}) \cup \Sigma)^+$.
5. $S \in V$ start variable.

Rule $X \to \varepsilon$ is only allowed if $X = S$
and S never occurs on a right-hand side.

Repetition: Context-free Grammars

Definition (Context-free Grammar)

A context-free grammar is a 4-tuple $\langle \Sigma, V, P, S \rangle$ with

1. Σ finite alphabet of terminal symbols,
2. V finite set of variables (with $V \cap \Sigma = \emptyset$),
3. $P \subseteq (V \times (V \cup \Sigma)^+) \cup \{\langle S, \varepsilon \rangle\}$ finite set of rules,
4. If $S \to \varepsilon \in P$, then all other rules in $V \times ((V \setminus \{S\}) \cup \Sigma)^+$.
5. $S \in V$ start variable.

Rule $X \to \varepsilon$ is only allowed if $X = S$
and S never occurs on a right-hand side.

With regular grammars, this restriction could be lifted.
How about context-free grammars?

ε-Rules

Theorem

For every grammar G with rules $P \subseteq V \times (V \cup \Sigma)^*$ there is a context-free grammar G' with $\mathcal{L}(G) = \mathcal{L}(G')$.

ε-Rules

Theorem

For every grammar G with rules $P \subseteq V \times (V \cup \Sigma)^*$ there is a context-free grammar G' with $\mathcal{L}(G) = \mathcal{L}(G')$.

Proof.

Let $G = \langle \Sigma, V, P, S \rangle$ be a grammar with $P \subseteq V \times (V \cup \Sigma)^*$.

Let $V_\varepsilon = \{A \in V \mid A \Rightarrow^* \varepsilon\}$. We can find this set V_ε by first collecting all variables A with rule $A \rightarrow \varepsilon \in P$ and then successively adding additional variables B if there is a rule $B \rightarrow A_1 A_2 \dots A_k \in P$ and the variables A_i are already in the set for all $1 \leq i \leq k$. ...

ε-Rules

Theorem

For every grammar G with rules $P \subseteq V \times (V \cup \Sigma)^*$ there is a context-free grammar G' with $\mathcal{L}(G) = \mathcal{L}(G')$.

Proof (continued).

Let P' be the rule set that is constructed from P by

- adding rules that obviate the need for $A \to \varepsilon$ rules: for every existing rule $B \to w$ with $B \in V, w \in (V \cup \Sigma)^+$, let I_ε be the set of positions where w contains a variable $A \in V_\varepsilon$. For every non-empty set $I' \subseteq I_\varepsilon$, add a new rule $B \to w'$, where w' is constructed from w by removing the variables at all positions in I'.
- removing all rules of the form $A \to \varepsilon$ (after the previous step).

...

ε-Rules

Theorem

For every grammar G with rules $P \subseteq V \times (V \cup \Sigma)^*$ there is a context-free grammar G' with $\mathcal{L}(G) = \mathcal{L}(G')$.

Proof (continued).

Then $\mathcal{L}(G) \setminus \{\varepsilon\} = \mathcal{L}(\langle \Sigma, V, P', S \rangle)$ and P' contains no rule $A \to \varepsilon$. If the start variable S of G is not in V_ε, we are done.

□

ε-Rules

Theorem

For every grammar G with rules $P \subseteq V \times (V \cup \Sigma)^*$ there is a context-free grammar G' with $\mathcal{L}(G) = \mathcal{L}(G')$.

Proof (continued).

Then $\mathcal{L}(G) \setminus \{\varepsilon\} = \mathcal{L}(\langle \Sigma, V, P', S \rangle)$ and P' contains no rule $A \to \varepsilon$. If the start variable S of G is not in V_ε, we are done.

Otherwise, let S' be a new variable and construct P'' from P' by

1. replacing all occurrences of S on the right-hand side of rules with S',
2. adding the rule $S' \to w$ for every rule $S \to w$, and
3. adding the rule $S \to \varepsilon$.

Then $\mathcal{L}(G) = \mathcal{L}(\langle \Sigma, V \cup \{S'\}, P'', S \rangle)$. □

Questions

Questions?

Chomsky Normal Form

Chomsky Normal Form: Motivation

As in logical formulas (and other kinds of structured objects), normal forms for grammars are useful:

- they show which aspects are critical for defining grammars and which ones are just syntactic sugar
- they allow proofs and algorithms to be restricted to a limited set of grammars (inputs): those in normal form

Hence we now consider a normal form for context-free grammars.

Chomsky Normal Form: Definition

Definition (Chomsky Normal Form)

A context-free grammar G is in Chomsky normal form (CNF) if all rules have one of the following three forms:

- $A \to BC$ with variables A, B, C, or
- $A \to a$ with variable A, terminal symbol a, or
- $S \to \varepsilon$ with start variable S.

German: Chomsky-Normalform

in short: rule set $P \subseteq (V \times (VV \cup \Sigma)) \cup \{\langle S, \varepsilon \rangle\}$

Chomsky Normal Form: Theorem

Theorem

For every context-free grammar G there is a context-free grammar G' in Chomsky normal form with $\mathcal{L}(G) = \mathcal{L}(G')$.

Chomsky Normal Form: Theorem

Theorem

For every context-free grammar G there is a context-free grammar G' in Chomsky normal form with $\mathcal{L}(G) = \mathcal{L}(G')$.

Proof.

The following algorithm converts the rule set of G into CNF:

Step 1: Eliminate rules of the form $A \to B$ with variables A, B.

If there are sets of variables $\{B_1, \dots, B_k\}$ with rules
$B_1 \to B_2, B_2 \to B_3, \dots, B_{k-1} \to B_k, B_k \to B_1$,
then replace these variables by a new variable B.

Then rename all variables to $V = \{A_1, \dots, A_n\}$ in a way that $A_i \to A_j \in P$ implies that $i < j$. For $k = n - 1, \dots, 1$: Eliminate all rules of the form $A_k \to A_{k'}$ with $k' > k$ and add a rule $A_k \to w$ for every rule $A_{k'} \to w$ with $w \in (V \cup \Sigma)^+$. ...

Chomsky Normal Form: Theorem

Theorem

For every context-free grammar G there is a context-free grammar G' in Chomsky normal form with $\mathcal{L}(G) = \mathcal{L}(G')$.

Proof (continued).

Step 2: Eliminate rules with terminal symbols on the right-hand side that do not have the form $A \to a$.

For every terminal symbol $a \in \Sigma$ add a new variable A_a and the rule $A_a \to a$.

Replace all terminal symbols in all rules that do not have the form $A \to a$ with the corresponding newly added variables. ...

Chomsky Normal Form: Theorem

Theorem

For every context-free grammar G there is a context-free grammar G' in Chomsky normal form with $\mathcal{L}(G) = \mathcal{L}(G')$.

Proof (continued).

Step 3: Eliminate rules of the form $A \to B_1 B_2 \dots B_k$ with $k > 2$

For every rule of the form $A \to B_1 B_2 \dots B_k$ with $k > 2$, add new variables $C_2, \dots, C_{k-1}$ and replace the rule with

$$\begin{aligned} A &\to B_1 C_2 \\ C_2 &\to B_2 C_3 \\ &\vdots \\ C_{k-1} &\to B_{k-1} B_k \end{aligned}$$

□

Chomsky Normal Form: Length of Derivations

Observation

Let G be a grammar in Chomsky normal form, and let $w \in \mathcal{L}(G)$ be a non-empty word generated by G.

Then all derivations of w have exactly $2|w| - 1$ derivation steps.

Proof.

$\rightsquigarrow$ Exercises □

Derivation Trees: General

Definition (Derivation Trees)

Let G be a context-free grammar, and let $S \Rightarrow w_1 \Rightarrow w_2 \Rightarrow \dots w_n$ be a derivation for a non-empty word $w_n \in \mathcal{L}(G)$.
The derivation tree T for this derivation is defined as follows:

- The root of the tree is associated with the start variable S.
- If the i-th derivation step replaces the variable A with the word z, then the corresponding A-node has $|z|$ children associated with the symbols of z (in the same order).

German: Ableitungsbaum

Note: The leaves of a derivation tree are in 1:1 correspondence to the symbols in the derived word.

Example: $\rightsquigarrow$ blackboard

Derivation Trees for Chomsky Normal Form Grammars

Observation

Let G be a grammar in Chomsky normal form,
and let $w \in \mathcal{L}(G)$ be a non-empty word generated by G.

All inner nodes in the derivation tree of w are binary,
except for the nodes whose children are leaves (which are unary).

(Obvious from the definitions of derivation trees
and Chomsky normal form.)

Pumping Lemma for Context-free Languages

Pumping lemma for context-free languages:

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

Key Ideas for Pumping Lemma for Context-free Language

We do not state or prove the pumping lemma
for context-free languages formally.

key proof ideas:

- Consider a Chomsky normal form grammar for the given language.
- The observation on Chomsky normal form derivation trees gives us bounds on the minimal depth of the derivation tree given the length of the generated word.
- In any sufficiently long word, there must be a sufficiently deep branch of the tree such that a variable symbol repeats on the branch.
- At such places, the tree (and hence the word) can be "pumped up" or "pumped down" by cloning or removing parts of the tree.

Questions

Questions?

Closure Properties

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 $\langle \Sigma_1 \cup \Sigma_2, V_1 \cup V_2 \cup \{S\}, P_1 \cup P_2 \cup \{S \to S_1, S \to S_2\}, S \rangle$ (where $S \notin V_1 \cup V_2$) is a context-free grammar for $\mathcal{L}(G_1) \cup \mathcal{L}(G_2)$ (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 $\langle \Sigma, V_1 \cup V_2 \cup \{S\}, P_1 \cup P_2 \cup \{S \to S_1 S_2\}, S \rangle$ (where $S \notin V_1 \cup V_2$) is a context-free grammar for $\mathcal{L}(G_1)\mathcal{L}(G_2)$ (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
$P' = (P \cup \{S' \to \varepsilon, S' \to S, S' \to SS'\}) \setminus \{S \to \varepsilon\}$
is a context-free grammar for $\mathcal{L}(G)^*$ after rewriting ε-rules. $\square$

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. □

Questions

Questions?

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 $w_\mathsf{l} \to w_\mathsf{r}$ of G we have $|w_\mathsf{l}| \leq |w_\mathsf{r}|$, 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 $A \rightarrow w$ 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 $\mathrm{P}_{\not\infty}$ 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 D (computability theory).

Questions

Questions?

Summary

Summary

- Every context-free language has a grammar in Chomsky normal form.
- Derivations in context-free languages have associated derivation trees. For grammars in Chomsky normal form, these are almost binary trees.
- 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.