CS 4410

Automata, Computability, and Formal Language

Dr. Xuejun Liang

Spring 2019

Chapter 8

Properties of Context-Free Languages

1. Two Pumping Lemmas
   - A Pumping Lemma for Context-Free Languages
   - A Pumping Lemma for Linear Language
2. Closure Properties and Decision Algorithms for Context-Free Languages
   - Closure of Context-Free Languages
   - Some Decidable Properties of Context-Free Languages

Learning Objectives

At the conclusion of the chapter, the student will be able to:

- Apply the pumping lemma to show that a language is not context-free
- State the closure properties applicable to context-free languages
- Prove that context-free languages are closed under union, concatenation, and star-closure
- Prove that context-free languages are not closed under either intersection or complementation
- Describe a membership algorithm for context-free languages
- Describe an algorithm to determine if a context-free language is empty
- Describe an algorithm to determine if a context-free language is infinite

A Pumping Lemma for Context-Free Languages

Theorem 8.1: (A Pumping Lemma for Context-Free Languages) Let L be an infinite context-free language. Then there exists some positive integer m such that any $w \in L$ with $|w| \geq m$ can be decomposed as

$$w=uvxyz, \text{ with } |vxy| \leq m \text{ and } |vy| \geq 1,$$

such that

$$uv^ixy^iz \in L, \text{ for all } i=0, 1, 2, \ldots$$

Example 8.1: Show that the language $L=\{a^nb^nc^n : n\geq 0\}$ is not context-free.

Example 8.2: The language $L=\{ww : w\in\{a,b\}^*\}$ is not context-free.

Example 8.3: The language $L=\{a^{n!}: n\geq 0\}$ is not context-free.

Example 8.4: The language $L=\{a^nb^j: n=j^2\}$ is not context-free.

A Pumping Lemma for Linear Languages

Definition 8.1: A context-free language is said to be linear if there exists a linear context-free grammar G such that L=L(G)

Example 8.5: The language $L=\{a^n b^n : n \geq 0\}$ is linear.

Theorem 8.2: (A Pumping Lemma for Linear Languages) Let L be an infinite linear language. Then there exists some positive integer m such that any $w \in L$ with $|w| \geq m$ can be decomposed as

$$w=uvxyz, \text{ with } |uvyz| \leq m \text{ and } |vy| \geq 1,$$

such that

$$uv^i xy^i z \in L, \text{ for all } i=0, 1, 2, \ldots$$

Example 8.6: The language $L=\{w : n_a(w)=n_b(w)\}$ is not linear.

Closure of Context-Free Languages

Theorem 8.3: The family of context-free language is closed under union, concatenation, and star-closure.

Theorem 8.4: The family of context-free language is not closed under intersection and complementation.

Theorem 8.5: Let L_1 be a context-free language and L_2 be a regular language. Then $L_1 \cap L_2$ is context-free.

Example 8.7: The language $L=\{a^n b^n : n \geq 0, n \neq 100\}$ is context-free.

Example 8.8: Show that the language

$$L=\{w \in \{a,b,c\}^* : n_a(w)=n_b(w)=n_c(w)\}$$

is not context-free.

Proof of Closure under Union

- Assume that L_1 and L_2 are generated by the context-free grammars $G_1 = (V_1, T_1, S_1, P_1)$ and $G_2 = (V_2, T_2, S_2, P_2)$
- Without loss of generality, assume that the sets V_1 and V_2 are disjoint
- Create a new variable S_3 which is not in $V_1 \cup V_2$
- Construct a new grammar $G_3 = (V_3, T_3, S_3, P_3)$ so that
  - $V_3 = V_1 \cup V_2 \cup \{ S_3 \}$
  - $T_3 = T_1 \cup T_2$
  - $P_3 = P_1 \cup P_2$
- Add to P_3 a production that allows the new start symbol to derive either of the start symbols for L_1 and L_2
  - $S_3 \rightarrow S_1 \mid S_2$
- Clearly, G_3 is context-free and generates the union of L_1 and L_2, thus completing the proof

Proof of Closure under Concatenation

- Assume that L_1 and L_2 are generated by the context-free grammars $G_1 = (V_1, T_1, S_1, P_1)$ and $G_2 = (V_2, T_2, S_2, P_2)$
- Without loss of generality, assume that the sets V_1 and V_2 are disjoint
- Create a new variable S_4 which is not in $V_1 \cup V_2$
- Construct a new grammar $G_4 = (V_4, T_4, S_4, P_4)$ so that
  - $V_4 = V_1 \cup V_2 \cup \{ S_4 \}$
  - $T_4 = T_1 \cup T_2$
  - $P_4 = P_1 \cup P_2$
- Add to P_4 a production that allows the new start symbol to derive the concatenation of the start symbols for L_1 and L_2
  - $S_4 \rightarrow S_1S_2$
- Clearly, G_4 is context-free and generates the concatenation of L_1 and L_2 , thus completing the proof

Proof of Closure under Star-Closure

- Assume that L_1 is generated by the context-free grammars $G_1 = (V_1, T_1, S_1, P_1)$
- Create a new variable S_5 which is not in V_1
- Construct a new grammar $G_5 = (V_5, T_5, S_5, P_5)$ so that
  - $V_5 = V_1 \cup \{ S_5 \}$
  - $T_5 = T_1$
  - $P_5 = P_1$
- Add to P_5 a production that allows the new start symbol S_5 to derive the repetition of the start symbol for L_1 any number of times
  - $S_5 \rightarrow S_1S_5 \mid \lambda$
- Clearly, G_5 is context-free and generates the star-closure of L_1, thus completing the proof

No Closure under Intersection

- Unlike regular languages, the intersection of two context-free languages L_1 and L_2 does not necessarily produce a context-free language
- As a counterexample, consider the context-free languages

$$L_1 = \{ a^n b^n c^m : n \geq 0, m \geq 0 \}$$

$$L_2 = \{ a^n b^m c^m : n \geq 0, m \geq 0 \}$$

- However, the intersection L_1 and L_2 is the language

$$L_3 = \{ a^n b^n c^n : n \geq 0 \}$$

- L_3 can be shown not be context-free by applying the pumping lemma for context-free languages

A Membership Algorithm for Context-Free Languages

- The combination of Theorems 5.2 and 6.5 confirms the existence of a membership algorithm for context-free languages
- By Theorem 5.2, exhaustive parsing is guaranteed to give the correct result for any context-free grammar that contains neither λ-productions nor unit-productions
- By Theorem 6.5, such a grammar can always be produced if the language does not include λ
- Alternatively, a npda to accept the language can be constructed as established by Theorem 7.1

Some Decidable Properties of Context-Free Languages

Theorem 8.6: Given a context-free grammar G=(V,T,S,P), there exists an algorithm for deciding whether or not L(G) is empty.

- For simplicity, assume that λ is not in L(G)
- Apply the algorithm for removing useless symbols and productions
- If the start symbol is found to be useless, then L(G) is empty;
- Otherwise, L(G) contains at least one string

Theorem 8.7: Given a context-free grammar G=(V,T,S,P), there exists an algorithm for deciding whether or not L(G) is infinite.

- Apply the algorithms for removing λ-productions, unit-productions, and useless productions
- If G has a variable A for which there is a derivation that allows A to produce a sentential form xAy, then L(G) is infinite
- Otherwise, L(G) is finite

Determining Whether Two Context-Free Languages are Equal

- Given two context-free grammars G_1 and G_2, is there an algorithm to determine if $L(G_1) = L(G_2)$?
- If the languages are finite, the answer can be found by performing a string-by-string comparison
- However, for general context-free languages, no algorithm exists to determine equality