CS 4410

Automata, Computability, and Formal Language

Dr. Xuejun Liang

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

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

- Every sufficiently long string *w* in L can be broken into five parts
  - $w = uvxyz$, with $|vxy| \leq m$ and $|vy| \geq 1$,
- An arbitrary, but equal number of repetitions of v and y yields another string in L
  - $w_i = uv^ixy^iz \in L$, for all $i = 0, 1, 2, \ldots$
- The pumping lemma can be used to show that, by contradiction, a certain language is not context-free

An Illustration of the Pumping Lemma for Context-Free Languages

As shown in Figure 8.1, the pumping lemma for context-free languages can be illustrated by sketching a general derivation tree that shows a decomposition of the string into the required components

$$S \stackrel{*}{\Rightarrow} uAz \stackrel{*}{\Rightarrow} uvAyz \stackrel{*}{\Rightarrow} uvxyz$$

$$A \stackrel{*}{\Rightarrow} vAy \qquad A \stackrel{*}{\Rightarrow} x$$

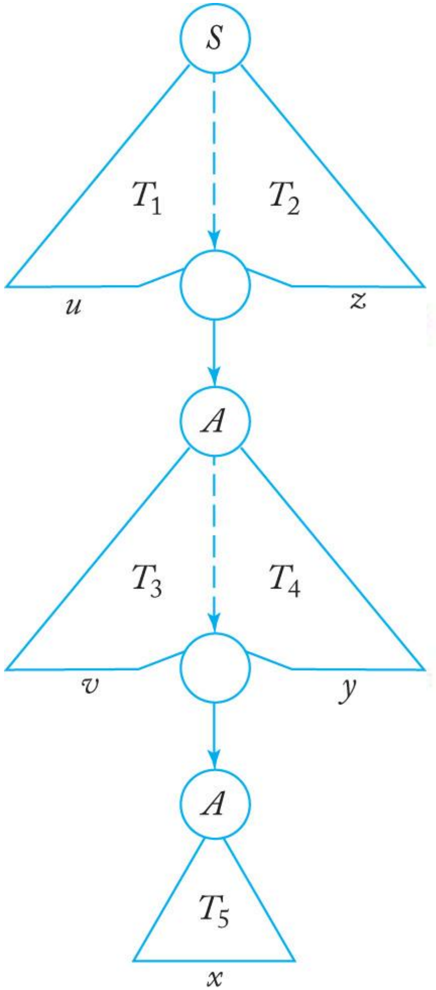


FIGURE 8.1

Examples: Using Pumping Lemma

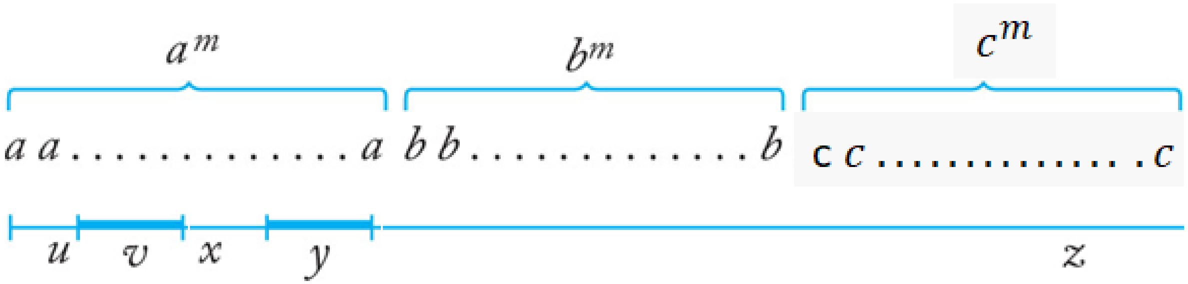
Example 8.1: Show that the language L={$a^n b^n c^n$: n≥0} is not context-free.

Given m > 0

Pick $w = a^m b^m c^m \in L$

Given $w = uvxyz$ with $|vxy| \le m$ and $|vy| \ge 1$

Case 1:



So, $v = a^k$ and $y = a^l$, $k + l > 0$

Then $w_0 = uv^0xy^0z = a^{m-(k+l)} b^m c^m \notin L$, as $m - (k + l) \neq m$

This is a contradiction. So L is not context-free

Examples: Using Pumping Lemma (Cont.)

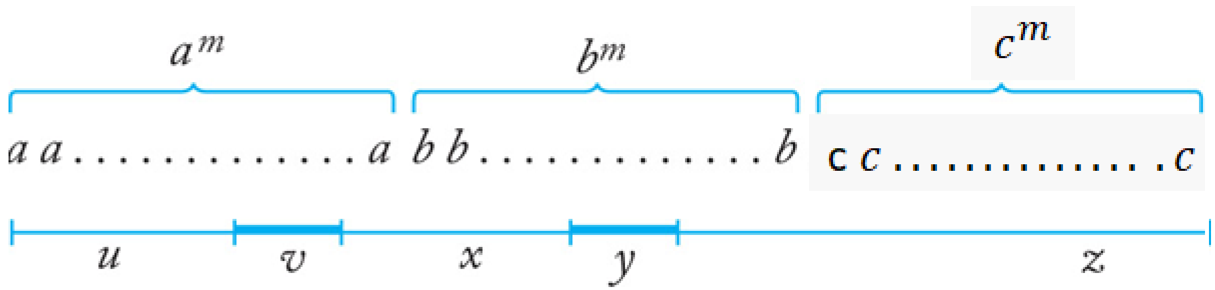
Example 8.1: Show that the language $L=\{a^n b^n c^n : n \geq 0\}$ is not context-free.

Given m > 0

Pick $w = a^m b^m c^m \in L$

Given $w = uvxyz$ with $|vxy| \leq m$ and $|vy| \geq 1$

Case 2:



So, either $v = a^k b^p (k \geq 1 \text{and } p \geq 0)$ and $y = b^l (l \geq 1)$

or $v = a^k\ (k \geq 1)$ and $y = a^p b^l (l \geq 1 \text{ and } p \geq 0)$

Assume $v = a^k\ (k \geq 1)$ and $y = a^p b^l (l \geq 1 \text{ and } p \geq 0)$

Then $w_0 = uv^0xy^0z = a^{m-k-p} b^{m-l} c^m \notin L$

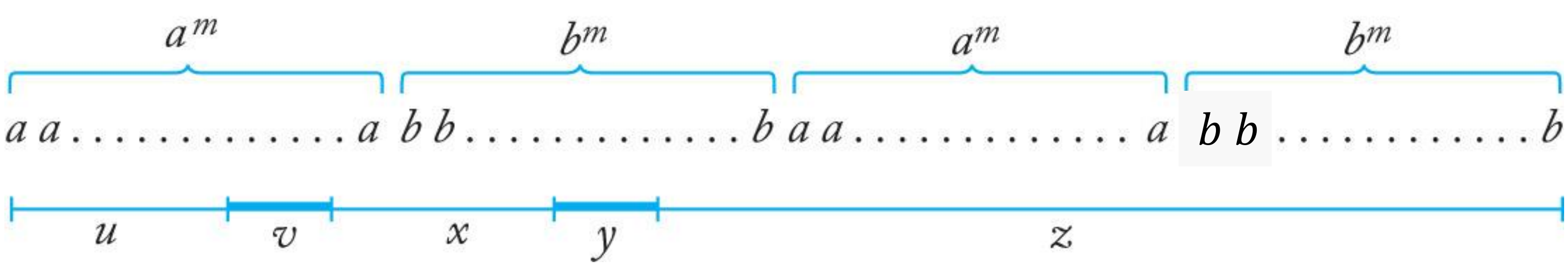
Examples: Using Pumping Lemma (Cont.)

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

Given m > 0

Pick $w = a^m b^m a^m b^m \in L$

Given $w = uvxyz$ with $|vxy| \le m$ and $|vy| \ge 1$



So, either $v = a^k b^p (k \ge 1$ and $p \ge 0)$ and $y = b^l (l \ge 1)$

or $v = a^k$ $(k \ge 1)$ and $y = a^p b^l (l \ge 1$ and $p \ge 0)$

Assume $v = a^k$ $(k \ge 1)$ and $y = a^p b^l (l \ge 1$ and $p \ge 0)$

Then $w_0 = uv^0xy^0z = a^{m-k-p} b^{m-l} a^m b^m \notin L$

Examples: Using Pumping Lemma (Cont.)

Example 8.3: The language L={$a^{n!}$: n≥3} is not context-free.

Given m > 2

Pick $w = a^{m!} \in L$

Given $w = uvxyz$ with $|vxy| \leq m$ and $|vy| \geq 1$

So, $v = a^k$ and $y = a^l$, $1 \leq k + l \leq m$

Then, $w_0 = uv^0xy^0z = a^{m!-(k+l)} \notin L$,

Because $m! > m! - (k + l) \geq m! - m > (m - 1)!$

This is a contradiction. So L 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

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

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.

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
  - $S_5 \rightarrow S_1 S_5 \mid \lambda$
- 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

No Closure under Complementation

- The complement of a context-free language L_1 does not necessarily produce a context-free language
- The proof is by contradiction: given two context-free languages L_1 and L_2, assume that their complements are also context-free
- By Theorem 8.3, the union of the complements must also produce a context-free language L_3 ($L_3 = \overline{L_1} \cup \overline{L_2}$)
- Using our assumption, the complement of L_3 is also context-free.
- However, using the set identity below, we conclude that the complement of L_3 is the intersection of L_1 and L_2, which has been shown not to be context-free, thus contradicting our assumption.

$$\overline{L_3} = \overline{\overline{L_1} \cup \overline{L_2}} = L_1 \cap L_2$$