The Pumping Lemma and Closure Properties

Mridul Aanjaneya

Stanford University

July 5, 2012
Tentative Schedule

- HW #1: Out (07/03), Due (07/11)
- HW #2: Out (07/10), Due (07/18)
- HW #3: Out (07/17), Due (07/25)
- Midterm: 07/31 (in class)
- HW #4: Out (07/31), Due (08/08)
- Tentative grades out by 08/12.
- Final: 08/18 (?)
The Infiniteness Problem

Question
Is a given regular language $L$ infinite?

- Start with a DFA for the language.
- **Key idea:** If the DFA has $n$ states, and the language contains any string of length $n$ or more, then the language is infinite.
- Otherwise, the language is surely finite.
  - **Limited** to strings of length $n$ or less.
Proof of Key Idea

- If an \( n \)-state DFA accepts a string \( w \) of length \( n \) or more, then there must be a state that appears twice on the path labeled \( w \) from the start state to a final state.
  - **Note:** Pigeonhole principle! 😊
- Because there are at least \( n + 1 \) states along the path.

Since \( y \) is not \( \varepsilon \), we see an infinite number of strings in \( L \) of the form \( xy^i z \) for all \( i \geq 0 \).
The Pumping Lemma

For every regular language \( L \), there is an integer \( n \) such that for every string \( w \in L \) of length \( \geq n \), we can write \( w = xyz \) such that:

- \( |xy| \leq n \).
- \( |y| > 0 \).
- For all \( i \geq 0 \), \( xy^i z \) is in \( L \).
The Pumping Lemma: Examples

Question
Prove that the language $L = \{0^k1^k \mid k \geq 1\}$ is not regular.

• Proof by contradiction. Suppose it were, and let a DFA with $n$ states accept all strings in $L$.
• Choose the string $w = 0^n1^n$. We can write $w = xyz$ where $x$ and $y$ consist of 0’s, and $y \neq \varepsilon$.
• But then $xyyz$ would be in $L$, and this string has more 0’s than 1’s!
• Choice of the proper string is very important!
  • Example: $w = (01)^n$ does not work!
• Same argument as above also works for $L = \{w \mid w \text{ has equal number of 0’s and 1’s}\}$
The Pumping Lemma: Examples

Question

Prove that the language \( L = \{ww \mid w \in \{0,1\}^*\} \) is not regular.
The Pumping Lemma: Examples

Question

Prove that the language $L = \{ww \mid w \in \{0,1\}^*\}$ is not regular.

- Suppose it were, and let a DFA with $n$ states accept all strings in $L$.
- Choose the string $w = 0^n10^n1$. We can write $w = xyz$ where $x$ and $y$ consist of 0’s, and $y \neq \epsilon$.
- But then $xyyz$ would be in $L$!
- Note that the string $w = 0^n0^n$ does not work.
Question

Prove that the language $L = \{1^{k^2} \mid k \geq 0\}$ is not regular.
Question

Prove that the language $L = \{1^{k^2} \mid k \geq 0\}$ is not regular.

- Suppose it were, and let a DFA with $n$ states accept all strings in $L$.
- Choose the string $w = 1^{n^2}$. We can write $w = xyz$. Consider the string $s = xyyz$.
- We know that $|xy| \leq n$ and thus $|y| \leq n$. So no. of 1’s in $xyyz$ is $n^2 + n < n^2 + 2n + 1$!
- The parameter $n$ is often called the pumping length.
The Pumping Lemma: Examples

Question

Prove that the language $L = \{0^i1^j \mid i > j\}$ is not regular.
The Pumping Lemma: Examples

• Sometimes pumping down is useful as well!

Question

Prove that the language \( L = \{0^i1^j | i > j\} \) is not regular.

• Let \( n \) be the pumping length, i.e., suppose there exists a DFA with \( n \) states that accepts all strings in \( L \).
• Choose the string \( w = 0^{n+1}1^n \). We can write \( w = xyz \) where \( x \) and \( y \) consist of 0’s, and \( y \neq \epsilon \).
• The pumping lemma states that all strings \( xy^iz \in L \), even for \( i = 0 \)!
• The string \( w = xz \) cannot have more 0’s than 1’s.
The Pumping Lemma: Examples

Question
Prove that the language \( L = \{ 1^p \mid \text{where } p \text{ is prime} \} \) is not regular.
The Pumping Lemma: Examples

Question

Prove that the language \( L = \{1^p \mid \text{where } p \text{ is prime} \} \) is not regular.

- Consider some prime \( q \geq n + 2 \), where \( n \) is the pumping length.
- Choose the string \( w = 1^q \). We can write \( w = xyz \) such that \( y \neq \varepsilon \) and \( |xy| \leq n \).
- Let \( |y| = m \). Then \( |xz| = q - m \). Consider the string \( s = xy^{q-m}z \) which is in \( L \) by the pumping lemma.
- \( |xy^{q-m}z| = |xz| + (q-m)|y| = q-m + (q-m)m = (m+1)(q-m) \).
- Note that \( m+1 > 1 \), as \( y \neq \varepsilon \).
- Also note that \( q \geq n + 2 \), and so \( q - m > 1 \).
A closure property of a language class says that given languages in the class, an operator (e.g., union) produces another language in the same class.

Example: We saw that regular languages are closed under union, concatenation and Kleene closure (star) operations.

We will see more examples: intersection, difference, reversal, homomorphism, inverse homomorphism.
Closure Properties: Intersection

- Construct the **product DFA** from DFA’s for \( L \) and \( M \).
- Let these DFA’s have sets of states \( Q \) and \( R \) respectively.
- Product DFA has set of states \( Q \times R \).
  - i.e., pairs \([q, r]\) with \( q \in Q \) and \( r \in Q \).
- Start state = \([q_0, r_0]\) (the start states of the DFA for \( L \) and \( M \)).
- **Transitions:** \( \delta([q, r], a) = [\delta_L(q, a), \delta_M(r, a)] \).
  - \( \delta_L, \delta_M \) are the transition functions for the DFA’s of \( L \) and \( M \).
  - i.e., we simulate the two DFA’s in the two state components of the product DFA.
- Make final states be pairs consisting of final states of both DFA’s of \( L \) and \( M \).
Product DFA: Example

- States: A, B, C, D
- Transitions:
  - A: 0 → B, 1 → B
  - B: 0,1 → B, 1 → C
  - C: 0 → D, 1 → D
  - D: 0 → D, 1 → D

States [A,C], [B,C], [B,D], [A,D] are also shown with their transitions.
Closure Properties: Difference

- If $L$ and $M$ are regular, then so is $L - M = \text{strings in } L \text{ but not in } M$.
- **Proof:** Let $A$ and $B$ be DFA’s whose languages are $L$ and $M$.
- Construct the product DFA $C$ of $A$ and $B$.
- Make the final states of $C$ be the pairs where $A$-state is final but $B$-state is not.
Closure Properties: Containment

- If $L$ and $M$ are regular, then so is $L - M = \text{strings in } L \text{ but not in } M$.
- **Proof:** Let $A$ and $B$ be DFA’s whose languages are $L$ and $M$.
- Construct the product DFA $C$ of $A$ and $B$.
- Make the final states of $C$ be the pairs where $A$-state is final but $B$-state is not.
- **Note:** Can also be used to test containment.
  - If $L - M = \emptyset$, then $L \subseteq M$.
  - How did we test if the language of a DFA is empty?
Closure Properties: Complement

- The complement of a language $L$ is $\Sigma^* - L$.
- Since $\Sigma^*$ is regular, the complement is always regular.
• Given language $L$, $L^R$ has all strings whose reversal is in $L$.

• **Example:** $L = \{0, 01, 100\}$; 
  $L^R = \{0, 10, 001\}$

• **Proof:** Let $E$ be a regular expression for $L$.

• We show how to reverse $E$, to provide a regular expression $E^R$ for $L^R$. 
• **Basis:** If $E$ is a symbol $a$, $\varepsilon$, or $\emptyset$, then $E^R = E$.

• **Induction:** If $E$ is:
  - $F + G$, then $E^R = F^R + G^R$
  - $FG$, then $E^R = G^R F^R$
  - $F^*$, then $E^R = (F^R)^*$
Let $E = 01^* + 10^*$.

\[
E^R = (01^* + 10^*)^R \\
= (01^*)^R + (10^*)^R \\
= (1^*0^R + 0^*_1^R) \\
= (1^R)^0 + (0^R)^1 \\
= (1^*0 + (0^*)1
\]
A homomorphism on an alphabet is a function that gives a string for each symbol in that alphabet.

Example: \( h(0) = ab, \ h(1) = \varepsilon. \)

Extend to strings by \( h(a_1a_2\ldots a_n) = h(a_1)h(a_2)\ldots h(a_n). \)

Example: \( h(01010) = ababab. \)
• If $L$ is regular, and $h$ is a homomorphism on its alphabet, then $h(L) = \{ h(w) \mid w \in L \}$ is also regular.

• **Proof:** Let $E$ be a regular expression for $L$.
• Apply $h$ to each symbol in $E$.
• Language of resulting RE is $h(L)$.
• Let $h(0) = ab$, $h(1) = \varepsilon$.
• Let $L$ be the language of a regular expression $01^* + 10^*$.
• Then $h(L)$ is the language of regular expression $ab\varepsilon^* + \varepsilon(ab)^*$.  
• **Note:** $ab\varepsilon^* + \varepsilon(ab)^* = ab + (ab)^* = (ab)^*$.  

Mridul Aanjaneya  
Automata Theory 27/ 27