Pumping Lemma for Regular Languages (Pre Lecture)

Dr. Neil T. Dantam

CSCI-561, Colorado School of Mines

Fall 2020

Application: XML

Matching XML Tags

- Consider the language of XML documents with matched tags.
- Can we write a regular expression to check for matching XML tags?
- Is this language regular?

Example

```
<div>
  <p>
    <em>Hello</em>,
    <strong>World!</strong>
  </p>
</div>
```

Introduction

Pumping Lemma

- We can write regular expressions and simulate NFA/DFA for *regular languages*.
- Are all languages regular languages? No!
- How do we know if a language is regular (and we should try to think of a regular expression) or not regular (and we should try something else)?

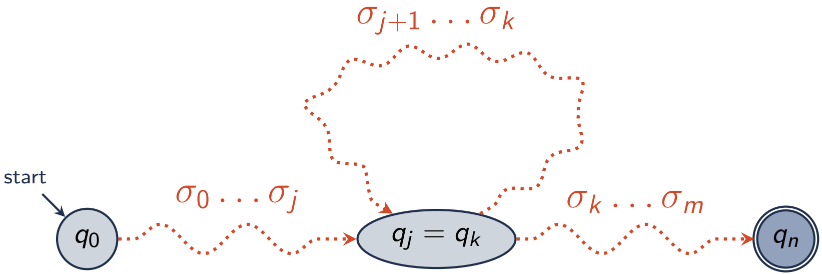
Outcomes

- Know the definition pumping lemma
- Understand why the pumping lemma holds (its proof)
- Apply *pumping lemma* to prove languages are not regular

Language size

Review

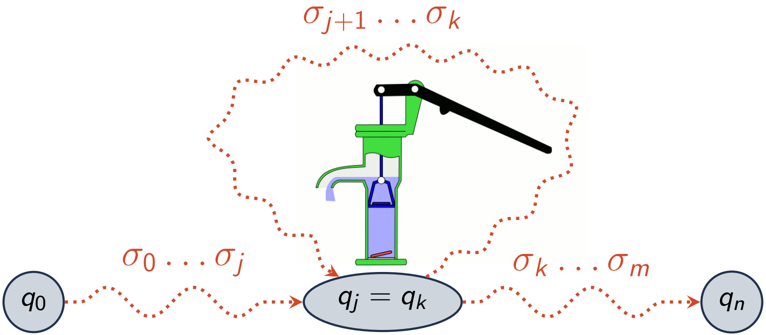
- Given finite automaton A and string $\sigma \in \mathcal{L}(A)$
- What if $|\sigma| > |Q|$?
  What does this tell us about the size of the the language, $|\mathcal{L}(A)|$?



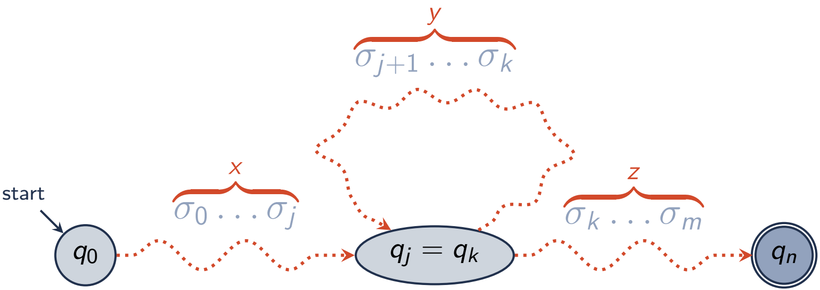
Pumping Lemma Intuition

Finite Automaton Cycles

- *Pumping Lemma*: Prove nonregularity, based on a property of all regular languages
- Any string in the language longer than the *pumping length* can be *pumped*
- I.e., the string contains a section that can be repeated an arbitrary number of times



Pumping Lemma Parts



We can repeat part y any number of times.

Theorem: Pumping Lemma

Theorem (Pumping Lemma)

If A is a regular language, there is a number p (the pumping length) where if:

- $\sigma \in A$
- $|\sigma| > p$

then we can divide σ into three pieces $\sigma = xyz$ such that:

- *for each* $i \geq 0$, $xy^iz \in A$ *(we can pump y)*
- $|y| > 0$
- $|xy| \leq p$

Proof Outline: Pumping Lemma

$$\sigma \;=\; \underset{q^{[0]}}{\uparrow}\;\sigma_0\;\underset{q^{[1]}}{\uparrow}\;\sigma_1\;\underset{q^{[2]}}{\uparrow}\;\sigma_2\;\underset{q^{[3]}}{\uparrow}\;\cdots\;\underset{q^{[n-1]}}{\uparrow}\;\sigma_{n-1}\;\underset{q^{[n]}}{\uparrow}\;\sigma_n\;\underset{q^{[n+1]}}{\uparrow}$$

Proof.

- $A = \mathcal{L}(M)$ and $M = (Q, \Sigma, \delta, q_0, F)$
- Assume $p = |Q|$
- Pigeonhole Principle:
  - p pigeonholes and more than p pigeons
  - Some hole has more than one pigeon
- For $\sigma \in A$, if $|\sigma| = n$, we visit $n+1$ states of a DFA to accept σ
- If $|\sigma| > p$, we must revisit some state

□

Using the Pumping Lemma to Prove a Language is not Regular

Proofs by Contradiction

Proofs by Contradiction

Theorem

Property P is true.

Proof Outline

- Assume P is false
- Show this assumption leads to a false consequence
- Contradiction. QED.

Using the pumping lemma

Theorem

Language L is not regular.

Proof Outline

- Assume L is regular
- Show that some string longer that p cannot be "pumped".
- Contradiction. QED.

Example 0: Pumping Lemma Proof

Theorem

Let $L = \{0^n1^n \mid n \geq 0\}$. *L is not regular*

Proof.

1. Assume L is regular and apply the pumping lemma
2. Choose $\sigma = 0^p1^p = xyz$
3. There are three possible cases:
   3.1 $y = 0 \ldots 0 \implies xyyz$ has more 0s than 1s $(xyyz \notin L)$
   3.2 $y = 1 \ldots 1 \implies xyyz$ has more 1s than 0s $(xyyz \notin L)$
   3.3 $y = 0 \ldots 01 \ldots 1 \implies xyyz$ has 0s after 1s $(xyyz \notin L)$
4. Contradiction, L is not regular.

□

Example 1: False Pumping Lemma "Proof"

False Theorem

Let $L = \mathcal{L}(0^*1^*)$. L is not regular.

False Proof

1. Assume L is regular and apply the pumping lemma.
2. Choose $\sigma = 0^p1^p = xyz$.
3. The previous example showed that 0^p1^p cannot be pumped.
4. Contradiction, L is not regular.

Explanation: Step 3 is wrong

- $(|xy| \leq p) \implies (y = 0 \ldots 0)$
- $xyyz \in L$ and $xz \in L$, so we actually can pump σ in this case.

Example 2: Pumping Lemma Proof

Theorem

Let $L = \{\omega\omega \mid \omega \in \{0,1\}^*\}$. L is not regular.

Proof.

1. Assume L is regular and apply the pumping lemma
2. Choose $\sigma = \overbrace{0^p 1}^{\omega} \overbrace{0^p 1}^{\omega} = xyz$.
3. $(|xy| \leq p) \implies (y = 0 \ldots 0)$.
4. $xyyz = 0^{p+k}10^p1$: first half ends in 0; second half ends in 1.
5. $xyyz \notin L$.
6. Contradiction, L is not regular.

□

Common False Steps in Applying the Pumping Lemma

- *I declare that the pumping length $p = 5$.*
  A regular language must have some constant p, but we do not know what that p based only on the fact that a language is regular. Instead, we treat p symbolically when using the pumping lemma to prove non-regularity.

- *"I found a string I could pump. Therefore, the language is regular."*
  There could still be some other string we cannot pump.

- *"I couldn't find a string that, when pumped, was not in the language. Therefore, the language is regular."*
  Maybe we just couldn't find the right string.

Example 3: False Pumping Lemma "Proof"

False Theorem

Let $L = \mathcal{L}(0001^*)$. L is not regular.

False Proof

1. Assume L is regular and apply the pumping lemma.
2. Choose pumping length $p = 3$ and $\sigma = 0001 = xyz$.
3. $(|xy| \leq p) \implies (y = 0 \ldots 0)$
4. $xyyz \notin L$ since more than 3 zeros. Contradiction, L is not regular.

Explanation: Step 2 is wrong

- 3 is not the pumping length of L.
- We generally do not know p numerically, so treat symbolically.

Example 4: False Pumping Lemma "Proof"

Theorem

Let $B = \{\omega \mid \omega \text{ has equal number of 0s and 1s}\}$. *B is regular.*

False Proof

1. Assume B is regular and apply the pumping lemma
2. Choose $\sigma = 0^{p/2}1^{p/2} = xyz$ and $y = 01$
3. xy^kz (or xz) will increase (or decrease) 0s and 1s by the same
4. Therefore, B must be regular.

Explanation: Step 4 is wrong

- We can indeed pump $0^{p/2}1^{p/2}$ and the result is in B.
- But, there may be (is) some other string in B we cannot pump.

Exercise 0: Pumping Lemma

Theorem

Let $B = \{\omega \mid \omega$ has equal number of 0s and 1s$\}$. B is not regular.

Proof.

□

Exercise 1: Pumping Lemma

Theorem

Let $C = \left\{ 1^{n^2} \mid n \geq 0 \right\}$. *C is not regular.*

Proof.

□

So what?

L isn't regular. Who cares?

- We prove/do many things about regular languages.
- We have fewer capabilities with non-regular languages.
- Don't waste your time trying to write a regular expression for a non-regular language.

Exercise: XML

Theorem

It is impossible to check matching XML tags using a regular expression.

Proof.

□

References

Textbook: Sipser, 3rd ed.

- Ch 1.4 Nonregular Languages

Alt. Textbook: Hopcroft, 3rd ed.

- Ch 4.1 Proving Languages Not to Be Regular