Harvard CS 121 and CSCI E-207

Lecture 8: Pumping and Other Aspects of Regular Languages

Salil Vadhan

September 27, 2012

- **Reading:** Sipser, §1.4.

Pumping Lemma (Basic Version)

If L is regular, then there is a number p (the pumping length)
such that
every string $s \in L$ of length at least p
can be divided into $s = xyz$, where $y \neq \varepsilon$ and
for every $n \geq 0$, $xy^n z \in L$.

$n = 1$	x	y	z	
$n = 0$	x	z		
$n = 2$	x	y	y	z
$\ldots$				

- Why is the part about p needed?
- Why is the part about $y \neq \varepsilon$ needed?

Pumping Lemma Example

- Consider

$$L = \{x : x \text{ has an even \# of } a\text{'s and an odd \# of } b\text{'s}\}$$

- Since L is regular, pumping lemma holds.

  (i.e, every sufficiently long string s in L is “pumpable”)

- For example, if $s = aab$, we can write $x = \varepsilon$, $y = aa$, and $z = b$.

Pumping the even a's, odd b's language

- Claim: L satisfies pumping lemma with pumping length $p = 4$.

- Proof:

- **Q:** Can the Pumping Lemma be used to prove that L is regular?

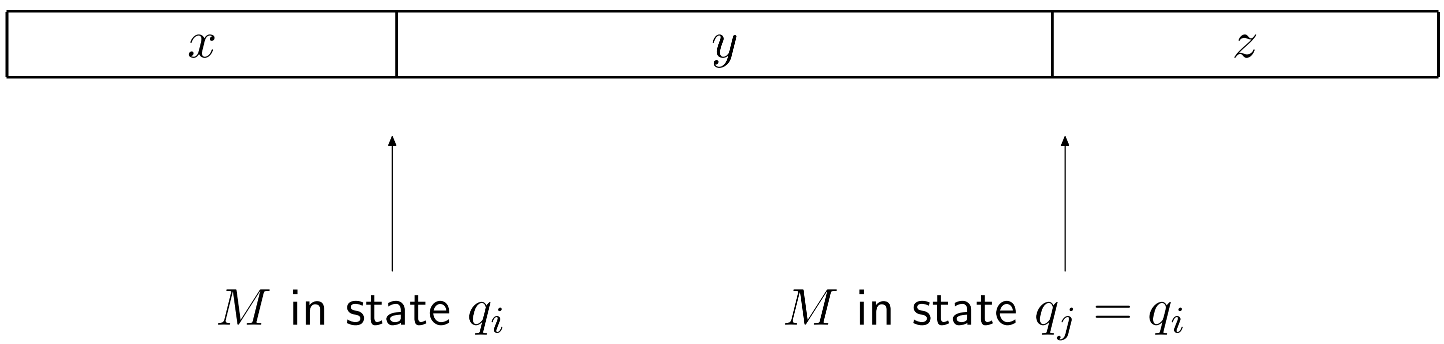
Proof of Pumping Lemma

(Another fooling argument)

- Since L is regular, there is a DFA M recognizing L.
- Let $p =$ # states in M.
- Suppose $s \in L$ has length $l \geq p$.
- M passes through a sequence of $l + 1 > p$ states while accepting s (including the first and last states): say, $q_0, \ldots, q_l$.
- Two of these states must be the same: say, $q_i = q_j$ where $i < j$

Pumping, continued

- Thus, we can break s into x, y, z where $y \neq \varepsilon$ (though x, z may equal ε):



- If more copies of y are inserted, M "can't tell the difference," i.e., the state entering y is the same as the state leaving it.

- So since $xyz \in L$, then $xy^nz \in L$ for all n.

Proof also shows (why?):

- We can take p = # states in smallest DFA recognizing L.

- Can guarantee division $s = xyz$ satisfies $|xy| \leq p$ (or $|yz| \leq p$).

Use PL to Show Languages are NOT Regular

Claim: $L = \{a^n b^n : n \geq 0\} = \{\varepsilon, ab, aabb, aaabbb, \ldots\}$ is not regular.

Proof by contradiction:

- Suppose that L is regular.
- So L has some pumping length $p > 0$.
- Consider the string $s = a^p b^p$. Since $|s| = 2p > p$, we can write $s = xyz$ for some strings x, y, z as specified by lemma.
- Claim: No matter how s is partitioned into xyz with $y \neq \varepsilon$, we have $xy^2 z \notin L$.
- This violates the conclusion of the pumping lemma, so our assumption that L is regular must have been false.

Strings of exponential lengths are a nonregular language

Claim: $L = \{w : |w| = 2^n \text{ for some } n \geq 0\}$ is not regular.

Proof:

“Regular Languages Can’t Do Unbounded Counting”

Claim: $L = \{w : w$ has the same number of a’s and b’s$\}$ is not regular.

Proof #1:

- Use pumping lemma on $s = a^p b^p$ with $|xy| \leq p$ condition.

"Regular Languages Can't Do Unbounded Counting"

Claim: $L = \{w : w$ has the same number of a's and b's$\}$ is not regular.

Proof #1:

- Use pumping lemma on $s = a^p b^p$ with $|xy| \leq p$ condition.

Proof #2:

- If L were regular, then $L \cap a^* b^*$ would also be regular.

Reprise on Regular Languages

Which of the following are necessarily regular?

- A finite language
- A union of a finite number of regular languages
- A union of a countable number of regular languages
- $\{x : x \in L_1$ and $x \notin L_2\}$, L_1 and L_2 are both regular
- A cofinite language (a set is *cofinite* if its complement is finite)
- The reversal of a regular language

Algorithmic questions about regular languages

Given X = a regular expression, DFA, or NFA, how could you tell if:

- $x \in L(X)$, where x is some string?
- $L(X) = \emptyset$?
- $x \in L(X)$ but $x \notin L(Y)$?
- $L(X) = L(Y)$, where Y is another RE/FA?
- $L(X)$ is infinite?
- There are infinitely many strings that belong to both $L(X)$ and $L(Y)$?