

Harvard CS 121 and CSCI E-207

Lecture 8: Non-Regular Languages

Harry Lewis

September 29, 2009

- **Reading:** Sipser, §1.4.

Cardinality of Languages

- An alphabet Σ is finite by definition
- **Proposition:** Σ^* is countably infinite
- So every language is either finite or countably infinite
- $P(\Sigma^*)$ is uncountable, being the set of subsets of a countably infinite set.

i.e. There are uncountably many languages over any alphabet

Q: Even if $|\Sigma| = 1$?

Existence of Non-regular Languages

Theorem: For every alphabet Σ , there exists a non-regular language over Σ .

Proof:

- There are only countably many regular expressions over Σ .
 \Rightarrow There are only countably many regular languages over Σ .
- There are uncountably many languages over Σ .
- Thus at least one language must be non-regular.

\Rightarrow In fact, “almost all” languages must be non-regular.

Q: Could we do this proof using DFAs instead?

Q: Can we get our hands on an *explicit* non-regular language?₂

Goal: Explicit Non-Regular Languages

It appears that a language such as

$$\begin{aligned} L &= \{x \in \Sigma^* : |x| = 2^n \text{ for some } n \geq 0\} \\ &= \{a, b, aa, ab, ba, bb, aaaa, \dots, bbbb, aaaaaaaaaa, \dots\} \end{aligned}$$

can't be regular because the “gaps” in the set of possible lengths become arbitrarily large, and no DFA could keep track of them.

But this isn't a proof!

Approach:

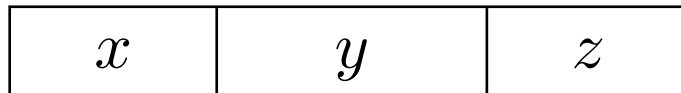
1. Prove some general property P of all regular languages.
2. Show that L does not have P .

Pumping Lemma (Basic Version)

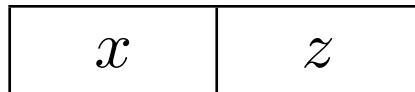
If L is regular, then there is a number p (the pumping length) such that

every string $s \in L$ of length at least p can be divided into $s = xyz$, where $y \neq \varepsilon$ and for every $n \geq 0$, $xy^n z \in L$.

$n = 1$



$n = 0$



$n = 2$



...

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

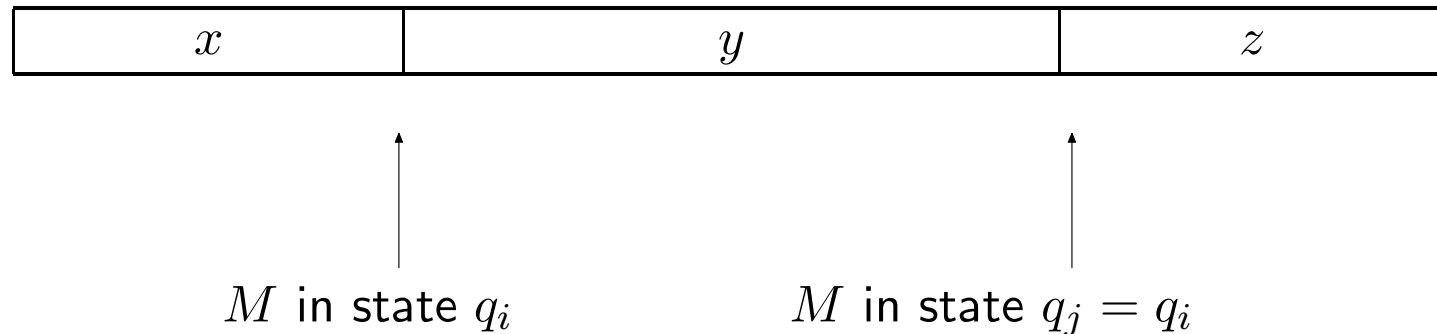
Proof of Pumping Lemma

(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, \dots, 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 $xy^n z \in L$ 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 $|yz| \leq p$).

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 $z = b$.

Use PL to Show Languages are NOT Regular

Claim: $L = \{a^n b^n : n \geq 0\} = \{\varepsilon, ab, aabb, aaabbb, \dots\}$ 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 $xy^2z \notin L$.
- 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: $L = \{w : w \text{ has the same number of } a\text{'s and } b\text{'s}\}$ 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: $L = \{w : w \text{ has the same number of } a\text{'s and } b\text{'s}\}$ 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
- $\{x : x \in L_1 \text{ and } x \notin L_2\}$, L_1 and L_2 are both regular
- A subset of a regular language

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