Lecture 8: Proving that a language is not regular

Example: $L = \{0^n 1^n | n \geq 0\}$ is not a regular language.

- Intuitively, this language L is not regular because any machine that can recognize L must remember how many 0's have been seen so far (i.e. unlimited number of possibilities).
- But, this argument is not a proof!
- Our intuition can sometimes lead us astray.

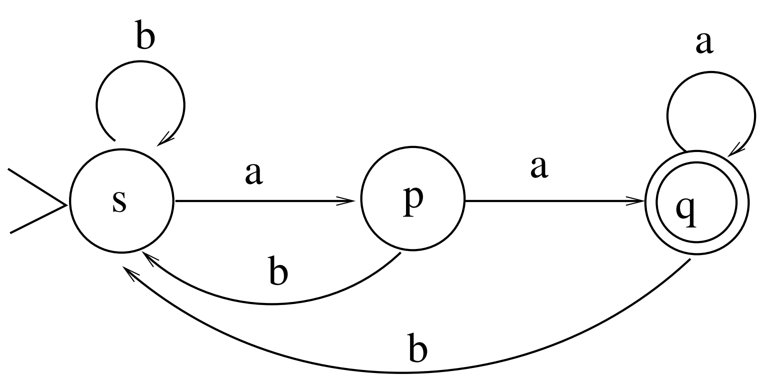
$$L_1 = \{w | w \text{ has an equal number of 0's and 1's } \}$$

$$L_2 = \{w | w \text{ has an equal number of occurrences of 01 and 10 as substrings } \}$$

Are they regular?

Pumping Theorem (or Pumping Lemma)

- It is a property of all (infinite) regular languages.
- Informally, the P.T. says that, for any regular language L, every string in L that is longer than a certain *special length* can be "*pumped*" (with the resulting string always in L).
- That is, for any regular language L that is infinite, there must be some repetitive pattern(s) that correspond to a Kleene star in a regular expression or a cycle in a state diagram.



The Pigeonhole principle

Suppose there are $n \geq 1$ pigeonholes and $m > n$ pigeons. Then at least one of the pigeonhole must contain at least two pigeons.

Equivalently,
If A and B are nonempty finite sets and $|A| > |B|$, then there is no one-one function from A to B.

Proof: Let $f : A \to B$.

- *Basis step:* $|B| = 0$.
  No function can be defined. Hence no one-one function can be defined.
- *Induction hypothesis:* Suppose f is not one-one whenever $|A| > |B|$, $|B| \leq n$ for some $n \geq 0$
- *Inductive step:* Consider the case $|B| = n + 1$.
  Choose some $a \in A$.

Case i) There is another $a' \in A$ s.t. $f(a) = f(a')$.
Then f is not one-one.

Case ii) a is the only element mapped to $f(a)$.

  * Define $g : A - \{a\} \to B - \{f(a)\}$ as $g(x) = f(x)$ for all $x \in A - \{a\}$.

- $|A - \{a\}| = |A| - 1 > |B| - 1 = |B - \{f(a)\}|$
- $|B - \{f(a)\}| = n$; by induction hypothesis, g is not one-one.
- Consequently, f is also not one-one. $\square$

From Pigeonhole principle, we have:
In a graph of n vertices, if there is a path from vertex a to vertex b, then there must be a path from a to b of length $\leq n$.

Pumping Theorem

Theorem 1 *Let L be a regular language. Then there exists an integer $n \geq 1$ such that every string w in L of length at least n can be written as $w = xyz$ (i.e., w can be divided into three substrings), satisfying the following conditions:*

1. $|y| > 0$ *(equivalently, $y \neq e$)*
2. $|xy| \leq n$
3. *for all $i \geq 0$, $xy^i z \in L$*

Note: y is the substring that can be pumped (removed or repeated any number of times, and the resulting string is always in L). (i) means the loop y to be pumped must be of length at least one; (ii) means the loop must occur within the first n characters. There is no restriction on x and z, they can be e.

Proof:

- Since L is regular, there exists a DFA M that accepts L.
- Let n be the number of states in M.
- Let w be *any* string in L of length $\geq n$.
  Denote the first n symbols of w by $\sigma_1 \ldots. \sigma_n$, i.e., $w = \sigma_1 \ldots. \sigma_n v$ where $v \in \Sigma^*$. Suppose the string $\sigma_1 \ldots. \sigma_n$

drives the machine through the sequence of states (not necessarily distinct) $q_0, q_1, ..., q_n$:

$$(s = q_0, \sigma_1\sigma_2...\sigma_n) \vdash_M (q_1, \sigma_2...\sigma_n) \vdash_M ... \vdash_M (q_n, e)$$

- Since M has n states, by Pigeonhole Principle, there exist i and j, $0 \leq i < j \leq n$ such that $q_i = q_j$.
  That is, the substring $\sigma_{i+1}\sigma_{i+2}...\sigma_j$ forms a loop.
- Let $x = \sigma_1...\sigma_i, \quad y = \sigma_{i+1}...\sigma_j, \quad z = \sigma_{j+1}...\sigma_n v$.
  Since i cannot be equal to j, we have $y \neq e$.
  Since $j \leq n$, we have $|xy| \leq n$.
- y could either
  - be removed from w, or
  - be repeated many times
  and M would still accept the resulting string.
  Hence, M accepts $xy^i z$ for all $i \geq 0$.

Note:

1. If no strings in L are of length $\geq n$ (i.e., L is finite), then the theorem is vacuously true.
2. Without the condition $|xy| \leq n$, the statement would still be true. This condition means that the *first* repetition is guaranteed to appear within the first n symbols of the string.

Using Pumping Theorem

The pumping theorem is one of the most sophisticated theorems in this course because it involves several quantifiers. Writing the theorem using quantifiers provides a clearer picture:

$\exists n \geq 1$, such that
$\forall w \in L, |w| \geq n$,
$\exists x, y, z$ such that $w = xyz$ and $y \neq e$, $|xy| \leq n$, and
$\forall i \geq 0$, $xy^i z \in L$.

We use the pumping theorem to prove that a language is NOT regular. To prove that L is not regular, we show the following, which contradicts the P.T.:

$\forall n \geq 1$,
$\exists w \in L, |w| \geq n$,
$\forall x, y, z$ such that $w = xyz$ and $y \neq e$, $|xy| \leq n$,
but $\exists i \geq 0$, $xy^i z \notin L$.

When you read the next few examples, pay attention to how the proofs are carefully written to correctly consider *for all* and *there exists* quantifiers.

A sketch of how to use P.T. to prove that a language is not regular.

- Assume L is regular.
- Then, by Pumping Theorem, there exists an integer n such that *all* strings of length $\geq n$ in L can be pumped.
- Find *one* string w in L that has length $\geq n$ and is not pumpable no matter how you split it into 3 parts, i.e.,:
  - Choose *one* string $w \in L$ with $|w| \geq n$.
  - Consider *all* valid ways of splitting w into x, y, z.
  - For each way of splitting, demonstrate *one* value i such that $xy^i z \notin L$
- The existence of one string w that is not pumpable contradicts the Pumping Theorem.
- Hence L is not regular.

Use of the Pumping Theorem

Example 1: $L = \{a^i b^i : i \geq 0\}$ is not regular.

- Suppose L is regular. Then the Pumping Theorem applies.
- Let n be the integer given in the Pumping Theorem.
- Choose $w = a^n b^n$ (Note: $w \in L$ and $|w| \geq n$).
- By P.T, w can be written as $w = xyz$ such that (i) $|xy| \leq n$, (ii) $y \neq e$, and (iii) for all $i \geq 0$, $xy^i z \in L$.
- Since $|xy| \leq n$, the string y can only consist of a's. Since $y \neq e$, y must contain at least one a. That is, $y = a^k$ for $0 < k \leq n$. (Note: here we are considering **all** valid ways of splitting w into 3 parts, satisfying conditions (i) and (ii).)
- Consider $i = 0$, we have $xy^0 z = xz = a^{n-k} b^n \notin L$. (Note: this result applies to all values of k. For any value of k, we can pick $i = 0$ to demonstrate that that particular splitting doesn't work). Contradicting the Pumping Theorem.
- Hence, L is not regular.

Example 2: $L = \{a^i : i \text{ is a prime number}\}$ is not regular.

- Assume it is regular. Then the Pumping Theorem applies.
- Let n be the integer in the Pumping Theorem.
- Consider $w = a^s \in L$ where s a prime and $s \geq n$.
- By P.T, w can be written as $w = xyz$ such that $|xy| \leq n$, $y \neq e$, and for all $i \geq 0$, $xy^iz \in L$.
- Since $|xy| \leq n$ and $y \neq e$, thus

  $$x = a^p, \; y = a^q, \; z = a^r$$

  where $p + q + r = s$, $p + q \leq n$, $q > 0$, and $p, r \geq 0$.
- Consider $xy^{p+2q+r+2}z$. Then, the number of a's $= p + q(p + 2q + r + 2) + r = (q + 1)(p + 2q + r)$, which is a product of two natural numbers each greater than 1. Non-prime! Hence this string is not in L. Contradicting the P.T.

  Note: Had you chosen to pump y $p + r$ times, then $p + q(p + r) + r = (q + 1)(p + r)$ would not work since $p + r$ may be 0 or 1!
- Hence L is not regular.

Proving a language is not regular

We can also prove that a language is not regular by using closure properties of regular languages to arrive at a contradiction.

Example:
Prove that the following L is not regular:
$L = \{w \in \{0, 1\}^* : w$ has an equal number of 0's and 1's $\}$

Proof:
Observe that $L \cap L(0^*1^*) = \{0^n 1^n : n \geq 0\}$.

If L is regular, then so would be $L \cap L(0^*1^*)$ by closure under intersection. But, we have shown that $\{0^n 1^n : n \geq 0\}$ is not regular. Contradiction.

Question: Can we prove this using P.T. by picking the same string $0^n 1^n$ where n is the number of states like in Example 1 and proceed in exactly the same way?

P.T. is a necessary, but not sufficient condition for regular languages.

One example (from Wikipedia): $\Sigma = \{1, 2, 3\}$, $L = \{uvwxy : u, y \in \Sigma^*, v, w, x \in \Sigma, v = w \text{ or } v = x \text{ or } x = w\} \cup \{w : w \in \Sigma^* \text{ and precisely } 1/7 \text{ of the symbols in } w \text{ are 3's}\}$.

This L is non-regular but satisfies the P.T.

Logic questions:

- Can you use P.T. to prove that a language is regular? No.
- For any given non-regular language L, can you use P.T. to prove that it is non-regular? No.