Course 9
Properties of Regular Languages

The structure and the content of the lecture is based on http://www.eecs.wsu.edu/~ananth/CptS317/Lectures/index.htm

Topics

1) How to prove whether a given language is not regular?

Some languages are *not* regular

When is a language is regular?

if we are able to construct one of the following: DFA *or* NFA *or* ε -NFA *or regular expression* or *grammar of type 3*

When is it not?

If we can show that no FA can be built for a language.

How to prove languages are ***not*** regular?

What if we cannot come up with any FA?

A) Can it be language that is not regular?

B) Or is it that we tried wrong approaches?

How do we *decisively* prove that a language is not regular?

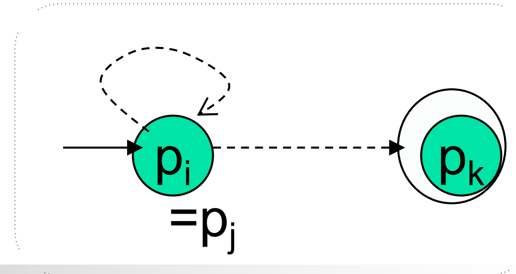
> *"The hardest thing of all is to find a black cat in a dark room, especially if there is no cat!" -Confucius*

Example of a non-regular language

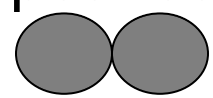
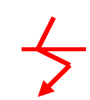
Let L = {w | w is of the form 0^n1^n , for all $n \geq 0$}

- *Hypothesis:* *L is not regular*
- Intuitive rationale: How do you keep track of a running count in an FA?
- A more formal rationale:
  - By contradition, if L is regular then there should exist a DFA for L.
  - Let k = number of states in that DFA.
  - Consider the special word $w = 0^k1^k \Rightarrow w \in L$
  - DFA is in some state p_i, after consuming the first i symbols in w

Uses Pigeon Hole Principle



Rationale…

- Let $\{p_0, p_1, \ldots p_k\}$ be the sequence of states that the DFA should have visited after consuming the first k symbols in w which is 0^k
- But there are only k states in the DFA!
- ==> at least one state should repeat somewhere along the path (by  +  Principle)
- ==> Let the repeating state be $p_i = p_j$ for $i < j$
- ==> We can fool the DFA by inputing $0^{(k-(j-i))}1^k$ and still get it to accept (note: k-(j-i) is at most k-1).
- ==> DFA accepts strings w/ unequal number of 0s and 1s, implying that the DFA is wrong! 

The Pumping Lemma for Regular Languages

What it is?

The Pumping Lemma is a property of all regular languages.

How is it used?

A technique that is used to show that a given language is **not** regular.

It can not be used to show that a given language is regular.

Pumping Lemma for Regular Languages

Let L be a regular language.

Then *there exists* some constant ***N*** such that *for every* string $w \in L$ s.t. $|w| \geq N$, *there exists* x, y, z, $w=xyz$, such that:

1. $y \neq \varepsilon$
2. $|xy| \leq N$
3. For all $k \geq 0$, all strings of the form $xy^kz \in L$

This property should hold for all regular languages.

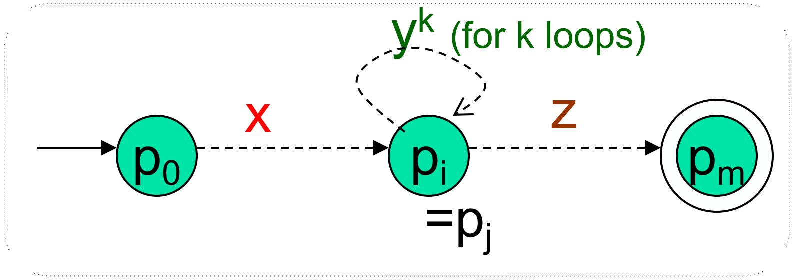
Definition: *N* is called the "Pumping Lemma Constant"

Pumping Lemma: Proof

- L is regular => it should have a DFA.
  - Set N := number of states in the DFA
- Any string $w \in L$, s.t. $|w| \geq N$, should have the form: $w = a_1 a_2 \ldots a_m$, where $m \geq N$
- Let the states traversed after reading the first N symbols be: $\{p_0, p_1, \ldots p_N\}$
  - ==> There are N+1 p-states, while there are only N DFA states
  - ==> at least one state has to repeat i.e, $p_i = p_j$ where $0 \leq i < j \leq N$ (by PHP)

Pumping Lemma: Proof…

- => We should be able to break w=xyz as follows:
  - $x=a_1a_2..a_i$; $y=a_{i+1}a_{i+2}..a_J$; $z=a_{J+1}a_{J+2}..a_m$
  - x's path will be $p_0..p_i$
  - y's path will be $p_i\, p_{i+1}..p_j$ (but $p_i=p_j$ implying a loop)
  - z's path will be $p_jp_{J+1}..p_m$
- Now consider another string $w_k=xy^kz$, where k≥0

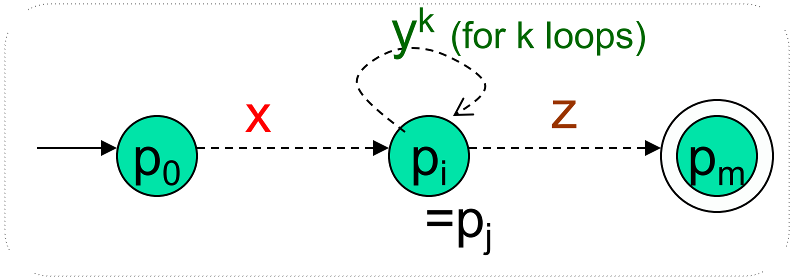


- Case k=0
  - DFA will reach the accept state p_m
- Case k>0
  - DFA will loop for y^k, and finally reach the accept state p_m for z
- In either case, $w_k \in L$

This proves part (3) of the lemma

Pumping Lemma: Proof…

- For part (1):
  - Since i<j, y $\neq \varepsilon$



- For part (2):
  - By PHP, the repetition of states has to occur within the first N symbols in w
  - ==> $|xy| \leq N$

□

Using P.L. to prove non-regularity

1. L regular $\Rightarrow$ L satisfies P.L.
2. L non-regular $\Rightarrow$?

Idea of 3.: Instead of proving A⟹B, prove ¬B⟹ ¬A (proof by contrapositive)

3. L non-regular $\Leftarrow$ L does not satisfy P.L.

Idea of 3.: Instead of proving A⟹B, assume A and derive a contradiction, i.e. ¬B

1. L regular and L does not satisfy P.L.

Note: This N can be anything (need not necessarily be the #states in the DFA.)

Example 1

Claim L = {w | $w = 0^n 1^n, n \geq 1$} is non-regular.

- Proof:

  By contradiction, assume L is regular and derive a contradiction

  - Then, there exists *N* s.t. for all *w* ∈ *L*. Let w = $0^N 1^N$
  - There exists x, y, z, w=xyz, such that:
    1. y≠ ε
    2. |xy|≤N
    3. For all k≥0, the string xy^kz is also in L

- w satisfies (1) and (2) above but:

(3) does not hold

w = $0^N 1^N$ = 0…00...00...01…1 ==> #0>#1
← x →← y → ←—— z ——→

w = $0^N 1^N$ = 0…01...11...11...1 ==> #0<#1
← x →← y → ←—— z ——→

w = $0^N 1^N$ = 0…00..1...0...1...1…1==> w has not the required shape
← x →←—— y ——→← z →

Example 2

Claim L_{eq} = {w | w is a binary string with equal number of 1s and 0s} is non-regular.

Assume L_{eq} be regular. Then there exists N such that for every string $w \in L$ s.t. $|w| \geq N$, there exists x, y, z with w=xyz, such that: (1) $y \neq \varepsilon$, (2) $|xy| \leq N$ (3) Exists $k \geq 0$ and $xy^kz \in L$.

Take N=N*, and $w=0^{N^*}1^{N^*}$ ($|w|=2N^* \geq N^*$), $w \in L_{eq}$.

Proof proceeds like in Example 1.

Example 3

Prove $L = \{0^n10^n \mid n \geq 1\}$ is not regular.

Assume L_{eq} be regular and derive a contradiction. Then there exists N such that for every string $w \in L$ s.t. $|w| \geq N$, there exists x, y, z with w=xyz, such that: (1) $y \neq \varepsilon$, (2) $|xy| \leq N$ (3) For all $k \geq 0$, all strings of the form $xy^kz \in L$.

Take N=N*, and $w=0^{N^*}10^{N^*}$ ($|w|=2N^*+1 \geq N^*$). Then w can be divided into 3 parts: x=0…0 (length N*-2), y=01,($|xy|= N^*-2 +2 \leq N^*$), z=0…0 (length N*). Then $|xy|=N^* \leq N^*$. For k=0 we have xz=0…0 (no 1). So not in L_{eq}.

We found a counterexample for which the PL does not hold. Hence L_{eq} is not regular.

Example 4

Prove $L = \{1^n \mid n \text{ is prime}\}$ is not regular.

Assume L_{eq} be regular. Then there exists N such that for every string $w \in L$ s.t. $|w| \geq N$, there exists a way to break w into three parts, w=xyz, such that: (1)$y \neq \varepsilon$, (2) $|xy| \leq N$ (3) For all $k \geq 0$, all strings of the form $xy^kz \in L$.

Take N=p, and $w=1^p$($|w|=p \geq p$ and p - prime). Then w can be divided into 3 parts: $|y|=l \geq 1$ (cond. (1) – is satisfied, and assume $|xy| <= p$ s.t. (2) is satisfied).

Trying to prove (3): Let k=p+1. We have $| xy^{p+1}z | = | xyz | + |y^p| = p+p|y| = p(1+|y|)$ which is not always a prime number, e.g. p=3, |y|=1, $| xy^{p+1}z| = 3(1+1) = 6$.