CSC-257
Theory Of Computation
(BSc CSIT, TU)

Ganesh Khatri
kh6ganesh@gmail.com

Proving a Language not to be Regular

- It is shown that the class of language known as regular language has at least four different descriptions.
- They are the languages accepted by DFAs, by NFAs, by Є-NFAs, and defined by RE.
- Not every language is Regular.
- To show that a language is not regular, the powerful technique used is known as Pumping Lemma

Pumping Lemma

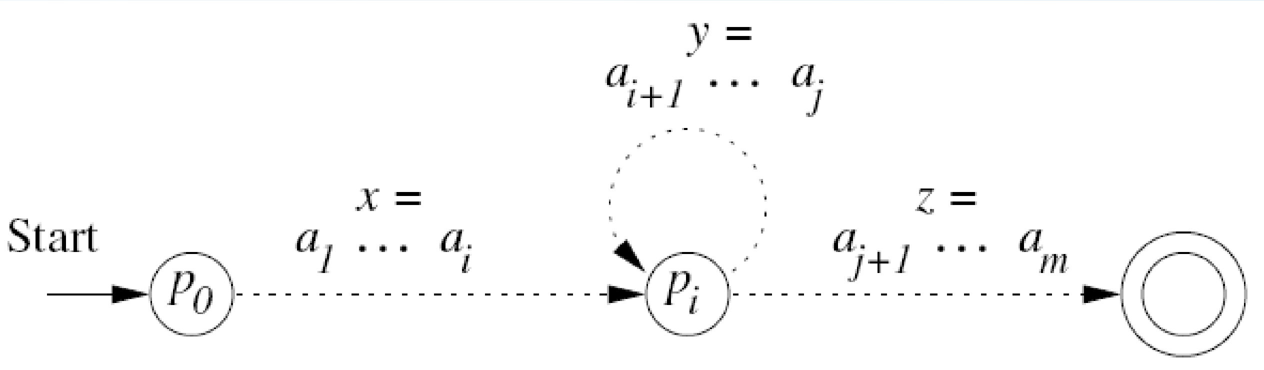
- **Statement :** Let L be a regular language. Then there exists a constant n such that for every string w in L such that $|w| >= n$, we can break w into three strings, $w = xyz$, such that :
  - $y \neq \varepsilon$
  - $|xy| <= n$
  - for all $k >= 0$, the string xy^kz is also in L
- That is, we can always find a nonempty string y not too far from the beginning of w that can be "pumped"; that is, repeating y any number of times, or deleting it (the case k=0) keeps the resulting string in language L

Pumping Lemma

- **Proof :** Suppose, L is a regular language for some DFA A. Suppose, A has n states.
- Now, consider any string w of length n or more, say $w = a_1a_2.... a_m$, where $m>=n$ and each a_i is an input symbol.
- For $i = 0,1,.....n$, define state p_i to be $\delta^*(q_0, a_1a_2...a_i)$, where δ be the transition function of A, and $q_0$is the start state of A.
- That is, p_i, is the state A is in after reading the first i symbols of w. Note that $p_0 = q_0$.
- By the pigeonhole principle, it is not possible for the n+1 different p_i's for $i = 0,1,2,\dots,n$ to be distinct, since there are only n different states.
- Thus, we can find two different integers i and j, with $0<=i<=j<=n$, such that $p_i=p_j$

Pumping Lemma

- Now, we can break w = xyz as follows :
  - $x = a_1a_2....a_i$
  - $y = a_i+1a_{i+2}......a_j$
  - $z = a_{j+1}a_{j+2}.....a_m$
- That is, x takes us to p_i once; y takes us from p_i back to p_i (since $p_i = p_j$), and z is the balance of w



Every string longer than the number of states must cause a state to repeat

- So we can say A accepts $a_1a_2............a_i(a_{i+1}............a_j)^K a_{j+1}...............a_m$ for all $k\geq0$
- Hence, xy^kz ε L for all $k\geq0$.

Exercises

- Show that language, $L=\{0^n1^n|n \geq 0\}$ is not a regular language
- **Solution :** Let L is a regular language. Then by pumping lemma, there are strings u, v, w with $v \geq 1$ such that $uv^kw \varepsilon L$ for $k \geq 0$
- **Case I:** Let v contain 0's only. Then, suppose $u = 0^p$, $v = 0^q$, $w = 0^r1^s$; Then we must have $p+q+r = s$ (as we have 0^n1^n) and $q>0$
- Now, $uv^kw = 0^p(0^q)^k0^r1^s = 0^{p+qk+r}1^s$
- Only these strings in $0^{p+qk+r}1^s$ belongs to L for k=1 otherwise not.
- Hence, L is not regular.

Exercises

- Show that language, $L=\{0^n1^n|n \geq 0\}$ is not a regular language
- **Solution :** Let L is a regular language. Then by pumping lemma, there are strings u, v, w with $v \geq 1$ such that $uv^kw \in L$ for $k \geq 0$
- **Case II:** Let v contains 1's only. Then $u= 0^p1^q$, $v= 1^r$, $w=1^s$, then $p= q+r+s$ and $r>0$
- Now, $0^p1^q(1^r)^k1^s = 0^p1^{q+rk+s}$
- Only those strings in 0^p1^{q+rk+s} belongs to L for $k = 1$ otherwise not
- Hence, L is not regular.

Exercises

- Show that language, $L=\{0^n1^n | n \geq 0\}$ is not a regular language
- **Solution :** Let L is a regular language. Then by pumping lemma, there are strings u, v, w with $v \geq 1$ such that $uv^kw \in L$ for $k \geq 0$
- **Case III:** Let, V contains 0's and 1's both. Then, suppose, $u = 0^p$, $v = 0^q1^r$, $w = 1^s$; $p+q = r+s$ and $q+r>0$
- Now, $uv^kw = 0^p(0^q1^r)^k1^s = 0^{p+qk}1^{rk+s}$
- Only those strings in $0^{p+qk}1^{rk+s}$ belongs to L for k=1, otherwise not. (As it contains 0 after 1 for k>1 in the string)
- Hence, L is not regular

Exercises

- Prove that $L = \{0^i \mid i \text{ is a perfect square}\}$ is not a regular language.
- **Proof :** Assume that L is regular and let m be the integer guaranteed by the pumping lemma.
- Now, consider the string $w = 0^{m\text{\^{}}2}$ where $m <= |w|(m^2)$
- Clearly $w \in L$, so w can be written as $w = xyz$ with $|xy| \leq m$ and $|y| > 0$.
- Consider what happens when $i = 2$. That is, look at xy^2z.
- Then, we have $m^2 = |w| < |xy^2z| \leq m^2 + m = m(m + 1) < (m + 1)^2$.
- That is, the length of the string xy^2z lies between two consecutive perfect squares.
- This means $xy^2z \notin L$ contradicting the assumption that L is regular.

Exercises

1. Prove that $L = \{ a^n : n \text{ is a prime number} \}$ is not regular
2. Prove that $L = \{ (10)^p 1^q : p, q \in \mathbf{N}, p \geq q \}$ is not regular

Exercises

1. Prove that $L = \{ a^n : n \text{ is a prime number} \}$ is not regular

- **Proof :** For the sake of contradiction, assume that L is regular.
- Let, $w = a^n \in L$ and $|w| = n$
- The Pumping Lemma must then apply; let k be the pumping length
- Let n be any prime number at least as large as k (since $|w| >= k$)
- Since $|w| \geq k$, it must be possible to split w into three pieces xyz satisfying the conditions of the Pumping Lemma.
- Now consider the string xy^iz. The string xy^iz has length $n + (i - 1)|y|$
- Letting $i = n + 1$, we have
- $n + (i - 1)|y| = n + ((n + 1) - 1)|y|$
- $= n + n|y| = n(1+|y|)$ which is a composite number.
- So, $xy^iz \notin L$. Here we got a contradiction. Hence, L is not regular.

Minimization of DFA

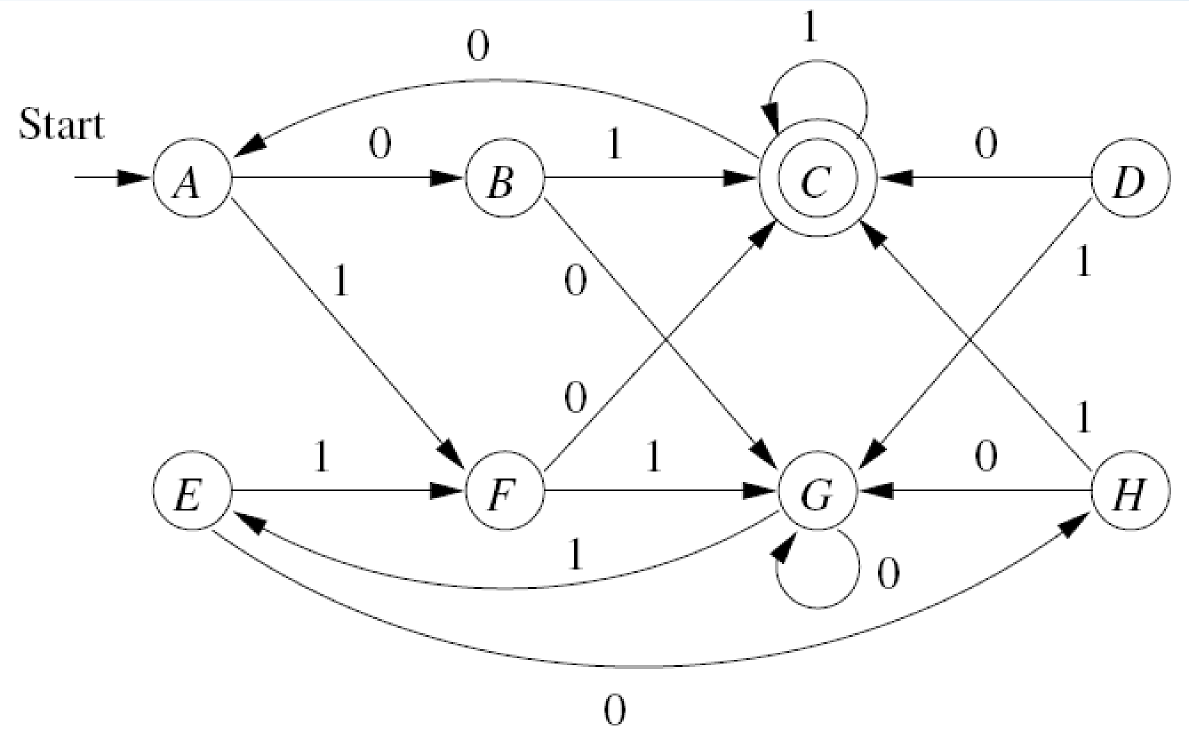
- **Equivalent States :**
  - Two states p & q are called equivalent states, denoted by $p \equiv q$ if and only if for each input string x, $\delta^*(p, x)$ is a final state if and only if $\delta^*(q, x)$ is a final state
- **Distinguishable States :**
  - Two states p & q are said to be distinguishable states if (for any) there exists a string x, such that $\delta^*(p, x)$ is a final state $\delta^*(q, x)$ is not a final state
- Suppose a DFA M, that accepts a language L (M).
- For minimization of M, the table filling algorithm is used.

Minimization of DFA

- For identifying the pairs of states (p, q) with $p \neq q$;(distinguishable)
  - List all the pairs of states for which $p \neq q$
  - Make a sequence of passes through each pairs
  - On first pass, mark the pair for which exactly one element is final (F)
  - On each sequence of pass, mark the pair (r, s) if for any $a \in \Sigma$, $\delta(r, a) = p$ and $\delta(s, a) = q$ and (p, q) is already marked
  - After a pass in which no new pairs are to be marked, stop
  - Then marked pairs (p, q) are those for which $p \equiv q$ and unmarked pairs are distinguishable.
- Note : any pair of states that we do not find distinguishable are equivalent and distinguishable pairs should be marked.

Minimization of DFA : Example

- Minimize the following DFA.



- Let us first find which pairs of states are distinguishable and equivalent using table filling algorithm as:
- Here, For the basis, since C is the only accepting state, so we put x in each pair that involves C. [p ≠ q]
- Now we know some distinguishable pairs, we can discover others.
- For instance, since {C,H} is distinguishable, and states E and F go to H and C, respectively on input 0.
- Again, pair {E,F} is also distinguishable because, on input 0, {E,F} goes to {H,C} which is distinguishable pair and so on.

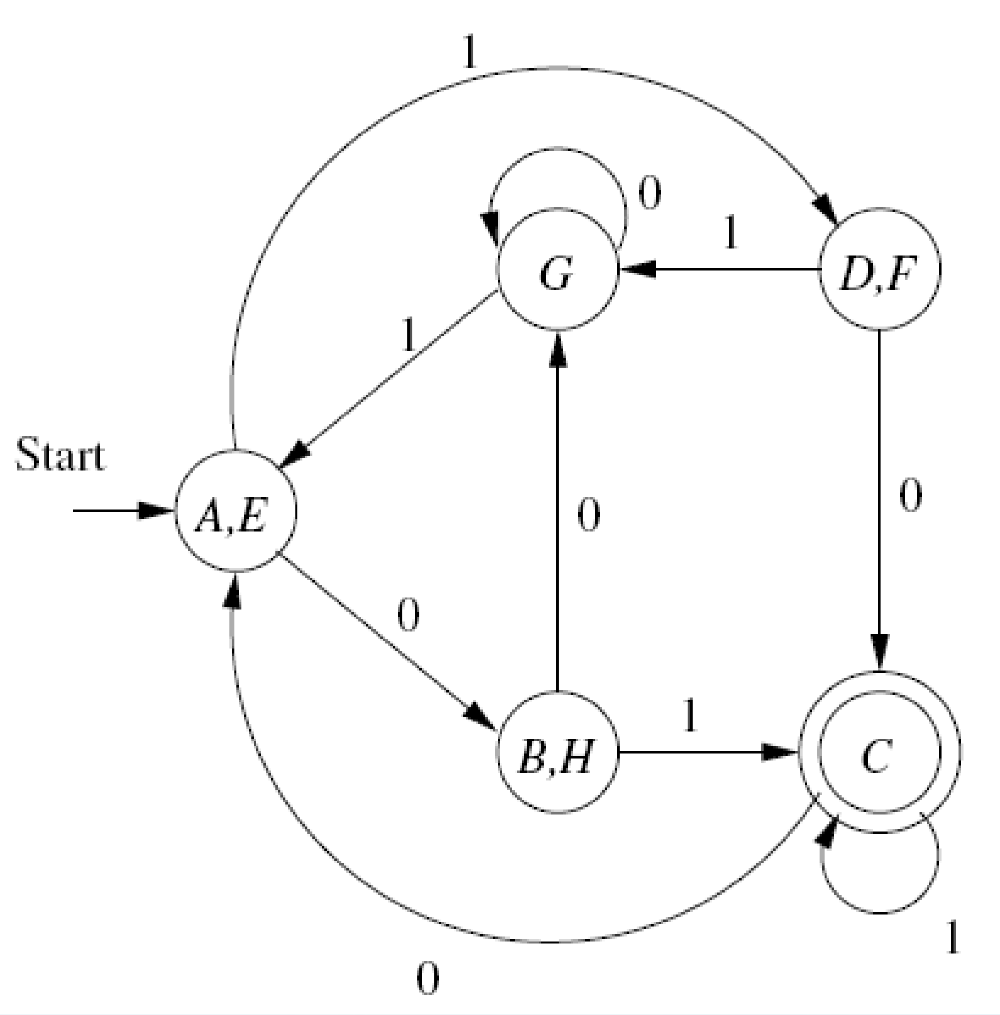
Minimization of DFA : Example

- Minimize the following DFA.
- ($\delta(A, 1), \delta(B, 1)$) = (F, C) – marked [since, F ≠ C]
- ($\delta(A, 0), \delta(D, 0)$) = (B, C) – marked [since, B ≠ C]
- And so on.
- At last, those pairs of states which are not marked will be equivalent states.

B	x						
C	x	x					
D	x	x	x				
E		x	x	x			
F	x	x	x		x		
G	x	x	x	x	x	x	
H	x		x	x	x	x	x
	A	B	C	D	E	F	G

Minimization of DFA : Example

- So, minimized DFA will look like :



	A	B	C	D	E	F	G
B	x						
C	x	x					
D	x	x	x				
E		x	x	x			
F	x	x	x		x		
G	x	x	x	x	x	x	
H	x		x	x	x	x	x