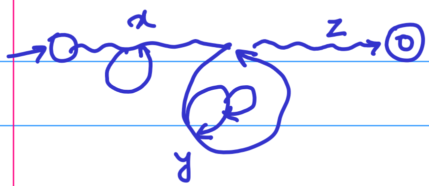
Pumping Lemma for regular languages revisited: Lecture 21

Let L be a reg. language # states in DFA for L

$|w| \geq p$ $w \in L$

$w = x.y.z$

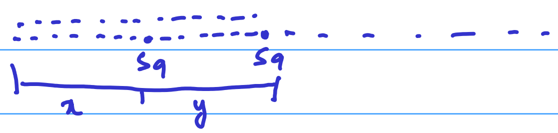


$\forall i \geq 0 \quad x y^i z \in L$

$|xy| \leq p$

$|y| \geq 1$

$s_0 \xrightarrow{a_0} s_7 \xrightarrow{a_1} s_6 \xrightarrow{a_1} s_1 \rightarrow \cdots$



$\neg$ (L is regular)

$\Uparrow \quad \Downarrow$

believer   believer

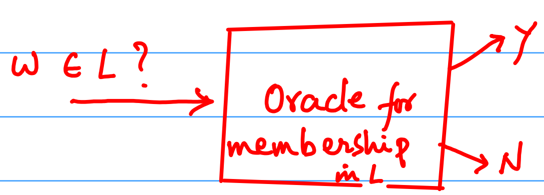
$\neg\left(\exists p > 0 \ \forall w \ (w \in L) \wedge (|w| \geq p) \ \exists x,y,z \ (w = x.y.z) \ \forall n \ (n \geq 0) \rightarrow x y^n z \in L \right)$

Adv.   Adv

$\forall p > 0 \ \exists w \ (w \in L) \wedge (|w| \geq p) \ \forall x,y,z \ (w = x.y.z) \ \exists n \ (n \geq 0) \wedge x y^n z \notin L$

1. L is regular # states in DFA for L = n.

Is L is infinite?

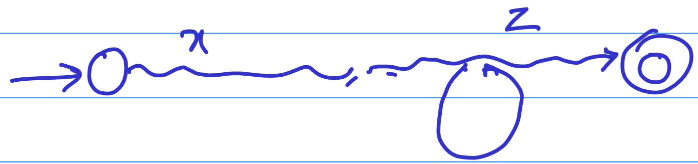


$$\sum_{i=n+1}^{2n} |\Sigma|^i$$

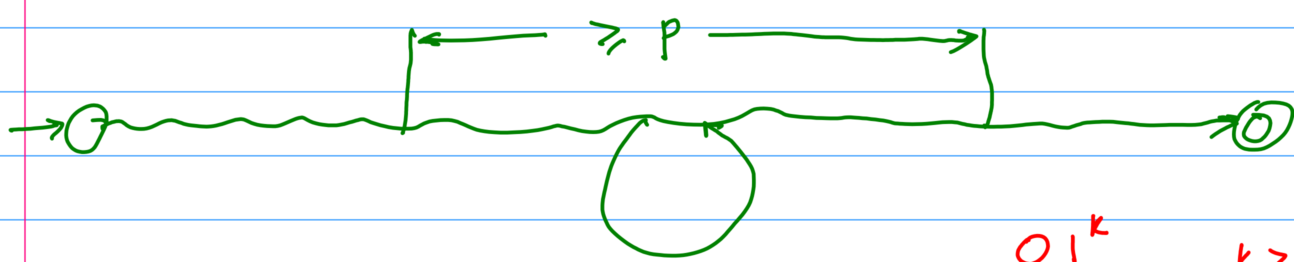
Feed all words w, $n < |w| \leq 2n$ to oracle.

✓ If $w \in L$ for some such w, then L is infinite.

If $w \notin L$ for every such w, then (L is finite)?



Angluin's L^* algorithm for learning DFAs



$0 1^k$ $\qquad$ $k \geq p$

$\underline{0}\, 1^k$

$y \rightarrow (0+1)^* \, 0 \, \underbrace{1^k}$ $\qquad$ $1^k \, 0 \, (0+1)^*$

$k =$ #1's is prime

$x = \varepsilon, \; y = 0$

001 0 11111

0^i 01 0111 11

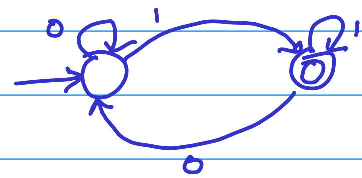
Minimization of DFAs: Lecture 22

NFA w/ ε

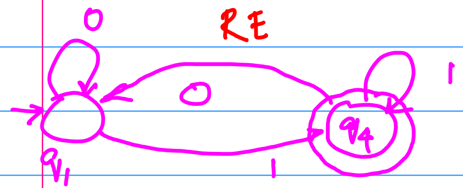
NFA w/o ε

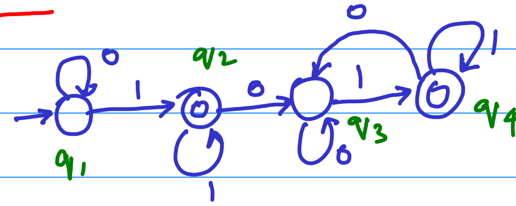
DFA

RE



$(0+1)^*1$





$\{q_1, q_3\}$

$\{q_2, q_4\}$

Indistinguishability : $\equiv$

$q_i \equiv q_j$ $\quad \forall w \in \Sigma^*$ $\quad q_i \xrightarrow{w} q_i'$

$q_j \xrightarrow{w} q_j'$

either both $q_i', q_j' \in F$

or both $\notin F$

Reflexive, symmetric ✓

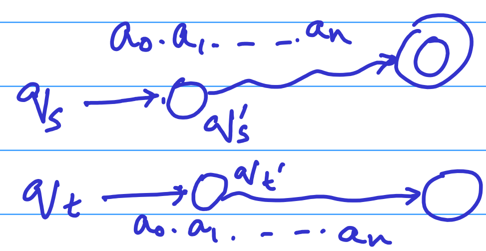
Transitive : $q_i \equiv q_j \;\wedge\; q_j \equiv q_k \Rightarrow q_i \overset{?}{\equiv} q_k$

$\exists w \in \Sigma^*$: $\quad q_i \xrightarrow{w} q_i' \qquad q_j \xrightarrow{w} q_j'$

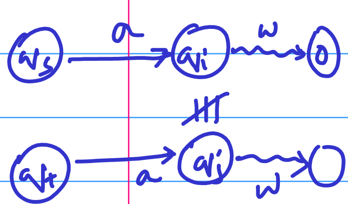
$q_k \xrightarrow{w} q_{k'}$

$\equiv$: equivalence relation

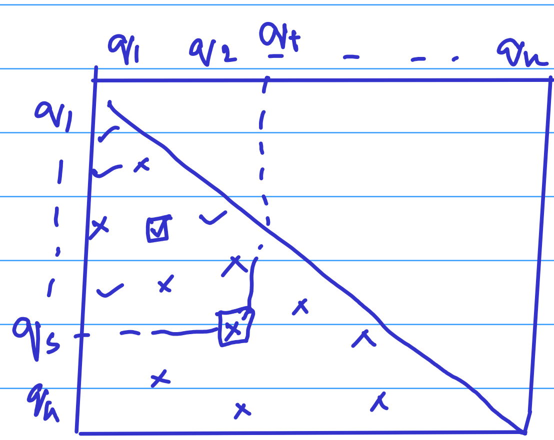
① • $q_i \not\equiv q_j \quad \forall q_i, q_j \in F \times (Q \setminus F)$ $\qquad q_i \equiv q_i \quad \forall q_i \in Q$



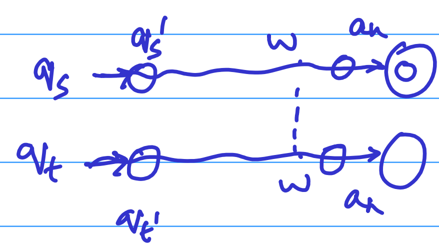
• $\forall (q_i, q_j)$ s.t. $q_i \not\equiv q_j$



$\forall (q_s, q_t)$, if $\exists a \in \Sigma$ s.t. $q_s \xrightarrow{a} q_i$, $q_t \xrightarrow{a} q_j$, then (q_s, q_t) must be in $\not\equiv$



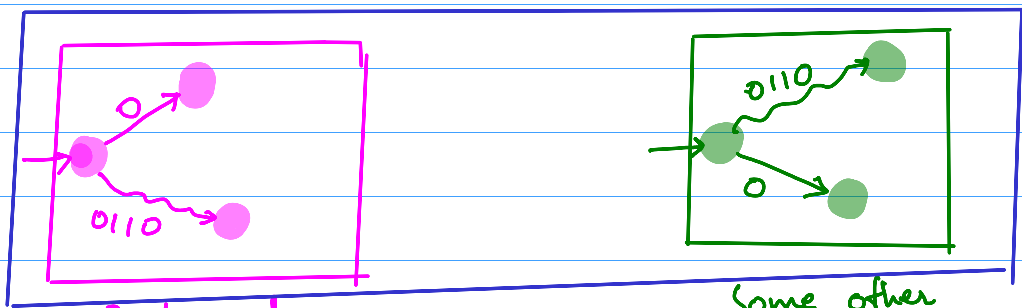
Suppose $q_s \not\equiv q_t$



$(0+1)^* \underbrace{1011011011}_{\text{10 letters}}$

Minimization of DFA and Nerode equivalence: Lecture 23

$|\equiv|$: no. of eq. classes of $\equiv$



Reduced aut. using our method
k states

Some other (DFA) aut. recognizing same language
l states

$l = k$ ~~$l < k$~~.

$L(e_1) = L(e_2)$

DFA A_1 DFA A_2

A^* B^*

$L(A_1) \subseteq L(A_2)$ $L(A_1) \cap \overline{L(A_2)} \stackrel{?}{=} \phi$

$\Sigma = \{0, 1\}$ $L = (0+1)^* 1$

$q_i \equiv q_j$ $\boxed{DFA}$

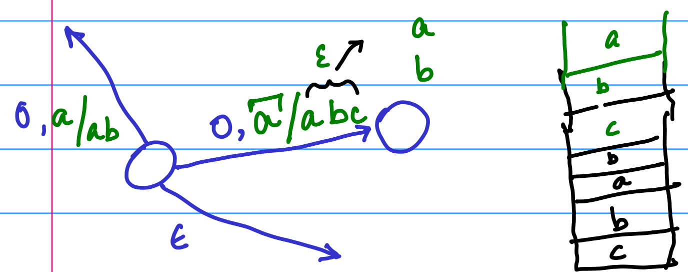
$\forall w_1, w_2 \in \Sigma^*$ $w_1 \sim_L w_2$

iff $\forall x \in \Sigma^*$ $w_1 . x$ & $w_2 . x$

are both in L

or both not in L

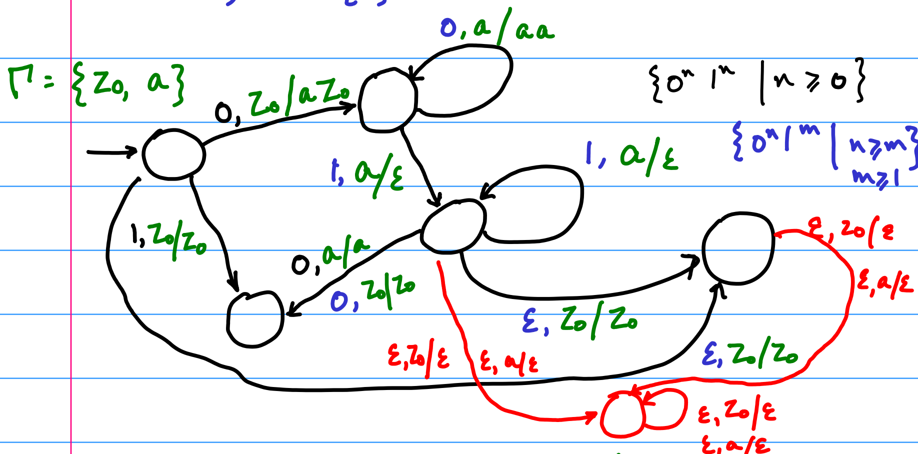
Pushdown Automata basics: Lecture 25



$$(Q, \Sigma, \Gamma, q_0, Z_0, \delta, F)$$

$$\delta : Q \times \Sigma \cup \{\varepsilon\} \times \Gamma \rightarrow 2^{Q \times \Gamma^*}$$

$$Q \times \Sigma \cup \{\varepsilon\} \rightarrow 2^{Q}$$



$\{0^n 1^n \mid n \geq 0\}$

$\{0^n 1^m \mid n \geq m, m \geq 1\}$

Pushdown Automata (PDA)

A

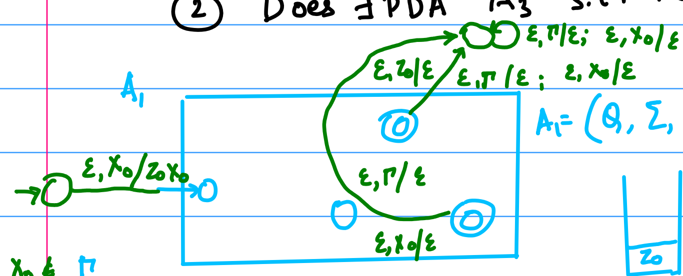
Acceptance by final state $L(A)$

" " empty stack. $N(A)$
or null

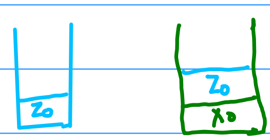
Given PDA A_1,

✓ (1) Does $\exists$ PDA A_2 s.t. $L(A_1) = N(A_2)$?

(2) Does $\exists$ PDA A_3 s.t. $N(A_1) = L(A_3)$?



$A_1 = (Q, \Sigma, \Gamma, q_0, Z_0, \delta, F)$



Class note scribblings for lecture 26 (equivalence of PDA acceptance by final state and empty stack) seems missing