5/10/2010

REG = $\{ L \mid L$ is regular i.e.
$L = L(M)$ for some DFA $M \}$

CFL = $\{ L \mid L$ is context free i.e.
$L = L(G)$ for some CFG $G \}$

THM REG $\subseteq$ CFL ?

IDEA: DFA: $q_i \xrightarrow{a} q_j$ | $q_k \xrightarrow{b} q_m$

CFG:

$A_i \longrightarrow a A_j$ | $A_k \longrightarrow \varepsilon \mid b A_m$

To accept: the symbol that corresponds to the accept state generates $\varepsilon$. (It can also generate other things, if it has transitions going out of it.)

Pf Let $L \in REG$

$\exists$ DFA $M$ s.t. $L(M) = L$ $\quad M = (Q, \Sigma, \delta, q_0, F)$

Create CFG $G_M$ $(V, \Sigma, R, S)$

$V = \{ A_0, A_1, \dots, A_{|Q|-1} \}$

$\Sigma = \Sigma$

$S = A_0$

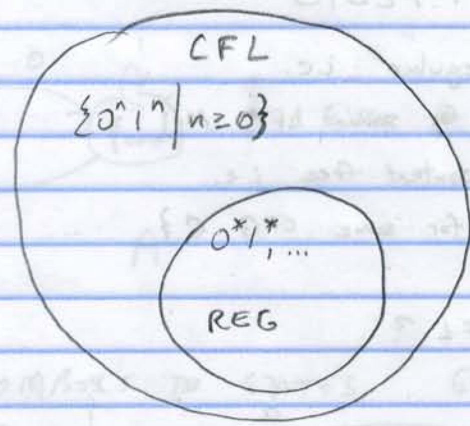
$R = \bigcup \{ A_i \rightarrow a A_j \mid \delta(q_i, a) = q_j \}$

$\bigcup \{ A_k \rightarrow \varepsilon \mid q_k \in F \}$

$L = L(M) = L(G_M) \Rightarrow REG \subseteq CFL$

$\cancel{CFL \subsetneq REG}$ ?

$CFL \subsetneq REG$




Question: Is there some kind of machine that can recognize CFL's?

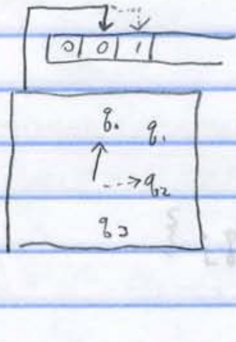
Can we add some data structure to DFA / NFA to accept L?

How about a stack?

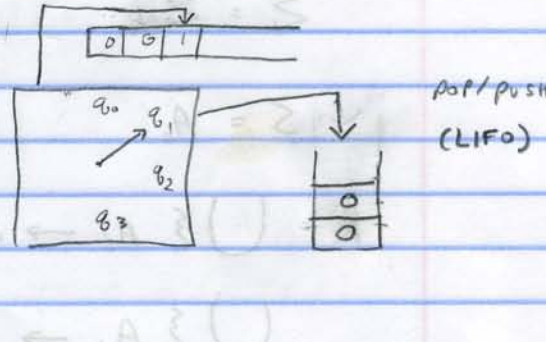
EXAMPLE: $L = \{0^n 1^n \mid n \geq 0\}$

Put all the zeroes on the stack. For every 1, pop a 0. If stack empty at end (and you didn't run out too soon), accept.

DFA / NFA




Add a Stack (PUSHDOWN AUTOMATA (PDA))



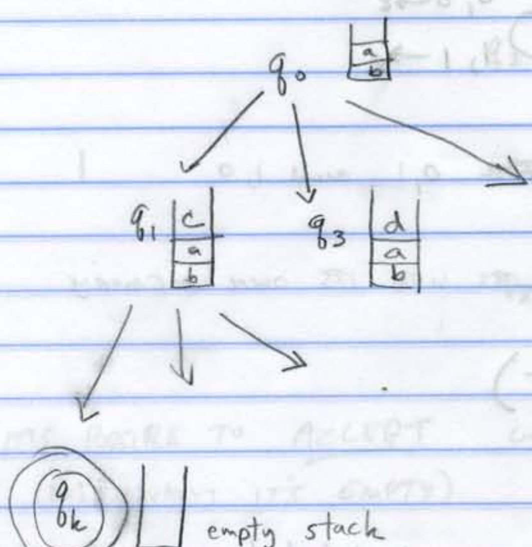
## NFA

$$\delta(q_i, a \text{ or } \varepsilon) = \{q_j, q_k, \ldots\}$$

## PDA

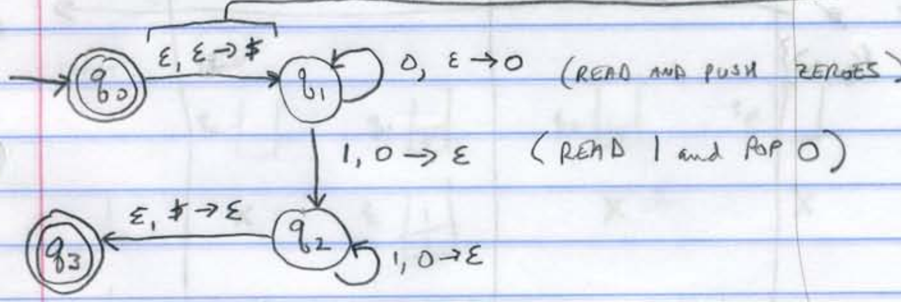
$$\delta(\underbrace{q_i}_{\text{state}}, \underbrace{a \text{ or } \varepsilon}_{\text{input}}, \underbrace{b \text{ or } \varepsilon}_{\text{popped symbol}}) = \{(\underbrace{q_j}_{\text{new state}}, \underbrace{c \text{ or } \varepsilon}_{\text{push}}), (q_k, d \text{ or } \varepsilon), \ldots\}$$

Each state has its own copy of the stack.




| | | |
|---|---|---|
| input ↓ $q_i \xrightarrow{a} q_j$ <br> NFA | input ↓ $q_i \xrightarrow{a,\, b \to c} q_j$ <br> pop push <br> PDA | NOTATION |

EXAMPLE: $L = \{0^n 1^n \mid n \geq 0\}$




HOW TO TEST FOR EMPTY STACK? INSERT SPECIAL SYMBOL, \$, AT THE BEGINNING