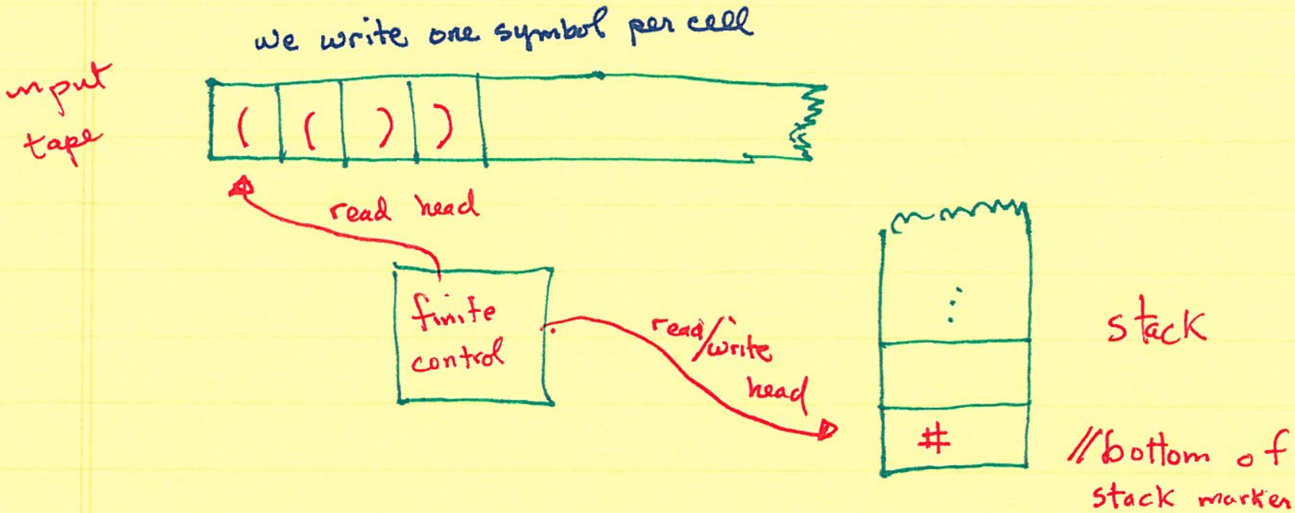
XIV

Pushdown Automata (pda)



A pushdown automaton

$P = \langle Q, \Sigma, \Gamma, \delta, q_0, F \rangle$

Q = finite set of states // consult definition for fa.

$q_0 \in Q$ = start state

F, where $F \subseteq Q$ = set of accept state // def. for fa again

however, pda have an alternate model for acceptance.

Σ = input alphabet

Γ = tape alphabet

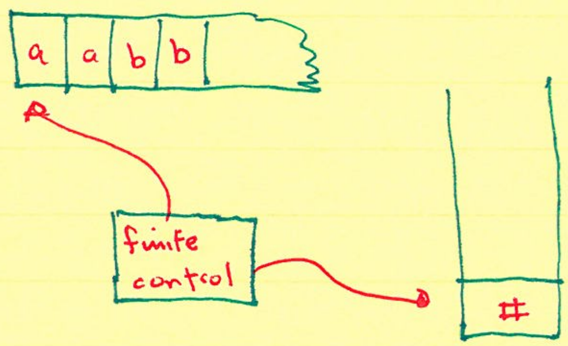
Transition function $\delta : (Q \times \Sigma \times \Gamma) \to$ (move right, change state, push or pop)

pda

- $L_1 = \{ a^n b^n \mid n \geq 1 \}$

  i.e., $L_1 = \{ ab, aabb, \ldots \}$

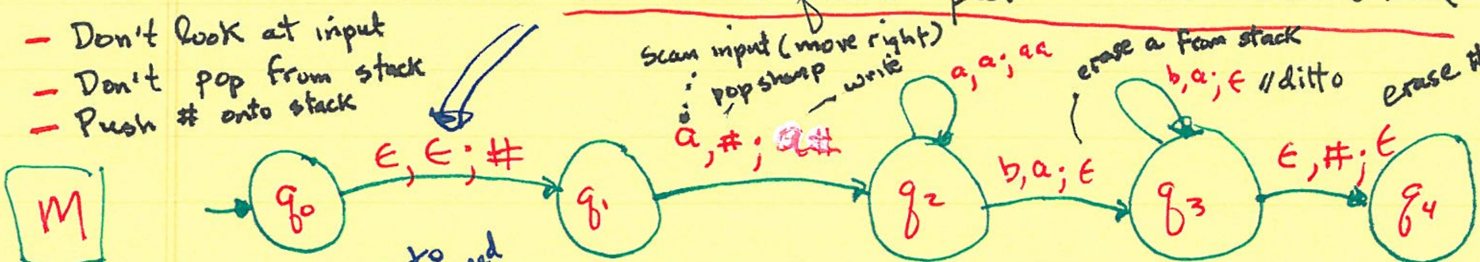
- A pda to accept L_1.



Strategy: encounter 'a' put stack symbol on stack
encounter 'b' pop stack symbol (also change state - why?).

- Our machine will accept by empty stack
  note $F = \emptyset$ here.

Let us assume we must first put # on the stack



state / input to be processed / stack contents

A trace $(q_0, aabb, \epsilon)$

$\vdash (q_1, aabb, \#)$

$\vdash (q_2, abb, a\#)$ // we place top of stack on the left.

$\vdash (q_2, bb, aa\#)$

$\vdash (q_3, b, a\#)$ change state, pop an 'a'.

$\vdash (q_3, \epsilon, \#)$ 'ϵ' in middle = we have processed w.

$\vdash (q_4, \epsilon, \epsilon)$ string processed & stack empty

— M has accepted $w = aabb$ by empty stack. ⇐

XIV

pda con't.

In general, when constructing a pda - we do not fully specify machine

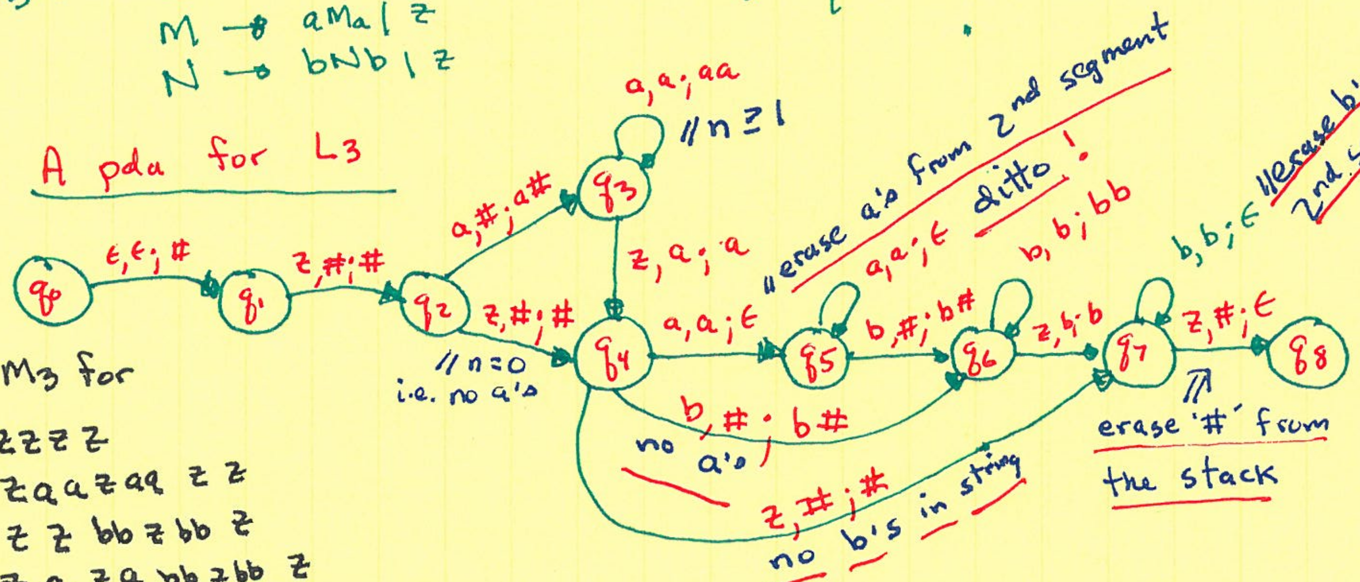
L3 revisited

G_3: $S \to zMNz$
$M \to aMa \mid z$
$N \to bNb \mid z$

$L(G_3) = \{ z a^n z a^n b^m z b^m z \mid m, n \geq 0 \}$

A pda for L3

$\boxed{M_3}$



Also, here $F = \emptyset$; accepting by empty stack

trace M_3 for

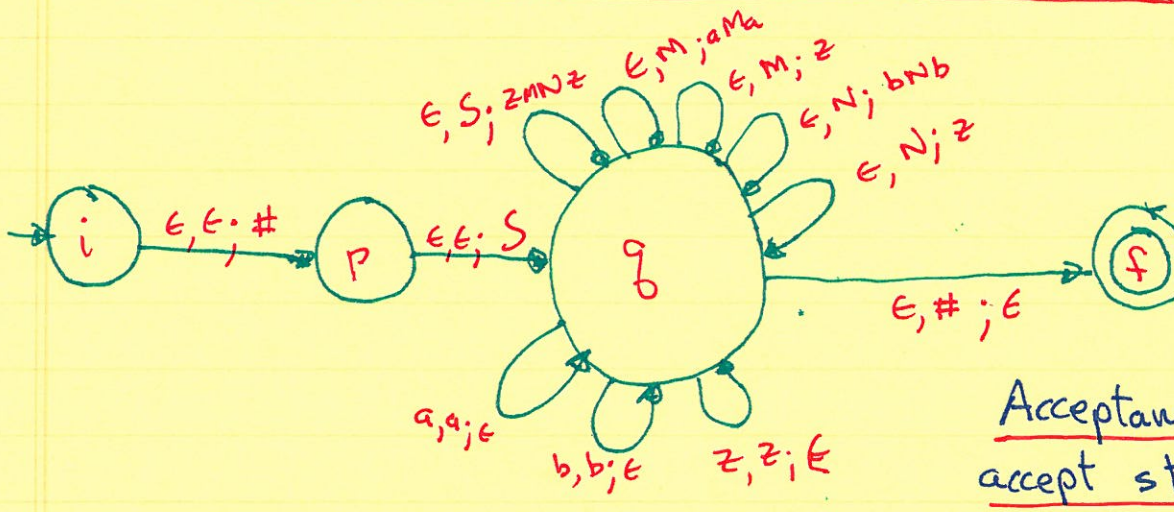
- w = z z z z
- w = z a a z a a z z
- w = z z bb z bb z
- w = z a z a bb z bb z
- w = z a z b z b z
- w = z z b z z

pda & cfg

- L_3 ... one more time

$$G_3: \quad \begin{aligned} S &\to zMNz \\ M &\to aMa \mid z \\ N &\to bNb \mid z \end{aligned}$$

- $L(G_3) = \{ z a^n z a^n b^m z b^m z \mid m, n \geq 0 \}$



Acceptance by accept state
$F = \{f\}$.

- let $w = zaazaabzbz$

$(i, zaazaabzbz, \epsilon)$
$\vdash (p, zaazaabzbz, \#)$
$\vdash (q, zaazaabzbz, S\#)$
$\vdash (q, zaazaabzbz, zMNz\#)$
$\vdash (q, aazaabzbz, MNz\#)$
$\vdash (q, aazaabzbz, aMaNz\#)$
$\vdash (q, azaabzbz, MaNz\#)$
$\vdash (q, azaabzbz, aMaaNz\#)$

$(q, zaabzbz, MaaNz\#) \vdash$
$(q, zaabzbz, zaaNz\#) \vdash$
$(q, aabzbz, aaNz\#) \vdash^*$
$(q, bzbz, Nz\#) \vdash$
$(q, bzbz, bNbz\#) \vdash$
$(q, zbz, Nbz\#) \vdash$
$(q, zbz, zbz\#) \vdash$
$(q, bz, bz\#) \vdash$
$(q, z, z\#) \vdash (q, \epsilon, \#) \vdash$
(f, ϵ, ϵ) **Accept**

XIV

pushdown acceptors con't.

more practice ...

$$L_4 = \{ w \in \{0,1\}^* \mid w = x c x^R \}$$

i.e., $L_4 = \{ c, 0c0, 1c1, 01c10, 10c01, 00c00, \ldots \}$

- G_4: $S \rightarrow$

- Now ... a pda for L_4

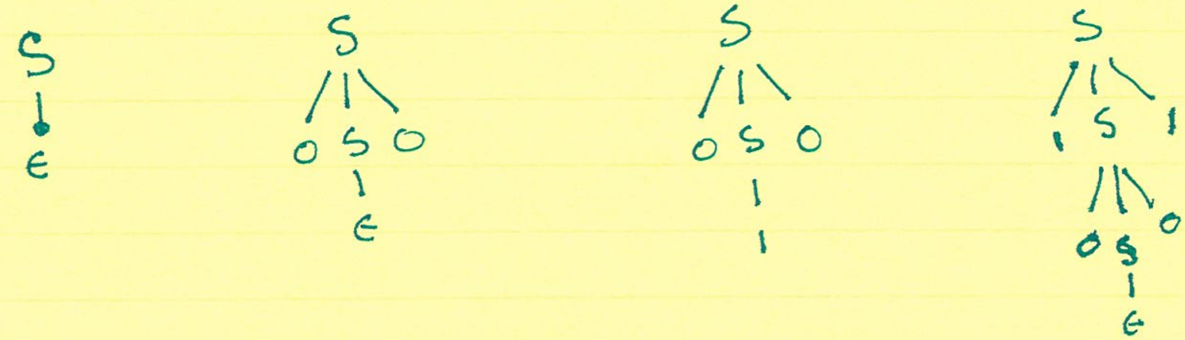
pda's con't.

- $L_5 = \{ w \in \{0,1\}^* \mid w = w^R \}$

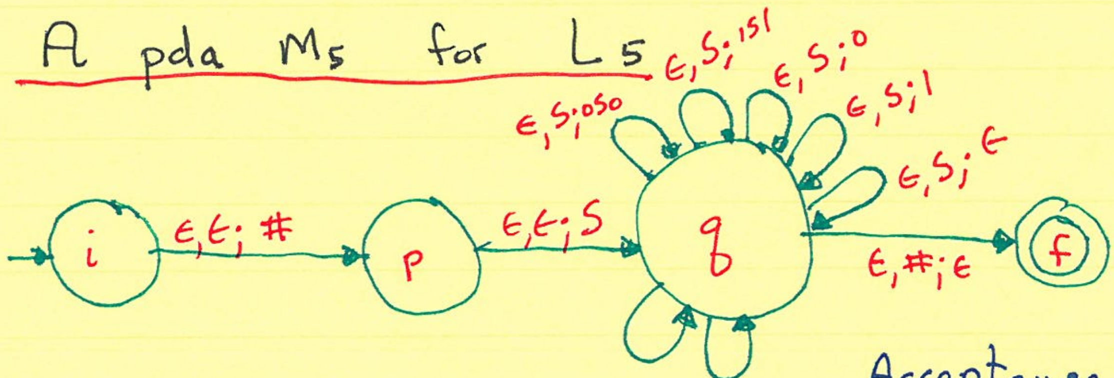
i.e. $L_5 = \{ \epsilon, 0, 1, 00, 11, 010, 101, 0000, 1111, \ldots \}$

G_5: $S \to 0S0 \mid 1S1 \mid 0 \mid 1 \mid \epsilon$

$w = \epsilon$ $w = 00$ $w = 010$ $w = 1001$



A pda M_5 for L_5



Acceptance by accept state

$F = \{f\}$

$W = 01010$

$(i, 01010, \epsilon)$
$\vdash (p, 01010, \#)$
$\vdash (q, 01010, S\#)$
$\vdash (q, 01010, 0S0\#)$
$\vdash (q, 1010, S0\#)$
$\quad (q, 1010, 1S10\#)$
$\vdash (q, 010, S10\#)$
$\vdash (q, 010, 010\#)$
$\vdash (q, 10, 10\#)$
$\vdash (q, 0, 0\#)$
$\quad (q, \epsilon, \#) \vdash$
$\quad (f, \epsilon, \epsilon) \vdash$

in accept state

input entirely read

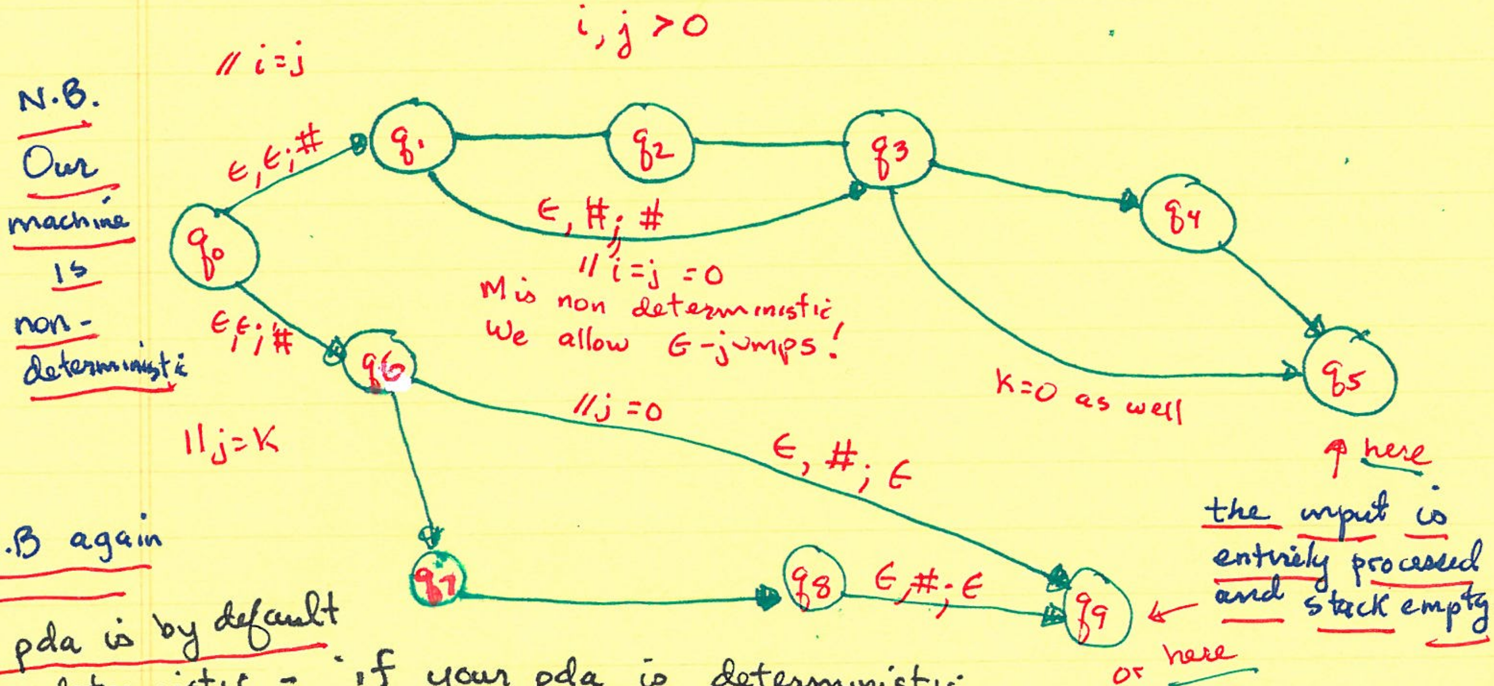
pushdown automata con't.

- $L6 = \{ a^i b^j c^k \mid ((i=j) \text{ or } (j=k)),\ i,j,k \geq 0 \}$

i.e., $L6 = \{ \epsilon, c, a, aabb, aabbc, abc, \dots \}$

- Design a pda for L6 : Complete M_6 below

N.B. Our machine is non-deterministic



N.B again

A pda is by default non-determistic - if your pda is deterministic you must say so !!!

→ $\boxed{\text{pda} \neq \text{dpda}}$!!

- Also produce a grammar G_6 for L6 as well!

- We note that M_6 is nondeterministic hence L6 is non-deterministic

A dcfl : $\{ L = wcw^R \mid w \in \{0,1\}^* \}$

A cfl : $\{ L = ww^R \mid w \in \{0,1\}^* \}$