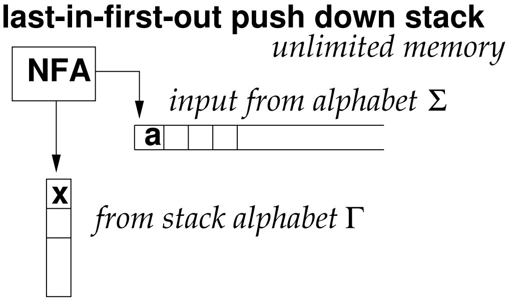
Pushdown Automata

NFA with single, last-in-first-out push down stack

unlimited memory



Stack operations: **Read top**
Remove top *(Pop)*
Write top *(Push)*

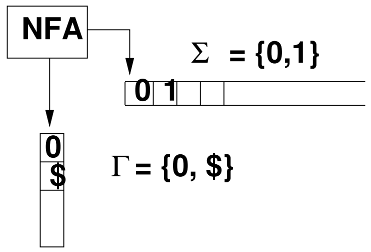
a, x → y *Reading input* **a** *with* **x** *on top, replace* **x** *with* **y**

(a, x = ε)

PDA ⟷ **CFG**

Example PDA

PDA to recognize $\{ 0^n 1^n \mid n \geq 0\}$ *(non-regular)*



Push $ on stack.

Start reading input. As read 0's, push them on stack.
When reach 1, for each 1, pop a 0 off stack. If no 0 to pop, reject.
If reach another 0, after you've read a 1, reject.
Accept if finish input, and stack has $ on top.
Finish by popping $ off stack. **(Deterministic)**

Formal Definition of PDA

A *PDA M* is a 6-tuple (Q, Σ, Γ, δ , q_0, F) where,

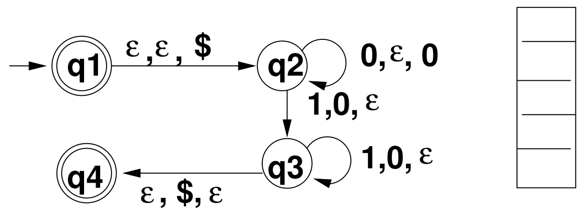
1. Q is a finite set of *states*
2. Σ is the *input alphabet*
3. Γ is the *stack alphabet*
4. $\delta : Q \times \Sigma_\varepsilon \times \Gamma_\varepsilon \rightarrow P(Q \times \Gamma_\varepsilon)$
5. q_0 in Q, the *start state*
6. F $\subseteq$ Q, set of *accept (final) states*

Ex. Q = {q1, q2, q3, q4} Σ = {0,1} Γ = {0, \$}
F = {q1, q4}

input	0			1			ε		
stack	0	\$	ε	0	\$	ε	0	\$	ε
q1									q2, \$
q2			q2, 0	q3, ε					
q3				q3, ε				q4,ε	
q4									

State Diagram of PDA

to recognize { $0^n 1^n$ | n $\geq$ 0} 0011 010



a,b,c: input a, top stack b, replace top stack c

a = ε means don't read input symbol

b = ε means don't read or pop top of stack

c = ε means don't write on top

Formal Definition of PDA Acceptance

A PDA M *accepts* w = w1 w2 .... wm in Σ_ε

if there is a sequence of states r0, r1, ... rm in Q

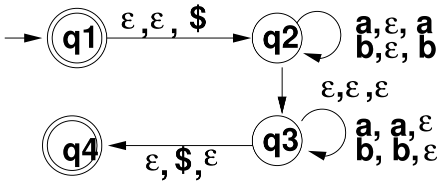
and strings s0 s1 s2 ... sm, si in Γ^* with

1. r0 = q0 , s0 = ε *(starts properly)*
2. (ri+1, b) is an element of δ (ri,wi+1,a) with si = at and si+1 = bt, with a, b in Γ_ε, t in Γ^* *(moves properly)*
3. rm is in F *(ends in final state)*

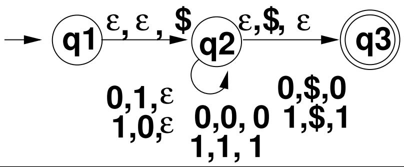
Note: Don't require stack to be empty to accept.

Example PDA

L = { w w^R | w in {a,b } *}



L = { w | w has the same number of 0's and 1's}



Example PDA

L = { $a^i b^j c^k$ | i,j,k $\geq$ 0 and i=j or i = k}

Non-deterministically check that # a's = # b's OR
 # a's = # c's

7

Pumping Lemma for CFL's

If A is a context-free language, then there is a no. p *(pumping length)* **where, if s is any string in A of length at least p, s may be divided into 5 pieces u,v,x,y,z, s = uvxyz, such that all of the following hold:**

1. **for each i $\geq$ 0, uv^ixy^iz is in A**
2. **|vy| > 0**
3. **|vxy| $\leq$ p**

Condition 1 lets us "pump out" elements in A
Note that either v or y can be ε, but both cannot be (by 2)
(Without 2, the lemma is trivially true, with v= y = ε)
Condition 3 assures us we can make vxy small, if needed.

8

Pumping Lemma Example

$L = \{ a^n b^n c^n \mid n \geq 0\}$ is not context-free.

Suppose L were CF. Then let p be the pumping length given by the pumping lemma.
Let $s = a^p b^p c^p$ in L.

Note that |s| > p, so s = uvxyz as in the lemma.

Case 1: **v and y contain only 1 type of symbol.**
(E.g., v contains only a's, and y only b's)

Since there are 3 symbols in the alphabet of L, there is a third type of symbol not in v or y.
Consider uvvxyyz. It contains more of the first two symbols than the third; contradiction.

Pumping Lemma Example, Continued

Case 2: **Either v or y contains at least 2 symbols.**
(E.g., v contains a's and b's)

Consider uvvxyyz. It contains symbols out of order in the part that repeats and contains at least 2 symbols. Contradiction.

These are the only cases. End of proof.

Ex. v = ab. Then u ababxyyz has substring abab, which is not allowed.

Pumping Lemma Example

L = { ww | w in {0,1}* } is not context-free.

Suppose L were CF. Then let p be the pumping length given by the pumping lemma.
Let s = $0^p1^p0^p1^p$ in L.

Note that |s| > p, so s = uvxyz as in the lemma.

Case 1: vxy is in the first half of the string
Then uvvxyyz will have the second half of the string start with a 1; but the first half starts with 0. So it cannot be of form ww. Contradiction.
Case 2: vxy is in the second half.
Follows similarly to case 1.

Pumping Lemma Example, continued

Case 3: vxy is in the middle of the string.

Then uxz is of the form $0^p1^i\,0^j1^p$, where

i and j are not both p. This string is not

of form ww. Contradiction.

Qu: What about uvvxyyz?

Could we have used 0^p10^p1 for s in the proof?