	read	pop	push
1	a	S	S1
2	b	1	ε
3	ε	S	ε
4	ε	S	11

$\delta(a, S) = \{S1\}$
$\delta(a, 1) = \emptyset$
$\delta(b, S) = \emptyset$
$\delta(b, 1) = \{\varepsilon\}$
$\delta(\varepsilon, S) = \{\varepsilon, 11\}$
$\delta(\varepsilon, 1) = \emptyset$

What does $\mapsto$ mean?

$$(aba, S11) \mapsto_1 (ba, S111)$$

when is it legal to write

$$(\sim, \sim) \mapsto (\sim, \sim)$$

---

when have a rule like

read	pop	push
a	B	y

$\rightarrow \; y \in \delta(a, B)$

then we can write:

$$(ax, Bz) \mapsto (x, yz)$$

ID's

We also write

$$(x, y) \mapsto^* (z, w)$$

when (x,y) can become (z,w) after several steps.

Then the language of the stack machine is:

$$L(M) = \{ x \in \Sigma^* \mid (x, S) \mapsto^* (\varepsilon, \varepsilon) \}$$

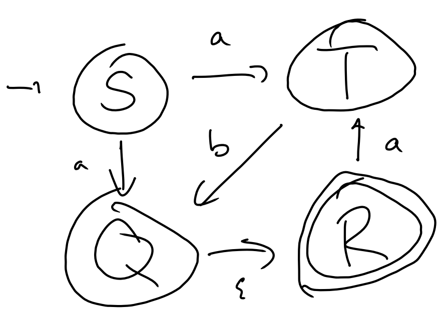
What kinds of ~~of~~ languages can we accept with stack machines?

Regular? Equivalently, can we convert NFA → Stack

CFL? CFG → Stack

We <u>can</u> convert any NFA to stack machine.

(so any reg. lang. can be done on a stack machine)



We can build a stack machine which remembers the state

read	pop	push
a	S	T
a	S	Q
b	T	Q
a	R	T
ε	Q	R
ε	R	ε

any arrow

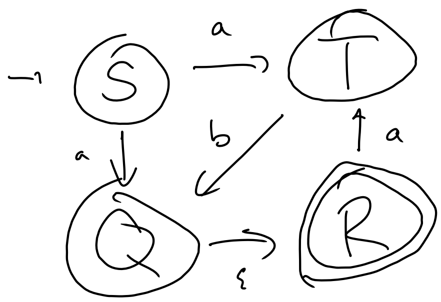
$X \xrightarrow{a} Y$

becomes a rule

<u>read</u>	<u>pop</u>	<u>push</u>
a	X	Y

any accepting state Z:

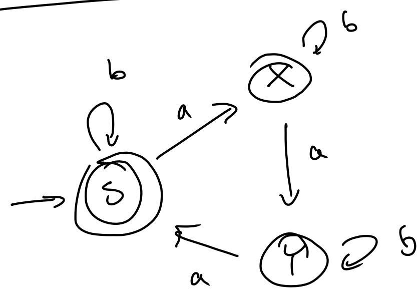
ε	Z	ε



Show a derivation for ab

$(ab, S) \mapsto (b, T) \mapsto (\varepsilon, Q)$

$\mapsto (\varepsilon, R) \mapsto (\varepsilon, \varepsilon)$.



convert to stack machine, and show a derivation for abaab.

read	pop	push
a	S	X
b	S	S
a	X	Y
b	X	X
a	Y	S
b	Y	Y
ε	S	ε

$(abaab, S) \mapsto (baab, X)$

$\mapsto (aab, X) \mapsto (ab, Y)$

$\mapsto (b, S) \mapsto (\varepsilon, S)$

$\mapsto (\varepsilon, \varepsilon)$

<u>Thm</u> If L is a regular languages, then there is a stack machine accepting L.

We can also convert CFG $\to$ Stack!

$S \to abSa \mid T$

$T \to bT \mid \varepsilon$

lang. is $\{ (ab)^n b^m a^n \}$

strategy for derivation: ~~abab~~bbbaa

$(ababbbbaa, S) \mapsto (ababbbbaa, abSa)$

$\mapsto (babbbbaa, bSa) \mapsto (abbbbaa, Sa)$

$\mapsto (abbbbaa, abSaa) \mapsto (bbbbaa, bSaa)$

$\mapsto (bbbaa, Saa) \mapsto (bbbaa, Taa) \mapsto (bbbaa, bTaa)$

$\mapsto (bbaa, Taa) \mapsto (bbaa, bTaa) \mapsto (baa, Taa)$

$\mapsto (baa, bTaa) \mapsto (aa, Taa) \mapsto (aa, aa)$

$\mapsto (a, a) \mapsto (\varepsilon, \varepsilon)$.

Any grammar rule like

$X \to Z$

makes:

read	pop	push
ε	X	Z

plus rules like

a	a	ε
b	b	ε
$\vdots$		

Only works for context-free:

X must be a single letter, since we did "pop X".

$a^n b^n$

CFG: $S \to aSb \mid \varepsilon$

stack:

read	pop	push		
ε	S	aSb	$\leftarrow$	$S \to aSb$
ε	S	ε	$\leftarrow$	$S \to \varepsilon$
a	a	ε		
b	b	ε		

derive $aabb$

CFG: $S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow aabb$

Stack: $(aabb, S) \mapsto (aabb, aSb) \mapsto (abb, Sb)$

$\mapsto (abb, aSbb) \mapsto (bb, Sbb)$

$\mapsto (bb, bb) \mapsto (b, b) \mapsto (\varepsilon, \varepsilon)$.