$$L = \{ x \in \{a,b\}^* \mid \#\text{ of } a\text{'s} = \#\text{ of } b\text{'s} \}$$

Store on stack the # of as as we read them,
cancel on the stack when we read b's.

Need to store X for extra as
Y for extra b's

it'll work like:

$(abbbabaa, S) \mapsto (bbbabaa, XS) \mapsto (bbabaa, S)$

$\mapsto (babaa, YS) \mapsto (abaa, YYS) \mapsto (baa, YS)$

$\mapsto (aa, YYS) \mapsto (a, YS) \mapsto (\varepsilon, S) \mapsto (\varepsilon, \varepsilon)$

	read	pop	push
1	a	S	XS
2	b	S	YS
3	a	Y	ε
4	b	X	ε
5	a	X	XX
6	b	Y	YY
7	ε	S	ε

Give a derivation showing abbaba is accepted.

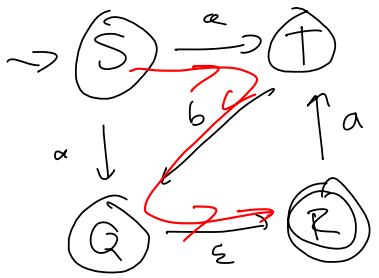
$(abbaba, S) \mapsto (bbaba, XS) \mapsto (baba, S)$

$\mapsto (aba, YS) \mapsto (ba, S) \mapsto (a, YS) \mapsto (\varepsilon, S)$

$\mapsto (\varepsilon, \varepsilon)$

Converting NFA / Grammars to Stack & vice-versa.

Converting NFA → Stack



Use the states as stack symbols one at a time,

When we follow an arrow, change the stack.

read	pop	push	
a	S	T	arrows
a	S	Q	
b	T	Q	
ε	Q	R	
a	R	T	
ε	R	ε	← R is accepting

derive ab

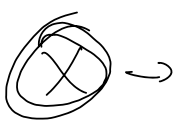
$(ab, S) \mapsto (b, T) \mapsto (\varepsilon, Q)$

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

Generally:

becomes

read	pop	push
a	X	Y



ε	X	ε

So any NFA can be converted to stack mach.

So any regular lang. can be computed by stack machine.

Not every stack machine can be converted to NFA like for $a^n b^n$

Grammars? Can we do

stack → grammar?

grammar → stack?

We can convert any CFG to a stack machine

$S \to abSa \mid T$ $\quad (ab)^n \; b^m \; a^n$

$T \to bT \mid \varepsilon$

derive abababbbaa:

$S \Rightarrow abSa \Rightarrow ababSaa \Rightarrow ababTaa \Rightarrow ababbTaa$

$\Rightarrow ababbbTaa \Rightarrow ababbbbaa \Rightarrow abababbbaa$

The stack alphabet will be $\{a, b, S, T\}$

each grammar production becomes an ε-read

read pop push

$S \to abSa$ becomes $\varepsilon \quad S \quad abSa$

we also use

a	a	ε
b	b	ε

$S \to abSa \mid T$

$T \to bT \mid \varepsilon$

becomes:

	read	pop	push	
1.	ε	S	abSa	productions
2.	ε	S	T	
3.	ε	T	bT	
4.	ε	T	ε	
5.	a	a	ε	read-pop letters
6.	b	b	ε	

derive abababbbaa: follow the same grammar steps, in the same order, pop letters when possible.

$(abababbbaa, S) \vdash_1 (ababbbbaa, abSa)$

$\vdash_5 (babbbbaa, bSa) \vdash_6 (abbbbaa, Sa)$

$\vdash_1 (abbbbaa, abSaa) \vdash_5 (bbbbaa, bSaa)$

$\vdash_6 (bbbaa, Saa) \vdash_2 (bbbaa, Taa)$

$\vdash_3 (bbbaa, bTaa) \vdash_6 (bbaa, Taa)$

$\vdash_3 (bbaa, bTaa) \vdash_6 (baa, Taa)$

$\vdash_3 (baa, bTaa) \vdash_6 (aa, Taa) \vdash_4 (aa, aa)$

$\vdash_5 (a, a) \vdash_5 (\varepsilon, \varepsilon)$