

Stacks vs Grammars.

CFG $\rightarrow$ Stack machine

$$S \rightarrow aSb \mid a$$

	read	pop	push
1	ϵ	S	aSb
2	ϵ	S	a
3	a	a	ϵ
4	b	b	ϵ

derive aaabbb

$$S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow aaabbb$$

Stack:

$$(aaabbb, S) \mapsto_1 (aaabbb, aSb)$$

$$\mapsto_2 (aaabbb, Sb) \mapsto_3 (aaabbb, aSbb)$$

$$\mapsto_4 (abb, Sbb) \mapsto_5 (abb, abb)$$

$$\mapsto_6 (bb, bb) \mapsto_7 (b, b) \mapsto_8 (\epsilon, \epsilon)$$

$$S \rightarrow aSc \mid aS \mid bS \mid \epsilon$$

convert to stack mach,

show a stack derivation for abaacc

$$S \Rightarrow aSc \Rightarrow abSc \Rightarrow abasc \Rightarrow abaaSc \Rightarrow abaaacc$$

read	pop	push
ϵ	S	aSc
ϵ	S	aS
ϵ	S	bS
ϵ	S	ϵ
a	a	ϵ
b	b	ϵ
c	c	ϵ

$$(abaaacc, S) \mapsto (abaaacc, aSc)$$

$$\mapsto (baacc, Sc) \mapsto (baacc, bSc)$$

$$\mapsto (aacc, Sc) \mapsto (aacc, aSc)$$

$$\mapsto (acc, Sc) \mapsto (acc, aSc)$$

$$\mapsto (cc, Sc) \mapsto (cc, cc) \mapsto (c, c) \mapsto (\epsilon, \epsilon)$$

This only works for CFG, since
the pop column is only allowed to
have single letters in it.

Stack machine $\rightarrow$ Grammar

$\{x \mid \# \text{ of } a\text{'s} = \# \text{ of } b\text{'s}\}$ $\left| \begin{array}{c|c|c} \text{read} & \text{pop} & \text{push} \\ \hline x & y & z \end{array} \right. \text{ becomes } y \rightarrow xz$

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



$S \rightarrow aXS$

$X \rightarrow aXX$

$Y \rightarrow a$

$S \rightarrow bYS$

$X \rightarrow b$

$Y \rightarrow bYY$

$S \rightarrow \epsilon$

$S \rightarrow aXS \mid bYS \mid \epsilon$

$\Rightarrow X \rightarrow aXX \mid b$

$Y \rightarrow bYY \mid a$

derive abba

stack: $(abba, S) \mapsto (bba, XS) \mapsto (ba, S) \mapsto (a, YS) \mapsto (\epsilon, S) \mapsto (\epsilon, \epsilon)$

grammar: $S \Rightarrow aXS \Rightarrow abS \Rightarrow abbYS \Rightarrow abbaS \Rightarrow abba$

each grammar step looks like

(string read so far) (stack)

A stack rule like

read	pop	push
x	y	z

This applies when the starts with y ,
 in which case it looks like
 $\rightarrow$ (string read so far) y (rest of the stack)
 $\Downarrow$
 $(\quad) x z (\quad)$

This is why $y \rightarrow xz$ works!