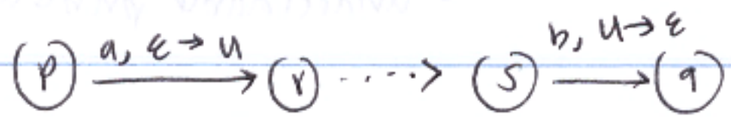
10-05-17

LAST CLASS **Lemma 2.27** if a PDA recognizes a language L, then L has a CFG

CFG rules: ① For each $p \in Q$, add rule $A_{pp} \to \varepsilon$ to R.

② For each $p, q, r \in Q$, add rule $A_{pq} \to A_{pr}A_{rq}$ to R.

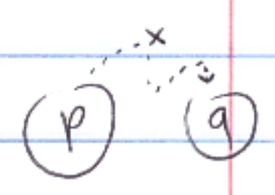
③ For each path in P:



add rule $A_{pq} \to a A_{rs} b$ to R

Claim 2.31 if x can bring PDA P from state p with empty stack (WES) to q WES, then A_{pq} generates x, i.e. $A_{pq} \Rightarrow^* x$.

Pf/ induction on # of steps in P from p to q. WES on input x.



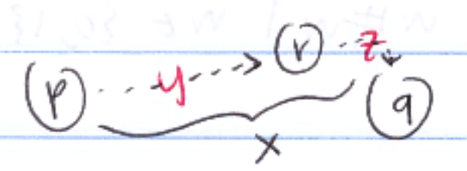
Base case: path takes 0 transitions

then $p = q \Rightarrow x = \varepsilon$ But $A_{pp} \to \varepsilon$ generates x

inductive hyp: (IH) Assume true for paths of length at most k.

inductive step: prove for path of length k+1.

case 1: stack becomes empty at some state $r \notin \{p, q\}$ during computation.



paths $p \to r$ & $r \to q$ have $\leq k$ steps.

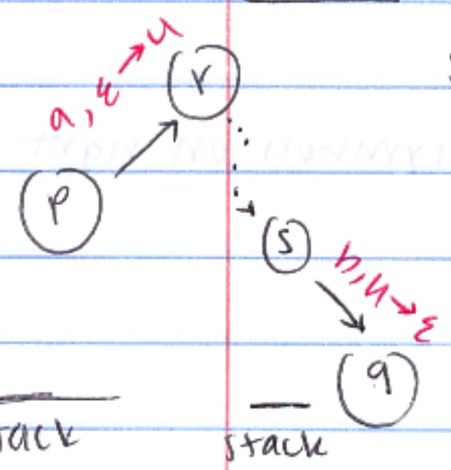
∴ IH implies $A_{pr} \Rightarrow^* y$

$A_{rq} \Rightarrow^* z$

$A_{pq} \to A_{pr}A_{rq} \Rightarrow yz = x$

this rule is in CFG. ✓

case 2: stack empty only at p & q



Since we push or pop in each step:

① First step has form:

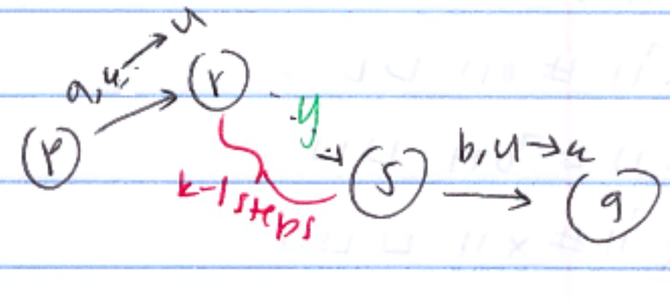
$\delta(p, a, \varepsilon) \to (r, u)$

state p; read a; pop nothing; state r; push u

② Last step has form:

$\delta(s, b, u) \to (q, \varepsilon)$

read b; pop u; push nothing



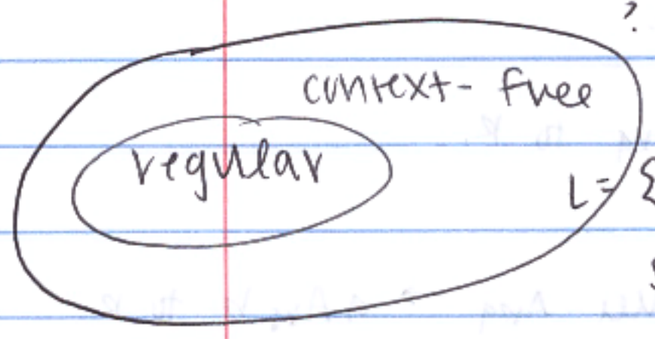
But we added rule $A_{pq} \to a A_{rs} b$ to R

Suppose $x = a y b$. since P goes from p to q WES, we know y takes P from r to s WES in k-1 steps.

∴ Apply IH to get $A_{rs} \Rightarrow^* y$

$\therefore A_{pq} \to a A_{rs} b \to^* a y b = x$

3. Turing Machines



$L = \{0^n 1^n \mid n \geq 0\}$

$S \to 0S1 \mid \varepsilon$

- invented in 1936 by Turing
- "can do everything" your cpu can do at home.
- do not exist
- "universal computing"
- undecidable problems!

Designing a turing machine

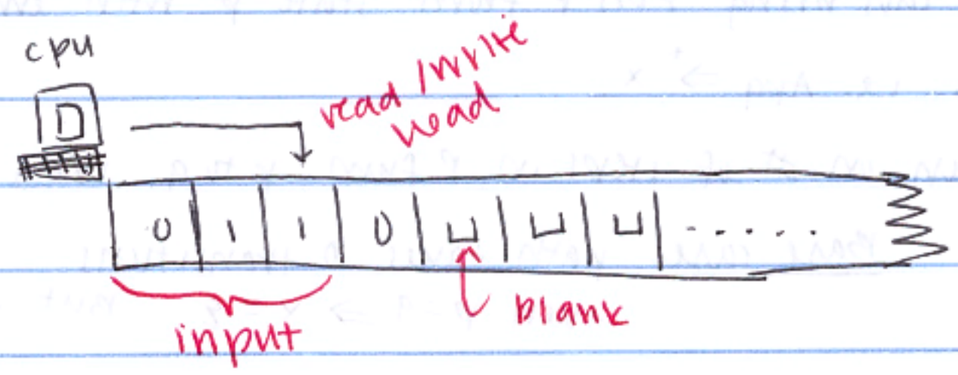
DESIGN

- memory
- read/write memory
- cpu/ brain

(1) Head moves left or right

(2) Head can read from & write to tape

(3) Tape is infinite.



e.g. TM to decide membership in language $B = \{w\#w \mid w \in \{0,1\}^*\}$

Description of TM:

on input string $x \in \{0,1\}^*$:

0. if x doesn't contain a #, reject
1. Zig zag across tape to corresponding positions on either side of #. If these contain different symbols, reject

"cross off" symbol as they're checked ⇐ so you don't check the same symbol again.

2. when all symbols to left of # are crossed off, if non-blank symbols on right of #, reject
3. Accept.

eg. $\overset{\neg}{0}11\ \#\ 011\ \sqcup\sqcup\ldots$

$x\overset{\neg}{1}1\ \#\ 011\ \sqcup\sqcup\ldots$

$x11\ \#\ \overset{\neg}{0}11\ \sqcup\sqcup\ldots$

$x\overset{\neg}{1}1\ \#\ x11\ \sqcup\sqcup\ldots$