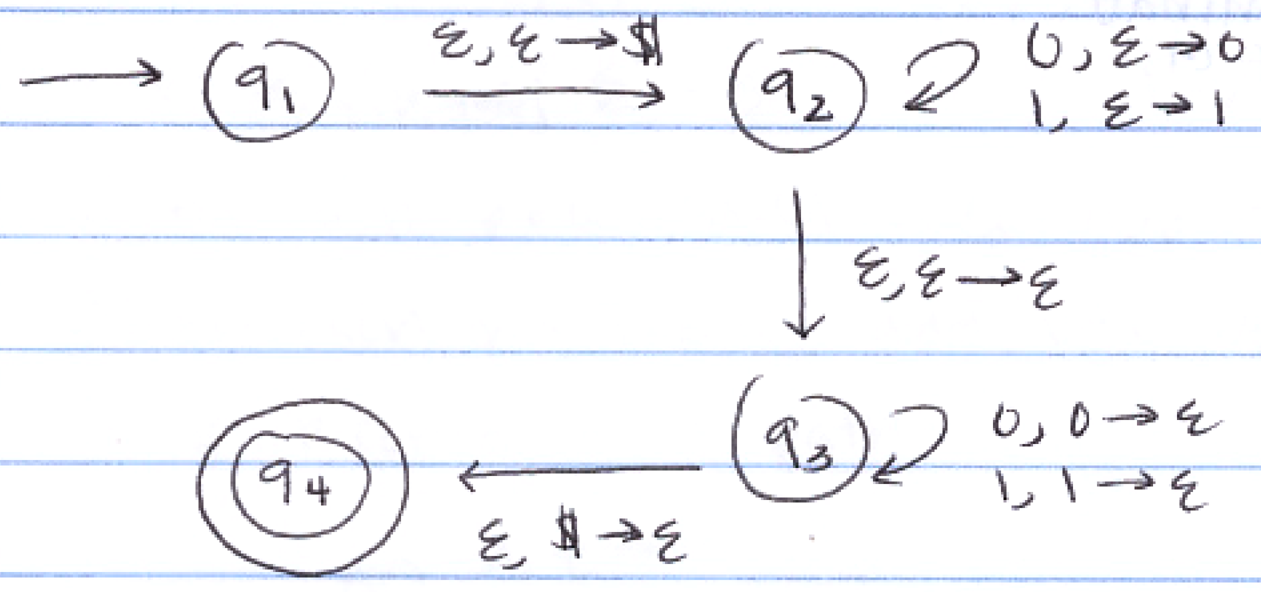
OCTOBER 3, 2017
Class Notes.

LAST CLASS PDAs.

Ex. 2.18 $L = \{ww^R \mid w \in \{0,1\}^*\}$

Thm 2.20 → a language is context-free iff a PDA recognizes it

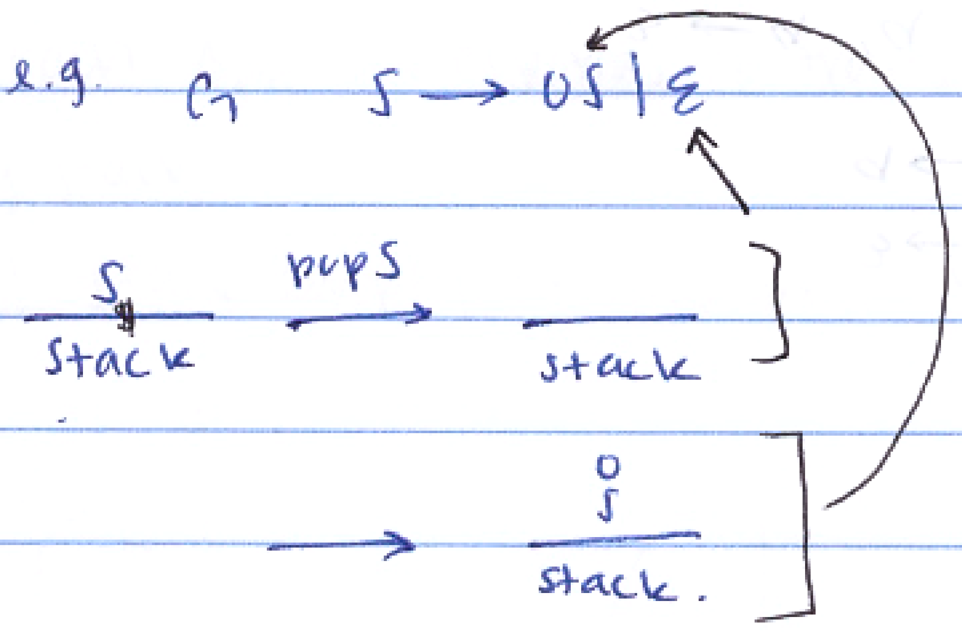


⇒ Lemma 2.21 - if a language is context-free, then some PDA recognizes it

Pf/ Let L be a CFL with CFG, $G = (V, \Sigma, R, S)$

We construct a PDA $M = (Q, \Sigma, \Gamma, \delta, q_0, F)$ to recognize L.

Idea: e.g. G $S \to 0S \mid \varepsilon$

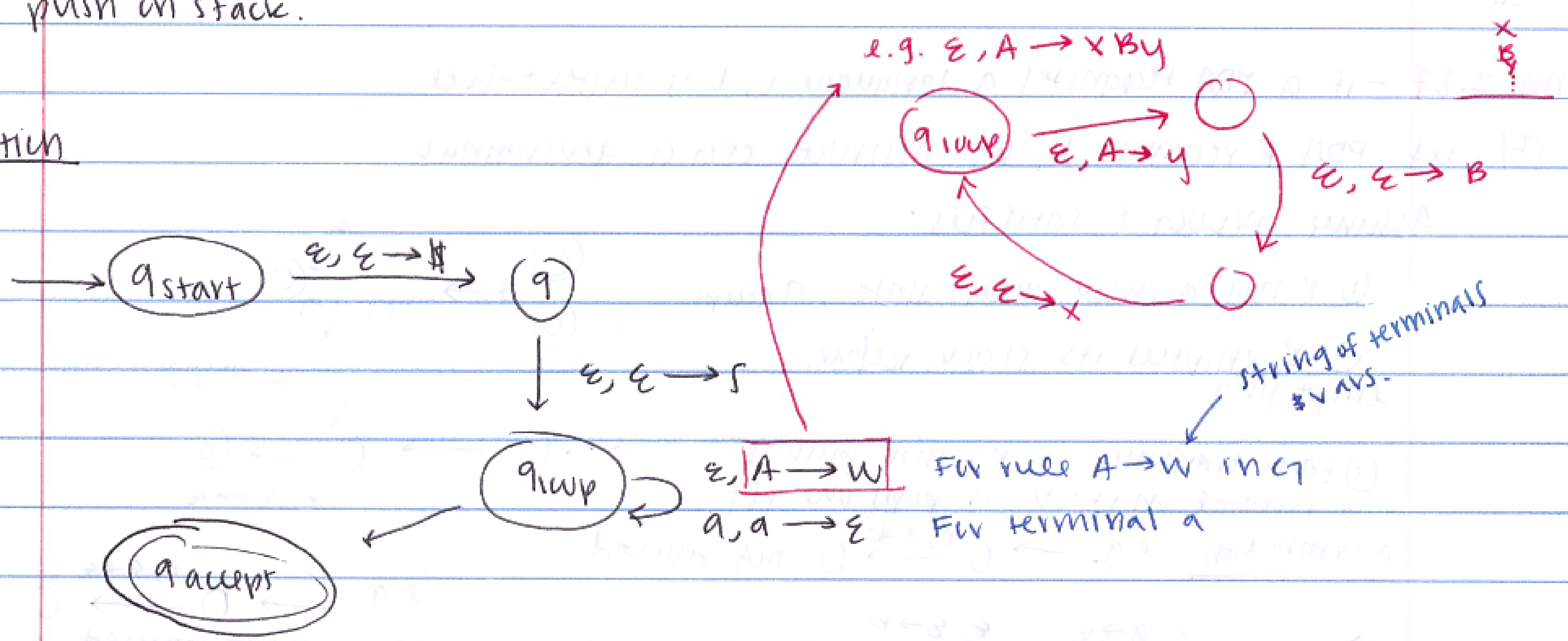


In words:

① Design M to non-deterministically pick derivation in G to "follow" to see if its input x can be derived.

② Use stack to store variables & terminals during derivation.
When a variable is popped, non-deterministically choose which substitution rule to push on stack.

Construction



$Q = \{q_{start}, q_{loop}, q_{accept}, q\} \cup E$ ← set of states to implement $\varepsilon, A \to w$ transitions.

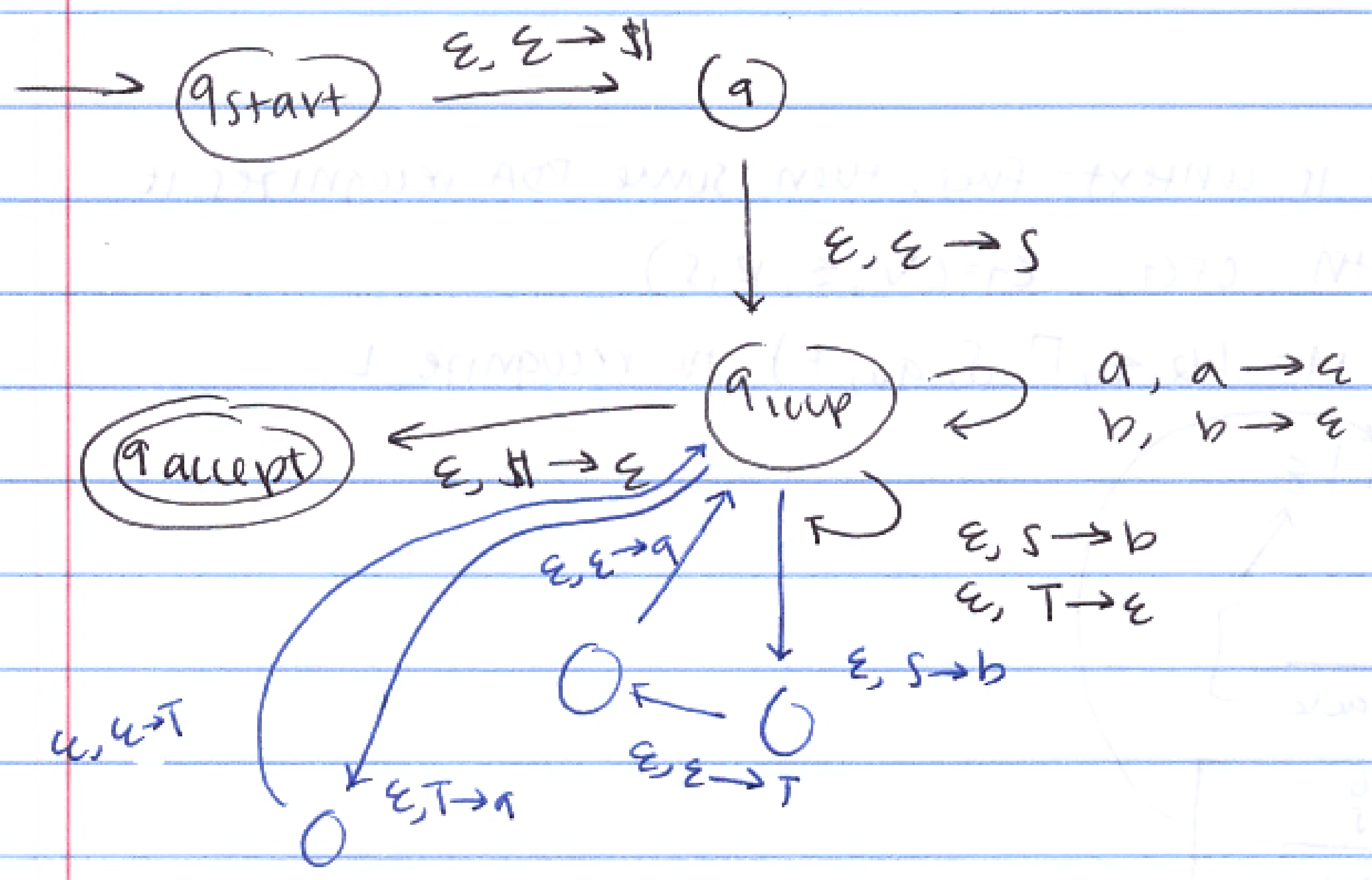
$q_0 = q_{start}$ $\quad \Sigma' = \Sigma$

$F = \{q_{accept}\}$ $\quad \Gamma = \Sigma \cup V \cup \{\$\}$ $\quad \delta$ defined as in figure.

(Σ: terminals of G. V: vars of G)

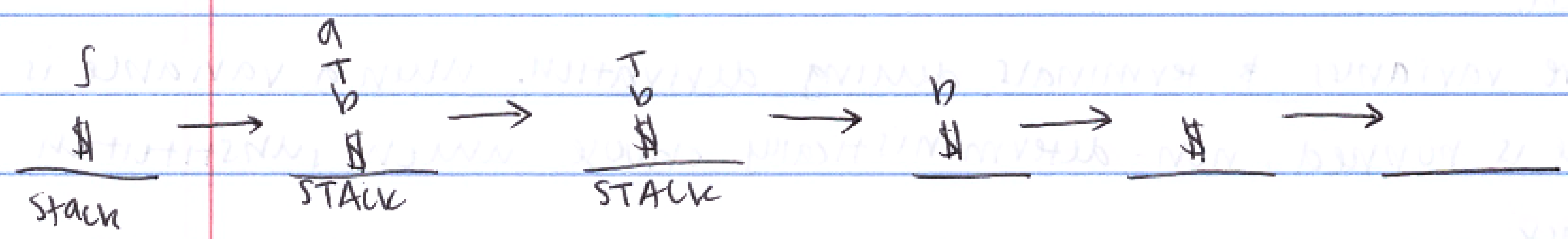
<u>Example 2.25</u>

G: $S \to aTb \mid b$
$\quad T \to Ta \mid \varepsilon$



e.g. $S \to aTb \to ab$

let's feed into our PDA

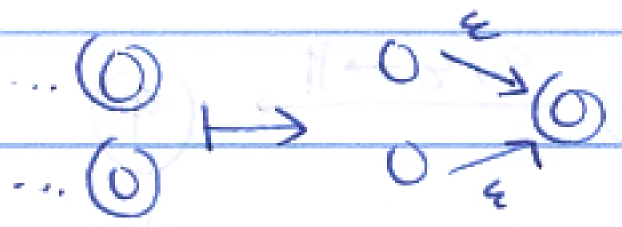


Lemma 2.27 – if a PDA recognizes a language L, L is context-free.

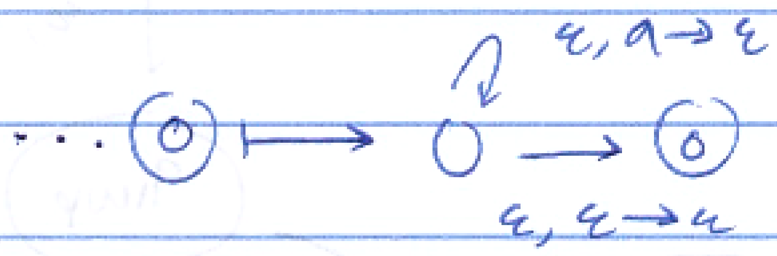
Pf| let PDA P recognize L. we construct CFG G generating L

Assume WLOG P satisfies:

① P has a single accept state, q_{accept}



② P empties its stack before emptying



③ Each transition of P either pushes a symbol on stack or pops one off <u>but not both</u> e.g. $\to \bigcirc \xrightarrow{c, a \to b} \bigcirc$ not allowed.

$\Rightarrow \bigcirc \xrightarrow{c, a \to \varepsilon} \bigcirc \xrightarrow{\varepsilon, \varepsilon \to b} \bigcirc$

e.g. $\to \bigcirc \xrightarrow{c, \varepsilon \to \varepsilon} \bigcirc$ NOT allowed.

$c, \varepsilon \to \$$ $\quad \varepsilon, \$ \to \varepsilon$

Idea: $\rightarrow (p) \xrightarrow{abc} (q)$ in CFG "add rule" $R_p \rightarrow abc\, R_q$

Let PDA $P = (Q, \Sigma, \Gamma, \delta, q_0, \{q_{accept}\})$.

We construct CFG $G = (V, \Sigma, R, S)$ as follows:

$V = \{A_{p,q} \mid p, q \in Q\}$ $\Sigma' = \Sigma$ $S = A_{q_0, q_{accept}}$, R as follows:

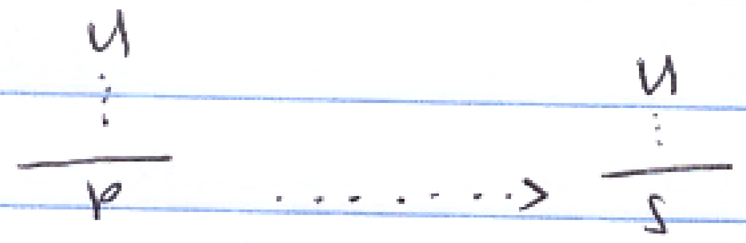
(1) For each $p \in Q$, add rule $A_{pp} \rightarrow \varepsilon$ to R.

(2) For each $p, q, r \in Q$, add rule $A_{pq} \rightarrow A_{pr} A_{rq}$

(3) If P contains a path:

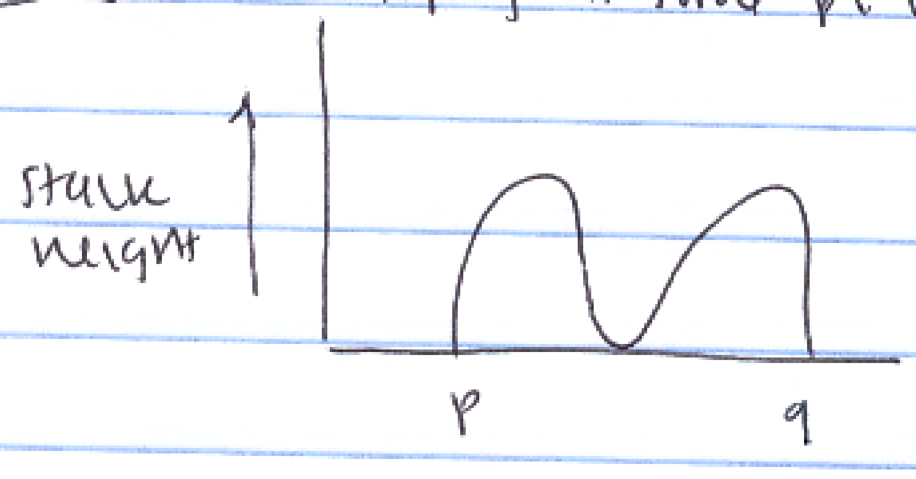
$(p) \xrightarrow{a, \varepsilon \rightarrow u} (r) \longrightarrow (s) \xrightarrow{b, u \rightarrow \varepsilon} (q)$

then add $A_{pq} \rightarrow a A_{rs} b$ to R.



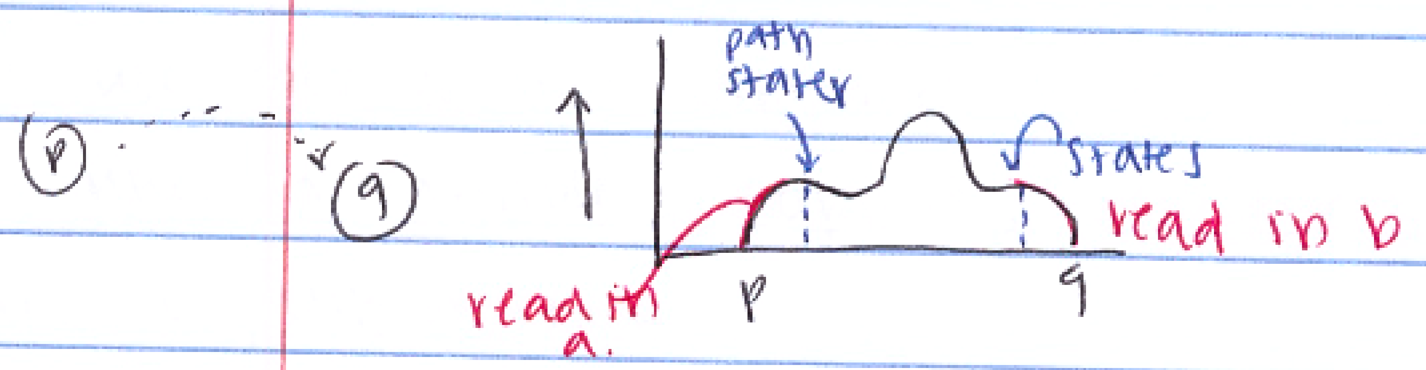
Idea 2: Each A_{pq} corresponds to transitions in P from p to q which start & end with empty stack.

Case 1: stack empty at some pt r b/t p & q



rule in G: $A_{pq} \rightarrow A_{pr} A_{rq}$

Case 2: stack never empty b/t p & q



Rule in G: $A_{pq} \rightarrow a R_s b$