CONTEXT-FREE
=
PDA-RECOGNIZABLE
PART II

Recall: Given a CFG G where every production rule is of the form $A \to \underbrace{a}_{T\cup\{\varepsilon\}}\overbrace{B_1 \cdots B_k}^{NT}$,

we construct a PDA M which accepts $\mathcal{L}(G)$ by empty stack.

$(q_0, \omega, \perp) \xrightarrow[M]{*} (q, \varepsilon, \perp)$ for some $q \in Q$.

$M = (Q, \Sigma, \Gamma, \Delta, q_0, \emptyset)$, where

$Q = \{q_0, q_1\}$, $\Sigma = T$, $\Gamma = NT \cup \{\perp\}$ and

$$\Delta = \{((q_0, \varepsilon, \varepsilon), (q_1, S)),$$

$$\{((q_1, a, A), (q_1, B_1 \cdots B_k)) \mid A \to aB_1 \cdots B_k \in R\}$$

Thm: If a terminal string x is obtained by applying rules in R to the leftmost nonterminal symbol, then M accepts x.

Thm: For any $x, y \in \Sigma^*$, $\gamma \in NT^*$, and $C \in NT$, one can generate $x\gamma$ by n applications of rules in R to C in leftmost-first order iff $(q_1, xy, C\bot) \xrightarrow[M]{n} (q_1, y, \gamma\bot)$.

Proof: By induction on n.

$n=0$: $C \longrightarrow x\gamma$ in 0 applications of any rule in R
iff $C = x\gamma$ iff $x = \varepsilon$ and $\gamma = C$
iff $(q_1, xy, C\bot) = (q_1, \varepsilon y, \gamma\bot) = (q_1, y, \gamma\bot) \xrightarrow[M]{0} (q_1, y, \gamma\bot)$.

$n = m+1$: Suppose $C \longrightarrow x\gamma$ in $m+1$ applications of rules from R.
Consider the $m+1^{\text{th}}$ rule applied. It must be of the form $D \longrightarrow c\beta$, where $c \in T \cup \{\varepsilon\}$, $D \in NT$, and $\beta \in NT^*$.

Then, there is some $z \in T^*$ and $\alpha \in NT^*$ s.t. $C \xrightarrow{m} zD\alpha$.

Then, $zD\alpha \xrightarrow{1} zc\beta\alpha = x\gamma$. So, $x = zc$, and $\gamma = \beta\alpha$.

By IH, $(q_1, zcy, C\bot) \xrightarrow[M]{m} (q_1, cy, D\alpha\bot)$ — (a)

We map each rule in R to a transition in Δ, so

$((q_1, c, D), (q_1, \beta)) \in \Delta$, since the $m+1^{th}$ rule was $D \to c\beta$

So, $(q_1, cy, D\alpha\bot) \xrightarrow[M]{1} (q_1, y, \underbrace{\beta\alpha}_{\gamma}\bot)$ — (b)

Combining (a) and (b),

$$(q_1, zcy, C\bot) \xrightarrow[M]{m+1} (q_1, y, \gamma\bot)$$

Now, suppose $(q_1, xy, C\bot) \xrightarrow[M]{m+1} (q_1, y, \gamma\bot)$.

Let $((q_1, c, D), (q_1, \beta)) \in \Delta$ be the final transition taken.

Then, $x = zc$ for some $z \in \Sigma^*$, $\gamma = \beta\alpha$ for some $\alpha \in \Gamma^*$, and

$$(q_1, zcy, C\bot) \xrightarrow[M]{m} (q_1, cy, D\alpha\bot) \xrightarrow[M]{1} (q_1, y, \beta\alpha\bot).$$

By IH, one can generate $zD\alpha$ by m applications of rules to C.

By the definition of Δ, $D \rightarrow c\beta \in R$.

So, $C \xrightarrow{m} zD\alpha \xrightarrow{1} zc\beta\alpha = x\gamma$.

So $x\gamma$ can be generated from C by $m+1$ applications of rules.

Thm: $\mathcal{L}(M) = \mathcal{L}(G)$.

Proof: $x \in \mathcal{L}(G)$

iff $S \longrightarrow x$ via some sequence of "leftmost" applications of rules in R

iff $(q_0, x, \perp) \xrightarrow[M]{*} (q, \varepsilon, \perp)$ for some $q \in Q$

iff $(q_0, x, \perp) \xrightarrow[M]{1} \underbrace{(q_1, x, S\perp) \xrightarrow[M]{*} (q, \varepsilon, \perp)}_{\text{above}}$ for some $q \in Q$

iff $x \in \mathcal{L}(M)$.

We now look at the other direction.

2 Given a PDA recognizing $\mathcal{L}$, construct a CFG which generates $\mathcal{L}$.

Suppose we are given $M = (Q, \Sigma, \Gamma, \Delta, q_0, \emptyset)$ which recognizes L by empty stack.

Recall that each transition in Δ is of the form $((q, a, \gamma), (q', \gamma'))$

We modify transitions so that each transition either only pushes a symbol onto the stack, or pops one off the stack.

Δ'

- Push+pop: split into multiple transitions
- Neither push nor pop: push arbitrary symbol, pop it off. F'

We also add a transition which takes M to state f if the input word is read and the stack is empty.

$M' = (Q, \Sigma, \Gamma, \Delta', q_0, \{f\})$ Prove that $L(M') = L(M)$.

We now employ a strategy similar to how we constructed an equivalent regular expression given a DFA.

For each pair of states $p, q \in Q$, define $A_{pq} \in NT$.

A_{pq} should generate all strings which take M from state p with empty stack to state q with empty stack.

If M goes from p with empty stack to q with empty stack on some string x, what is the first move of M?

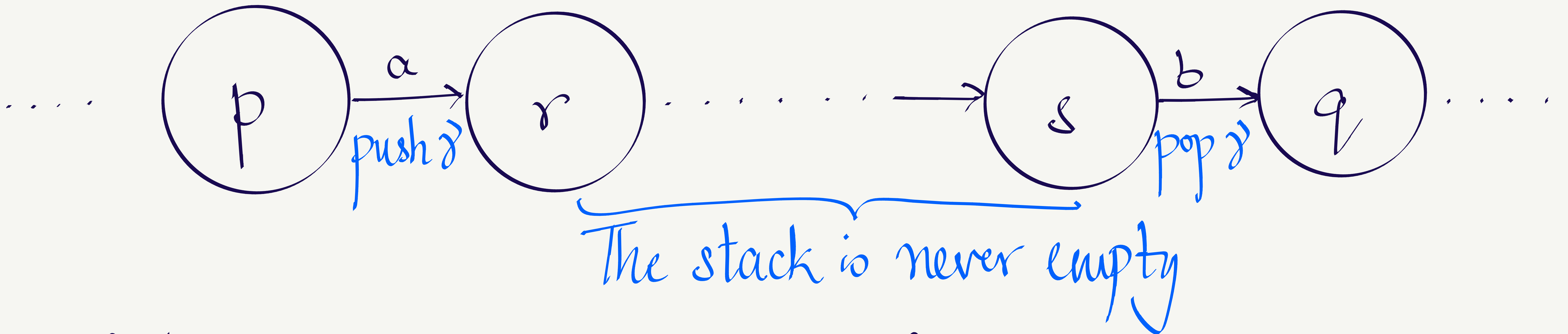
Has to be a push, since we cannot pop from an empty stack!

Similarly, what is the last move? A pop.

We will define the production rules for A_{pq} inductively.

Suppose $x = a\omega b$ for some ω. Two possibilities arise

(a) The symbol that is initially pushed onto the stack, say γ, is only popped off the stack at the end of ω. Then, the operation of M looks as follows:



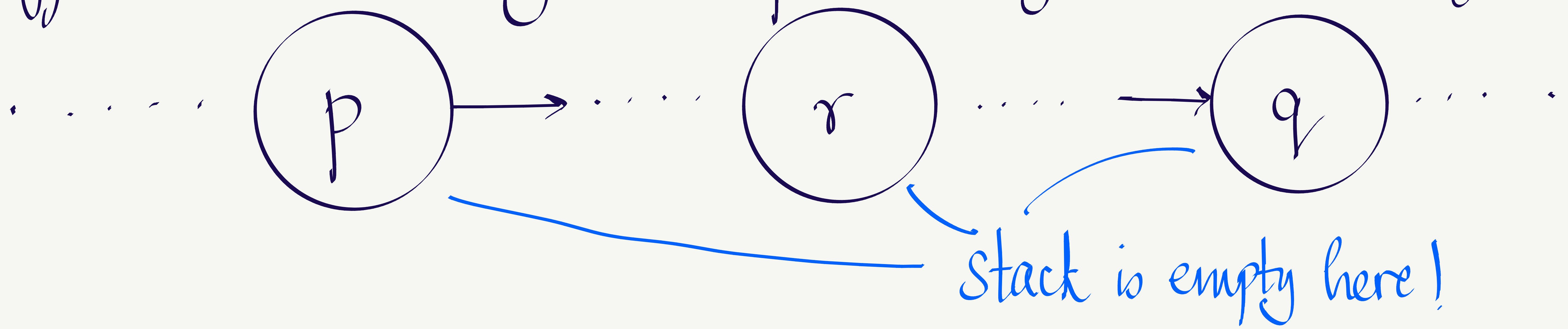
What does the stack contain when the machine enters r? γ

What does the stack contain when the machine enters r? γ

Whatever else was pushed onto the stack between r and s is popped off by the end of that section of the run!

So, A_{pq} can be replaced by $aA_{rs}b$, i.e. $A_{pq} \rightarrow aA_{rs}b$.

(b) The symbol that is initially pushed onto the stack is popped off the stack midway. The operation of M looks as follows



$x = yz$ s.t. y takes M from (p, empty stack) to (r, empty stack), and z takes M from (r, empty stack) to (q, empty stack)

$A_{pq} \longrightarrow A_{pr} A_{rq}$