Context-Free ⇔ Pushdown Recognition

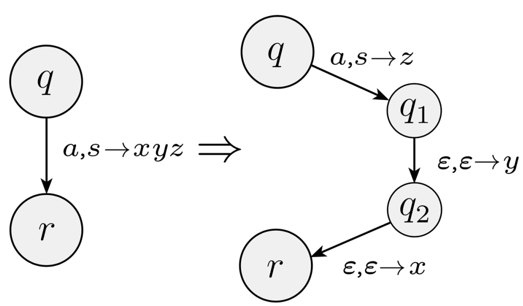
Sipser: Section 2.2 pages 117 - 125

PDA Example

Show the state diagram for a PDA that recognizes { 0^n1^n | $n \geq 0$ }.

Writing Strings to Stack

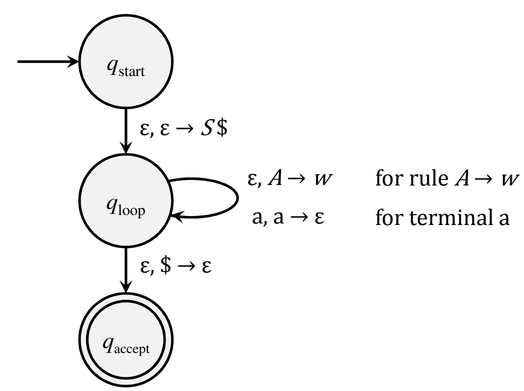
Suppose we want to write strings (multiple characters) to a PDA stack…



Recognizing Context-Free Languages

Lemma. If a language is context-free, then some pushdown automaton recognizes it.

Proof.



For Example

We apply this construction to $G = (V, \Sigma, R, S)$, where

$V = \{S\}$,

$\Sigma = \{[,]\}$,

$R = \{ S \rightarrow \varepsilon \mid SS \mid [S] \}$.

Generating Pushdown Languages

Lemma. If a pushdown automaton recognizes some language, then it is context-free.

Let $P = (Q, \Sigma, \Gamma, \delta_0, F)$ be given…

Assume WLOG:

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

2. P empties its stack before accepting.

3. Each transition does either a *push* or a *pop*, but not both.

The Variables of G

Proof. Given $P = (Q, \Sigma, \Gamma, \delta, q_0, \{q_{accept}\})$ construct

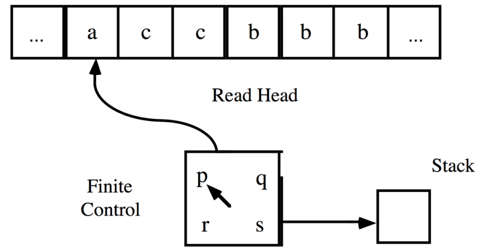
$G = (V, \Sigma, R, S)$, where

$V = \{A_{pq} \mid p, q \in Q\}$ (A_{pq} generates all strings that take P from p with empty stack to q with empty stack.)

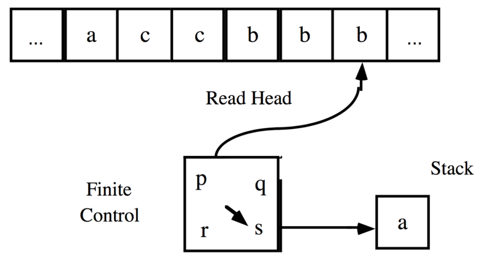
$S = A_{q_0, q_{accept}}$

P's Operation on Strings of A_{pq}

P's first move had better be a *push*.

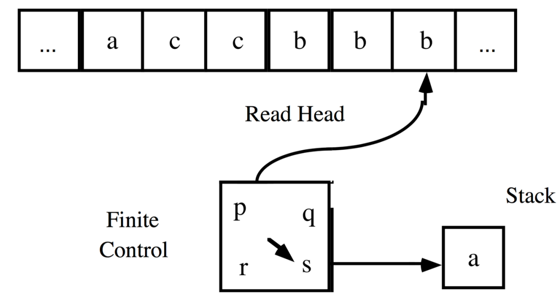


P's last move had better be a *pop*.



For Strings of A_{pq}, Either ...

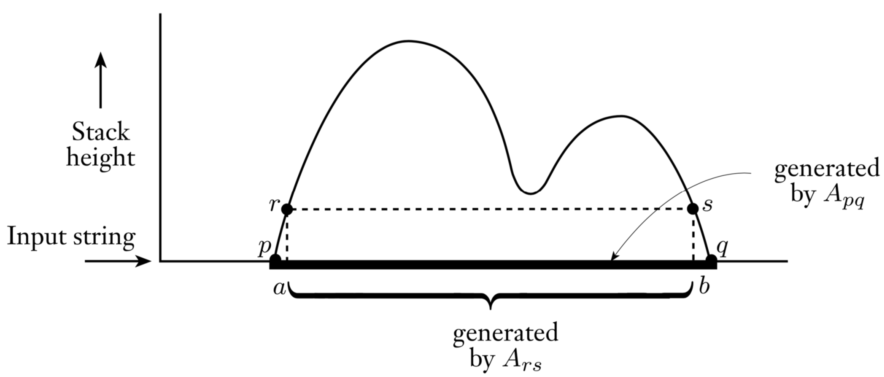
The symbol pushed at the beginning is the symbol popped at the end.



We model this with a grammar rule

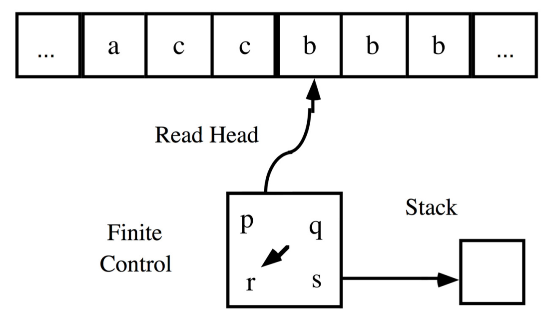
$$A_{pq} \longrightarrow aA_{rs}b$$

Intuitively, $A_{pq} \longrightarrow aA_{rs}b$ ...



Or ...

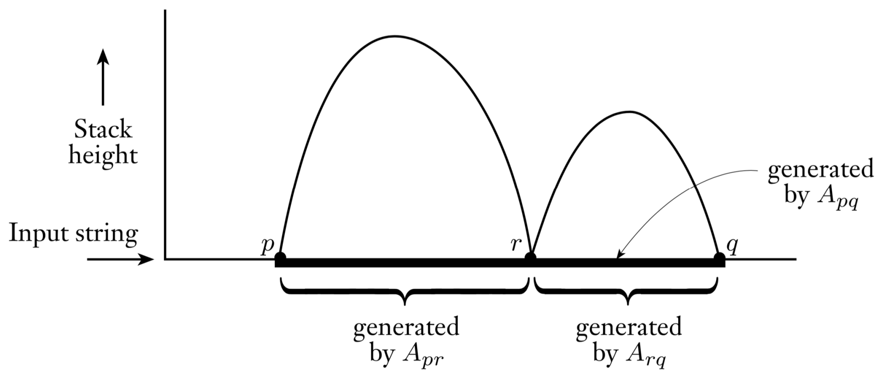
The symbol initially pushed onto the stack is popped somewhere before the end of the string.



We model this with a grammar rule

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

Intuitively, $A_{pq} \longrightarrow A_{pr}\, A_{rs}$ ...



Formally

If $(r, t) \in \delta(p, a, \varepsilon)$ and $(q, \varepsilon) \in \delta(s, b, t)$ then add the rule

$$A_{pq} \longrightarrow aA_{rs}b$$

For each $p, q, r \in Q$, add the rule

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

For each $p \in Q$, add the rule

$$A_{pp} \longrightarrow \varepsilon$$

Chomsky Hierarchy of Languages (Partial)

Corollary. Every regular language is context-free.