Pushdown Automata

Introduction

Automata with finite control state + pushdown stack

- Control state: handles viewing current state in FA, are there transitions going out that matches current symbol in string, does it end with an accept.
- Pushdown stack: idk, probably will know about it after this though

Pushdown automata equivalent to context-free grammar

Outline

- Pushdown Automata
- Context-Free Grammar to PDA
- PDA to Context-Free Grammar

Pushdown Automata (PDA)

A **pushdown automaton** is a 6-tuple $M = (Q, \Sigma, \Gamma, \delta, q_0, F)$ where:

- Q is the finite set of states
- Σ is the input alphabet
- Γ is the stack alphabet
- δ $Q \times \Sigma \times \Gamma \mapsto P(Q \times \Gamma)$ is the nondeterministic transition function
- q_o is the start state
- $F \subseteq Q$ is the set of accept states

PDA State

PDA Transition

- Read next input symbol σ
- Pop top stack symbol γ_k
- Push new stack symbol γ_l
- Move from predecessor q_i to successor state q_j

Stack Symbols:

- ϵ push/pop nothing
- $: marks bottom of stack
- $\gamma \in \Gamma$: any other symbol

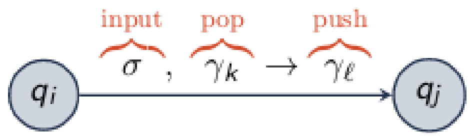
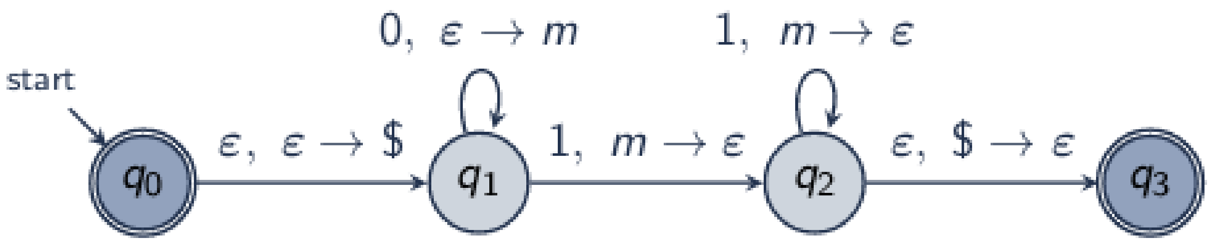


Figure 1: Format of PDA transitions

Example PDA for 0^n1^n



Explanation:

For each transition on q_1 (adding 0's) we push **m** to the stack.

For each transition on q_2 (adding 1's) we pop **m** from the stack.

When the stack is empty again we will go to q_3 and end. This gives us an even number of 0's 1's because the amount of **m**'s pushed must be identical to **m**'s popped.

Example, matching 0011 on above PDA

Start with pushing two m's on the stack

Pop two m's from the stack

End on q_3

Context-Free Grammar to PDA

Given: A CFG

Find: An equivalent PDA

Approach: Construct a PDA corresponding to a left-to-right scan, left-most derivation:

- Store the current RHS of productions on the stack
- Pop terminals on top of the stack, reading from the input string
- Nondeterministically expand nonterminals on top of the stack by popping the nonterminal pushing its RHS

Extended PDA Diagrams

We can rewrite PDA diagrams with multiple transitions. Easier to represent when doing CFG representations.

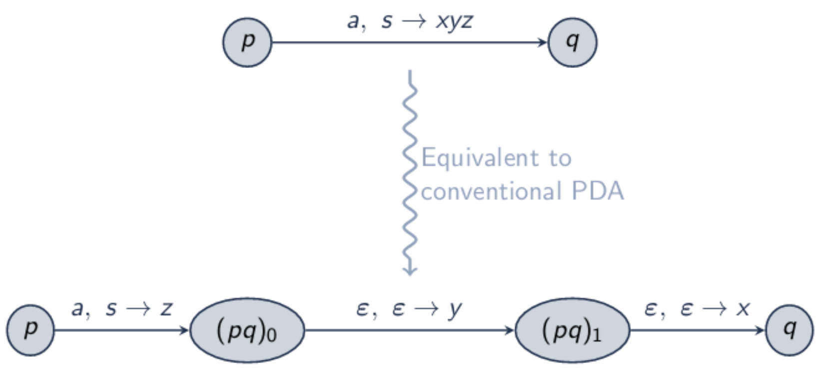
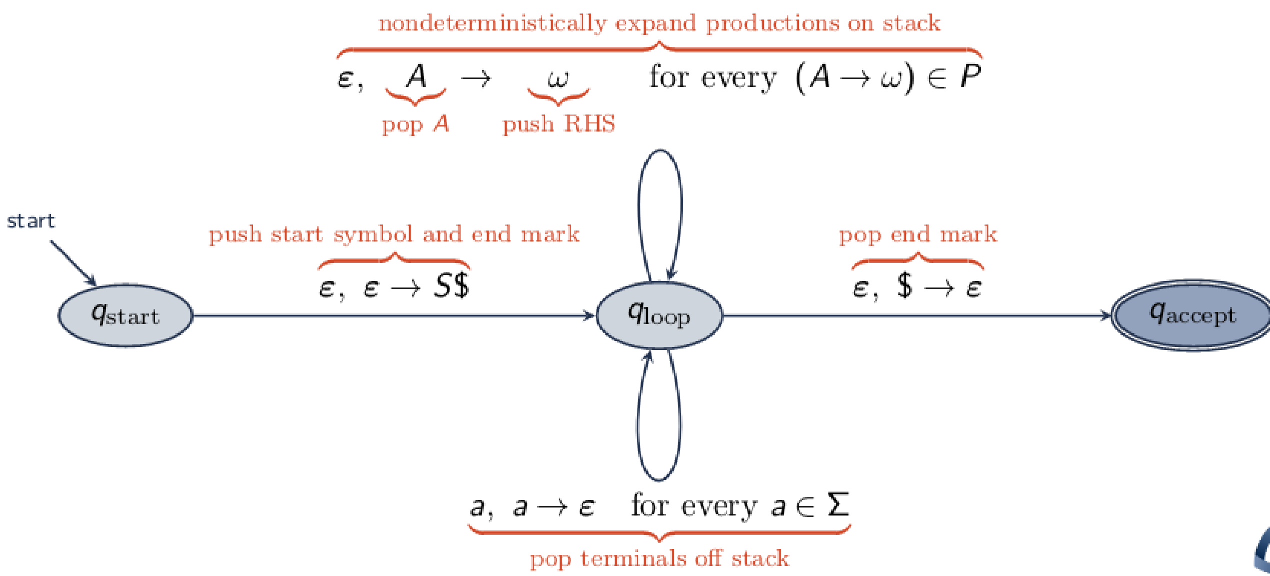


Figure 2: Example of PDA with multiple stack additions for a single transition

CFGs to PDAs



Explanation:

- For the transition (from start) and (to accept). They use a dollar sign to determine when to stop for the CFG.
- One of the self transitions for q_{loop} is for all the possible terminal transitions.
  - For example a,a $\rightarrow \epsilon$ is used to pop the terminal a off the stack
- The other self transitions is for the expansion of nonterminals.

PDA to Context-Free Grammar

Idea: CFG derivations simulate PDA transitions

Given: PDA M = (Q, Σ, Γ, δ, q_0, F)

Find: Equivalent CFG G = (V, T, P, S)

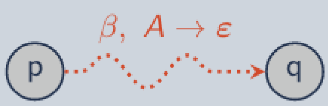
Assume: M has a single start state pushes that pushes $ and a single accept state with empty stack. We can always convert M to this form with symbols/state push/empty the stack

Approach: Construct nonterminals (p, A, q) indicating PDA traces that:

- Begin in state p
- End in state q
- Erase A from the stack

PDA to CFG Nonterminals

PDA Transitions



- Begin in state p
- End in state q
- Pop A from the stack
- Consume terminal (sub)string β

Grammar Nonterminal

$$\langle p, A, q\rangle$$

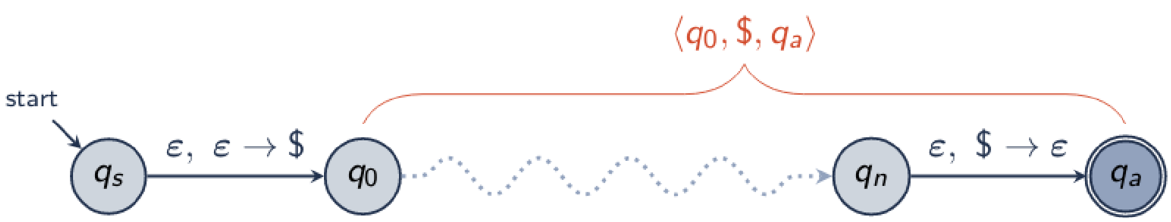
$p \in Q$: initial state in PDA

$q \in Q$: ending state in PDA

$A \in \Gamma$: stack symbol popped

$\beta \in \Sigma^*$: derived string

Start Symbol Production



Start Symbol $\langle S\rangle$

$$\langle S\rangle \to \langle q_0, \$, q_a\rangle$$

- Begin in q_0
- End in q_a
- Pop stack bottom marker $

Productions for <p,X,q>

Pop Edge: Direct

$$p \xrightarrow{a,\ X \to \varepsilon} q \quad \rightsquigarrow \quad \langle p, X, q \rangle \to a$$

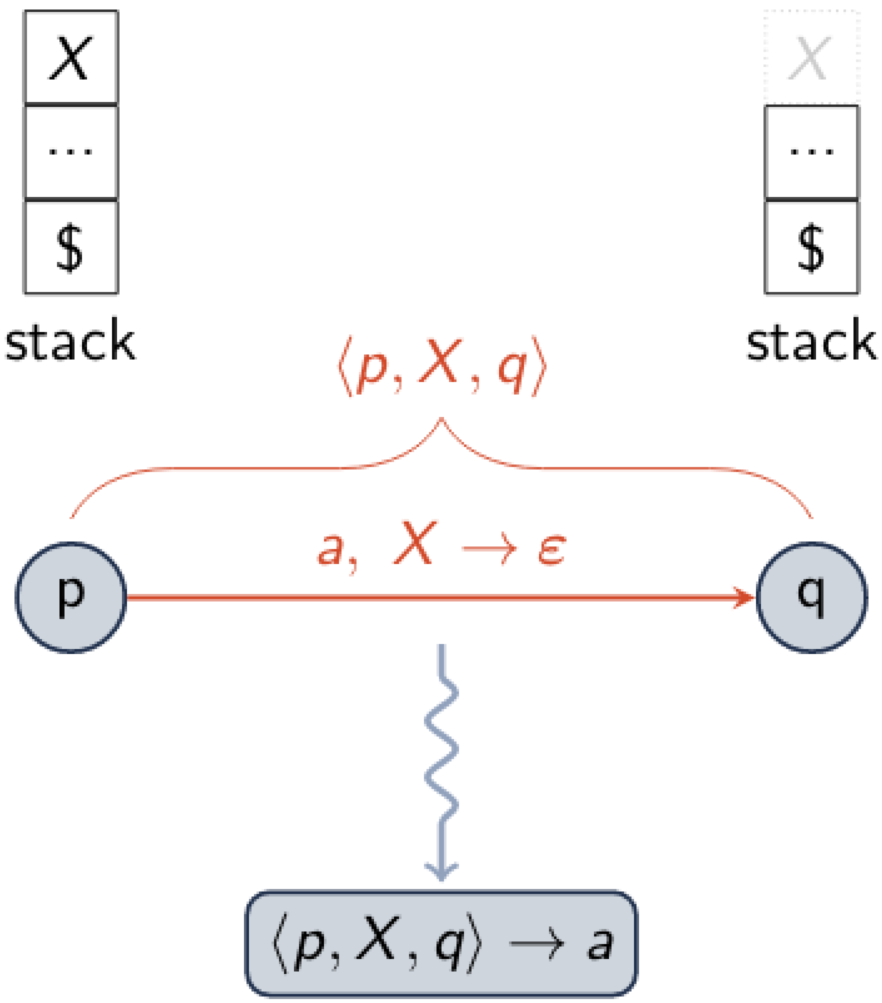
Replace Edge: One intermediate nonterminal through r

$$p \xrightarrow{a,\ X \to Y} r \quad \rightsquigarrow \quad \langle p, X, q \rangle \to a \langle r, Y, q \rangle$$

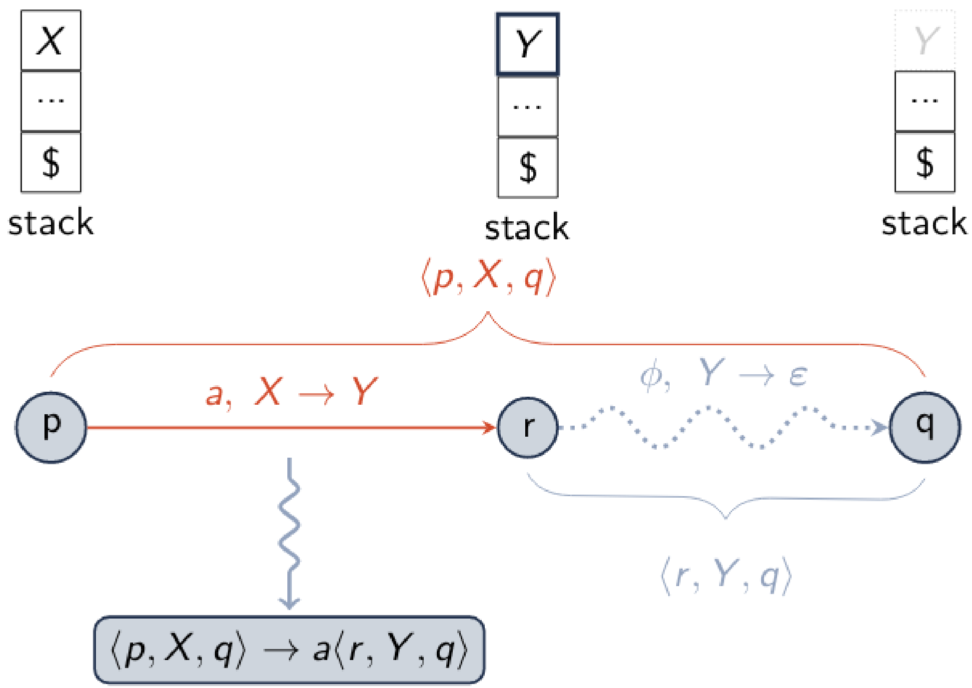
Push Edge: Two intermediate nonterminals through r and s

$$p \xrightarrow{a,\ \varepsilon \to Y} r \quad \rightsquigarrow \quad \langle p, X, q \rangle \to a \langle r, Y, s \rangle \langle s, X, q \rangle \text{ for all } s \in Q$$

Pop Case



Replace Case



Push Case