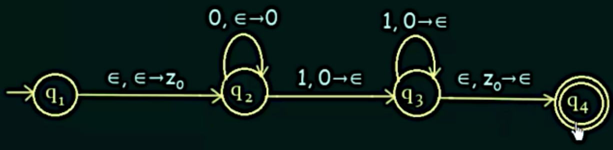
Example: Construct a PDA that accepts $L = \{ 0^n 1^n \mid n \geq 0 \}$



Pushdown Automata (Formal Definition)

A Pushdown Automata is formally defined by 7 Tuples as shown below:

$P = (Q, \Sigma, \Gamma, \delta, q_0, z_0, F)$

where,

- Q = A finite set of States
- Σ = A finite set of Input Symbols
- Γ = A finite Stack Alphabet
- δ = The Transition Function
- q_0 = The Start State
- z_0 = The Start Stack Symbol
- F = The set of Final / Accepting States

δ takes as argument a triple $\delta(q, a, X)$ where:

(i) q is a State in Q

(ii) a is either an Input Symbol in Σ or $a = \epsilon$

(iii) X is a Stack Symbol, that is a member of Γ

he output of δ is finite set of pairs (p, γ) where:

p is a new state

γ is a string of stack symbols that replaces X at the top of the stack

g. If $\gamma = \epsilon$ then the stack is popped

If $\gamma = X$ then the stack is unchanged

f $\gamma = YZ$ then X is replaced by Z and Y is pushed onto the stack

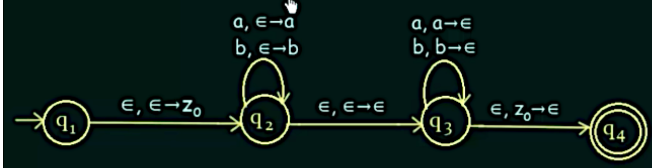
Pushdown Automata - Example (Even Palindrome) PART-1

Construct a PDA that accepts Even Palindromes of the form

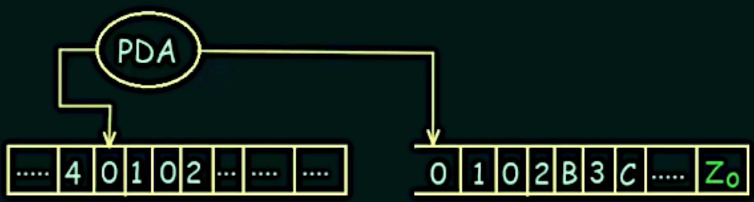
$L = \{ ww^R \mid w = (a+b)^+ \}$

PALINDROMES: A word or sequence that reads the same backwards as forwards.

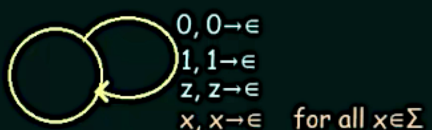
Examples: NOON
NO LEMON NO MELON
123321
abba
RACECAR



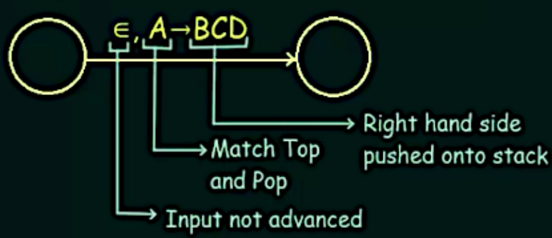
Rule: $A \rightarrow 0\,1\,0\,2\,B\,3\,C$

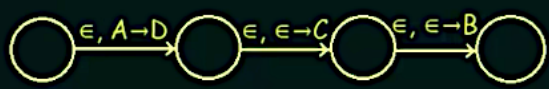


MATCH TERMINAL SYMBOLS TO THE STACK TOP



Rule: $A \rightarrow BCD$ Add this to the PDA





Turing Machine (Formal Definition)

A Turing Machine can be defined as a set of 7 tuples

$(Q, \Sigma, \Gamma, \delta, q_0, b, F)$

$Q \rightarrow$ Non empty set of States

$\Sigma \rightarrow$ Non empty set of Symbols

$\Gamma \rightarrow$ Non empty set of Tape Symbols

$\delta \rightarrow$ Transition function defined as

$Q \times \Sigma \rightarrow \Gamma \times (R/L) \times Q$

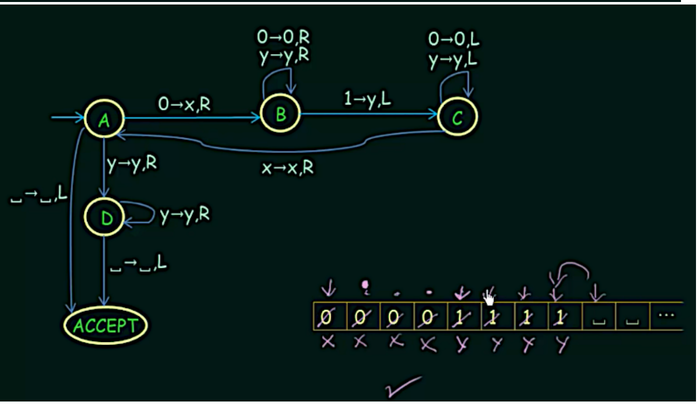
$q_0 \rightarrow$ Initial State

$b \rightarrow$ Blank Symbol

$F \rightarrow$ Set of Final states (Accept state & Reject State)

Thus, the Production rule of Turing Machine will be written as

$\delta(q_0, a) \rightarrow (q_1, y, R)$



Pumping Lemma (Context-Free)

If L is a context-free language, them there exists a number p (pumping length) where any string $w \in L$ where $|w| \geq p$ the string can be divided into five pieces $w = uvxyz$ where the following conditions are satisfied:

1. $|vxy| \leq p$
2. $|vy| > 0$
3. $uv^ixy^iz \in L$ for all $i \geq 0$

Context Free Grammar

A context free grammar is a 4-tuple (V, Σ, R, S)

- V: is a finite set of **variables**
- Σ: is a finite set of **terminals** - disjoint from V
- R: finite set of **rules**
- $S \in V$ the **start variable**

uAv **yields** uwv, written as $uAv \Rightarrow uwv$

Means you can get from uAv to uwv in one "step" by applying a rule on A

u **derives** v, written as $u \overset{*}{\Rightarrow} v$

Means either $u = v$

Or starting at u then applying a series of rules, you can get to v

(ie. there exists a sequence $u_1, u_2, u_3, \ldots, u_k$ for $k \geq 0$ such that

$u \Rightarrow u_1 \Rightarrow u_2 \Rightarrow u_3 \Rightarrow \ldots \Rightarrow u_k \Rightarrow v$)

4) Consider the following CFG $G = (\{S, A, B\}, \{a, b\}, R, S)$ where the rules in R are given as follows.

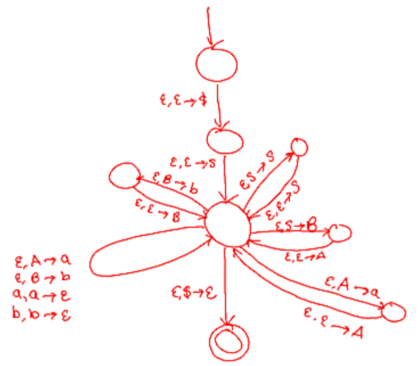
$$S \to SS \mid AB$$
$$A \to Aa \mid a$$
$$B \to Bb \mid b$$

(a) Show that G is ambiguous by giving two leftmost derivations of a string in $L(G)$.

1) $S \Rightarrow SS \Rightarrow SSS \Rightarrow ABSS \Rightarrow aBSS \Rightarrow abSS \Rightarrow abABS \Rightarrow abaBS \Rightarrow ababS \Rightarrow ababAB \Rightarrow ababaB \Rightarrow ababab$

2) $S \Rightarrow SS \Rightarrow ABS \Rightarrow aBS \Rightarrow abS \Rightarrow abSS \Rightarrow abABS \Rightarrow abaBS \Rightarrow ababS \Rightarrow ababAB \Rightarrow ababaB \Rightarrow ababab$

(b) Convert G to an equivalent PDA following the steps of the CFG to PDA conversion.



6) Prove that the language $L = \{0^n \mid n > 0,\ n \text{ is a prime number}\}$ is not context free using the pumping lemma for context free languages.

Assume for a contradiction that L is context free.

Let p be the pumping length given by the PL and let p' be the smallest prime number $\geq p$.

Choose $s = 0^{p'}$. $s \in L$ and $|s| \geq p$, so should be able to write $s = uvxyz$ such that all 3 properties of the PL are satisfied.

By property 1 of the PL, $|vy| > 0$. Let $m = |vy|$.

By property 3 of the PL, $uv^ixy^iz \in L$ for each i. uv^ixy^iz has form $0^{p' + (i-1)m}$

However, not all numbers $p' + (i-1)m$ are prime. Consider $i = p'+1$. Then uv^ixy^iz is the string $0^{p' + p'k} = 0^{p'(k+1)}$. The number of 0's in this string has factors p' and $(k+1)$, where $p' \geq 2$ since p' is prime and $k+1 > 1$ since $k > 0$. So, $uv^ixy^iz \notin L$ for $i = p'+1$.

Thus, we cannot satisfy both properties 1 and 3 in any rewriting of s as $uvxyz$.

Therefore, L is not context free.

Chomsky Normal Form

A context-free grammar $G = (V, \Sigma, R\ S)$ is in chomsky normal form (CNF) if every rule is in the form:

$A \to BC$ where $A, B, C \in V$ (B and C cannot be he start variable)

or

$A \to a$ where $A \in V$ and $a \in \Sigma$

and

$S \to \varepsilon$ is only permitted where S is the start variable

CNF Steps

1. Add a new start variable
2. Eliminate all ε-rules $(A \to \varepsilon)$
3. Eliminate all unit rules $(A \to B)$
4. Convert remaining rules to be in the form $A \to BC$ or $A \to a$

Push-down Automaton

A push-down automaton is defined as a 6-tuple $(Q, \Sigma, \Gamma, \delta, q_0, F)$

Q: finite set of states

Σ: finite set of input alphabet

Γ: finite stack alphabet $(\Sigma \subseteq \Gamma)$

δ: $Q \times \Sigma_\varepsilon \times \Gamma_\varepsilon \to P(Q \times \Sigma_\varepsilon)$

$q_0 \in Q$: start state

$F \subseteq Q$: set of accept states

(c) Convert G into Chomsky Normal Form. Show all your steps.

Step 1: $S_0 \to S$
$S \to SS \mid AB$
$A \to Aa \mid a$
$B \to Bb \mid b$

Step 2: (No ε-rules)

Step 3: $S_0 \to SS \mid AB$
$S \to SS \mid AB$
$A \to Aa \mid a$
$B \to Bb \mid b$

Step 4: $S_0 \to SS \mid AB$
$S \to SS \mid AB$
$A \to AX \mid a$
$B \to BY \mid b$
$X \to a$
$Y \to b$

5) Consider the language $L = \{0^i 1^j 2^k \mid i, j, k \geq 0 \text{ and } i + k = j\}$.

(a) Give a context free grammar G with $L(G) = L$.

$S \to AB$
$A \to 0A1 \mid \varepsilon$
$B \to 1B2 \mid \varepsilon$

(b) Give a state diagram for a PDA which recognizes L (without using the CFG to PDA conversion).

7) Give a high-level description of a Turing machine which recognizes the following language:

$$L = \{0^i 1^j 2^k \mid i \times j = k \text{ and } i, j, k \geq 1\}$$

1. Scan the tape from left to right and check if the input string has form some 0's followed by some 1's followed by some 2's. If not, reject. If so, return tape head to beginning.
2. Mark a 0 then move the tape head to the start of 1's. Alternate between 1's and 2's and mark each one until all 1's are marked. If any 1's remain after all 2's have been marked, reject.
3. Unmark all of the 1's and repeat step 2 if there are more 0's left unmarked. If all 0's are marked:
   - If all 2's are marked, accept
   - Otherwise, reject.