TOC

Home Assignment-3.

1) PDA is a 6 tuple machine

$P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0)$

Q -> set of states Γ -> stack

Σ -> I/P symbols δ -> transition.

q_0 -> start Z_0 -> initial stack

transitions

pushing A for each a

$\delta(q_0, a, Z_0) \rightarrow (q_0 A Z_0)$

$\delta(q_0 a A) \rightarrow (q_0 A A)$

popping phase.

$\delta(q_0 b A) \rightarrow (q_1, \epsilon)$

pop A for each b

$\delta(q_1 b A) \rightarrow (q_1, \epsilon)$

stack is empty I/P is done.

$\delta(q_1 \epsilon Z_0) \rightarrow (q\_accept, \epsilon)$

Stack operations

I/P : aaabbb

state.	I/P	stack	action.
q_0	aaabbb	Z0	a, PUSH A
q_0	aaabbb	AZ0	a, PUSH A
q_0	abbb	AAZ0	a, PUSH A
q_0	bbb	AAAZ0	b, q_1, POP A
q_1	bb	AAZ0	b, POP A
q_1	bb	AZ0	b, POP A
q_1	ϵ	Z0	POP Z0
q_{accept}	ϵ	ϵ	empty.

2) PDA design

q_0 : Start state

q_1 : midpoint

q_{accept}

I/P symbols (eg: a, b)

stack bottom (eg: Z)

PDA transitions

$(q_0\ a, Z) \rightarrow (q_0, aZ) \rightarrow$ push a

$(q_0\ b, Z) \rightarrow (q_0, bZ) \rightarrow$ push b

$(q_0\ a\ a) \rightarrow (q_0, aa) \rightarrow$ push a

$(q_0, b, b) \rightarrow (q_0, bb) \rightarrow$ push b

$(q_0, \epsilon, S) \rightarrow (q_1, S)$ -> switch to q_1

$(q_0, a, a) \rightarrow (q_1, \epsilon)$ -> match a with top stack

$(q_1, b, b) \rightarrow (q_1, \epsilon)$ -> math b with top stack

$(q_1, \epsilon, z) \rightarrow (q_{accept}, z)$ -> only z left

I/P :- abba

State q_0 : I/P :- abba

state	I/P	stack	action
q_0	abba	z	a, push az
q_0	bba	az	a, push baz
q_0	ba	baz	baz
q_1	ba	baz	b, az
q_1	a	az	a, z
q_1	ϵ	z	q_{accept} -> accepted.

PDA accepts the I/P abba by guessing the midpoint after ab and matching ba as reverse of ab.

3) prove $L_3 = \{a^n b^n c^n \mid n \geq 1\}$ is not "CFL"

Applying the pumping lemma to prove L_3 is not a CFL

let L_3 is a CFL

then by pumping lemma, there exists a length p such that $z \in L_3$ with $|z| \geq p$, z can be split into uvwxy statisfying conditions.

let $z = a^n b^n c^n \in L_3$

z has equal of a, b and c.

$z = uvwxy$.

~~let~~ $|uwx| \leq p$ and $vx \neq \epsilon$

→ uwx contains only a's and b's but not all three of a, b and c.

⇒ pump i ≠ 1 are modifying no. of a's and b's the no. of c's remain same.

let $i = 2$

uv^2wx^2y.

∴ Thus the no. of a's, b's and c's will no longer be equal so

$$uv^2wx^2y \notin L_3$$

∴ L_3 is context free, leads to a contradiction

-> ∴ $L_3 = \{a^n b^n c^n / n \geq 1\}$ is not a CFL

4) High level logic

=> Matches the first half of the string to the 2nd half, ensuring each character in 1st half as matching one in the 2nd half.

-> The total string length must be even.

Sketch of States and Transitions

i) q_start: Read a or b mark it as x, remember what was read, move right to find 2nd half

2) q_find_match a |q_find_match_b.

3) q_back -> return to start to mark
the next unmatched character

4) q_check_done

5) q_reject.

$L_4 = \{ww\}$

$w = ab \rightarrow ww = \{abab\}$

-> read a, mark as x, go to q_find_a

-> skip b, see a -> match found,
mark as x

-> no back to stat, stop at 1st b

-> q_start, read b, mark as x,

-> skip x, skip x, find b -> mark as x

-> no back, check any unmatched
a or b left -> accept.

5) Assumption (for contradiction)

If a function halts(P, I) exists.

halts(P, I) returns True.

If program P halts on I

and False if P runs forever on I

contradictory Program

paradox(P) does opposite of P(P)

-> If P(P) halts, then pardox(P) loops forever

-> If P(P) loops forever, the pardox(P) halts

our assumption that a function halts(P, I) exists must be false.

:- no such universal halting function can exist.