Name : Sindhuja Yerramalla

UUID : U00839259

email : syrrmlla@memphis.edu .

Foundations of Computing - HW3.

Scanned with CamScanner

Problem 1: Problem 6 (d), Sec 7.1 from Linz.

$L = \{w : 2n_a(w) \le n_b(w) \le 3n_a(w)\}$

~~Given~~ Let M be the NPDA such that

$M = [\{q_0, q_1, q_2, \cancel{q_3}\}, \Sigma, \{a, b, z\}, \delta, q_0, z, \{q_2\}]$

let the δ transition be as:

$\delta(q_0, a, z) = \{(q_0, aaz)\}$

$\delta(q_0, a, z) = \{(q_0, aaaz)\}$

$\delta(q_0, a, a) = \{(q_0, aaa)\}$

$\delta(q_0, a, a) = \{(q_0, aaaa)\}$

$\delta(q_0, b, z) = \{(q_0, bz)\}$

$\delta(q_0, b, b) = \{(q_0, bb)\}$

$\delta(q_2, \lambda, z) = \{(q_0, az)\}$

$\delta(q_0, b, a) = \{(q_0, \lambda)\}$

$\delta(q_0, a, b) = \{(q_1, \lambda)\}$

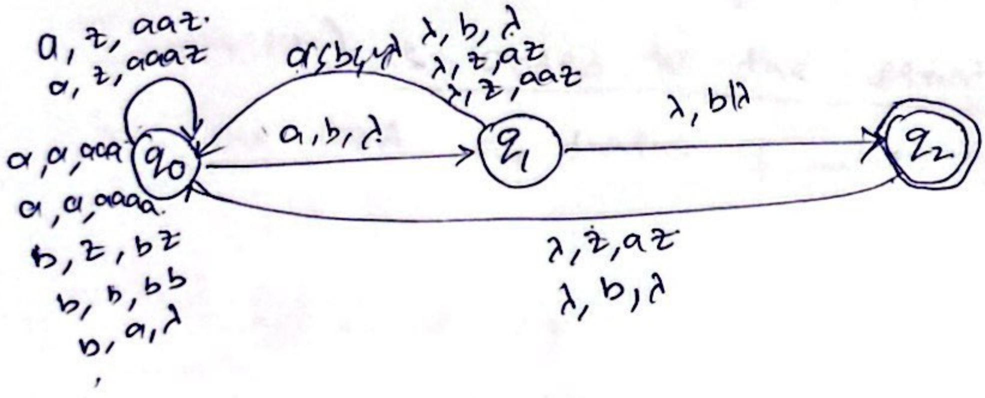
$\delta(q_1, \lambda, b) = \{(q_0, \lambda)\}$

$\delta(q_1, \lambda, z) = \{(q_0, az)\}$

$\delta(q_1, \lambda, z) = \{(q_0, aaz)\}$

$\delta(q_1, \lambda, b) = \{(q_2, \lambda)\}$

$\delta(q_2, \lambda, b) = \{(q_0, \lambda)\}$

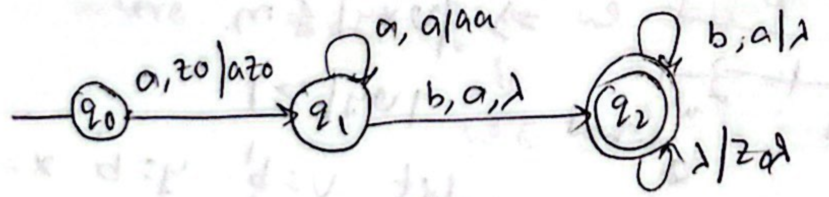


Problem 2 : Problem 19 from Sec 7.1 Linz.

As per given statement, we need to prove that if a language is accepted by final state method then that will be accepted by an empty stack method too. and viceversa. Consider that the 'M' be a Push down Automata that accepts language 'L' by empty stack method and 'M_2' be a push down Automata that accepts Language(L) by fsm. so we need to prove that if there is M then there is M_2 as well accepting the same Language 'L'.

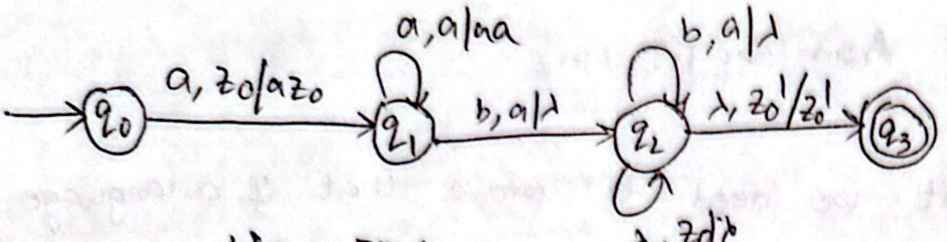
We Know that a stack can be empty at any control state. If we start the PDA with a new bottom of stack Symbol Z_0' and the PDA of normal operation of new PDA. The top as any state of z_0' shows up on stack. A new transition (λ transition) is added to the final state in new PDA The old PDA as drains of when ever by stack the new state q_f

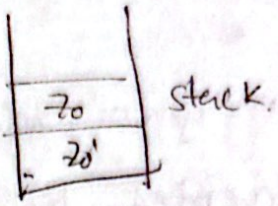
Ex $L = \{a^n b^n \mid n \geq 1\}$



Old PDA

The stack as z_0 as stack of empty by string As using accept by empty stack method Them M acept by finalstate.





New PDA

$\therefore$ we have an equivalent PDA that accepts same Language for empty stack.

Hence proved.

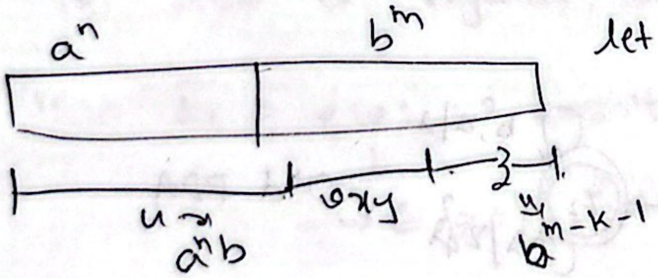
Problem 3 : Problem 7(i), Sec 8.1 from Linz.

given $L = \{a^n b^m : n \text{ is prime or } m \text{ is prime}\}$.

Let us Assume that given L is Context-free. then it should satisfy pumping lemma.

let m be a positive prime. and $n = m+i$. and n is prime. and

$\Rightarrow \omega = a^{m+i} b^m = uvxyz$, $|\omega| \geq m$. & $\omega \in L$.



let $vxy = b^K$. for $K \leq m$.

$\Rightarrow |vxy| \leq m$.

$|vy| \geq 1$

let $v = b^1$ $y = b^1$ $x = b^{K-2}$

As per pumping lemma $uv^ixy^iz \in L$ let $i = 2$.

the $uvvxyyz$

$a^n b \; b b \; b^{K-2} \; b b \; b^{m-K-1} \Rightarrow a^n b^{m+2}$.

we cannot say $m+2$ is always prime

$m+2$ might be not ~~make~~ always a prime, in that condition.

$uv^2xy^2z \notin L$

This is contradiction, so our assumption is wrong

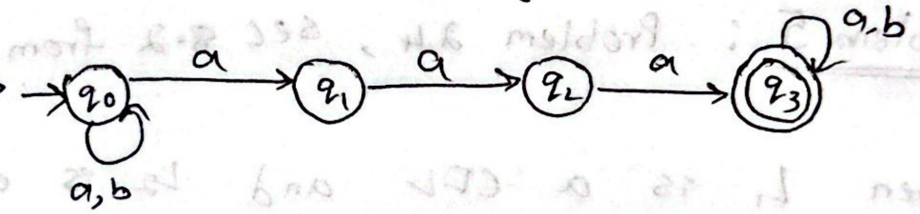
$\therefore$ $L = \{a^n b^m : n$ is prime or m is prime$\}$ is not CFL //.

<u>Problem 4</u> :- Problem 18, sec 8.2 from linz.

Given $L = \{w \in \{a,b\}^* : n_a(w) = n_b(w) : w$ does not contain a substring $aab\}$

Let $L_1 = \{w \in \{a,b\}^* : w$ has substring $aab\}$

NFA for L_1 =>



Since we have an NFA that acceps L_1, we can say that

L_1 is a regular language

=> $\overline{L_1}$ is also a regular language. ——(1)

where $\overline{L_1} = \{w : w$ doesn't have aab substring$\}$
$w \in \{a,b\}^*$

let $L_2 = \{w \in \{a,b\}^* ; n_a(w) = n_b(w)\}$. This is CFL as discussed in class we know that if L_1 is a CFL and L' be a regular language the $L \cap L'$ is also a CFL.

$\therefore$ $\overline{L_1} \cap L_2$ is a CFL, since $\overline{L_1}$ is regular [(1)]
& L_2 is CFL

$\overline{L_1} \cap L_2 = \{ w \in \{a,b\}^* : n_a(w) = n_b(w) ; w$ doesn't contain a substring aab$\}$

$\therefore$ The given language is CFL, hence proved //.

Problem 5 : Problem 24, sec 8.2 from linz

Given L_1 is a CFL and L_2 is a regular language.

from Theorem 8.5 (linz) we can say that

when L_1 is a CFL & L_2 is regular then $L_1 \cap L_2$ is CFL.

$\Rightarrow$ $L_1 \cap L_2$ is CFL —— (1)

let $L = L_1 \cap L_2$ & grammar for L be $G = (V, T, S, P)$

We then convert this grammar to CYK by removing all λ transitions & useless symbols and productions

In this process if S is found to be useless then $L(G)$ is empty. if not then $L(G)$ contains atleast 1 element. $\longrightarrow$ Theorem 8.6 (linz) —— (2)

$\therefore$ from (1) & (2) [i.e from Theorem 8.5 & 8.6] we can say that there exists an algorithm to determine or not L_1, L_2 has a common elements

$\therefore$ hence proved //.