ambiguous context free language.

5

Pumbing lemma for CFG

Let L be an infinite context free language. Then there exist some positive integer 'n' such that any $w \in L$ with $|w| \geq n$ can be decomposed as

$w = uvxyz$

with

$|vxy| \leq n$

and

$vy \neq \epsilon$

Such that $uv^ixy^iz \in L$, for all $i \geq 0$

4/10/2017

Application of Pumbing lemma

Pumbing lemma for CFL is used to check whether the language is context free or not

eg: To check the language is regular or not.

Q: Find out the language $L = \{a^nb^nc^n / n \geq 1\}$ is context free or not?

Solution:-

Step 1: Let assume L is context free language. Then L must satisfy pumbing lemma for CFL

Step 2: Choose a +ve number n of the pumbing lemma

Step 3: Take $W = a^nb^nc^n \in L$ ($\because |w| = 3n \geq n$)

Step 4: Decompose w into $uvxyz$ such that

$|vxy| \leq n$ and $vy \neq \epsilon$

$w = \underbrace{a_1a_2 \cdots a_n}_{vxy}\underbrace{b_1b_2 \cdots b_n}_{vxy}\underbrace{c_1c_2 \cdots c_n}_{vxy}$

vxy   vxy

Here we can't split W such a way that vxy contains both a's and c's, since last a and first b^c are atleast n+1 position apart

There are two cases

CASE - I

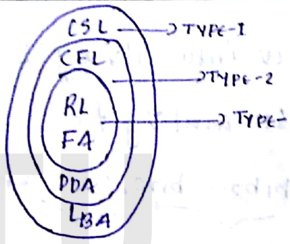
vxy has no 'c'. Then after pumbing vxy twice: uv^2xy^2z not an element of L.

CASE-II

vxy has no 'a' then after pumbing v & y twice $uv^2xy^2z \notin L$

CONTRADICTION to our assumption. Hence L is not CFL.

Context Sensitive Language and Context Sensitive Grammar:



Context sensitive Grammar is also known as Type-1 Grammar.

- Type-1 Grammar (context sensitive Grammar) generates context sensitive languages.
- A context sensitive Grammar $G = \{V, T, S, P\}$ is defined as V = set of variable
  - T = set of terminals = Σ
  - S = Starting symbol of the production
  - P = The production of the form $\alpha A \beta \rightarrow \alpha \gamma \beta$

where $A \in V$, $\alpha\beta \in (V \cup T)^*$, $\gamma \in (V \cup T)^+$

The strings α and β may be empty

* The rule $S \rightarrow \epsilon$ is allowed if S doesn't appear on the right hand side of any rule
* The language generated by these grammars are recognised by a linear bounded automaton

6/10/17 CSG Example

Let $G = \{V, T, S, P\}$ where

$V = \{S, A, B\}$

$T = \{a, b, c\}$

$S = S$

$$P = \begin{cases} S \rightarrow AB \\ AB \rightarrow AbBc \\ A \rightarrow a \\ B \rightarrow bcB \\ B \rightarrow b \end{cases}$$

LINEAR BOUNDED AUTOMATA

A LBA is a single TAPE non deterministic turing machine. Touring m/c with two Tape ~~different~~ Special symbols left end marker and right end marker "<" and ">"

The transition in LBA should satisfy these conditions

1) It shouldn't replace the end marked symbol by any other symbols.

It shouldn't write own cell beyond the marker symbol.

The initial configuration of LBA will be :

$\langle q_0 aaaaa \ldots a \rangle$

⊢	<	a	a	a	a	a	>	
	↑ q_0							

Formal definition

Formally a LBA is a non deterministic turing automata,

$$M = (Q, \Sigma, \Gamma, \delta, q_0, F, <, >)$$

where Q - set of states

Σ = set of input symbol, alphabets

Γ - set of all tape symbols

$\Sigma \subset T$ proper subset

δ = set of all transitions,

q_0 = ~~set of~~ initial state

F = set of final ~~set~~ state

$<$ = left end marker

$>$ = right end marker

10/10/17.

Turing Machine.

Turing machine is called as mathematical model of a computer in theoretical concept it is implemented in 1936 by Allen turing

Formal definition:

A turing machine is defined as 7 tuple

$$M = \{Q, \Sigma, \Gamma, \delta, B\!\!\!/, q_0, F\}$$

whose components have the following meaning.

Q = finite set of states

Σ = " " input symbols called alphabet

Γ = complete set of tape symbols, $\Sigma \subset \Gamma$

δ = transition function, that takes arguments $\delta(Q,a)$ and returns a triple (p,y,d) $\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{R, L\}$

where, q is the current state

a is state symbol

p - next state.

Y - replacing symbol of y
d - direction either left or right
$\cancel{B}$ - B is the blank symbol $B \in \Gamma$
q_0 - Starting state
F final or accepting ~~attept~~ state

Informal definition:

B	B	B	0	0	0	1	1	B	B	B	B	B

R/W head

Finite control unit

The m/c consists of FC which can be any one of finite set of states There is a tape divided into cells each cell can hold any of a finite no: of symbols, The input which is a finite length of string symbols chosen from the input alphabet Σ which is placed on tape all other tape cells extending infinitely to the left and right. A blank is a tape symbol but not a i/p symbol. A Tape head is always positioned at one of the tape cell The turing m/c is said to be scanning that cell if it reads any symbol from that cell.

- A move of a turing m/c is the function of state of the finite control and tape symbol scanned.

Q Turing machine that accept $L = \{a^n b^n \mid n \geq 1\}$

B	B	a	a	a	b	b	b	B	B

q_0

$\Sigma = \{a, b\}$ $\Gamma = \{a, b, X, Y, B\}$ $a \to X$ $b \to Y$

$$\delta(q_0, a) = (q_1, X, R)$$

B	B	X	a	a	b	b	b	B	B

$\delta(q_1, a) = (q_1, a, R)$

$\delta(q_1, b) = (q_2, Y, L)$

B	B	X	a	a	$\cancel{Y}$	b	b	B	B

$\delta(q_2, a) = (q_2, a, L)$

$\delta(q_2, X) = (q_0, X, R)$

$(q_1, Y) = (q_1, Y, R)$

$(q_2, Y) = (q_0, Y, L)$

B	B	X	X	X	y	y	y	B	B

11/10/17

Instantaneous Description of TM (ID)

An ID is used to represent one move or more than one move in turing machine we can use '$\vdash$' this symbol to represent a move in TM and '$\vdash^*$' to repsent multiple moves.

We shall use the string,

$X_1 X_2 X_3 \dots X_{i-2} X_{i-1}\, q X_i\, X_{i+1} \dots X_n$ to represnt an ID. In which,

(*) q is the ~~sta~~ current state of the turing m/c

* The Tape head is scanning i^{th} symbol from the left

* $X_1 X_2 X_3 \dots X_n$ is the portion of the tape b/w leftmost and rightmost blank.

A move in ID ~~can be~~ is represented ~~as~~ in below

CASE-1 :-

If $\delta(q, X_i) = (P, Y, R)$ then next move can be represented as

$X_1 X_2 X_3 \dots X_{i-2} X_{i-1} \boxed{q X_i} X_{i+1} \dots X_n \vdash X_1, X_2 \dots X_{i-1} \boxed{P X_{i+1}} X_{i+2} \dots X_n$

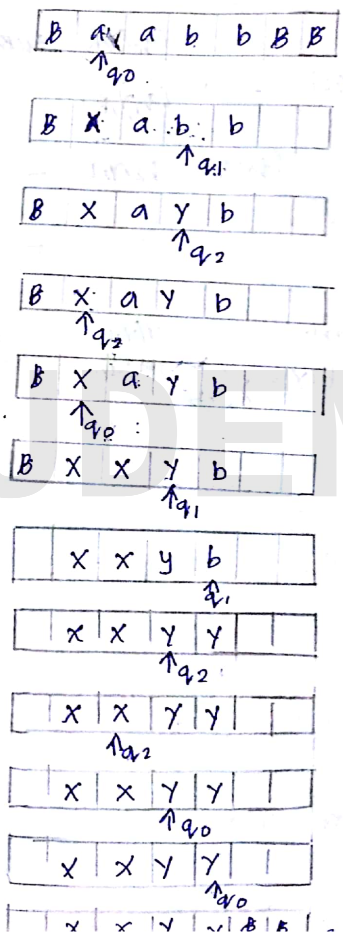
CASE-1

If $\delta(q, X_i) = (P, Y, L)$ then next move can be represented as

$X_1 X_2 X_3 \dots X_{i-2} X_{i-1}\, q X_i\, X_{i+1} \dots X_n \vdash$

B	X_1	X_2	...	X_{i-1}	X_i	X_{i+1}	X_{i+2}	...	X_n	B	B

Q Construct TM that accept language $L = \{a^n b^n \mid n \geq 1\}$



$w c w^R$

$a^n b^n c^n$

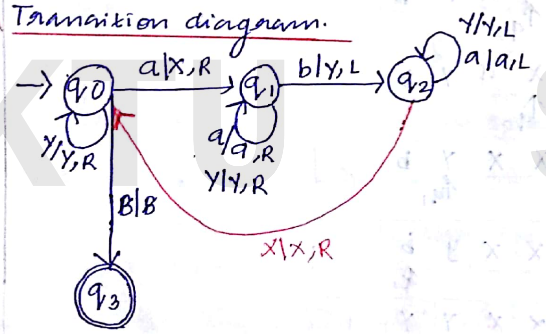
$a^n b^n$

Transition Table.

States ↓	a	b	X	Y	B
→ q_0	(q_1, X, R)	—	—	(q_0, Y, R)	(q_3, B, R) H
q_1	(q_1, a, R)	(q_2, Y, L)	—	(q_1, Y, R)	—
q_2	(q_2, a, L)	—	(q_0, X, R)	(q_2, Y, L)	—
* q_3	—	—	—	—	—

(Tape Alphabet: a, b, X, Y, B)

Transition diagram.



Instantanious discription of a string 'aabb'

w = 'aabb'

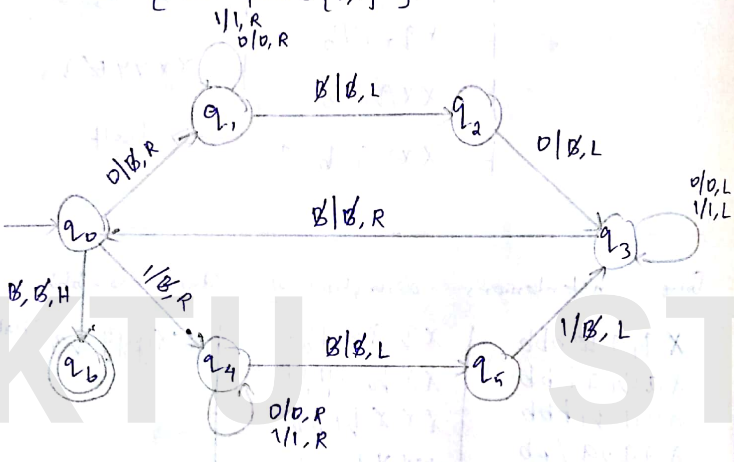
Initial configeration = $q_0 w$

$q_0 aabb \vdash X\boxed{q_1 a}bb$
$\vdash Xa\boxed{q_1 b}b$
$\vdash X\boxed{q_2 a}Yb$
$\vdash \boxed{q_2 X}aYb$
$\vdash X\boxed{q_0 a}Yb$
$\vdash XX\boxed{q_1 Y}b$
$\vdash XXY\boxed{q_1 b}$
$\vdash X\boxed{q_2 X}YY$
$\vdash XX\boxed{q_0 Y}Y$
$\vdash XXY\boxed{q_0 Y}$
$\vdash XXYY\boxed{q_0 B}$
$\vdash XXYYBq_3$
$\Rightarrow$ Halt

Give instantanious description of string aaabbb

$Xq_1aaabbb$
Xaq_1abbb
$Xaaq_1bbb$
Xaq_2aYbb
$q_2Xaaybb$
Xq_0aaYbb
$XXqaYbb$
$XXaqYbb$
$XXaYq_1bb$
$XXaq_2YYb$
XXq_2aYYb
Xq_2XaYYb
XXq_0aYYb
$XXXq_1YYh$
$XXXYq_1Yh$
$XXXYYq h$
$XXXYq_2YY$
$XXXq_2YYY$
XXq_2XYYY
$XXXq_0YYY$
$XXXYq_0YY$
$XXXYYq_0Y$
$XXXYYYq_0B$
$XXXYYYq_3 \rightarrow$ Halt

Q) Construct Turing machine that accepts Language,
$L = \{ww^R \mid w \in \{0,1\}^*\}$



$q_0 \rightarrow$ If 0 comes, make it as B and move right. state to q_1
~~q0~~ If 1 comes, make it as B and move right. state to q_4
$q_1 \rightarrow$ Skips 1's with 1's & 0's with 0's and move right untill it finds B.
$q_2 \rightarrow$ If 0 comes, replace 0 with B & move left state to q_3
$q_4 \rightarrow$ Same as q_1
$q_5 \rightarrow$ If 1 comes, make it as blank. state to q_3
$q_3 \rightarrow$ If B, then replace it with B and move right, state to q_0.

State	Tape symbol 0	1	B
$\rightarrow q_0$	(q_1, B, R)	(q_4, B, R)	(q_6, B, H)
q_1	$(q_1, 0, R)$	$(q_1, 1, R)$	(q_2, B, L)
q_2	(q_3, B, L)		
q_3	$(q_3, 0, L)$	$(q_3, 1, L)$ (q_5, B, L)	
q_4	$(q_4, 0, R)$	$(q_4, 1, R)$	(q_5, B, L)
q_5		(q_3, B, L)	
q_6			

13/10/2017

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

~~$(q_0, a) = q_0$~~

B a a a b b b c c c B

$(q_0, a) = (q_1, X, R)$

$(q_0, Y) = (q_0, Y, R)$ $\quad (q_0, B) = (q_f, B, Halt)$

$(q_0, Z) = (q_0, Z, R)$

$(q_1, a) = (q_1, a, R)$

$(q_1, Y) = (q_1, Y, R)$

$(q_1, b) = (q_2, Y, R)$

$(q_2, b) = (q_2, b, R)$

$(q_3, Z) = (q_2, Z, R)$

$(q_2, c) = (q_3, Z, L)$

$(q_3, Z) = (q_3, Z, L)$

$(q_3, b) = (q_3, b, L)$

$(q_3, Y) = (q_3, Y, L)$

$(q_3, a) = (q_3, a, L)$

$(q_3, X) = (q_0, X, R)$

Transition table

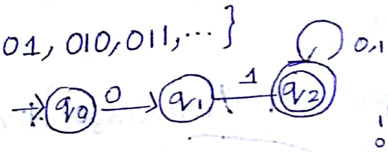
State	Tape Alphabet						
	a	b	c	X	Y	Z	B
q_0	$\{q_1, X, R\}$	~~$\{q_2\ b\ R\}$~~			$(q_0\ Y\ R)$	$(q_0\ Z\ R)$	$(q_f\ B$ Halt$)$
q_1	$\{q_1\ a\ R\}$	$(q_2\ Y\ R)$			$(q_1\ Y\ R)$		
q_2		$(q_2\ b\ R)$	$(q_3\ Z\ L)$			$(q_2\ Z\ R)$	
q_3		$(q_3\ b\ L)$				$(q_3\ Z\ L)$	

Q Construct TM that accept $L = \{w \# w^R / w \in (a,b)^*\}$

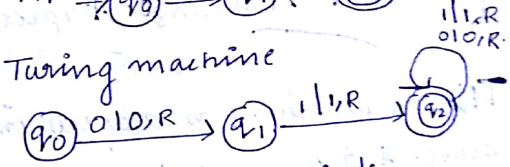
Q Construct TM that start with 01

A $L = \{01, 010, 011, \dots\}$

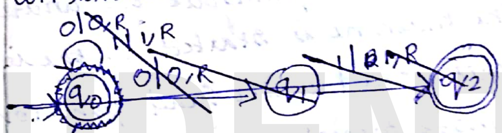
FA



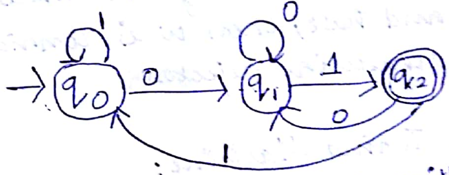
Turing machine



Q Construct TM that ~~start~~ ends with 0,1

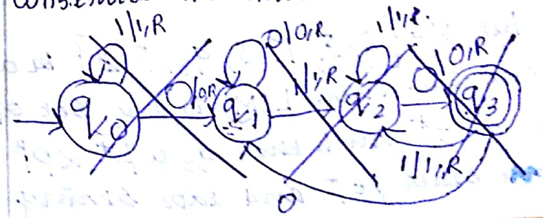


$L = \{01, 001, 101, 0001, 0101, 1001, 1101\}$

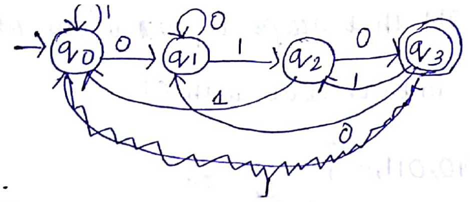


00101

Construct TM that ends with ~~0101~~ 010



0010
1010
00010
01010
10010
11010
01010



17/10/17.

Turing machine as Language Accepters :-

TM can be viewed as accepters in the following sense. A string w is written on the tape with blanks fillingout the unused portions. The machine is started in the initial state q_0 with read-write head position on the leftmost symbol of w. If after a sequence of moves the turing m/c enters a final state and halt, then w is considered as accepted otherwise rejected.

→ **Language of Turing Machine:**

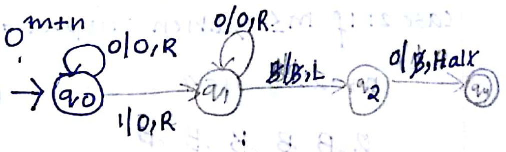
Let $M = \{Q, \Sigma, \Gamma, \delta, q_0, F, B\}$ be a turing machine. Then $L(M)$ is the set of strings $w \in \Sigma^*$ such that $q_0 w \vdash^* \alpha p \beta$ for some state $p \in F$ and tape string α and β then that string is accepted by turing machine. $L(M) = \{w \in \Sigma^* \mid q_0 w \vdash^* \alpha p \beta$ for some $p \in F$ and $\alpha, \beta \in \Gamma^*\}$

~~Danger~~ Turing Machine As Transducers

- It is an abstract computer.

$0^m 1 0^n \Rightarrow 0^{m+n}$

000100



Q Construct a TM to compute proper substraction of two numbers. Proper sub is defined by

$$m - n = \begin{cases} m-n & \text{if } m > n \\ 0 & \text{if } m \leq n \end{cases}$$

input:- $B\underbrace{0000}_{m}1\underbrace{0\ldots0}_{n}B$

19/10/17 **Proper Substraction**

$$m - n = \begin{cases} m-n & \text{if } m > n \\ 0 & \text{if } m \leq n \end{cases}$$

Input representation = $0^m 1 0^n$, eg: $0^5 1 0^3$

Tape Configuration = B 00000 1 000 0 ~~B B~~

output

Case 1: if $m > n$ then output = 0^{m-n}

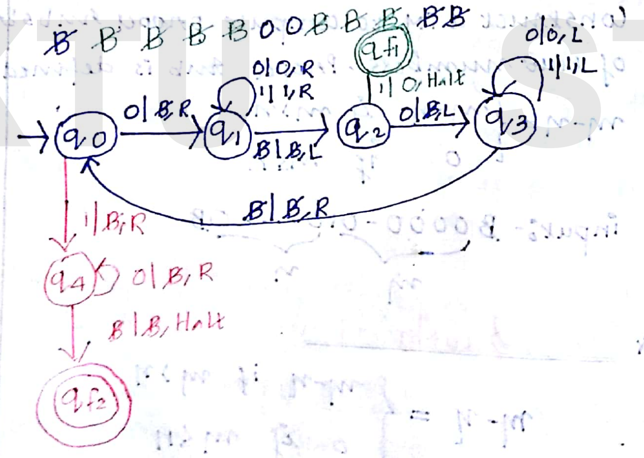
~~B~~ ~~B~~ ~~B~~ ~~B~~ 0 0 ~~B~~ ~~B~~ ~~B~~ ~~B~~

Case 2: If $m \leq n$, then output = B

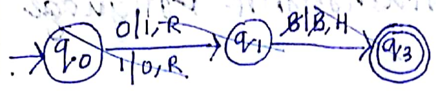
$m = 2$ & $n = 4$

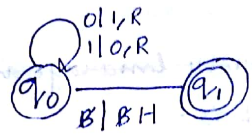
~~B B~~ B B B B

~~B~~ ~~B~~ ~~B~~ ~~B~~ B 0 0 B B B BB



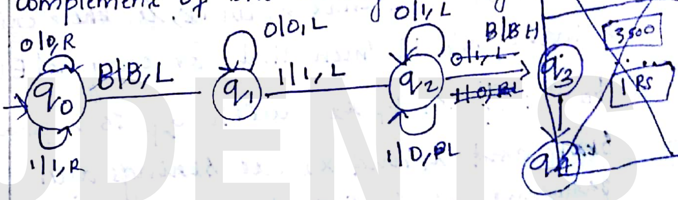
Construct TM that produce 1's complement of a given binary number





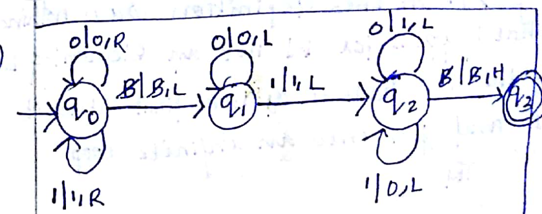
Q Construct TM that produce 2's complement of the binary string.

A



10100
01100 2's

A)



~~101011~~ B
~~010101~~

0101
1011

11001