TUTORIALSDUNIYA.COM

Theory of Computation Notes

Contributor: Abhishek Sharma

[Founder at TutorialsDuniya.com]

Computer Science Notes

Download FREE Computer Science Notes, Programs, Projects, Books for any university student of BCA, MCA, B.Sc, M.Sc, B.Tech CSE, M.Tech at https://www.tutorialsduniya.com

Please Share these Notes with your Friends as well

facebook WhatsApp twitter Telegram

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

Wednesday TOC

Alphabets :- no empty finite set of symbols is called as alphabet and it is denoted by 'Σ'

ex. $\Sigma = \{a, b, c, d \ldots z\}$

$\Sigma = \{$अ, आ ... ज्ञ$\}$

$\Sigma = \{0, 1\}$ Binary

$\Sigma = \{1\}$ Unary

String :- The sequence of symbol from the alphabet is called as string

ex. $\Sigma = \{0, 1\}$

$w = 0$

$w = 01$

$w = 010$

$w = 102$ // it is not string

Length of The string :- if w is any string define over the alphabet Σ then no. of symbol envolve in the string w is called as length of the string and it is denoted by $|w|$

ex. $w = \{1, 0\} \in |w| = 2$

$w = \{0, 1, 0\} \in |w| = 3$

$w = \{0, 1, 0, 0\} \in |w| = 4$

Empty string :- A string of length 0

OR

A string without symbol is called empty string and it is denoted by 'ε'

↳ epsailon

Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

ex. $w = |w| = |\epsilon| = 0$

$w = w.\epsilon = \epsilon.w$

$TOC = TOC.\epsilon = \epsilon.TOC$

Sub String :- Let u, w be two string define over the alphabet 'Σ' and it is said to be sub string if it is obtained from w as it is the same order.

ex. TOC

T, O, C

TO, OC

TC is not substring because it is not order

Notes

* every string 'ϵ' is a substring of every string
* every string is substring of its ~~string~~ self.
* if u is the substring of w then $|u| \leq |w|$

ex. TOC

substrings = TOC, T, O, C, TO, OC, ϵ

Prefix :- Sequence of starting Symbol is called Prefix.

Suffix :- Sequence of ending symbol is called Suffix.

N = TOC

Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

Prefix = ϵ, T, TO, TOC

Suffix = TOC, OC, C, ϵ

Language :- Collection of string from the alphabet 'Σ' is called as Language

ex. $\Sigma = \{0, 1\}$

Language $L = \{00, 01, 10, 11\}$

$L = \{0^n 1^n \mid n \geq 1\} = \{01, 0011, \ldots\}$

Power of N Alphabet :- Let Σ is any alphabet till Σ^a is the set of all the string of length a that is

$$\Sigma^K = \{w \mid |w| = K\}$$

ex. $\Sigma = \{a, b\}$

$\Sigma^1 = \{a, b\}$

$\Sigma^2 = \Sigma . \Sigma = \{a, b\} . \{a, b\}$

$\{aa, ab, ba, bb\}$

$\Sigma^3 = \Sigma . \Sigma . \Sigma = \{a, b\} . \{a, b\} . \{a, b\}$

$\{aa, ab, ba, bb\} . \{a, b\}$

$\{aaa, aba, baa, bba, aab, abb, bab, bbb\}$

$\Sigma^4 = \{w \mid |w| = 4\}$

⋮

Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

$\Sigma^k = \{ w \mid |w| = k \}$

Positive Closer :-

'Σ^+'

ex. $\Sigma = \{0,1\}$

$\Sigma^+ = \{ 0, 1, 00, 11, 10, 010, 100, 101 \ldots$

Clean Closser :-

'Σ^*'

ex. $\Sigma = \{0,1\}$

$\Sigma^* = \{ \epsilon, 0, 1, 00, 11, 10, 010, 100, 101 \ldots$

Notes :-

* if Σ is any alphabet then Σ^* is univarsal Language
* if L is any language from the alphabet Σ

$L \leq \Sigma^*$

Visit https://www.tutorialsduniya.com for Notes, books, programs, question papers with solutions etc.

Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

Language

- Empty
  - $L = \phi$
- Non-Empty
  - finite
  - ~~non~~ infinite

Empty language :- The Language 'L' that can't Contain any ~~[illegible]~~ string

$L = \phi \in |L| = 0$

$|L|$ = Number of strings in L

Non empty language :- The language 'L' that Contains at least one string

$L \neq \phi \Leftrightarrow |L| \neq 0$ $\qquad L = \{\epsilon\}$

ex. $L = \{a\} \in |L| = 1$

$L = \{\epsilon\} \in |L| = 1$

$L = \{\epsilon, a\} \in |L| = 2$

$L = \{a^n \mid n \geq 0\} \in |L| = \infty$ ~~$\{\epsilon, a, aa, aaa, aaaa, \ldots\}$~~

$L = \{\epsilon\}$ is non empty language

Note :- $\{\epsilon\} \neq \phi$

TutorialsDuniya.com

Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

Finite language :- The language which contains finite no. of string where the length of each and every string is finite is called finite language.

i.e. $L = \text{finite} = |L| = \text{finite}$

for ex.

$L = \{10, 00, 11\}$

$|L| = 3$

$L = \{101, 110, 010, 1111, 11110\}$

$|L| = 5$

Points:-

$L = \phi$

$|L| = \phi$

⇒ confusion!

⇒ point to be noted and compare with empty string.

An empty set ϕ is a finite Language

Infinite language :- The language which contains ~~different~~ infinite no. of strings where the length of string is infinite is called infinite language.

for ex.

$L = \{a^n \mid n \geq 0\}$

L is infinite

$|L| = \infty$

Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

i.e.

$L = \Sigma^+$

$L = \Sigma^*$

both are infinite Language

$L = \{a^n b^n \mid n \geq 0\}$

$|L| = \infty$ it is infinite Language

ex. $L = w \in \Sigma^* \mid |w| = \text{even}$ where as $\Sigma = \{0,1\}$

$(0,1)^* = \{\epsilon, 0, 1, 00, 01, 10, 11$

As per given condⁿ in the example even no. of string

$L = \{\epsilon, 00, 10, 11, \}$

$L = \{\epsilon, 00, 10, 11, 10, 0000, 1100 \ldots\}$

$|L| = \infty$

Notes :-

Every finite Language is accepted by finite Automata

So every finite language is regular language

Every Language

$\boxed{RL \subset CFL \subset CSL \subset REL}$

Every infinite Language can be finite or infinite

Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

RL = finite & infinite

Regular Language
- finite
- infinite

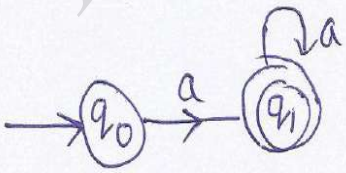
Non Regular Language
↓ always
infinite Language

Regular Language :-

Regular language
- Representation
- 1. (Finite Automata) FA
- 2. RE (Regular expression)
- 3. RG (Regular Grammar)

RL has 3 Representation FA, RE, RG

ex. Let $L = \{a^n \mid n \geq 1\}$

1. 

= FA

$\boxed{\delta(q_0, a) \rightarrow q_1}$

2. $r = a^+$ = RE

3. $S \rightarrow aS \mid a$ = RG

1. $S \rightarrow a$
2. $S \rightarrow aS$
   $\rightarrow aa$
3. $S \rightarrow aS$
   $\rightarrow aaS$
   $\rightarrow aaa$

Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

Finite Automata :- Mathematical Representation of Regular language is called as finite Automata

OR

Mathematical system that process the string is called as finite Automata

OR

Main Defn :-

Finite Automata is a collection of 5 tuple, (M) → Machine

$= (Q, \Sigma, \delta, q_0, F)$

where Q = Set of all states

Σ = Input Alphabets

δ = $Q \times \Sigma \rightarrow Q$ (Transation function)

q_0 = initial state

F = final state

ID (Instantoneous Description) :- Description movement of FA.

The moments of FA depends on two entities

(A) = current state

(B) = current input Alphabet / symbols

ex. $q_i \xrightarrow{x} q_j$

$$\boxed{\delta(q_i, x) \rightarrow q_j}$$

Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

(Transfer/ transition/ Configuration)

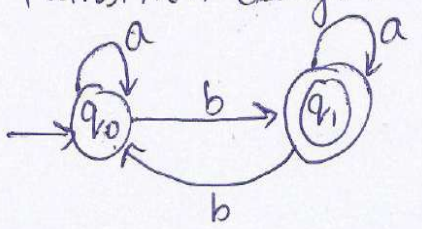
Representation of FA :-

FA can be represented in two ways

(a) Transition Diagram

(b) Transition Table

(a) Transition diagram



IDs :-

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

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

$\delta(q_1, a) \rightarrow q_1$

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

in this exp

$|\Sigma| = 2$ (a, b)

transition state = 2

q_0, q_1

(b) Transition Table

Q \ Σ	a	b
$\rightarrow q_0$	q_0	q_1
q_1	q_1	q_0

TutorialsDuniya.com

Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

Points :-

* In general finite Automata is ∈FA
* Finite Automata only one initial state
* Finite Automata can be Constructed with or without final state i.e. no. of final state in FA can be 0 or 1 or more.
* The transfer path for any state in DFA is Unique
* No. of transition state = $|\Sigma|$
* No. of transition in DFA = $|\Sigma| * |Q|$

Q. Construct the Language that Contains all the string of 0's and 1's where length of string is exactly 2 ?

Sol^n :-

$\Sigma = \{0, 1\}$

$\Sigma^* = \{\varepsilon, 0, 1, \underline{00}, \underline{01}, \underline{10}, \underline{11}, 001, 101, 100, 111 \ldots\}$

$\downarrow |w| = 2$

$\boxed{L = \{00, 01, 10, 11\}}$

Q. Construct the language Contains all the string of 0's and 1's where the length of string is atmost 2 ?

$\Sigma = \{0, 1\}$

$\Sigma^* = \{\varepsilon, 0, 1, 00, 01, 10, 11 \ldots\}$

Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

$|w| \leq 2$

$L = \{\varepsilon, 0, 1, 00, 01, 10, 11\}$

Q. Construct the Language that contains all the strings of 0's and 1's where every string

a) start with 0

b) Start with 10

Sol^n:- $\Sigma = \{0, 1\}$

$\Sigma^* = \{\varepsilon, 0, 1, 00, 10, 11, 01, 101, 000, 010, \ldots$

for condition $\downarrow |w| = 0 \times$

a) $L = \{0, 00, 01, 000, 001, 011, 010, 0001 \ldots\}$

b) $L = \{10, 101, 100, 1001, 1010, 10001 \ldots\}$

Q. Construct the language that contains all the strings of 0's and 1's where every string

a) start and end with 0

b) start and end with same symbol

c) start and end with different symbol

Sol^n $\Sigma = \{0, 1\}$

$\Sigma^* = \{\varepsilon, 0, 1, 00, 01, 10, 11, 001, 010, 100, 111, 101 \ldots\}$

$\downarrow |w| = 0 \times 0$

a) $L = \{00, 010, \cancel{0110, 01110}\ 000, 0110, 0010, 0100, \ldots\}$

Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

$\downarrow |w| = 0 \times 0, 0$

b) $L = \{000, 11, 010, 101, 0110, 1001, 1011, 1101, \ldots$

$1 \times 1, 1$

$000, 111$

$\downarrow |w| = 0 \times 1$

1×0

c) $L = \{01, 10, 011, 100, 110, 001, 1000, 1100, 1110, 1010$

$\ldots$

Q. Construct the language that contains all the strings of 0's and 1's where every string

a) where the length of string is even (mod 2)

b) odd 1 (Mod 2)

c) 1 (mod 3)

Sol^n. a) $\Sigma = \{0, 1\}$

$\Sigma^* = \{\varepsilon, 0, 1, 00, 01, 10, 11, 000, 001, 010, 100, 111, 101, 011 \ldots\}$

$|w| =$ ~~0 (mod 2)~~ even

0 (mod 2)

~~$L = \{\varepsilon, 0, 1, 00, 01, 10, 11, 0000, 0001, 0010, 0100, 1000$~~

~~0110, 0111, 1010~~

$(0, 2, 4, 6, 8 \ldots)$

$L = \{w \in \Sigma^* \mid w = 0, 2, 4, 6, 8 \ldots\}$

$L = \{\varepsilon, 00, 01, 10, 11, \ldots\}$

$\Sigma = \{0, 1\}$ $\Sigma^* = \{\varepsilon, 0, 1, 00, 01, 10, 11, 111 \ldots\}$

b) $|w| =$ odd

1 (mod 2)

$(1, 3, 5, 7, 9 \ldots)$

$L = \{w \in \Sigma^* \mid w = 1, 3, 5, 7, 9 \ldots\}$

$L = \{0, 1, 000, 001, 010, 100, 101, 110, 111, 011 \ldots\}$

Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

TutorialsDuniya.com

Download FREE Computer Science Notes, Programs, Projects, Books PDF for any university student of BCA, MCA, B.Sc, B.Tech CSE, M.Sc, M.Tech at https://www.tutorialsduniya.com

- Algorithms Notes
- Artificial Intelligence
- Android Programming
- C & C++ Programming
- Combinatorial Optimization
- Computer Graphics
- Computer Networks
- Computer System Architecture
- DBMS & SQL Notes
- Data Analysis & Visualization
- Data Mining
- Data Science
- Data Structures
- Deep Learning
- Digital Image Processing
- Discrete Mathematics
- Information Security
- Internet Technologies
- Java Programming
- JavaScript & jQuery
- Machine Learning
- Microprocessor
- Operating System
- Operational Research
- PHP Notes
- Python Programming
- R Programming
- Software Engineering
- System Programming
- Theory of Computation
- Unix Network Programming
- Web Design & Development

Please Share these Notes with your Friends as well

facebook WhatsApp twitter Telegram

Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

(c) $|w| = 1 \pmod 3$

$(1, 4, 7 - - - -)$

$L = \{0, 1, 0000, 0001, 0010, 0100, 1000, 1001, 1010, 1100, 1101, 1110, 1111 - - - -\}$

Q Construct the language that contains all the string of 0's and 1's where

a) The no. of ~~string~~ 0's in string is even

b) The no. of 1's in string is odd

c) No. of 1's in string is multiple of 3

Soln $\Sigma = \{0, 1\}$

$\Sigma^* = \{\epsilon, 0, 1, 00, 01, 10, 11, 100, 010, 101, 001 - - - -\}$

$|w|_0 =$ even

$0 \pmod 2$

$(0, 2, 4, 6, 8 - -)$

$L = \{\epsilon, 00, 11, 1111, 100, 010, 001, - - - -$

b) $\Sigma = \{0, 1\}$

$\Sigma^* = \{\epsilon, 0, 1, 00, 01, 10, 11, 100, 010, 101, 001 - - -\}$

$|w|_1 =$ odd

$1 \pmod 2$

$(1, 3, 5, 7 - - -)$

Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

$L = \{0, 0, 00, 000, 0000, \ldots 1, 111, 11111 \ldots\}$

c) $\Sigma = \{0, 1\}$

$\Sigma^* = \{\varepsilon, 0, 1, 00, 01, 10, 11, 100, 010, 101, 001 \ldots\}$
0000, 0001, 0010, 0100, 1000, 1001, 1010, 1100, 1101, 1110, 1111

$|w|_1 =$ 0 Multiple of 3

0 (Mod 3)

$L = \{\varepsilon, 0, 00, 111, 000, \ldots\}$

Q Construct the language that contain's all the string of 0's and 1's where all the string is palladrome

$\Sigma = \{0, 1\}$

$\Sigma^* = \{\varepsilon, 0, 1, 00, 01, 10, 11, 100, 010, 101 \ldots\}$

$L = \{w \in \Sigma^* \mid w = w^R\}$

$L = \{\varepsilon, 0, 1, 00, 11, 101, 010, 000, 111 \ldots\}$

TutorialsDuniya.com

Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

Regular Expression :- if Σ is any alphabet than the expression which is constructed over the symbol of Σ using operator $*, +, \cdot$ is called Regular expression.

A set $*$ is said to be klan clouser, $+$ is positive klause and $\cdot$ is contakanation

these three operator is said to be regular operator and expression that generate regular language is called regular expression.

ex. let us $\Sigma = \{0, 1\}$

$r = 0, 1$

$r = \cancel{0*1}\ 0^* 1$

$r = 0^+ 1$

$r = 1^* 0^*$

$r = (01)^*$

$r = 0^* + 1^* + (01)^*$

Points :- 1. Regular expression always reference regular language.

for ex. $r = 0.(0+1)$

$L = (00+01)$

2. Arthimatic expression always reference numerical values.

ex. $x = 0.(0+1)$

Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

3. Regular expression

if R is any regular expression L(R) is Regular Language generated by

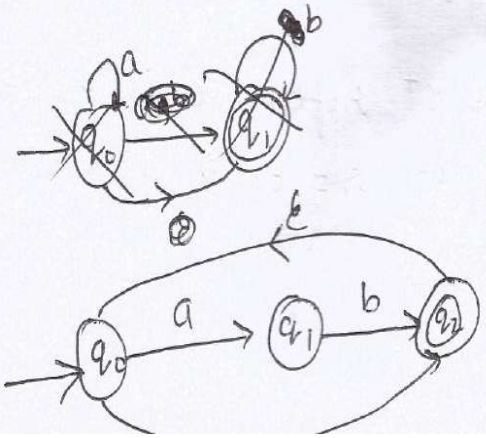
	R (RE)	RL
ex. 1.	$r = \phi$	$L = \phi$
2.	$r = \epsilon$	$L = \{\epsilon\}$
3.	$r = a$	$L = \{a\}$
4.	$r = ab$	$L = \{ab\}$
5	$r = a+b$	$L = \{a, b\}$
6.	$r = w_1 + w_2 + \dots w_n$	$L = \{w_1, w_2, \dots w_n\}$
7.	$r = a.(a+b)$	$L = \{aa, ab\}$
8.	$r = a^*$	$L = \{\epsilon, a, aa, aaa \dots\}$
9.	$r = a^+$	$L = \{a, aa, aaa \dots\}$

Regular Operator precedence

1 *
2 .
3 +

Deductions :-

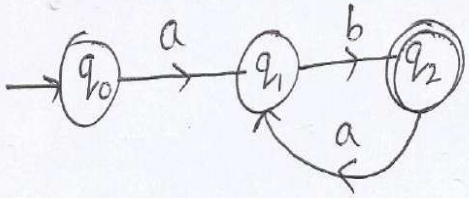
1. every finite language is regular.
2. $R = (ab)^* \Rightarrow L = \{\epsilon, ab, abab, ababab \dots\}$



Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

3. $r = (ab)^+ \Rightarrow L = \{ab, abab, ababab - - - - - -\}$



if r is regular expression that r^* and r^+ is also regular expression.

$r^* = \{\varepsilon, r, rr, rrr - - -\}$

$r^+ = \{r, rr, rrr - - -\}$

$L(r^*) = [L(r)]^*$

$L(r^+) = [L(r)]^+$

ex. $r = ab \Rightarrow L = \{ab\}$

$r^* = (ab)^* \Rightarrow L = \{\varepsilon, ab, abab - - - -\}$

$r^+ = (ab)^+ \Rightarrow L = \{ab, abab, ababab - - -\}$

4. if $r = \varepsilon$ then $r^* = \{\varepsilon\}$

$r^+ = \{\varepsilon\}$

if $r = \phi$ then $r^* = \varepsilon$

$r^+ = \phi$

In general both r^*, r^+ reference is finite language except for $r = \phi$ or $r = \varepsilon \Rightarrow r^*, r^+$ represents of finite language

TutorialsDuniya.com

Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

5. $r^* = r^+ \cup \varepsilon \Rightarrow r^+$ is subset of r^*

i.e. $r^+ + r^* = r^*$

$r^+ \cap r^* = r^+$

6. (i) $r^* . r = r^+ = r . r^*$

(ii) $r^+ . r^* = r^+ = r^* . r^+$

(iii) $r^* . r^* = r^*$

(iv) $r^+ + \varepsilon = r^*$

(v) $(r^*)^* = r^*$

(vi) $(r^+)^* = r^*$

(vii) $\left(\left((r^+)^*\right)^+\right)^{*/+} = r^*$

7. if r_1, r_2 with the regular expression both $(r_1 + r_2)$, $(r_1 . r_2)$ is also Regular expression

i.e. $L(R_1 + R_2) = L(R_1) + L(R_2)$

$L(R_1 . R_2) = L(R_1) . L(R_2)$

8. if R_1, R_2 two regular expression then

$$(r_1 + r_2)^* = (r_1^* + r_2^*)^*$$
$$= (r_1^* + r_2)^*$$
$$= (r_1 + r_2^*)^*$$
$$= (r_1^* . r_2^*)^*$$

TutorialsDuniya.com

Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

9 if r_1, r_2 be two regular expression then

$$(r_1 \cdot r_2)^* \cdot r_1 = r_1 \cdot (r_2 \cdot r_1)^*$$ // shifting Rule

10. $r_1 = (a^* + b^*)^* = (a^* \cdot b^*)^*$

$r_2 = (a^* + b^+ a^*)^*$

Justify $r_1 = r_2$

$r_1 = (a^* + b^*)^*$

$= \cancel{(a+b)^*} \; (a^* \cdot b^*)^*$

$r_2 = (a^* + b^+ a^*)^*$

$(a^* + b^+)^*$

$(a^*)^*$

a^*

<u>soln</u> :- $r_2 = (a^* + b^+ a^*)^*$

$(a^* \epsilon + b^+ a^*)^*$

$(a^* [\epsilon + b^+])^*$

$(a^* b^*)^*$

$r_1 = r_2$

TutorialsDuniya.com

Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

two regular expression are equal if and only if

$$L(r_1) = L(r_2)$$

Points

ex. $r = 0^*$

$L = \{\epsilon, 0, 00, 000 - - - \}$

$L = \{0^n \mid n \geq 0\}$

ex. $r = 0^* \cdot 1$

$L = \{0^n \cdot 1 \mid n \geq 0\}$

ex. $r = (01)^*$

$L = \{\epsilon, 01, 0101, 010101 - - - - \}$

$L = \{01^n \mid n \geq 0\}$

Q. $r = 0^* 1^*$

$L = \{\epsilon, 0, 00, 000 - - - , 1, 11, 111, - - - - \}$

$L = \{0^m 1^n \mid m, n \geq 0\}$

Q. $r = 0^+ 1^+$

$L = \{0^m 1^n \mid m, n \geq 1\}$

TutorialsDuniya.com

Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

Q $r = 0^* 1^+$

$L = \{ 0^m, 1^n \mid m \geq 0, n \geq 1 \}$

Q $r = (0^* 1^+)^*$

$L = \{ (0^m, 1^n)^p \mid m \geq 0, n \geq 1, p \geq 0 \}$

Q Construct RE that contain all string 0's and 1's

a) including ϵ

b) excluding ϵ

$\Sigma = \{0,1\}$

$r = (0+1)^*$

b) $r = (0+1)^+$

TutorialsDuniya.com

Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

Q Construct RE that contain all string 0's and 1's

a) where every string start with 0

b) Start with 10

c) Start with 011

L = {0, 01, 00, 011, 010, 000, - - - -}

a) $r = 0.(0+1)^*$

b) $r = 10.(0+1)^*$

c) $r = 011.(0+1)^*$

Q String end with 10

end with 011

end with 1010

a) $(0+1)^*.10$

b) $(0+1)^*.011$

c) $(0+1)^*.1010$

Q where every string contain substring

a) 101 → $(0+1)^* 101 (0+1)^*$

b) 0101 → $(0+1)^* 0101 (0+1)^*$

TutorialsDuniya.com

Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

Q where the no. of 0's in string is

a) excatly 2

b) at most 2

c) even 0,2,4,6,8

d) odd (1,3,5,7,9)

e) At least 2

Sol^n :- $\Sigma = \{0,1\}$ $|w|_0 = 2$

x0x0x

↓

a) $r = 1^* 0.1^* 0.1^*$

$|w|_0 = 0,1,2$

b) $r = (1^* + 1^* 0.1^* + 1^* . 0.1^* . 0.1^*)$

$|w|_0 = 2,4,6,8$

c) $r = (1^* . 0.1^* . 0.1^*)^* + 1^*$

(d) $|w|_0 = 1,3,5,7$

$(1^* 0.1^*) . (1^* 0.1^* . 0.1^*)^*$

(e) $|w|_0 = 2,3,4,5, \ldots$

$$\boxed{(1+0)^* . 0 . (1+0)^* . 0 . (1+0)^*}$$

Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

Q Construct RE for the following Language

$L = \{0^m 1^n \mid m+n = \text{even}\}$

$L = \{0^m 1^n \mid m+n = \text{odd}\}$

Sol^n $\Sigma = \{0, 1\}$

even

$m = 2x$, $n = 2x$

m+n=even

$0^{2x} 1^{2x}$

$(00)^x (11)^x$

$(00)^* (11)^*$

$m = 2x+1$, $n = 2x+1$

m+n=even

$0^{2x+1} \cdot 1^{2x+1}$

$(00)^x \cdot 0 \; (11)^x \cdot 1$

$(00)^* \cdot 0 \; (11)^* \cdot 1$

$r = (00)^* (11)^* + (00)^* \cdot 0 (11)^* \cdot 1$

(b) $\Sigma = \{0, 1\}$

odd

$m = \text{odd} = 2x+1$, $n = \text{even} = 2x$

$m = \text{even} = 2m$, $n = \text{odd} = 2m+1$

$(00)^x \cdot 0 \; (11)^x + (00)^x (11)^x \cdot 1$

$(00)^* \cdot 0 \; (11)^* + (00)^* (11)^* \cdot 1$

Tutorialsduniya.com

Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

Sol^n $0^{2x} 1^{2x+1}$ $0^{2x+1} 1^{2x}$

$(00)^x \cdot (11)^x \cdot 1$ $(00)^x \cdot 0 \cdot (11)^x$

$(00)^* (11)^* \cdot 1 + (00)^* \cdot 0 \cdot (11)^*$

Q Construct RE for the language

$L = \{ wxw^R \mid w, x \in (0+1)^+ \}$

Sol^n:- $w = 100 \Rightarrow 100x001 \Rightarrow 1x1$

$w = 001 \Rightarrow 001x100 \Rightarrow 0x0$?

$wxw^R = 0x0 + 1x1$

$0(0+1)^+ 0 + 1(0+1)^+ 1$

Visit https://www.tutorialsduniya.com for Notes, books, programs, question papers with solutions etc.

Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

TutorialsDuniya.com

Download FREE Computer Science Notes, Programs, Projects, Books PDF for any university student of BCA, MCA, B.Sc, B.Tech CSE, M.Sc, M.Tech at https://www.tutorialsduniya.com

- Algorithms Notes
- Artificial Intelligence
- Android Programming
- C & C++ Programming
- Combinatorial Optimization
- Computer Graphics
- Computer Networks
- Computer System Architecture
- DBMS & SQL Notes
- Data Analysis & Visualization
- Data Mining
- Data Science
- Data Structures
- Deep Learning
- Digital Image Processing
- Discrete Mathematics
- Information Security
- Internet Technologies
- Java Programming
- JavaScript & jQuery
- Machine Learning
- Microprocessor
- Operating System
- Operational Research
- PHP Notes
- Python Programming
- R Programming
- Software Engineering
- System Programming
- Theory of Computation
- Unix Network Programming
- Web Design & Development

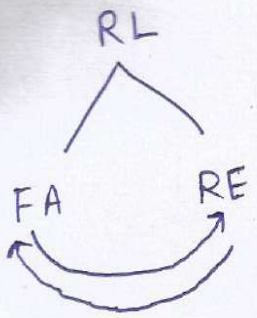
Please Share these Notes with your Friends as well

facebook WhatsApp twitter Telegram

Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

Equivalance b/w FA and RE :-



There are two techniques for the conversion of FA TO RE and RE TO FA

1. ARDEN'S Theoram
2. State ELIMANATION Method

1. ARDEN'S LEMMA Theoram :-

This Mechanism can be used for DFA and NFA and can't be used for ε NFA.

if P, Q, R be the three RE i.e.

$r = Q + RP$ than

(a) $r = Q + RP$ has unique solution if P does not contains ε

that is $r = Q + RP \Rightarrow \boxed{r = QP^*}$

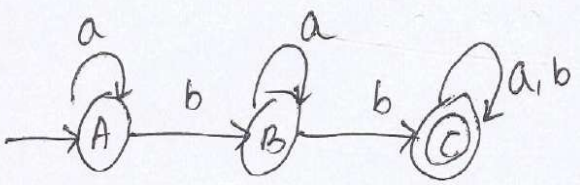
(b) if P contains ε than $r = Q + RP$ has infinite many solution

Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

$$\boxed{R = Q + RP \qquad R = QP^*}$$

Ex.



$$A = Aa + \varepsilon \quad \text{——— (i)}$$

$$\underset{R}{A} = \underset{Q}{\varepsilon} + \underset{R}{A}\underset{P}{a}$$

$$A = \varepsilon a^*$$

$$A = a^*$$

$$B = \cancel{Bb + Ba}\ Ab + Ba$$

$$\underset{R}{B} = \underbrace{a^* b}_{Q} + \underset{R}{B}\underset{P}{a}$$

$$B = a^* b a^*$$

$$C = Bb + Ca + Cb$$

$$C = a^* b a^* b + C(a+b)$$

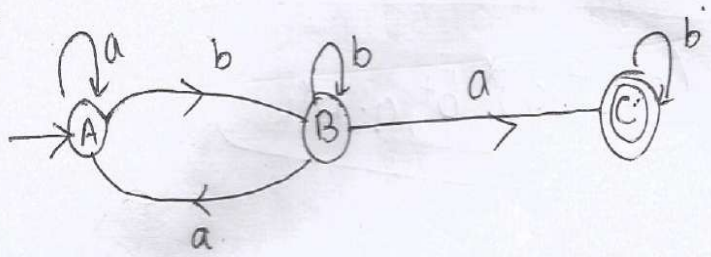
$$\boxed{C = a^* b a^* b (a+b)^*}$$

TutorialsDuniya.com

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

ex.



$A = Aa + Ba + \epsilon$ —— (1)

$B = Ab + Bb$ —— (2)

$\cancel{B = a^* b + Bb}$

$\cancel{B = a^* b . b^*}$

$B = (A+B)$

$\cancel{C = Ba + Ca + Cb}$

$= \cancel{a^* b . b^* . b + C(a+b)}$

$a^* b . \cancel{b^* . a} + Cb$

$\cancel{C = a^* b . b^* . a (a+b)^*}$

$$\boxed{\cancel{C = a^* b . b^* . a b^*}}$$

$C = Ba + Cb$

Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

$R = Q + RP$

$$\boxed{C = (a+b^+.a)^* b^+.a.b^*}$$

Ans

Soln :-

$A = Aa + Ba + \epsilon$ —— ①

$B = Bb + Ab$

$B = Ab + Bb$

$\downarrow \quad \downarrow \quad \downarrow$

$R \quad Q \quad RP$

$\Rightarrow B = QP^*$

$= Ab \cdot b^*$

$= Ab^+$

$A = Aa + (Ab^+)a + \epsilon$

$A = A(a + b^+a) + \epsilon$

$\downarrow \quad \downarrow \quad \uparrow \quad \downarrow$

$R \quad R \quad P \quad Q$

$A = QP^*$

$= \epsilon \cdot (a + b^+.a)^*$

$A = (a + b^+.a)^*$

$b.(\epsilon, b, bb$

$(b, bb, bb$ ~

b^+

$B = (a+b^+.a)^* \cdot B$

$C = Ba + Cb$ —— ③

$C = \underbrace{(a+b^+.a)^*}_{Q} \cdot \underbrace{b^+.a}_{P} + Cb$

$C = (a+b^+.a)^* \cdot b^+ \cdot b^*$

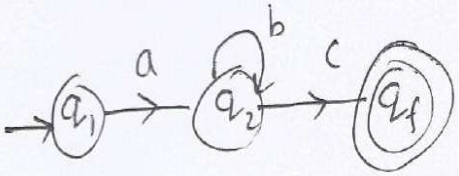
TutorialsDuniya.com

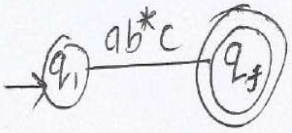
Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

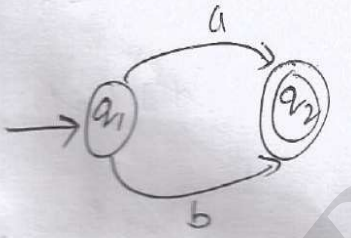
2. STATE ELIMINATION METHOD :-

Ex.

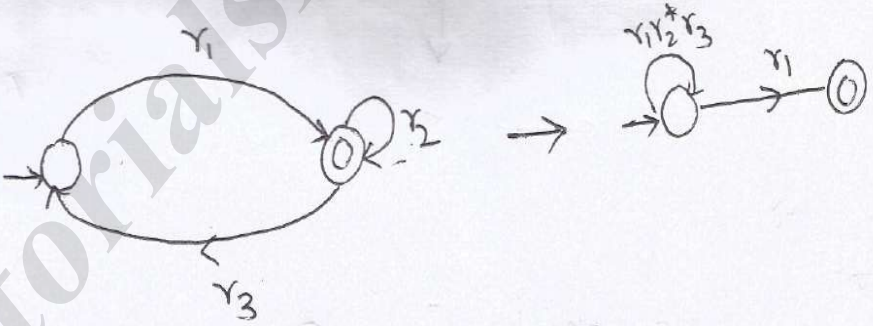


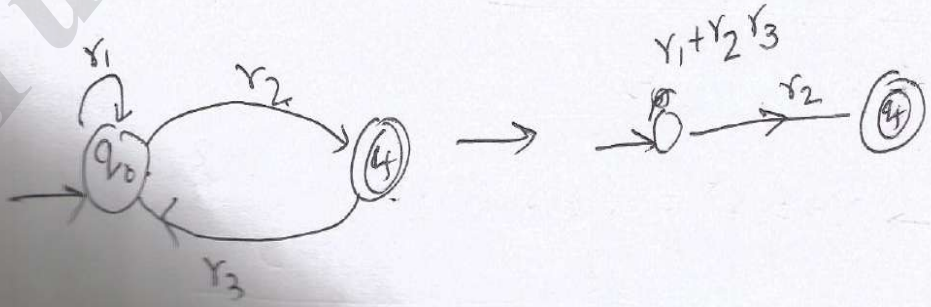


a+b is represented by



ex.





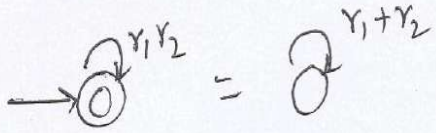
TutorialsDuniya.com

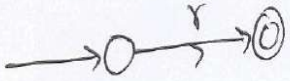
Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

untill following ~~who~~ you reach this is final

(1) $\Rightarrow r = r^*$

(2) 

(3) 

(4)

FA ⟷ RE

RE ⟶ ε NFA

↑ ↓

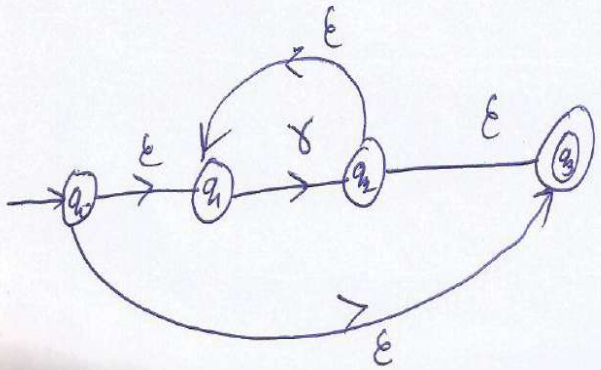
DFA ⟵ NFA

~~Equi~~

Method of Synthesis :-

i) Klean closure

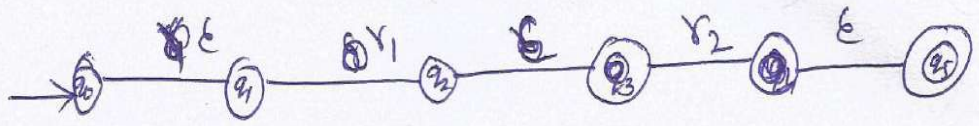
Rules → $re = r^* = \{\varepsilon, r, rr, rrr \ldots\}$



Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

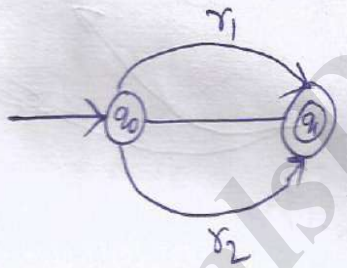
Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

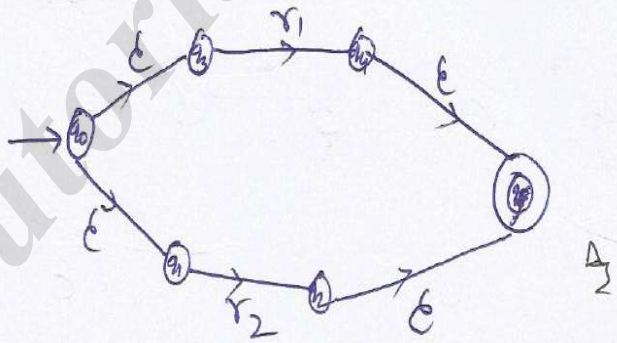
(ii) Concateration $(r_1 \cdot r_2)$



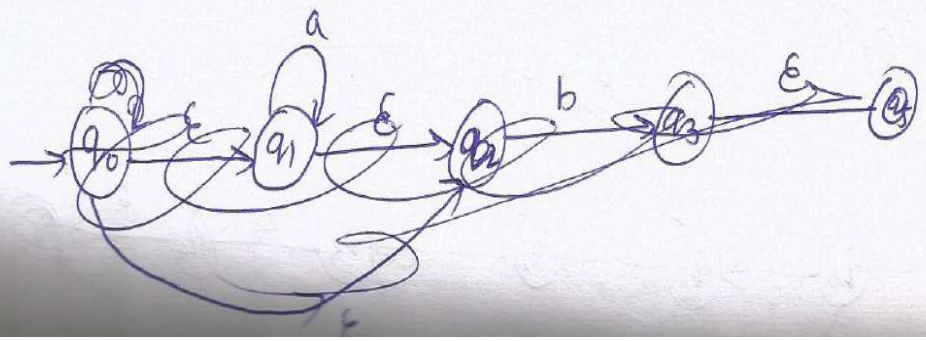
(iii) Union (U)

$r_1 + r_2 = (r_1 \cup r_2)$



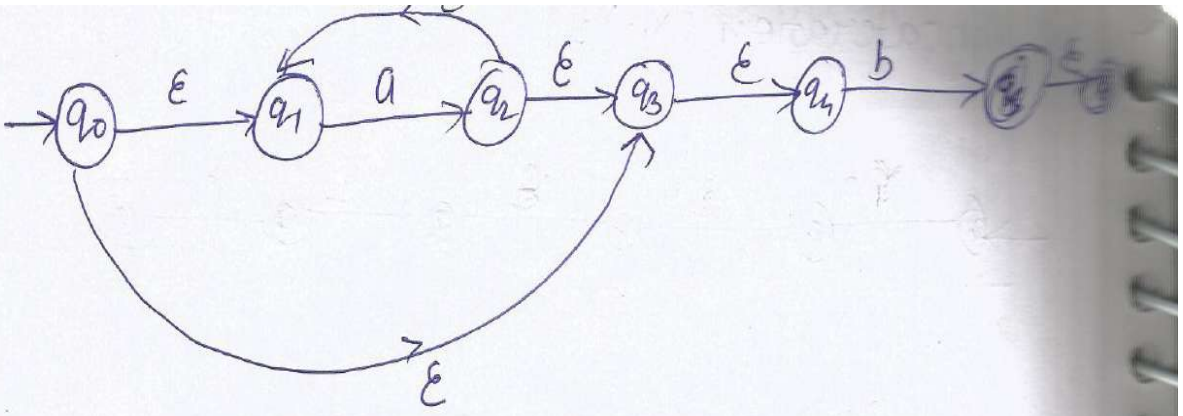


Ex. $a^* b$

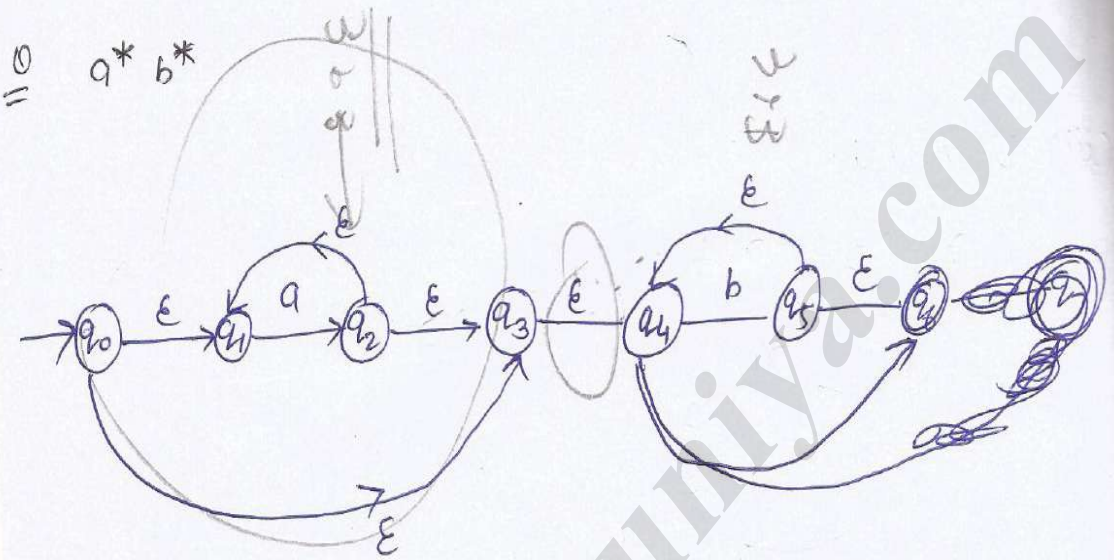


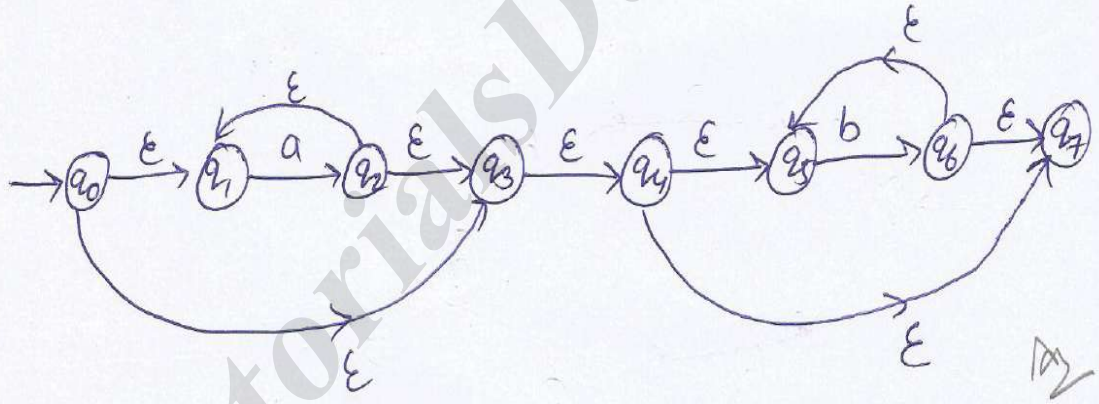
Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

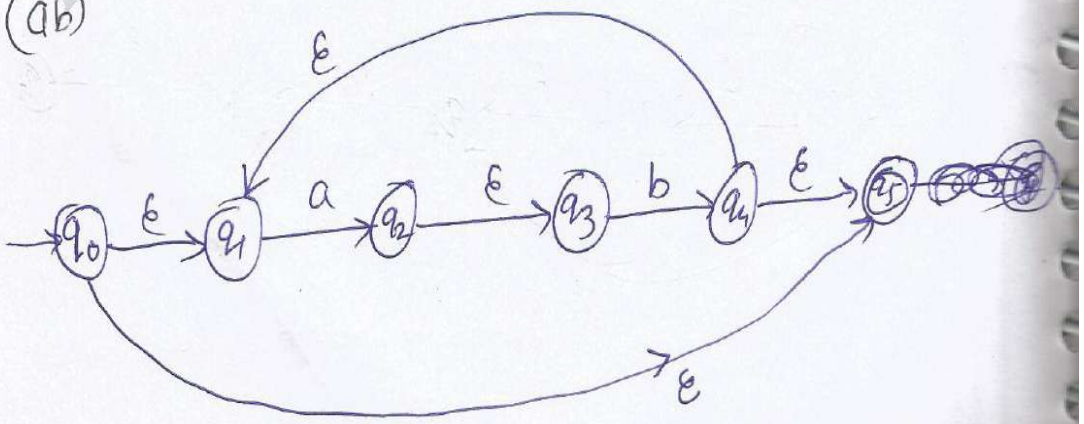


Q. $a^* b^*$





Q. $(ab)^*$

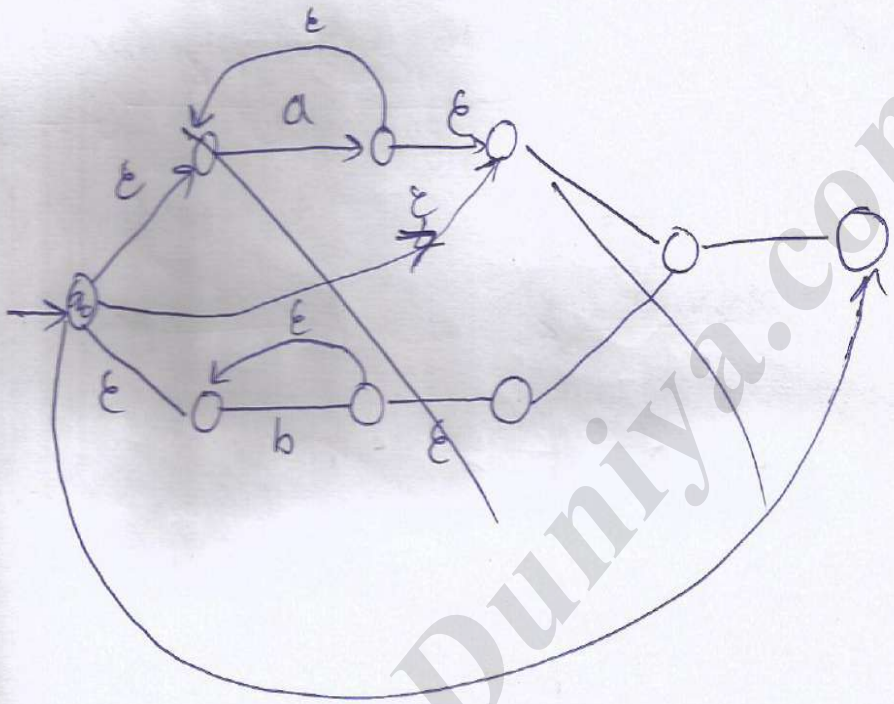


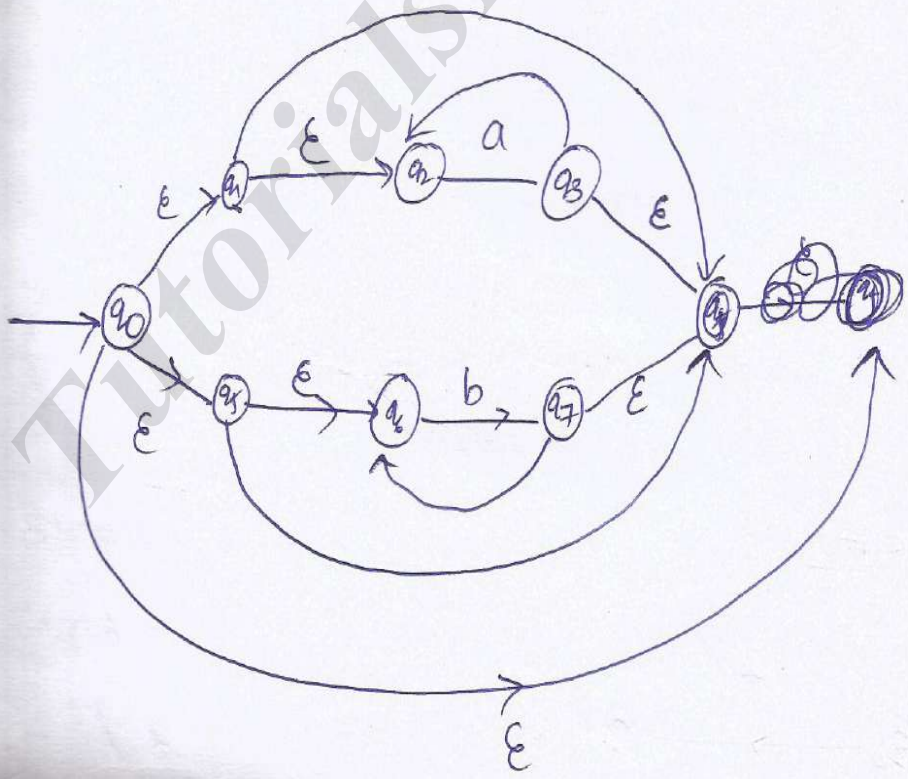
TutorialsDuniya.com

Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

Q $a^* + b^*$





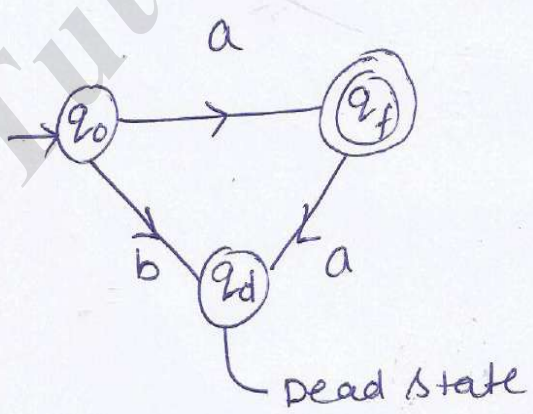
TutorialsDuniya.com

Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

Q $(a^* + b^*)^*$

Dead State :-



where No outgoing edges

TutorialsDuniya.com

Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

Closure properties of Regular Language :-

Regular language satisfy the clouse property under the following operator.

i) Complement
ii) Klean Closure
iii) Positive closure
(iv) Reversal operator
(v) prefix operator
(vi) Union operator($L_1 \cup L_2$)
(vii) Intersection operator($L_1 \cap L_2$)
(viii) Concatenation ($L_1 \cdot L_2$)
(ix) Difference operator($L_1 - L_2$)
(x) Symmetric difference $\left(\frac{L_1}{L_2}\right)$
(xi) Quotient operator
(xii) Subsitution

(iv) Reverse/Reversal operator :- if L is a Regular Language then L^R is also a Regular language

Process to get finite Automata for L^R :-

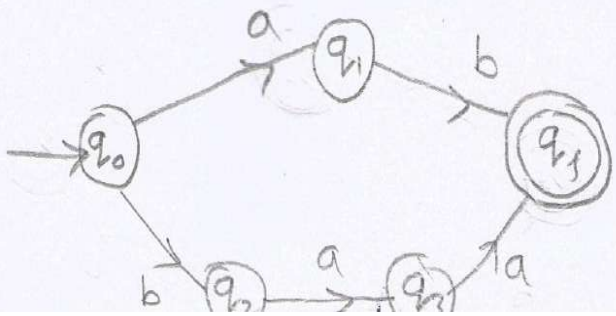
Step 1 :- Interchange initial and final states

Step 2 :- Change the direction of arrow (Adjust)

Step 3 :- If more than one initial state occur then convert the system to have only one initial state.

ex. $L = \{ab, baa\}$

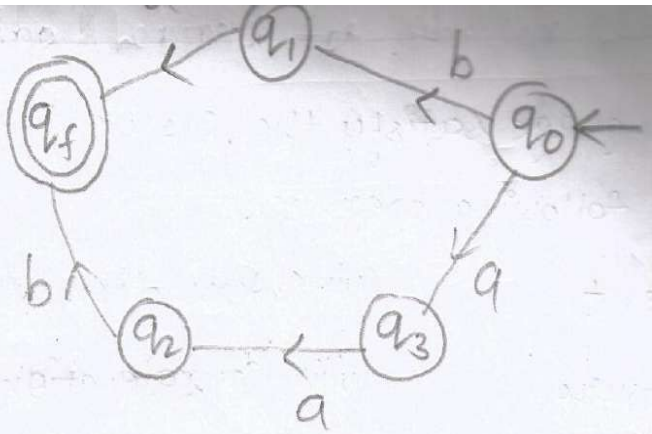
F.A. =



Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

L^R =



Prefix operator :- if L is a Regular Language than prefix of L^R is also a Regular language

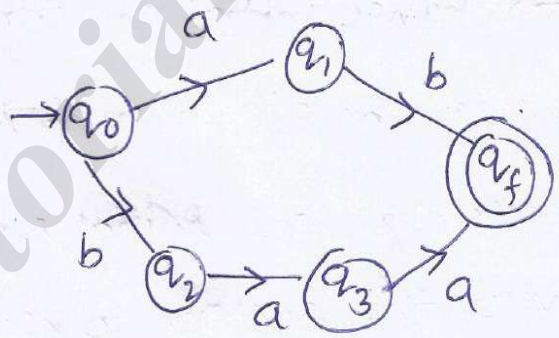
Process to get FA for prefix of L :-

Step 1 :- In case of NFA make all the state as final state

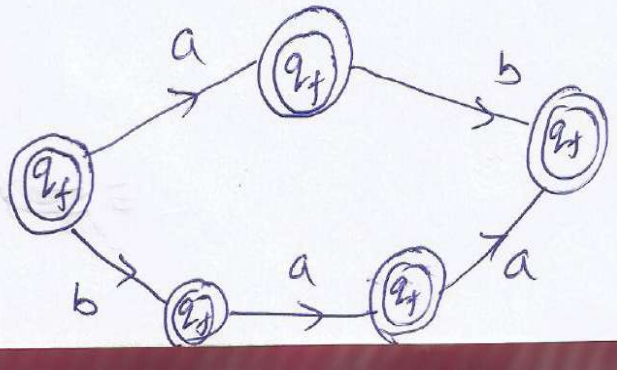
Step 2 :- In case of DFA except dead state and make the all state as final

ex. $L = \{ab, baa\}$

F.A. =



Prefix of L =



Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

xi) Quotient operator :- if L_1 and L_2 be Regular Language than $\frac{L_1}{L_2}$ is also Regular language

where

$$\frac{L_1}{L_2} = \{ x \mid xy \in L_1, y \in L_2 \} \Rightarrow \frac{xy}{y} = x$$

$$\frac{L_1}{L_2} = \{ y \mid xy \in L_1, x \in L_2 \} \Rightarrow \frac{xy}{x} = y$$

ex. $\frac{010}{10}$ Result $= 0$

$\frac{0101}{10} = 01$

$\frac{101}{00} = \phi$

$\frac{100}{100} = \epsilon$

TutorialsDuniya.com

Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

Imp.

Pumping Lemma for Regular Language

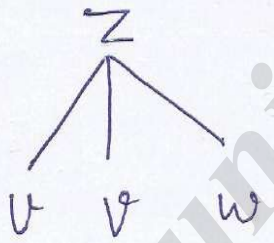
process of pumping Lemma :-

Step 1 :- Assume L is regular ⇒ L satisfy pumping Lemma property

Step 2 :- Choose $z \in L$ such that $|z| \geq n$

Step 3 :- Split z into three part such that

$$|uv| \leq |z| \quad \text{and} \quad |v| \neq 0$$



Step 4 :- if there exist at least one value of L such that $uv^iw \notin L$ that means u, v^i, w does not belongs to L.

if L does not satisfy pumping lemma property which is contradiction.

L is non regular Languag.

ex. $L = \{a^n b^n \mid n \geq 0\}$ Prove this is non regular language

sol^n :- Step 1 :- Assume L is regular language

Let $z \in L$

TutorialsDuniya.com

Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

TutorialsDuniya.com

Download FREE Computer Science Notes, Programs, Projects, Books PDF for any university student of BCA, MCA, B.Sc, B.Tech CSE, M.Sc, M.Tech at https://www.tutorialsduniya.com

- Algorithms Notes
- Artificial Intelligence
- Android Programming
- C & C++ Programming
- Combinatorial Optimization
- Computer Graphics
- Computer Networks
- Computer System Architecture
- DBMS & SQL Notes
- Data Analysis & Visualization
- Data Mining
- Data Science
- Data Structures
- Deep Learning
- Digital Image Processing
- Discrete Mathematics
- Information Security
- Internet Technologies
- Java Programming
- JavaScript & jQuery
- Machine Learning
- Microprocessor
- Operating System
- Operational Research
- PHP Notes
- Python Programming
- R Programming
- Software Engineering
- System Programming
- Theory of Computation
- Unix Network Programming
- Web Design & Development

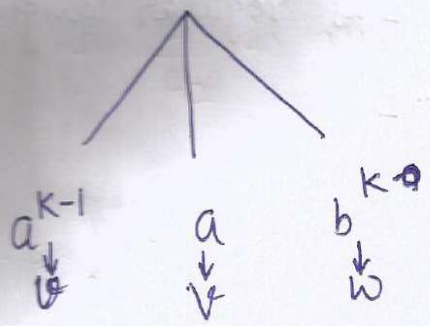
Please Share these Notes with your Friends as well

facebook WhatsApp twitter Telegram

Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

$z = a^k . b^k$



$uv^iw = a^{k-1} a^i b^k$ for $i = 2$

$a^{k+1} . b^k \notin L \quad [\neq a^k b^k]$

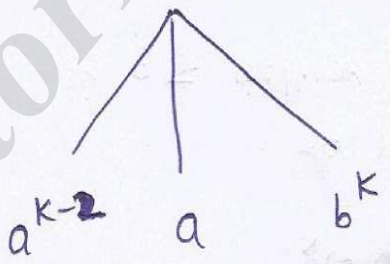
$\Rightarrow L = \{a^n b^n \mid n \geq 0\}$ is not regular langua

Q. $L = \{a^m b^n \mid m < n\}$

Step1:- Assume L is regular language

Let $z \in L$

$z = a^{k-1} b^k$



$uv^iw = a^{k-2} . a^i . b^k$

for $i = 2$

$a^k b^k \notin L \quad [\neq a^{k-1} . b^k]$

$\Rightarrow L = \{a^m b^n \mid m < n\}$ is a not a regular langua

Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

Q. $L = \{a^{n^2} \mid n \geq 0\}$

Soln:- Step 1:- Assume L is regular language

Step 2:- Let $z \in L$

$z = a^{k^2}$

$a^{k^2-1} \quad a \quad \epsilon$

$uv^iw = a^{k^2-1} \cdot a^i \cdot \epsilon$

for $i=2$

$a^{k^2+1} \cdot \epsilon$

$a^{k^2+1} \notin L$

L does not satisfy pumping lemma property which is contradiction.

L is not a regular language.

Q. Number of string length ≤ 3 generated by following RE will be - - - - -

$RE = (1+01)^* \, 0 \, (1+0)^*$

Soln:- $L=0 \Rightarrow 0$

$L=1 \Rightarrow 0$

$L=2 \Rightarrow 00, 01, 10,$

$L=3 \Rightarrow 001, 011, 110, 000, 100, 010, 101$

$L = \{0, 00, 01, 10, 001, 011, 110, 000, 100, 101, 010\}$

Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

Q. if $A = (0^* + 10)^*$

$B = (0 + (10)^*)^*$

a) A C B b) B C A c) A = B d) None of these

$[(R_1 + R_2^*)^* = (R_1^* + R_2)^*]$

Q. i) 0^*

ii) $(00)^*$

iii) $(0+\varepsilon)\cdot 0^*$

iv) $0.(00)^*$

i) $0^* =$ iii) $(0+\varepsilon)\cdot 0^*$

Q. Find the sets represented by following regular expression

i) $(a+b)^*(aa+bb+ab+ba)^*$

The set of all string having odd no. of symbols from $\Sigma^* = \{a, b\}$

ii) $(a+b)\cdot(a+b)^*$

$\{a, b, ba, bb, bab, bba, baa, bbb - - - -\}$

TutorialsDuniya.com

Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

Eqvivalance b/w DFA and NDFA

Conversion NFA To DFA

Q. Construct Deterministic Automata eqvivalant to

$$M = (\{q_0, q_1\}, \{0,1\}, \delta, q_0, \{q_0\})$$

where δ is define by its state table transition

Q \ Σ	0	1
$\rightarrow q_0$	q_0	q_1
q_1	q_1	q_1, q_0

Sol^n 1. for the deterministic ~~finite~~ Automata M_1, the states are the subsets of $\{q_0, q_1\}$

i.e $\phi, [q_0], [q_0, q_1], [q_1]$

2. q_0 is the initial state

3. q_0 and q_1 are the final states ~~the~~ Reason these are the the only state containing q_0 (which is final state) in the given table)

4. δ is defined by the state table given belong

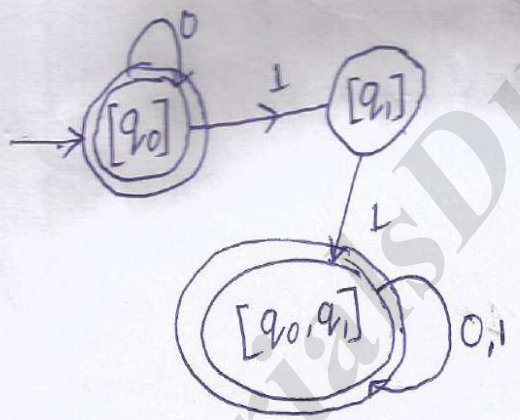
Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

Q \ Σ	0	1
ϕ	ϕ	ϕ
$[q_0]$	$[q_0]$	$[q_1]$
$[q_0, q_1]$	$[q_0, q_1]$	$[q_0, q_1]$
$[q_1]$	q_1	$[q_0, q_1]$

$\delta([q_0], 0) \longrightarrow [q_0]$

Construct DFA To given Table



Q. Construct DFA equivalent to

$M = (\{q_0, q_1, q_2\}, \{a, b\}, \delta, q_0, \{q_2\})$

where δ is given by the table

Q \ Σ	a	b
→ q_0	q_0, q_1	q_2
q_1	q_0	q_1
q_2		q_0, q_1

Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

1. For the determinstic Automata M_1, the states are the subsets of $\{q_0, q_1, q_2\}$

$\phi, [q_0], [q_0, q_1], [q_1]$

2. q_0 is the initial state

$Q \backslash \Sigma$	a	b
ϕ	ϕ	ϕ
$[q_0]$	$[q_0, q_1]$	q_2
$[q_0 q_1]$	$[q_0, q_1]$	$[q_1, q_2]$
$[q_1]$	$[q_0]$	$[q_1]$
$[q_2]$		$[q_0, q_1]$
$[q_1, q_2]$	$[q_0,]$	$[q_0, q_1]$
$[q_0, q_1, q_2]$	$[q_0, q_1]$	$[q_0 q_1, q_2]$
$[q_0, q_2]$	$[q_0 q_1]$	$[q_0, q_1, q_2]$

$2^Q = 2^3 = 8$

Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

Grammar

A grammar is collection of 4 variables or tuples

i.e. (V_n, Σ, P, S)

→ where V_n is a finite non empty set who's element are called variable

→ Σ is a finite non empty set who's elements are called Terminals. (Small letters)

$V_n \cap \Sigma = \phi$

→ S is a special variable (i.e. an element of V_n called the start symbol)

→ P is a finite set who's elements are $\alpha \rightarrow \beta$ where α and β are strings on $V_n \cup \Sigma$

α has at least one symbol from V_n

The elements of P are called Productions or production rule or rewritting rule.

Derivations and the language generated by the Grammar :-

Q. if $G = (\{S\}, \{0,1\}, \{S \rightarrow 0S1, S \rightarrow \Lambda\}, S)$ (NULL)

find $L(G) = ?$

Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

Sol^n:- $S \to \Lambda$ is the production so Λ is in L of G

also for $\forall\, n \geq 1$

$S \to 0S1$

$S \to 00S11$

$S \to 000S111$

$\vdots$

$S \to 0^n 1^n$

$\therefore \quad 0^n 1^n \in L(G)$ s.t. $n \geq 0$

description:- The derivation of w starts with S

if $S \to \Lambda$ is applied first, we get Λ

In this case ~~to~~ $w = \Lambda$.

The first production to be applied $S \to 0S1$

at any stage if we apply $S \to \Lambda$ we

get a terminal string. A terminal

string is obtain is only by applying $S \to \Lambda$

Thus the derivation of w is of the form

$S \to 0^n S 1^n \to 0^n 1^n \; \forall\, n \geq 1$

i.e. $L(G) = 0^n 1^n \in L(G)$ s.t. $n \geq 0$

TutorialsDuniya.com

Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

Q Let $G = (\{S, C\}, \{a, b\}, \{S \to aCa, C \to aCa \mid b\}, \{S\})$

find $L(G) = ?$

Solⁿ :- $S \to aCa$, $C \to aCa \mid b$

$S \to aaCaa$ by application $C \to aCa$

$S \to aaaCaaa$ "

$\vdots$

$S \to a^n C a^n$ by Application $C \to aCa$ for $(n-1)$ time

$S \to a^n b a^n$ $n \geq 1$

$L(G) = a^n b a^n$

Description :-

TutorialsDuniya.com

Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

Q $G = (\{S, a_1\}, \{0,1,2\}, \{S \to 0Sa_12, S \to 012, 2a_1 \to a_12, 1a_1 \to 11\}, \{S\})$

Sol^n:- $S \to 0Sa_12$

$S \to 0012a_12$ by application $S \to 012$

$S \to 001a_12$ by applicati

$S \to 00112$

$S \to 00011222a_12$

$00011a_122$

~~00011122~~ $00011a_1222$

000111222

0

$L(G)$ $0^n 1^n 2^n$ $n \geq 1$ A_2

Q. $G = (\{S, A_1, A_2\}, \{a, b\}, P, S)$

where P consists of

$S \to aA_1A_2a$, $A_1 \to baA_1A_2b$

$A_2 \to A_1ab$, $aA_1 \to baa$

$bA_2b \to abab$

Test w, $w = baabbabaaabbaba$

Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

Soln:-

$S \rightarrow \underline{a A_1} A_2 a$

~~$S \rightarrow aba A_1 A_2 b A_2 a$~~

~~$S \rightarrow aba \underline{A_1} A_1 abb A_1 aba$~~

~~$S \rightarrow \cancel{aba}\ abbaa ba A_1 A_2 b abb A_1 aba$~~

~~$S \rightarrow a\ baa A_1 aba$~~

~~$baaaaaba$~~

~~$abaA_1A_2bA_2a$~~

$S \rightarrow baa \underline{A_2} a$ (by using $aA_1 \rightarrow baa$)

$S \rightarrow baa \underline{A_1} aba$ (by using $A_2 \rightarrow A_1 ab$)

$S \rightarrow baa ba \underline{A_1} A_2 baba$ (by using $A_1 \rightarrow ba A_1 A_2 b$)

$S \rightarrow baa bbaa \underline{A_2} baba$ (by using $aA_1 \rightarrow baa$)

$S \rightarrow baa bbaa \underline{A_1} abbaba$ (by using $A_2 \rightarrow A_1 ab$)

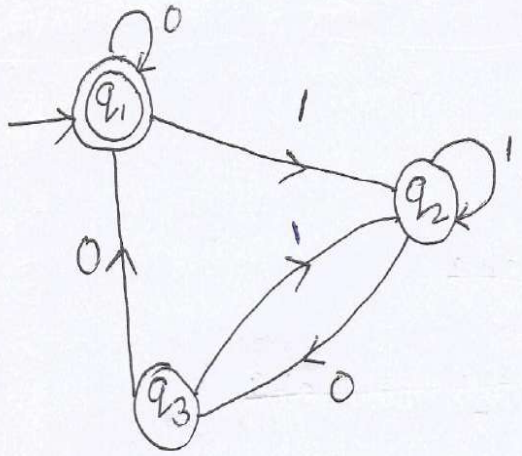
$\boxed{S \rightarrow baabbabaaabbaba}$ (by using $aA_1 \rightarrow baa$)

Accepted

Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

Q.



Construct RE

$Q \backslash \Sigma$	0	1
q_1	q_1	q_2
q_2	q_3	q_2
q_3	q_1	q_2

use ARDENS Theoram :-

$$q_1 = \underset{R}{q_1}\underset{P}{0} + \underset{Q}{\epsilon} \quad \text{—— (1)}$$

$$\cancel{q_1 = q_p 0}$$

$$q_1 = \epsilon 0^*$$

$$q_1 = 0^* \quad \text{—— (2)}$$

$$q_2 = q_1 \cdot 1 + q_2 \cdot 1 + q_3 \cdot 1$$

$$\cancel{q_2 = \underset{Q}{0^*} \cdot 1 + \underset{R}{q_2} \cdot \underset{P}{1}} \qquad q_2 = 0^* \cdot 1 + q_1 \cdot q_2 \cdot q_3 \cdot 1$$

$$q_2 = 0^* \cdot 1 + q_2 \cdot q_2 \cdot 1 \cdot 1$$

$$q_2 = 0^* \cdot 1 + q_2 \cdot 1$$

$$\cancel{q_2 = 0^* \cdot 1 \cdot 1^*}$$

TutorialsDuniya.com

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

$R = QP^*$

$q_3 = q_2 \cdot 0$ —— (2)

$q_2 = q_1 \cdot 1 + q_2 \cdot 1 + q_3 \cdot 1$

$q_2 = q_2 \cdot 1 + q_1 \cdot 1 + q_3 \cdot 1$

$q_2 = q_2 \cdot 1 + 0^* \cdot 1 + q_3 \cdot 1$ —— (3)

$q_3 = q_2 \cdot 0$ —— (4)

Put in q_3 in eq^n (3)

$q_2 = q_2 \cdot 1 + q_3 \cdot 1 + 0^* \cdot 1$

$q_2 = q_2 \cdot 1 + q_2 \cdot 0 + 0^* \cdot 1$

$q_2 = \underset{\substack{\uparrow \\ R}}{} \quad \underset{\substack{\uparrow \\ P}}{(1+01)} \underset{\substack{\uparrow \\ R}}{q_2} + \underset{\substack{\uparrow \\ Q}}{0^* \cdot 1}$

P does not contain ϵ

$R = QP^*$

$q_2 = 0^* \cdot 1 \cdot (1+01)^*$

$q_2 = 0^* \cdot 1 \cdot (1+01)^*$

Putting in eq^n (4)

$q_3 = q_2 \cdot 0$

$0^* 1 \cdot (1+01)^* \cdot 0$

Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

TutorialsDuniya.com

Download FREE Computer Science Notes, Programs, Projects, Books PDF for any university student of BCA, MCA, B.Sc, B.Tech CSE, M.Sc, M.Tech at https://www.tutorialsduniya.com

- Algorithms Notes
- Artificial Intelligence
- Android Programming
- C & C++ Programming
- Combinatorial Optimization
- Computer Graphics
- Computer Networks
- Computer System Architecture
- DBMS & SQL Notes
- Data Analysis & Visualization
- Data Mining
- Data Science
- Data Structures
- Deep Learning
- Digital Image Processing
- Discrete Mathematics
- Information Security
- Internet Technologies
- Java Programming
- JavaScript & jQuery
- Machine Learning
- Microprocessor
- Operating System
- Operational Research
- PHP Notes
- Python Programming
- R Programming
- Software Engineering
- System Programming
- Theory of Computation
- Unix Network Programming
- Web Design & Development

Please Share these Notes with your Friends as well

facebook WhatsApp twitter Telegram

Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

Equivalence to find the finite Automata

Comparison Method :-

Process :-

Let M and M' to be the finite Automata on Σ

We construct a Comparison table consisting of $(n+1)$ column where n is the no. of input symbols.

The first column consist of pairs of vertices of the form ~~q, q'~~ (q, q') where ~~q~~ q belong to M and $q' \in M'$

if q, q' appear in some row of the 1st column ~~with~~ in the corresponding entry A-column ($A \in \Sigma$) is q_A, q'_A where q_A, q'_A are reachable from q, q' respectively on application of A (i.e. by A ~~letter~~ paths)

Here are two cases which specify whether M and M' are equivalent or not

Case 1 :- if we reach a pair q, q' s.that q is the final state of M and q' is non-final state of M' or vise-varse q is non final and q' is the final we terminate the construction and conclude M and M' are not equivalent.

Case 2 :- Here the construction is terminated when No new element appears in the 2nd and the subsiquent column which are not in come first column (i.e. when all the elements ~~of~~ in 2nd are subsiquent column appear in the 1st column)

Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

In this case we conclude that M and M' are equivalent

Q. Consider the following two DFA's M, and M' over $\Sigma = \{c, d\}$ as given below determine whether M and M' are equivalent or not

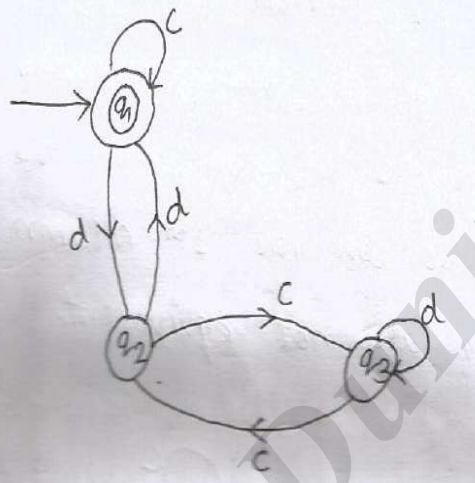


fig M

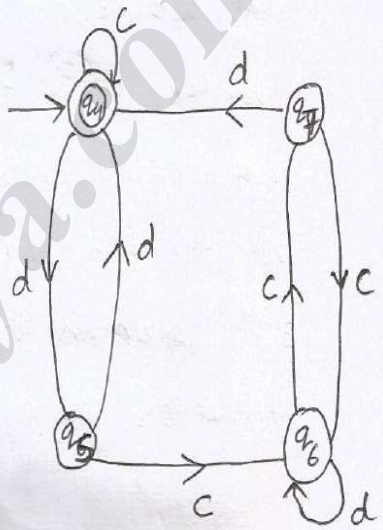


fig M'

Sol^n:-

(q, q')	(q_c, q'_c)	(q_d, q'_d)
(q_1, q_4)	(q_1, q_4)	(q_2, q_5)
(q_2, q_5)	(q_3, q_6)	(q_1, q_4)
(q_3, q_6)	(q_2, q_7)	(q_3, q_6)
(q_2, q_7)	(q_3, q_6)	(q_1, q_4)

Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

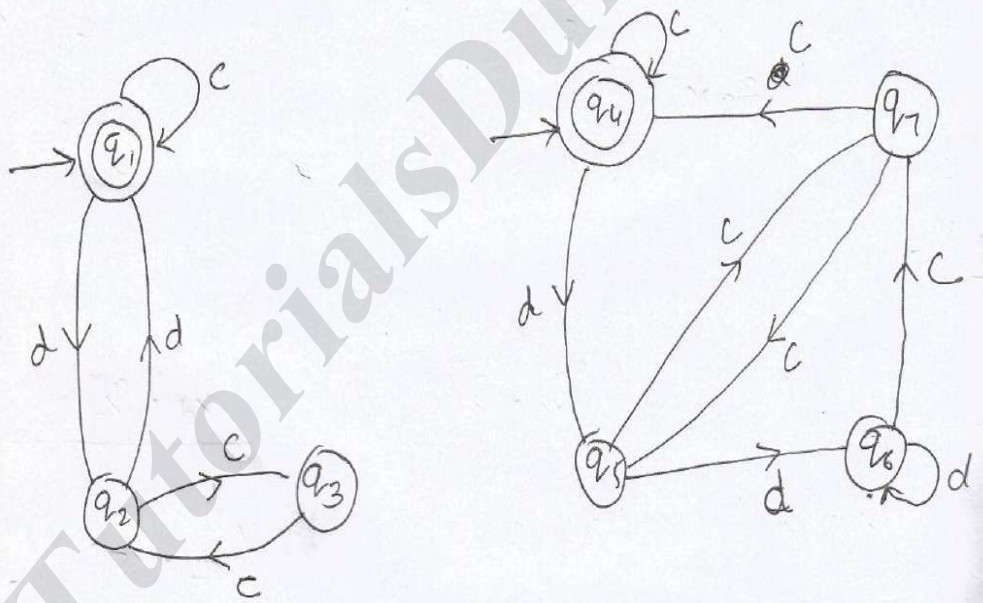
Description :-

initial state in M and M' are q_1 and q_4 respectively Hence the 1st element of the 1st column in the comparison table must be the 1st element in the 2nd column is (q_1, q_4) since both ~~q_1, q_4~~ q_1 and q_4 are ~~c~~ ~~highlen~~ reacable

As we do not get ~~copy~~ ~~here~~ a pair of (q, q') where q is the final state and q' is the non final state (or vise versa) at every row we proceed untill all the elements in the 2nd or 3rd columns are also in the 1st column

therefore M and M' are not equivalent.

Q.



Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

(q, q')	(q_c, q'_c)	(q_d, q'_d)
(q_1, q_4)	(q_1, q_4)	(q_2, q_5)
(q_2, q_5)	(q_3, q_6)	(q_1, q_6)

M and M' are not equivalent

Q Is $L = a^{2n}$ $n \geq 1$ Regular language ?

a^{2n}

By pumping lemma theorem

let L is RL

$z \in L$

$z = a^{2k}$

a^{2k-1} a ϵ

$uv^iw = a^{2k-1} . a^i . \epsilon$

$a^{2k-1} . a^i$

at $i = 1$

$a^{2k} \in L$

at $i > 1$

$a^{2k-1+i} \notin L$

at $i = 1$ L is RL

at $i > 1$ L is non Regular language

Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

CFL

Context free language

language - CFL
Corresponding Machine PDA
Grammar CFGA

CFG :- A Grammar G is said to be CFG if every production is of the form $A \rightarrow \alpha$ where $A \in V_n$ and $\alpha \in (V_n \cup \Sigma)^*$

Derivation Trees :- Derivations in a CFG can be represented using trees such trees representing derivation are called derivations trees. derivation trees are also said to be parse tree

Assumption / properties :- i) every vertex has the level which is ~~terminal~~ variabal or terminal or null.

ii) The root has level S

iii) The level of an internal vertex is variable

(iv) A vertex n is a leaf if its level is $a \in \Sigma$ or NULL.

ex. Let $G = (\{S, a\}, \{a, b\}, \{P\}, \{S\})$

where P Consist of $S \rightarrow aAS \mid a \mid SS$

$A \rightarrow SbA \mid ba$

We have to Construct derivation tree for $w = aabaa$

Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

$S \rightarrow aAS \mid a \mid SS$

$A \rightarrow SbA \mid ba$

~~$S \rightarrow SS$ aAS~~

~~aAS $aSbAS$ abA~~

~~ab $aabbaa$~~

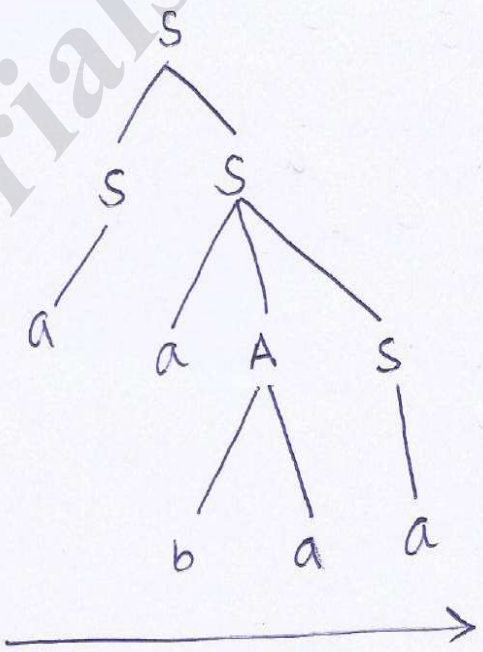
$S \rightarrow SS$

$S \rightarrow aS$ using $S \rightarrow a$

$S \rightarrow aaAS$ using $S \rightarrow aAS$

$S \rightarrow aabaS$ using $A \rightarrow ba$

$S \rightarrow aabaa$ using $S \rightarrow a$



it's a left most Derivation Trees

TutorialsDuniya.com

Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

Two Types of Derivations Tree :-

i) Left Most Derivation Tree

ii) Right Most Derivation Tree

Right Most Derivation Tree's

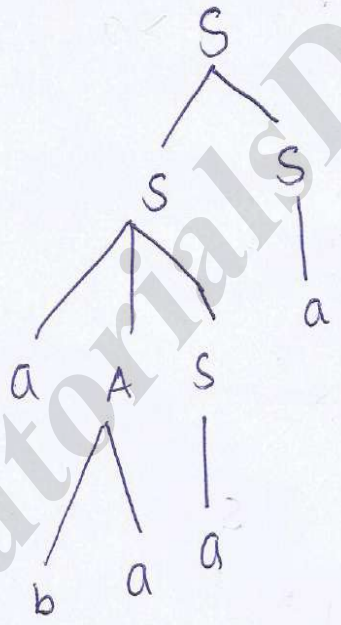
$S \rightarrow SS$

$S \rightarrow Sa$ using $S \rightarrow a$

$S \rightarrow aAS\,a$ using $S \rightarrow aAS$

$S \rightarrow aAaa$ using $S \rightarrow a$

$S \rightarrow abaaa$ using $A \rightarrow ba$



TutorialsDuniya.com

Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

Q.

Let G be the Grammar

$S \to 0B \mid 1A$

$A \to 0 \mid 0S \mid 1AA$

$B \to 1 \mid 1S \mid 0BB$

$w = 00110101$

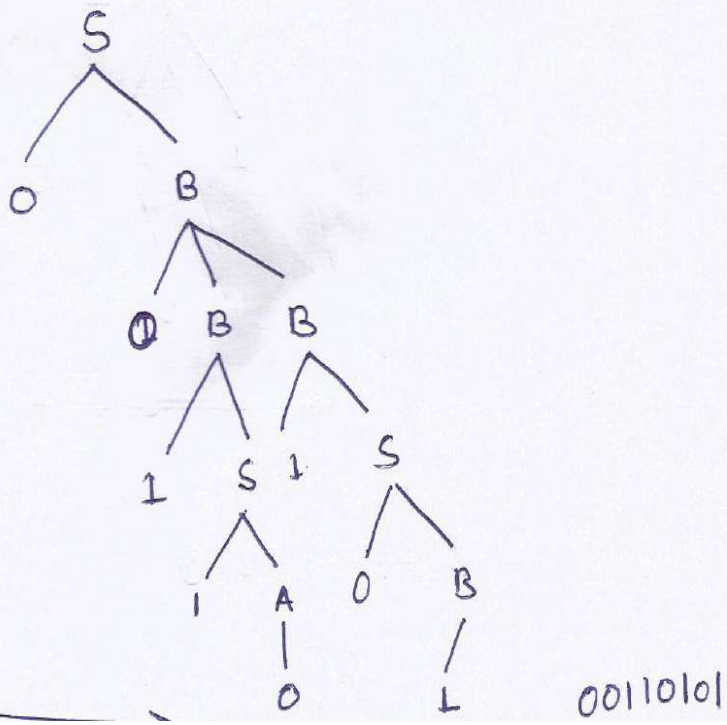
a) LDT

b) RMT

a) $S \to 0B$

$00BB$	using $B \to 0BB$
$001SB$	using $B \to 1S$
$0011AB$	using $S \to 1A$
$00110B$	using $A \to 0$, $B \to 1S$
$001101S$	
$0011010B$	using $S \to 0B$
00110101	using $B \to 1$

Left Derivation Tree



Tutorialsduniya.com

Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

Right Most DT

b) $S \rightarrow 0\underline{B}$

$S \rightarrow 0\underline{0BB}$ using $B \rightarrow 0BB$

$S \rightarrow 00B\underline{1S}$ $B \rightarrow 1S$

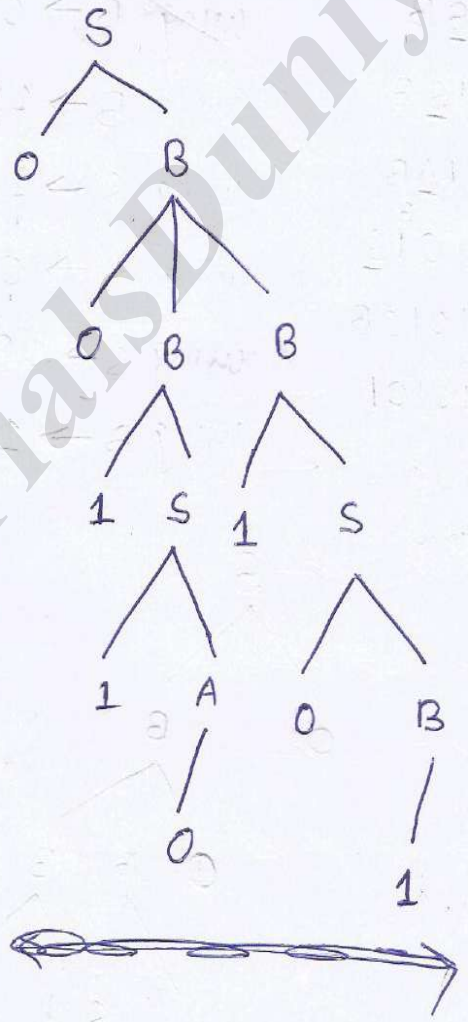
$S \rightarrow 00B1\underline{0B}$ $S \rightarrow 0B$

$S \rightarrow 00B10\underline{1}$ $B \rightarrow 1$

$S \rightarrow 00\underline{1S}101$ $B \rightarrow 1S$

$S \rightarrow 0011\underline{1A}101$ $S \rightarrow 1A$

$S \rightarrow 00110101$ $A \rightarrow 0$



~~10101100~~

00110101

Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

Imp.

Ambuigrity in CFG (Ambigous Grammmar)

A Terminal string $w \in L(G)$ is Ambigous if there exist two or more darivation Trees for w (or there exist two or more left Most darivations of w)

ex. $G = (\{S\}, \{a, b, +, *\}, P, S)$

$S \rightarrow S+S \mid S*S \mid a \mid b$

$w = a + a*b$

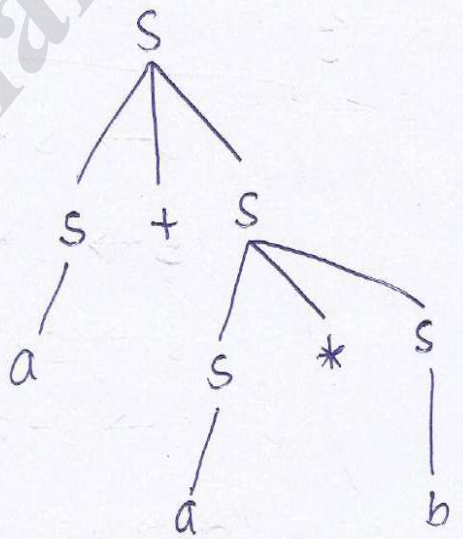
LDT:- $S \rightarrow S+S$

$a + S*S \quad S \rightarrow a$

$a + a*S \quad S \rightarrow a$

$a + a*b \quad S \rightarrow b$

RD



RDT:- $S \rightarrow S+S$

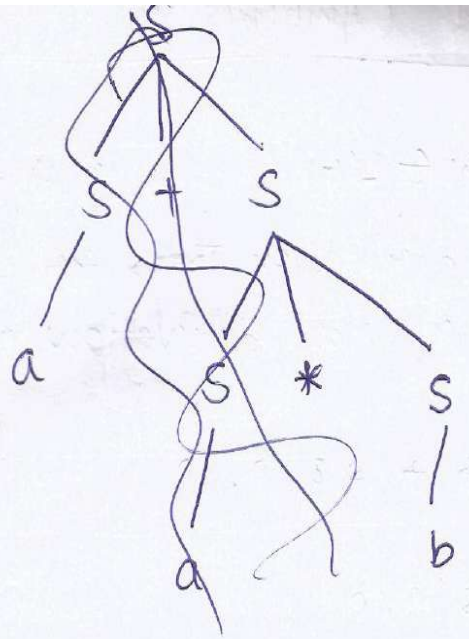
$S \rightarrow S+S*S$

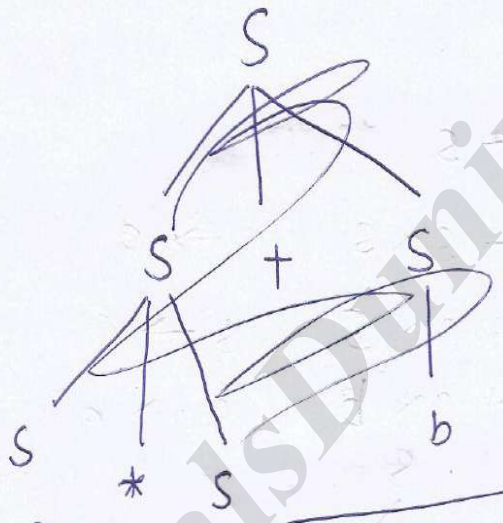
$S \rightarrow S+S*b$ using $S \rightarrow b$

Tutorialsduniya.com

Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com





$S \rightarrow S * S$

$S * b$

$\cancel{(S+S)*}$

$S + S * b$

$a + a * b$

RDT

$S \rightarrow S * S$

$S \rightarrow S * b \quad (S \rightarrow b)$

$S \rightarrow S + S * b \quad (S \rightarrow S+S)$

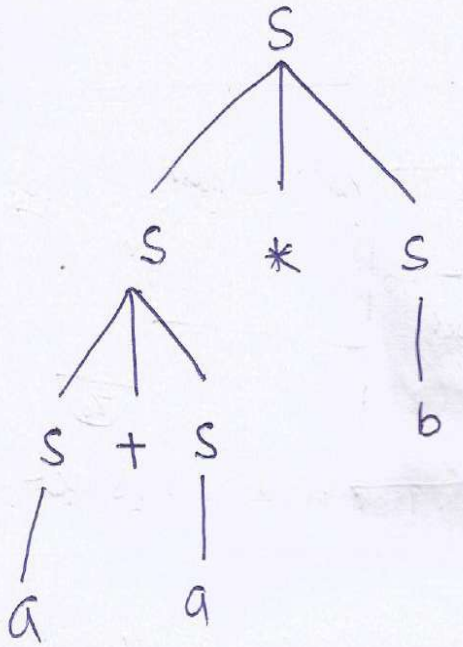
$S \rightarrow S + a * b \quad (S \rightarrow a)$

$S \rightarrow a + a * b \quad (S \rightarrow a)$

TutorialsDuniya.com

Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com



PDA

PDA is a Collection of 7 tuples

$M = Q, \Sigma, \Gamma, Z_0, q_0, q_f, \delta$

TOV

where

Q = Set of all states

Σ = set of all input alphabets

Γ = Set of all Tape Symbols

Z_0 = Top most symbol of stack

$\delta = Q \times \Sigma \times \Gamma \rightarrow Q \times \Gamma^*$

q_0 = initial state

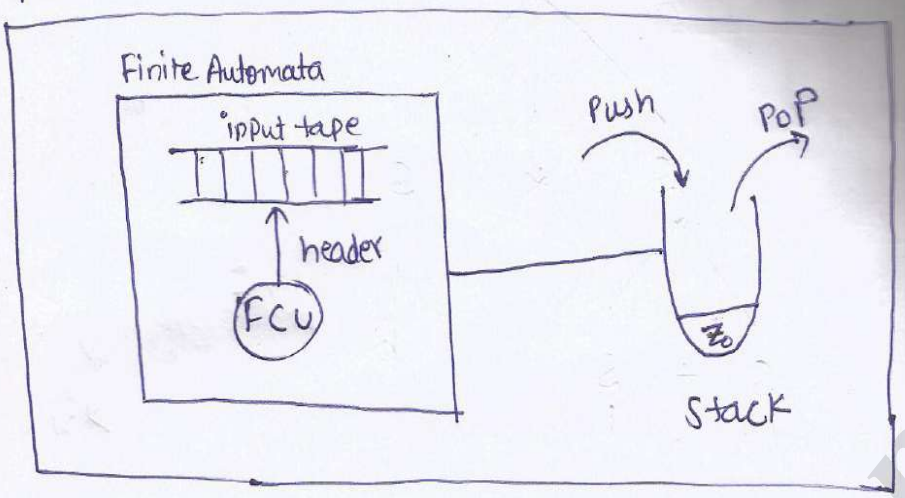
q_f = final state

Tutorialsduniya.com

Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

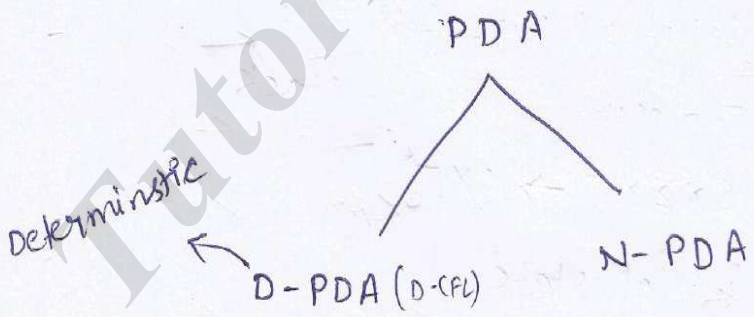
PDA



Structure of Push Down Automata

Points :-

1. PDA is more powerful than FA bcz of memory
2. The purpose of symbol Z_0 is to check that stack is empty or not
3. The language accepted by PDA is called CFL
4. PDA accept every Regular language and also accept sum of non-Regular language



$E(DPDA) \neq E(NPDA)$

$\therefore E(DPDA) < E(NPDA)$

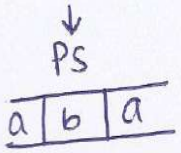
Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

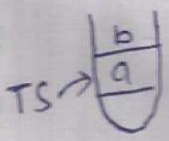
Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

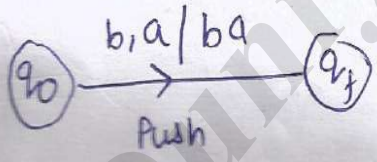
ID'S for PDA :-

RE Decribe the movement of PDA. it Depends on 3 entities

1. Current state
2. Current processing symbol (P.S.)
3. Current top most symbol (T.S.)







b,a/a → skip

b,a/ε → Pop

ID'S

$$\delta(q_0, b, a) = (q_f, ba)$$

q_0 → current state; b → P.S.; a → T.S.; q_f → Next state; ba → New T.S.

Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

TutorialsDuniya.com

Download FREE Computer Science Notes, Programs, Projects, Books PDF for any university student of BCA, MCA, B.Sc, B.Tech CSE, M.Sc, M.Tech at https://www.tutorialsduniya.com

- Algorithms Notes
- Artificial Intelligence
- Android Programming
- C & C++ Programming
- Combinatorial Optimization
- Computer Graphics
- Computer Networks
- Computer System Architecture
- DBMS & SQL Notes
- Data Analysis & Visualization
- Data Mining
- Data Science
- Data Structures
- Deep Learning
- Digital Image Processing
- Discrete Mathematics
- Information Security
- Internet Technologies
- Java Programming
- JavaScript & jQuery
- Machine Learning
- Microprocessor
- Operating System
- Operational Research
- PHP Notes
- Python Programming
- R Programming
- Software Engineering
- System Programming
- Theory of Computation
- Unix Network Programming
- Web Design & Development

Please Share these Notes with your Friends as well

facebook WhatsApp twitter Telegram

Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

Block Diagram of PDA

Component of PDA :-

1. input tape
2. tape header
3. FCU (Finite Control Unit)
4. Stack

Diagram :- PDA वाला

By Default PDA is NPDA

Consider Gr use productions are

Q. $S \rightarrow aAS \mid a$

$A \rightarrow SbA \mid SS \mid ba$

So that $S \rightarrow aabbaa$

construct Possible derivations trees which yield aabbaa

Sol^n :-

$S \rightarrow aAS$

$S \rightarrow aSbAS$ using $A \rightarrow SbA$

$S \rightarrow aabAS$ using $S \rightarrow a$

$S \rightarrow aabbaS$ using $A \rightarrow ba$

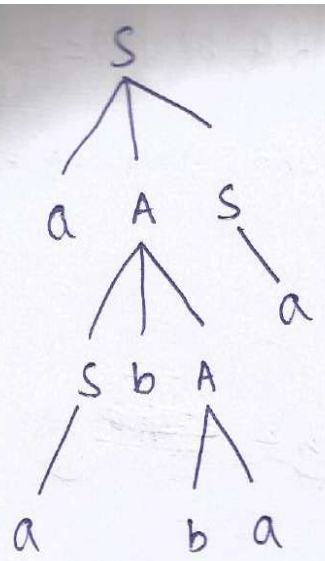
$S \rightarrow aabbaa$ using $S \rightarrow a$

LMD Tree :-

Tutorialsduniya.com

Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com



RMD Tree :-

$S \rightarrow aAS$

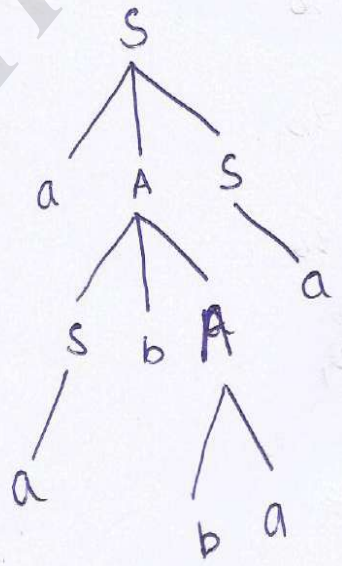
aAa using $S \rightarrow a$

~~aSbAba~~ $A \rightarrow Sba$

$aSbAa$

$asbaa$ using $A \rightarrow ba$

~~as~~ $aabbaa$ using $S \rightarrow a$



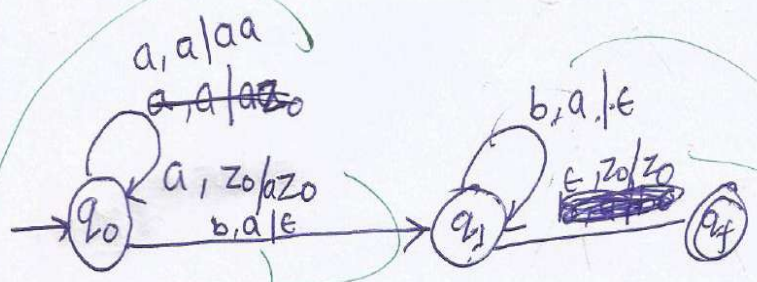
Not Ambigious

Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

Q. Construct PDA for $L = \{a^n b^n \mid n \geq 1\}$ check $w = aabb$ for its accepting ?

Soln:- $L = \{ab, aabb, aaabbb \text{ -----}\}$



$\delta(q_0, a, Z_0) = (q_0, aZ_0)$

$\delta(q_0, a, a) = (q_0, aa)$

$\delta(q_0, b, a) = (q_1, \epsilon)$

$\delta(q_1, b, a) = (q_1, \epsilon)$

$\delta(q_1, \epsilon, Z_0) = (q_f, Z_0)$

For check $w = aabb$:-

$aabb, Z_0, q_0$

abb, aZ_0, q_0

bb, aaZ_0, q_0

b, aZ_0, q_1

ϵ, Z_0, q_f

Accepted

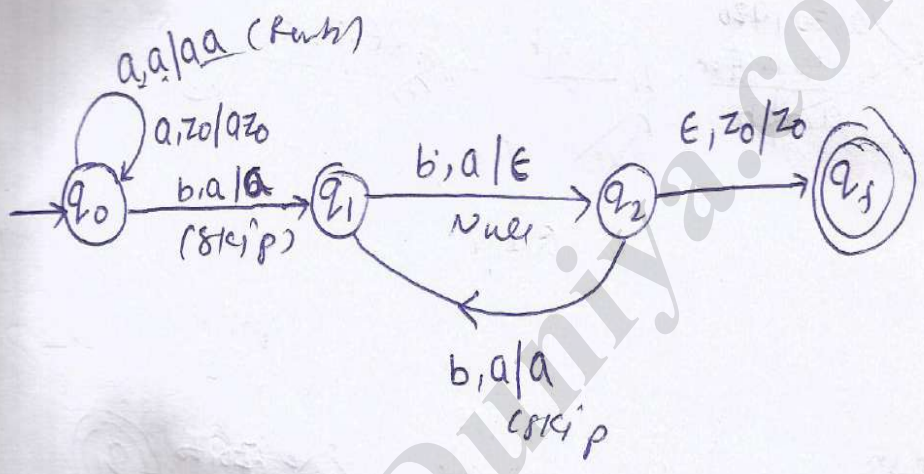
Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

Q. $L = \{a^n b^{2n} \mid n \geq 1\}$

Construct PDA ?

$L = \{abb, aabbbb, aaabbbbbb \ldots\}$



$\delta(q_0, a, z_0) = (q_0, az_0)$

$\delta(q_0, a, a) = (q_0, aa)$

$\delta(q_0, b, a) = (q_1, a)$

$\delta(q_1, b, a) = (q_2, \epsilon)$

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

$\delta(q_1, b, a)$ (q_2, ϵ)

$\delta(q_2, \epsilon, z_0) = (q_f, z_0)$

Ans

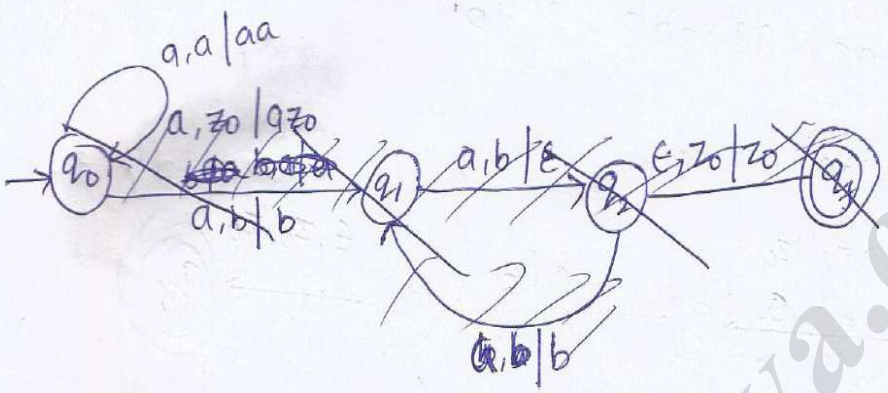
Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

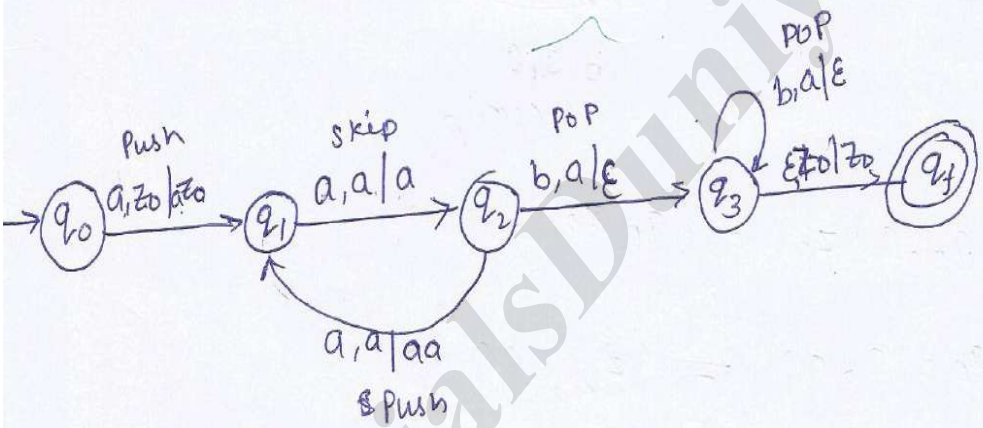
Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

Construct PDA ?

$L = \{a^{2n} b^{n} \mid n \geq 1\}$

$L = \{aab,\ aaaabb,\ aaaaaabbb,\ ----\}$





ID's :-

$\delta(q_0, a, Z_0) = (q_p, aZ_0)$

$\delta(q_1, a, a) = (q_2, a)$

$\delta(q_2, a, a) = (q_1, aa)$

$\delta(q_1, a, a) = (q_2, a)$

$\delta(q_2, b, a) = (q_3, \varepsilon)$

$\delta(q_3, \varepsilon, Z_0) = (q_f, Z_0)$

Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

for check aaaaaabbb :-

aaaaaabbb Z_0 , q_0

aaaaabbb, aZ_0 , q_1 (Push)

aaaabbb aaZ_0 , q_2 (Skip)

aaabbb, aaaZ_0 , q_1 (Push)

aabbb , aaZ_0 , q_2 (Skip)

abbb , aaaZ_0, q_1 (Push)

bbb , aaaZ_0 , q_2 (Skip)

bb, aaZ_0, q_3 (Pop)

b, aZ_0 , q_3 (Pop)

ϵ , Z_0 , q_f (Pop)

Accepted

TutorialsDuniya.com

Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

2. $L = \{a^m b^n \mid m < n\}$,

$m \geq 0$

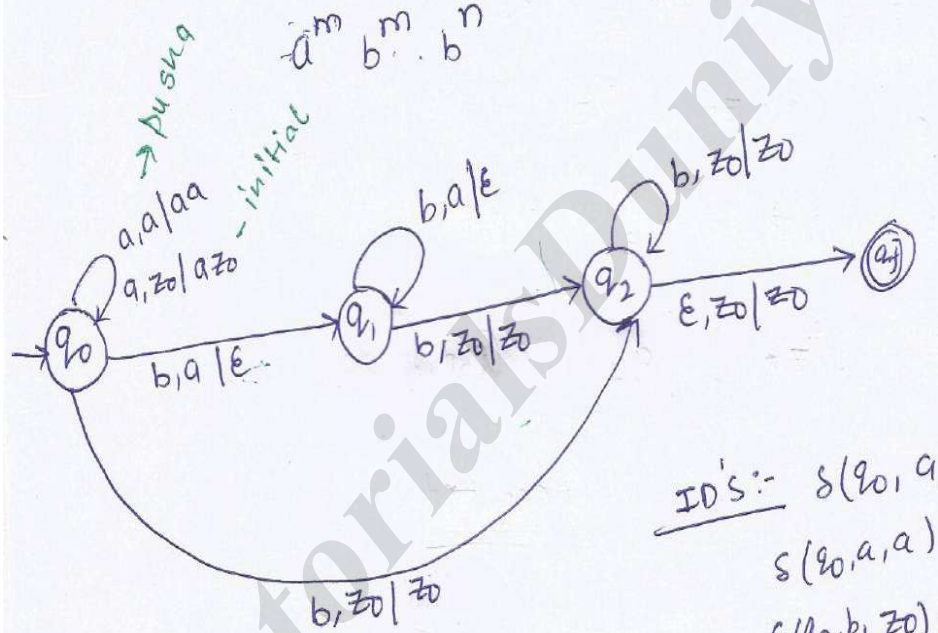
$n \geq 1\}$

Construct PDA

~~L = {b, abb, aabbb, aaabbbb ---}~~

$L = \{$ b, bb, bbb - - - -
abb, abbb, abbbb - - - -
aabbb, aaabbbb, aaaabbbbb - - - - $\}$

$a^m b^m . b^n$



ID's :- $\delta(q_0, a, Z_0) = (q_0, aZ_0)$

$\delta(q_0, a, a) = (q_0, aa)$

$\delta(q_0, b, Z_0) = (q_2, Z_0)$

$\delta(q_0, b, a) = (q_1, \epsilon)$

$\delta(q_1, b, a) = (q_1, \epsilon)$

$\delta(q_1, b, Z_0) = (q_2, Z_0)$

$\delta(q_2, \epsilon, Z_0) = (q_f, Z_0)$

Check Acceptness :-

aaabbb , Z0 , q0

aabbb , aZ0 , q0 (Push)

abbb , aaZ0 , q0 (Push)

bbb , aaaZ0 , q0 (Push)

bb , aaZ0 , q1 (Pop)

Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

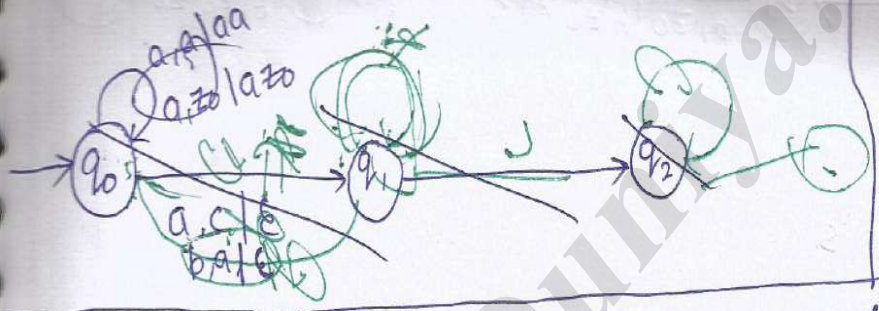
b , aZ_0 , q_1 (POP)

~~Z0~~ ε , Z_0 , q_1 (POP)

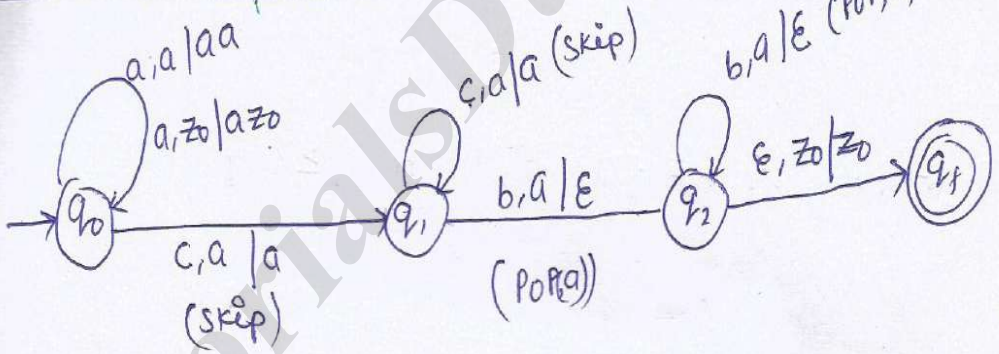
Not accepted

Q $L = \{a^n c^m b^n \mid n \geq 1, m \geq 1\}$ construct PDA?

L = { acb, aacbb, aaacbbb - - - - - , accb, aaccbb, aaaccbbb - - - - - , accb, aaccbb, aaaccbbb - - - - - }

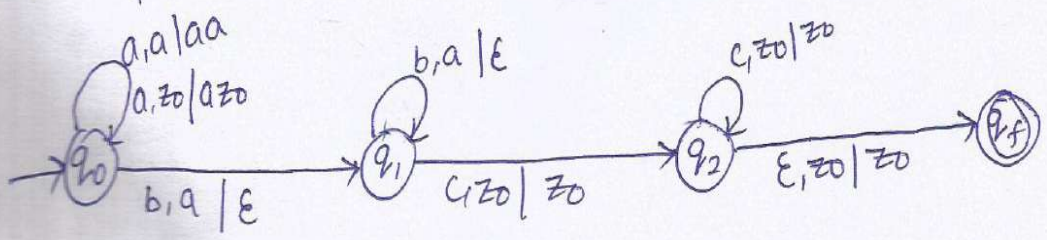


7 ID's



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

L = { abccc, aabbccc, aaabbbccc - - - - - - - - - , abcc, aabbcc, aaabbbccc - - - - - - - - - , abc, aabbcc, aaabbbc - - - - - - - - - }



7 ID's

Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

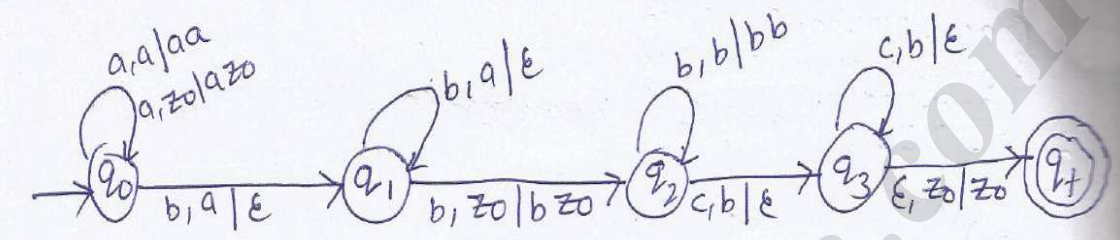
Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

Q $L = \{a^m b^n c^p \mid n = m+p\}$

$m, n, p \geq 1$

$a^m \; b^{m+p} \; c^p$

$a^m \; b^m . b^p \; c^p$

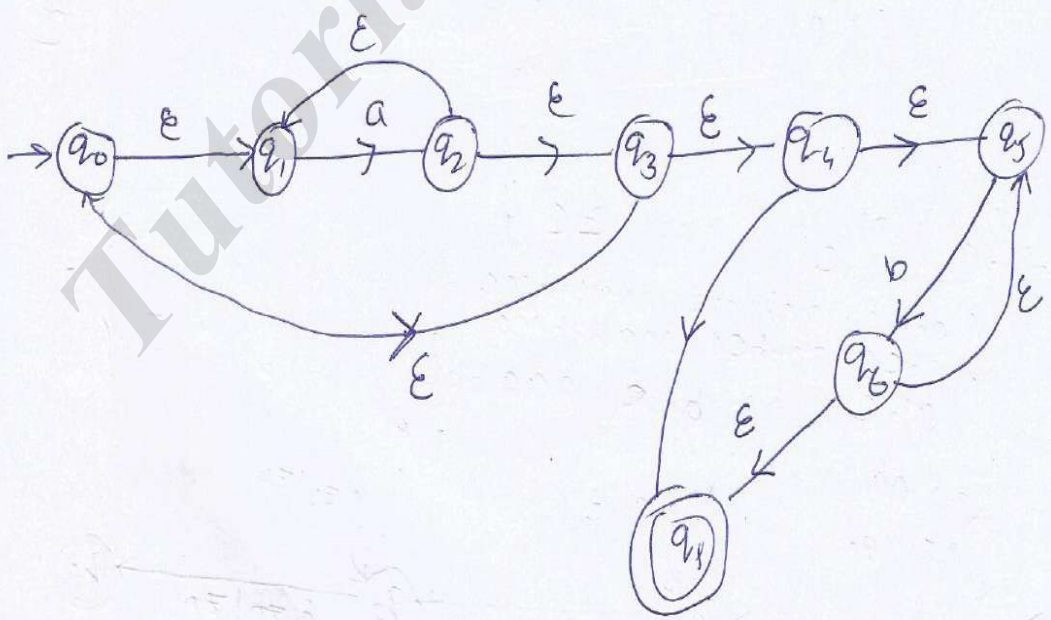


ID's :-

9 ID's

Imp.

Q Find ε-NFA for $L = \{a^m b^n \mid m, n \geq 0\}$



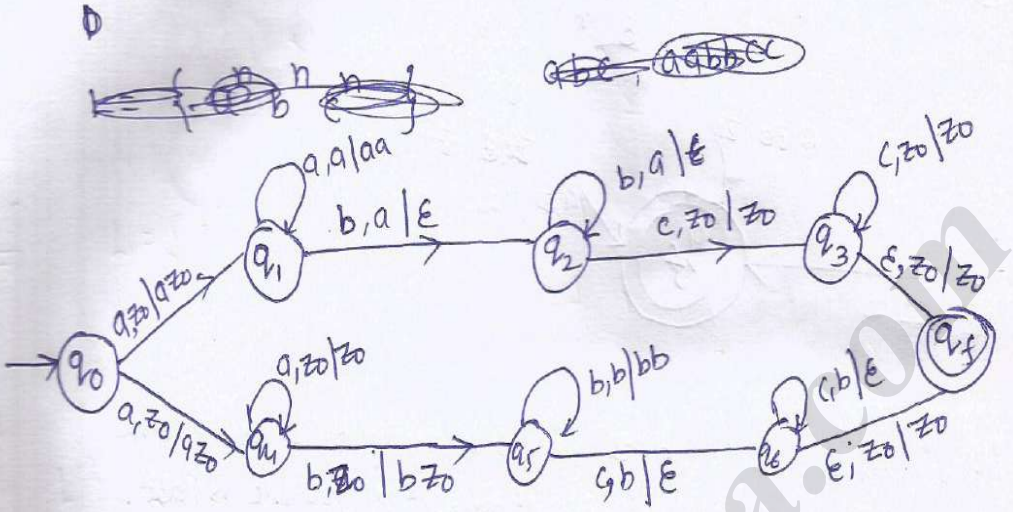
Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

Q $L = \{a^m b^n c^p \mid m = n \text{ or } n = p\}$

$mnp \geq 1$

D



$L = \{a^m b^n c^p \mid m = n \text{ or } mnp \geq 1\}$

$a^m b^m c^p$

$\{abc, aabbc, aaabbbc \ldots\}$

$L = \{a^m b^n c^p \mid n = p \text{ or } mnp \geq 1\}$

$a^m b^n c^p$

$L = \{abc, abbcc, abbbccc \ldots\}$

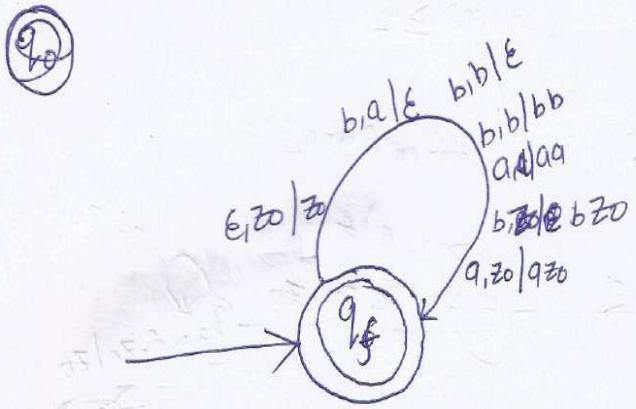
it is NPDA bcz it has two out for the input from the same state. 14 ID's

Q $L = \{w \in (a+b)^* \mid |a| = |b|\}$

Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

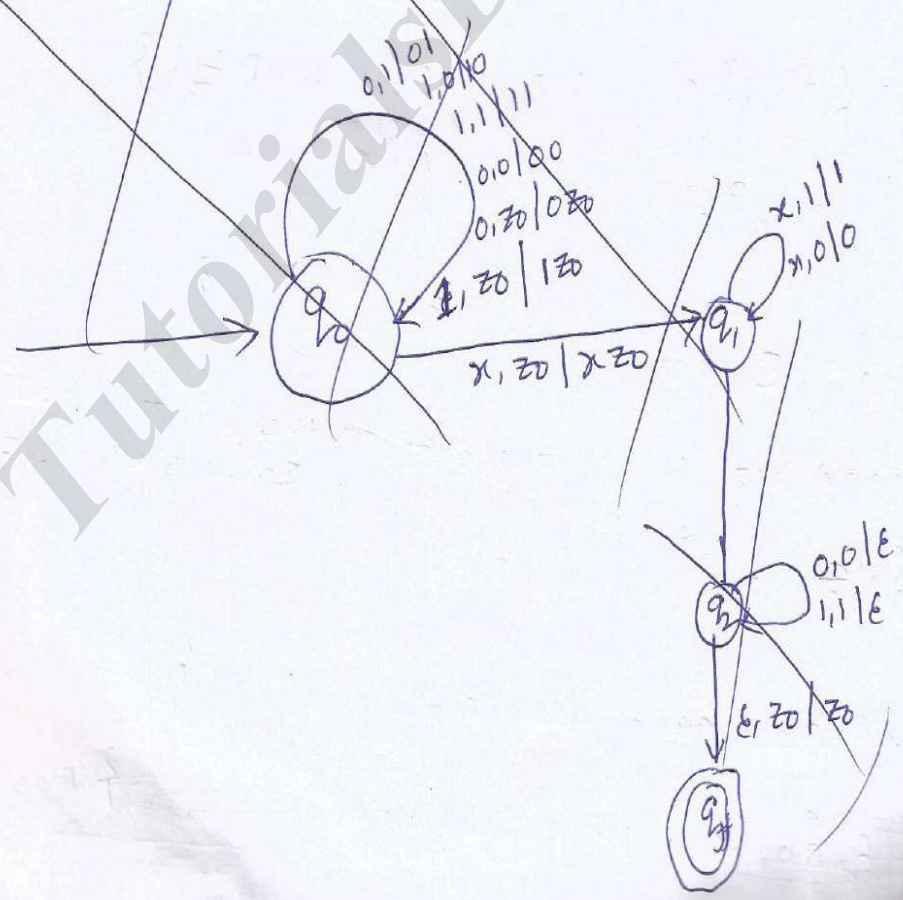
$L = \{ \varepsilon, ab, aabb, aaabbb, \ldots \}$



it is also D-PDA bcz it has only and only one output for given input from the same state.

Q. $L = \{ wxw^R \mid w \in (0+1)^* \}$

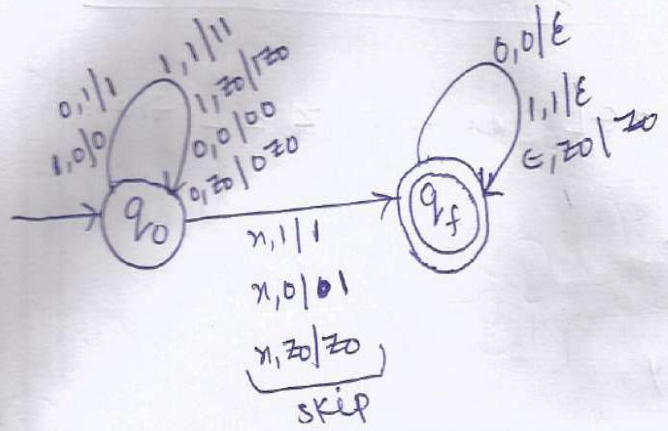
$L = \varepsilon, 00, 00, 000 \ldots$



TutorialsDuniya.com

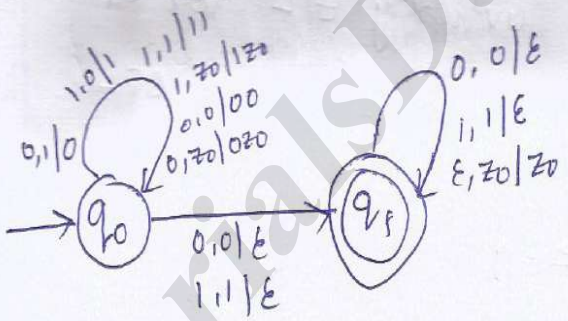
Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com



is is also D-PDA

Q $L = \{ ww^R \mid w \in (0+1)^* \}$



D-PDA Machine and Grammer CFL

TutorialsDuniya.com

Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

Pumping Lemma for CFL

if L is any CFL and $z \in L$ such that $|z| \geq L$ then

$u \cdot v^i \cdot w \cdot x^i \cdot y \in L$

where $z = uvwxy$

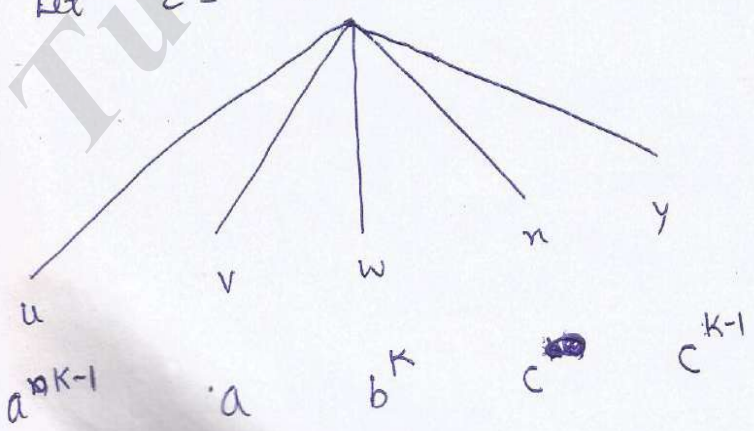
$|uvx| \leq |z|$

$|vx| \neq 0$

Pumping lemma for CFL is use to prove that sum of the language is NOT CFL. if there exist an integer i such that $uv^iwx^iy \notin L$ for at least one value of i then L is not CFL

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

Let $z = a^K b^K c^K$



Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

TutorialsDuniya.com

Download FREE Computer Science Notes, Programs, Projects, Books PDF for any university student of BCA, MCA, B.Sc, B.Tech CSE, M.Sc, M.Tech at https://www.tutorialsduniya.com

- Algorithms Notes
- Artificial Intelligence
- Android Programming
- C & C++ Programming
- Combinatorial Optimization
- Computer Graphics
- Computer Networks
- Computer System Architecture
- DBMS & SQL Notes
- Data Analysis & Visualization
- Data Mining
- Data Science
- Data Structures
- Deep Learning
- Digital Image Processing
- Discrete Mathematics
- Information Security
- Internet Technologies
- Java Programming
- JavaScript & jQuery
- Machine Learning
- Microprocessor
- Operating System
- Operational Research
- PHP Notes
- Python Programming
- R Programming
- Software Engineering
- System Programming
- Theory of Computation
- Unix Network Programming
- Web Design & Development

Please Share these Notes with your Friends as well

facebook WhatsApp twitter Telegram

Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

$uv^iwx^iy = a^{K-1} a^i b^K c^i c^{K-1}$

$\Rightarrow$ For $i = 2$

$a^{K-1} a^2 b^K c^2 c^{K-1}$

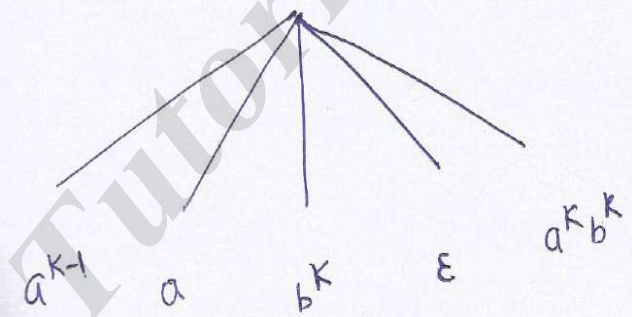
$a^{K+1} b^K c^{K+1} \notin L[a^K b^K c^K]$

here at least one value which contain at least one value $i = 2$ then L is not CFL

Q. $L = \{ww \mid w \in (a+b)^+\}$ Language is CFL or not ?

Let $z = (a+b)^+ (a+b)^+$

Let $z = a^K b^K a^K b^K$



$uv^iwx^iy = a^{K-1} a^i b^K \varepsilon^i a^K b^K$

for $i = 2$

$a^{K+1} b^K a^K b^K$

Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

Q $L = \{a^p \mid p \text{ is prime}\}$

Q $L = \{a^{2n} \mid n \geq 0\}$ chek CFL or not ?

TutorialsDuniya.com

Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

TutorialsDuniya.com

Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

Normal Forms for CFG :-

1. **Chomsky normal form :-** A CFG is in Chomsky normal form if every production is of the form

$A \to a$ or $A \to BC$ or $S \to \wedge$

is in G if $\wedge \in L(G)$

be Assume that S does not appear R.H.S. on the production

Reduction to Chomsky Normal Form :-

Step 1 :- elimination of null Productions and Unit Productions

Step 2 :- elimination of terminals on R.H.S.

Step 3 :- (i) Restricting the no. of variables on R.H.S.

TutorialsDuniya.com

Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

Q. Reduce the following Grammar G To CNF

G is $S \to aAD$

$A \to aB \mid bAB$

$B \to b$

$D \to d$

Sol^n Here there are no any NULL Production and ~~S → aabd~~ Unit Production then we can Process

Step2

Step2:- $G_1 = (V_N', \{a,b,d\}, P_1, S)$

where P_1 and V_N' are constructed as follow

(i) $B \to b$, $D \to d$ are included in P_1

(ii) $S \to aAD$ gives rise to $S \to C_aAD$ and $C_a \to a$

$A \to aB$ gives rise to $A \to C_aB$

$A \to bAB$ gives rise to $A \to C_bAB$ and $C_b \to b$

$V_N' = \{S, A, B, D, C_a, C_b\}$

Step3:- P_1 consist of $S \to C_aAD$, $A \to C_aB \mid C_bAB$, $B \to b$, $D \to d$, $C_a \to a$, $C_b \to b$

$A \to C_aB$, $B \to b$, $D \to d$, $C_a \to a$, $C_b \to b$

are included into P_2

$S \to C_aAD$ is replaced by $S \to C_aC_1$ and $C_1 \to AD$

Similarly

Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

$A \to C_b AB$ is replaced by $A \to C_b C_2$ where $C_2 \to AB$

Now $G_2 = (V_n'', \{a, b, d\}, \{P_2\}, \{S\})$

$V_n'' = \{S, A, B, D, C_a, C_b, C_1, C_2\}$

where P_2 consist of

$A \to C_a B$ $\quad$ $S \to C_a C_1$

$B \to b$ $\quad$ $A \to C_b C_2$

$D \to d$ $\quad$ $C_1 \to AD$

$C_a \to a$ $\quad$ $C_2 \to AB$

$C_b \to b$

Q. $S \to a \mid AbB$

$A \to aA \mid a$

$B \to bB \mid b$

Convered into CNF :

Step 1 :- Here there is no NULL and Unit Production we Can Process to Step 2

Step 2 :- $G_1 = (V_n', \{a, b\}, P_1, S)$

where P_1 and V_n' are constructed as follow

(i) $S \to a$, $A \to a$, $B \to b$ are included in P_1

Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

$S \to AbB$ gives rise to $S \to AC_bB$ and $C_b \to b$

$A \to aA$ gives rise to $A \to C_aA$ and $C_a \to a$

$B \to bB$ gives rise to $B \to C_bB$

$V_N' = \{S, A, B, C_a, C_b\}$

Step 3 :- P_1 consist of $S \to a$, $A \to a$, $B \to b$

$S \to AC_bB$, $A \to C_aA$, $B \to C_bB$

$C_b \to b$, $C_a \to a$

$S \to a$, $A \to a$, $B \to b$, $A \to C_aA$, $B \to C_bB$

$C_b \to b$, $C_a \to a$ are included in P_2

$S \to AC_bB$ is replaced by $S \to AC_1$

where $C_1 \to C_bB$

Now $G_2 = (\{V_N''\}, \{a, b\}, P_2, S)$

~~where P_2 and V_N'' are con~~

$V_N'' = \{S, A, B, C_a, C_b, C_1\}$

and P_2 consist of $S \to a$, $B \to C_bB$

$A \to a$, $C_a \to a$

$B \to b$, $C_b \to b$

$A \to C_aA$, $S \to AC_1$

$C_1 \to C_bB$

Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

Q. $S \to \sim S \mid [S\,]\,)S\,] \mid p \mid q$

Soln There is no null and unit production, we can proceed to step 2

Let $G_1 = \{ \{V'_N\}, \{\sim, [,],),], p, q\}, P, S\}$

where P_1 and V'_N are constructed as follows

i) $S \to p$, $S \to q$ are include in P_1

(ii) $S \to \sim S$, give rise to $S \to C_a S$, $C_a \to \sim$

$S \to [S\,]\,)S\,]$ give rise to $S \to C_b S C_c C_d S C_e$

where $C_b \to [$, $C_c \to]$, $C_d \to)$, $C_e \to]$

$V'_N = \{S, C_a, C_b, C_c, C_d, C_e\}$

P_1 consist of

$S \to C_a S \mid C_b S C_c C_d C_e \mid p \mid q$

$C_a \to \sim$, $C_b \to [$, $C_c \to]$, $C_d \to)$, $C_e \to]$

$S \to C_b S C_c C_d S C_e$ is replaced by $S \to C_b C_1$

where $C_1 \to S C_c C_d S C_e$

$C_1 \to S C_c C_d S C_e$ is replaced by $C_1 \to S C_2$

where $C_2 \to C_c C_d S C_e$

$C_2 \to C_c C_d S C_e$ is replaced by $C_2 \to C_c C_3$

where $C_3 \to C_d S C_e$

$C_3 \to C_d S C_e$ is replaced by $C_3 \to C_d C_4$

where $C_4 \to S C_e$

Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

Now

$G_2 = (\{V_N''\}, \Sigma, P_2, S)$

V_N'' and P_2 are constructed as follow

$V_N'' = \{S, C_a, C_b, C_c, C_d, C_e, C_1, C_2, C_3, C_4\}$

P_2 consist of $S \to P|q|C_aS|C_bC_1$

$C_a \to \sim$ $\quad C_c \to]$

$C_b \to [$ $\quad C_1 \to SC_2$

$C_c \to]$ $\quad C_2 \to C_cC_3$

$C_d \to)$ $\quad C_4 \to SC_c$

Q. $S \to aAbB$ — Convert the Grammer into
$A \to aA|a$ — chomsky normal form
$B \to bB|b$

Soln:- 1. There is no null and Unit production we can Procede to step 2

2 $G_1 = (\{V_N'\}, \{a, b\}, P, S)$

where P_1 and V_N' are constructed as follow

(i) $A \to a$, $B \to b$ are included in P_1

$A \to aA$ give rise to $A \to C_aA$ and $C_a \to a$

$B \to b \cdot B$ give rise to $B \to C_bB$ where $C_b \to a$

$S \to aAbB$ give rise to $S \to C_aAC_bB$

$V_N = \{S, A, B, C_a, C_b\}$

$P_1 =$ $\quad A \to a|C_aA$ $\quad C_a \to a$

$B \to C_bB|b$ $\quad C_b \to b$

$S \to C_aAC_bB$

Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

Step3 :- Production $A \to a \mid C_a A$

$B \to b \mid C_b B$

$C_a \to a, \; C_b \to b$

are included in P_2

$S \to C_a A C_b B$ is replaced by $S \to C_a C_1$ where $C_1 \to A C_b B$

$C_1 \to A C_b B$ is replaced by $C_1 \to A C_2$ where $C_2 \to C_b B$

$G_2 = (\{V_N''\}, \{a, b\}, P_2, S)$

where P_2 and V_N'' are constructed below

$P_2 =$ $A \to a \mid C_a A$

$B \to b \mid C_b B$

$S \to C_a C_1$

$C_2 \to C_b B$

$C_a \to a$

$C_b \to b$

$C_1 \to A C_2$

$V_N'' = \{S, A, B, C_a, C_b, C_1, C_2\}$

TutorialsDuniya.com

Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

GNF (Greibach Normal Form)

CFG is in CNF if every production is of the form

$A \to a\alpha$ where $\alpha \in V_n$

$a \in \Sigma$ ($\alpha \to \wedge$ may be)

& $S \to \wedge$ is in grammar if $\wedge \in L(G)$

when $\wedge \in L(G)$ we assume that S does not appear on RHS of any production.

Lemma (A) :- Let $G = (\{V_N\}, \Sigma, P, S)$ be a CFG

let the set of A Production be $A \to A\alpha_1 | A\alpha_2$

$A\alpha_3 \text{ --- } A\alpha_n | \beta_1 | \beta_2 | \beta_3 \text{ --- } | \beta_n)$

[β_i does not start with A]

let Z be a new variable here

$G_1 = (\{V_N'\}, \Sigma, P, S)$ & $V_N' = V_N \cup Z)$

where P_1 is define is as follows

(i) The set of A - Production in P_1 are

$A \to \beta_1 | \beta_2 | \beta_3 \text{ --- } | \beta_n$ with introduction of new variable

$A \to \beta_1 Z | \beta_2 Z | \beta_3 Z \text{ ---- } | \beta_n Z$

(ii) The set of Z production P_1 are

$Z \to \alpha_1 | \alpha_2 | \text{ --- } \alpha_n$ with introduction of new variable Z

Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

$Z \to \alpha_1 Z \mid \alpha_2 Z \mid \cdots \alpha_n Z$

(iii) The production for other variables are as S in P then G_1 is a CFG equivalent to G

Apply above lemma on Given CFG

$A \to aBD \mid bDB \mid C$

$A \to AB \mid AD$

$A \to \underbrace{aBD}_{\beta_1} \mid \underbrace{bDB}_{\beta_2} \mid \underbrace{C}_{\beta_3}$ | $A \to A\underbrace{B}_{\alpha_1} \mid A\underbrace{D}_{\alpha_2}$

Let Z be a new variable

$A \to aBDZ \mid bDBZ \mid CZ$

$Z \to \alpha_1 \alpha_2$

$Z \to B/D$

$Z \to BZ/DZ$

Reduction to CNF :-

(i) Elimination of NULL and Unit production
(ii) Renaming of Production
(iii) Convert A_n – Production using lemma A
(iv) modify A_i Production
(v) Modify Z_i Production
(vi) Construction of Grammar equivalent to given Grammar G.

Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

Q. Construct the Grammar in GNF

$S \to AA \mid a$

$A \to SS \mid b$

Sol^n :- (i) The given grammar is in CNF S and A Renamed as A_1 and A_2 respectively

So the production are

$A_1 \to A_2 A_2 \mid a$

$A_2 \to A_1 A_1 \mid b$

As their is no unit and NULL Production So we procede to Step 2

2

(i) A_1 Production are in required from. they are

$A_1 \to A_2 A_2 \mid a$

(ii) $A_2 \to b$ is in required form. Talking $A_2 \to A_1 A_1$ as the base Production using $A_1 \to A_2 A_2 \mid a$ in the base production.

then the resulting production are

$A_2 \to A_2 A_2 A_1$ and $A_2 \to a A_1$

then the input production are there

$A_2 \to b \mid a A_1 \mid A_2 A_2 A_4$

(iii) we have to apply lemma A to A_2 Production Introducing a new variable Z_2 to new product

set of A_2 Production in P_1 are

$A_2 \to \beta_1 \mid \beta_2$

$A_2 \to \beta_1 Z_2 \mid \beta_1 Z_2$

Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

$A_2 \rightarrow b \mid aA_1$

$A_2 \rightarrow bZ_2 \mid aA_1Z_2$

set of Z_2 Production in P_1 are

$Z_2 \rightarrow \alpha$

$Z_2 \rightarrow aZ_2$

$\therefore Z_2 \rightarrow A_2A_1$

$Z_2 \rightarrow A_2A_1Z_2$

4

(i) A_2 Production are

$A_2 \rightarrow aA_1 \mid b \mid aA_1Z_2 \mid bZ_2$

(ii) Among A_1 - Production we retain $A_1 \rightarrow a$ and replace all input Production in $A_1 \rightarrow A_2A_2$

$A_1 \rightarrow aA_1A_2 \mid bA_2 \mid aA_1Z_2A_2 \mid bZ_2A_2$ —— (1)

The set of all modify Production are

$A_1 \rightarrow a \mid aA_1A_2 \mid bA_2 \mid aA_1Z_2A_2 \mid bZ_2A_2$ —— (2)

5. Z_2 Production to be modified are

$Z_2 \rightarrow A_2A_1 \mid A_2A_1Z_2$

Modified Z_2

$Z_2 \rightarrow aA_1A_1 \mid bA_1 \mid aA_1Z_2A_1 \mid bZ_2A_1$

$Z_2 \rightarrow aA_1A_1Z_2 \mid bA_1Z_2 \mid aA_1Z_2A_1Z_2 \mid bZ_2A_1Z_2$ —— (3)

Hence the equivalent Grammar in 'c'

$G' = (\{V_N'\}, \Sigma, P_1, A_1)$

$V_N' = \{A_1, A_2, Z_2\}$

P_1 consist of (1), (2) (3)

Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

Q. consider a CFG with the following question :

$S \rightarrow ASA \mid B$

$B \rightarrow aCb \mid bCa$

$C \rightarrow ACA \mid A$

$A \rightarrow a \mid b$

i) Give three strings of length 7 in L(G)

ii) Are the following strings in L(G)

a) aaa b) bbb c) aba d) abb

(iii) is NULL(Λ) in L(G)?

Sol^n :- $S \rightarrow ASA$

~~$S \rightarrow AASAA$~~ $S \rightarrow ABA$ [$S \rightarrow B$]

~~$S \rightarrow AAASAAA$~~ $S \rightarrow AbCaA$ [$B \rightarrow bCa$]

~~$S \rightarrow aaa$~~ $S \rightarrow abACAaA$ [$C \rightarrow ACA$]

$S \rightarrow abaAaaa$ [$C \rightarrow a$]

$S \rightarrow abaaaaa$ [$A \rightarrow a$]

$S \rightarrow abaAaaa$

$S \rightarrow ababaaa$ [$A \rightarrow b$]

$S \rightarrow ASA$

$S \rightarrow ABA$ [$S \rightarrow B$]

$S \rightarrow AbCa$ [$B \rightarrow bCa$]

$S \rightarrow AbACAa$ [$C \rightarrow ACA$]

Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

TutorialsDuniya.com

Download FREE Computer Science Notes, Programs, Projects, Books PDF for any university student of BCA, MCA, B.Sc, B.Tech CSE, M.Sc, M.Tech at https://www.tutorialsduniya.com

- Algorithms Notes
- Artificial Intelligence
- Android Programming
- C & C++ Programming
- Combinatorial Optimization
- Computer Graphics
- Computer Networks
- Computer System Architecture
- DBMS & SQL Notes
- Data Analysis & Visualization
- Data Mining
- Data Science
- Data Structures
- Deep Learning
- Digital Image Processing
- Discrete Mathematics
- Information Security
- Internet Technologies
- Java Programming
- JavaScript & jQuery
- Machine Learning
- Microprocessor
- Operating System
- Operational Research
- PHP Notes
- Python Programming
- R Programming
- Software Engineering
- System Programming
- Theory of Computation
- Unix Network Programming
- Web Design & Development

Please Share these Notes with your Friends as well

facebook WhatsApp twitter Telegram

Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

$S \rightarrow ASA$

$S \rightarrow aSA \quad (A \rightarrow a)$

$S \rightarrow aASAA \quad (S \rightarrow ASA)$

$S \rightarrow aaSAA \quad (A \rightarrow a)$

$S \rightarrow aaBAA \quad (S \rightarrow b)$

$S \rightarrow aaacbAA \quad (B \rightarrow acb)$

$S \rightarrow aaaAbAA \quad (c \rightarrow A)$

$S \rightarrow aaaabAA \quad (A \rightarrow a)$

$S \rightarrow aaaabaa \quad (A \rightarrow a)$

$S \rightarrow aaaabaa \quad (A \rightarrow a)$

a) aaa Not matchi

b) bbb Not match

c) aba Not match

d) abb matched

$S \rightarrow B$

$S \rightarrow acb$

$S \rightarrow aAb$

$S \rightarrow aab$

NULL Production

TutorialsDuniya.com

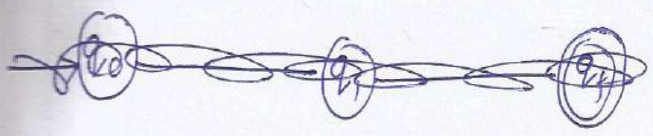
Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

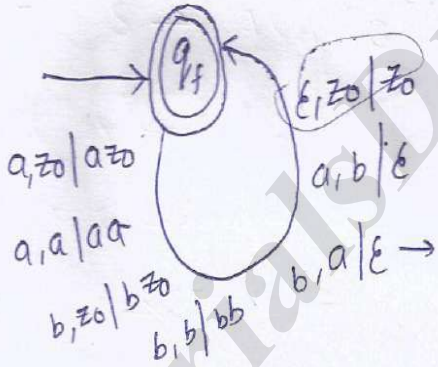
Q. Construct PDA Aexcepting the set of all string over a,b with equal no. of A's and B's

L = {ab, aabb, aaabbb, aaaabbbb - - - - - }

ba, bbaa, bbbaaa, bbbb, aaaa - - - - -



$L = \{ (a+b)^* \mid |w|_a = |w|_b \}$



Tutorialsduniya.com

Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

Push Down Automata and Context Free Grammer

Rule 1 :- $\delta(q, \wedge, A) = \{q, \alpha \mid A \rightarrow \alpha \text{ is in } P\}$

Rule 2 :- $\delta(q, a, a) = \{q, \wedge\}$ For every a in Σ

Question: Construct a PDA A equivalent is the following CFG :- CFG is

$S \longrightarrow 0BB$

$B \rightarrow 0S \mid 1S \mid 0$

Check whether 010^4 is in $N(A)$

Sol^n :- Define PDA as follow

$A = (\{q\}, \{S, B, 0, 1\}, \{0, 1\}, \delta, q, S, \phi)$

$A = \boxed{((q), V_N \cup \Sigma, \Sigma, \delta, q, S, \phi)}$

This is the general syntax for PDA

δ is define the following rule

R_1 $\delta(q, \wedge, S) = \{q, 0BB\}$

R_2 $\delta(q, \wedge, B) = \{q, 0S\}\{q, 1S\}\{q, 0\}$

R_3 $\delta(q, 0, 0) = \{q, \wedge\}$

R_4 $\delta(q, 1, 1) = \{q, \wedge\}$

Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

$(q, 010^4, S)$

$\vdash (q, 010^4, 0BB)$ By R_1

This symbol is used for check acceptency in CFG

$\vdash (q, 010^4, 01SB)$ By R_2

$\vdash (q, 010^4, 010BBB)$ By R_1

$\vdash (q, 010^4, 0100BB)$ By R_2

$\vdash (q, 010^4, 01000B)$ By R_2

$\vdash (q, 010^4, 010000)$ By R_2

$\vdash (q, 010^4, 010^4)$ B R_4

$\vdash (q, \wedge)$

$(q, 010^4, S)$

$\vdash (q, 010^4, 0BB)$ By R_1

$\vdash (q, 10^4, BB)$ By R_3

$\vdash (q, 10^4, 1SB)$ By R_2

$\vdash (q, 0^4, SB)$ By R_4

$\vdash (q, 0000, 0BBB)$ By R_1

$\vdash (q, 000, BBB)$ By R_3

$\vdash (q, 000, 000)$ By R_2

$\vdash (q, \wedge)$ By R_3

TutorialsDuniya.com

Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

Q. $S \rightarrow aSb / A$

$A \rightarrow bSa \mid S \mid \wedge$

whether check acceptency "aabbbab"

Solⁿ :-

R_1 $\delta(q, \wedge, S) = \{q, aSb\}\ \{q, A\}$

R_2 $\delta(q, \wedge, A) = \{q, bSa\}\ \{q, S\}, \{q, \wedge\}$

R_3 $\delta(q, a, a) = \{q, \wedge\}$

R_4 $\delta(q, b, b) = \{q, \wedge\}$

Acceptency check :-

$(q, aabbbab, S)$

$\vdash (q, aabbbab, aSb)$ By R_1

$\vdash (q, abbbab, Sb)$ By R_3

$\vdash (q, abbbab, aSbb)$ By R_1

$\vdash (q, bbbab, Sbb)$ By R_3

$\vdash (q, bbbab, Ab)$ By R_1

$\vdash (q, bbbab, bSab)$ By R_2

$\vdash (q, bbab, Sab)$ By R_4

Aab

Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

Q. $S \to ASA \mid bA$

$A \to B \mid S$

$B \to c$

$S \to ASA \mid bA$

$A \to c \mid ASA \mid bA$

$B \to c$

Soln :- Here, there are NO NULL production and Unit Production are then we can process step 2

$A \to B$

$A \to S$

$A \to c$

$A \to ASA \mid bA$

Now there is no unit production

Step 2 :- $G_1 = (V_N', \{b, c\}, P_1, S)$

where P_1 and V_N' constructed as follow

(i) $B \to c$, $A \to c$ are included in P_1

~~(ii) S → ASA~~

(ii) $S \to bA$ gives rise to $S \to C_bA$ where $C_b \to b$

$A \to bA$ gives rise to $A \to C_bA$

$V_N' = \{S, A, B, C_b\}$

P_1 : $S \to ASA \mid C_bA$, $C_b \to b$

$A \to c \mid ASA \mid C_bA$

$B \to c$

step3 :- $S \to C_bA$, $A \to c \mid C_bA$ and $B \to c$

$C_b \to b$ are included in P_2

$S \to ASA$ is replaced by $S \to AC_1$ where

$C_1 \to SA$

$A \to ASA$ is replaced by $A \to AC_2$ where

$C_2 \to SA$

Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

Now $G_2 = (\{V_N''\}, (b, c), P_2, S)$

Where P_2 and V_N'' are constructed below

$V_N'' = \{S, A, B, C_b, C_1, C_2\}$

P_2 : $S \to AC_1 \mid C_bA$, $C_1 \to SA$

$A \to c \mid AC_1 \mid C_bA$, $C_b \to b$

$B \to c$, $C_2 \to SA$

Ans

1. Find Equivalent Grammar and Equivalent PDA to the given language

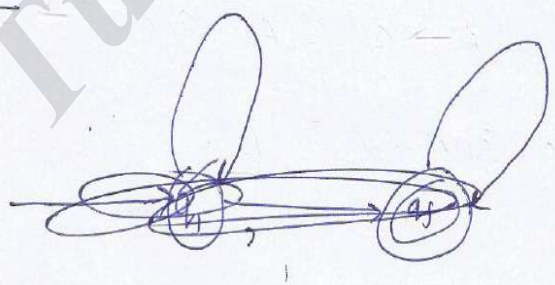
$L = \{ww^R \mid w \in (a+b)^*\}$

$S \to aSA \mid bSB \mid aA \mid bB$

$A \to a$

$B \to b$

PDA :-



R_1 : $\delta(q, \Lambda, S) = (q, aSA \mid bSB \mid aA \mid bB)$

R_2 : $\delta(q, \Lambda, A) = (q, a)$

R_3 : $\delta(q, \Lambda, B) = (q, b)$

Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

$R_4 \quad (q, a, a) = (q, \Lambda)$

$R_5 \quad (q, b, b) = (q, \Lambda)$

Q. find the RE for the set of L string in which every 0's followed by at least 2 one's

$\varepsilon, 1, 011, 0111,$

$\cancel{RE = (1+011)^*}$

$w \in L$

$1^* + (011)^*$

$(1^* + (011)^*)^*$

$(1+011)^*$

Q. no. of string of length ≤ 3 generated by RE $(0+10)^*$

$L = \{\varepsilon, 010, \cancel{001}, 000, \cancel{011}, 0, 10, 00, 100\}$

$L_0 = \varepsilon$

$L_1 = 0$

$L_2 = 10, 00$

$L_3 = 100, 000, 010$

TutorialsDuniya.com

Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

Q. which of the following RE does not contain the sub string 1101

a) $(10)^* 010^*$ NO → 10101001010 10

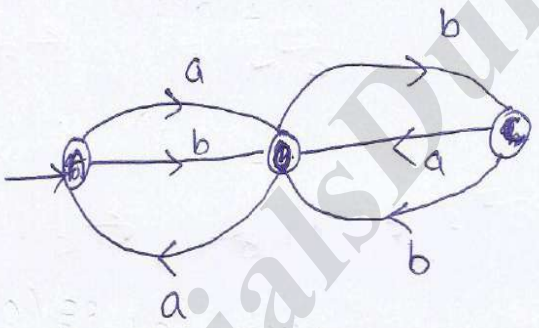
b) $01^* 01^*$ = {0 1101}, {0 1101}

c) $(11)^* 0 (01)^*$ NO

d) $(1+0)^*$

Ans a and c

Q find RE ?



$R = QP^*$

Arden's theorem :-

$A = B^a + \varepsilon$

$A = \varepsilon + Ba$

$\downarrow \quad \downarrow \quad \downarrow$

$R \quad Q \quad P$

$B = Aa + Ab$

$B = A(a+b)$

$A = \varepsilon a^*$

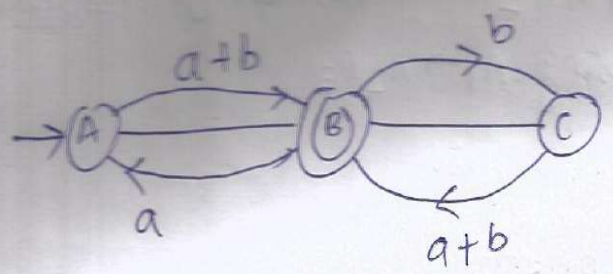
$A = a^*$

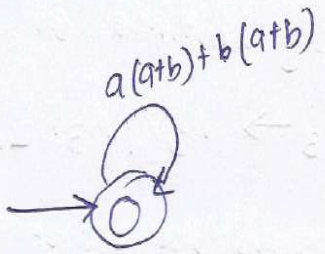
$B = Aa + Ab$

TutorialsDuniya.com

Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com





$a(a+b) + b(a+b)$

$$r = [a(a+b) + b(a+b)]^*$$

$$R = [(a+b)^2]^*$$

Q. Convert the Given Grammar into GNF

$S \to AB$

$A \to BS \mid b$

$B \to SA \mid a$

Solⁿ :- The Given Grammar is in CNF S, A and B are renaming as A_1, A_2, A_3 respectively, so the Production are -

$A_1 \to A_2 A_3$ $\qquad$ $A_3 \to A_1 A_2 \mid a$

$A_2 \to A_3 A_1 \mid b$

As their is no NULL or Unit Production so we can Proceed to step 2

Step 2:- (i) $A_1 \to A_2 A_3$ is in required form.

(ii) $A_2 \to b$ is in required form

(iii) $A_2 \to A_3 A_1 \mid b$ are in required form

(iii) $A_3 \to a$ is in required form

Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

Apply lemma to $A_3 \rightarrow A_1 A_2$. The Resulting Production are $A_3 \rightarrow A_2 A_3 A_2$ using $A_1 \rightarrow A_2 A_3$

Apply lemma on A_3 Again

$$A_3 \rightarrow A_3 A_1 A_3 A_2 \mid b A_3 A_2$$

Step3 Now we have to apply lemma A on A_3 Prod^n introducing a new variable Z_3

$$A_3 \rightarrow \underbrace{a}_{\beta_1} \mid \underbrace{b A_3 A_2}_{\beta_2} \mid A_3 \underbrace{A_1 A_3 A_2}_{\alpha}$$

$$A_3 \rightarrow a Z_3 \mid b A_3 A_2 Z_3$$

$$Z_3 \rightarrow A_1 A_3 A_2$$

$$Z_3 \rightarrow A_1 A_3 A_2 Z_3$$

Now A_3-Production are

$$A_3 \rightarrow a Z_3 \mid b A_3 A_2 Z_3 \mid a \mid b A_3 A_2 \quad \text{—} (1)$$

Step 4 Among A_2 Production we retain $A_2 \rightarrow b$ and replace all the input production in

$$A_2 \rightarrow A_3 A_1$$

$$A_2 \rightarrow a A_1 \mid b A_3 A_2 A_1 \mid a Z_3 A_1 \mid b A_3 A_2 Z_3 A_1$$

The set of all modify production are

$$A_2 \rightarrow b \mid a A_1 \mid b A_3 A_2 A_1 \mid a Z_3 A_1 \mid b A_3 A_2 Z_3 A_1 \quad \text{—} (2)$$

For A_1 Production we replace the A_2 in $A_1 \rightarrow A_2 A_3$

$$A_1 \rightarrow b A_3 \mid a Z_3 A_1 A_3 \mid b A_3 A_2 Z_3 A_1 A_3 \mid a A_1 A_3 \mid b A_3 A_2 A_1 A_3 \quad \text{—} (3)$$

Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

Step 5:- Z_3 Production to be modify are

$Z_3 \rightarrow A_1 A_3 A_2 \mid A_1 A_3 A_2 Z_3$

$Z_3 \rightarrow bA_3 A_3 A_2 \mid aZ_3 A_1 A_3 A_3 A_2 \mid aA_1 A_3 A_3 A_2 \mid$

$bA_3 A_2 Z_3 A_1 A_3 A_3 A_2 \mid bA_3 A_2 A_1 A_3 A_3 A_2$

$Z_3 \rightarrow bA_3 A_3 A_2 Z_3 \mid aZ_3 A_1 A_3 A_3 A_2 Z_3 \mid aA_1 A_3 A_3 A_2 Z_3$

$bA_3 A_2 Z_3 A_1 A_3 A_3 A_2 Z_3 \mid bA_3 A_2 A_1 A_3 A_3 A_2 Z_3$

Q. Find a Grammar in GNF equivalent to the Grammar

$E \rightarrow E + T / T$

$T \rightarrow T \times F / F$

$F \rightarrow (E) / a$

TutorialsDuniya.com

Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

Closure properties of D-CFL :-

Closed -

i) Compliment
ii) intersection in regular set
iii) quotient with regular set

Not Closed -

(i) union
(ii) intersection
(iii) Concentaition
(iv) Klean closer
v

Closure properties of CFL :-

Closed :-

(i) union
(ii) concentation
(iii) Klean closure
(iv) Reversal
(v) quotient with R set
(vi) intersection with R set

Not closed :-

(i) intersection (ii) compliment

TutorialsDuniya.com

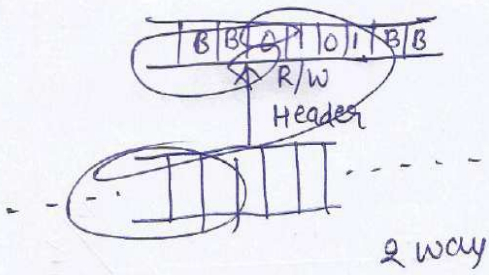
Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

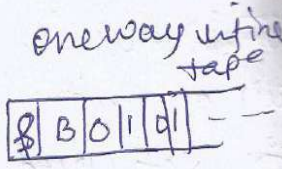
Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

Turing Machine

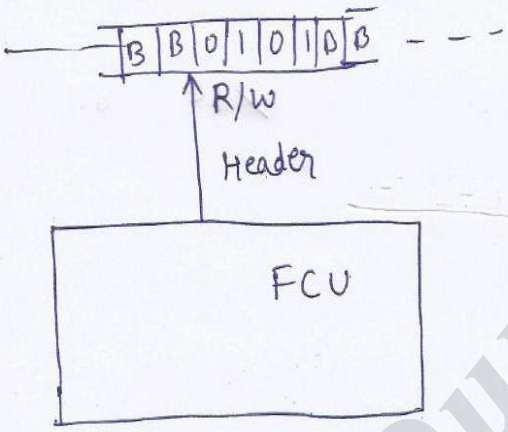
11-Sep-2018

Block diagram of Turing machine :-





multitape (infinite tape)



Q 3 components in Turing machine :-

(i) two way infinite tape

(ii) Read/write header

(iii) Finite control unit

infinite tape :-

TutorialsDuniya.com

Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

Read/write Header :- Input लेगा।

TM is DTM when by default

TM is accepted two language
i) Recursive
ii) Recursive innumerable language

Defination :- TM is collection of 7 tuples

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

where Q set of state

Σ input symbols

Γ ~~top most symbol~~ set of all tap Symbol

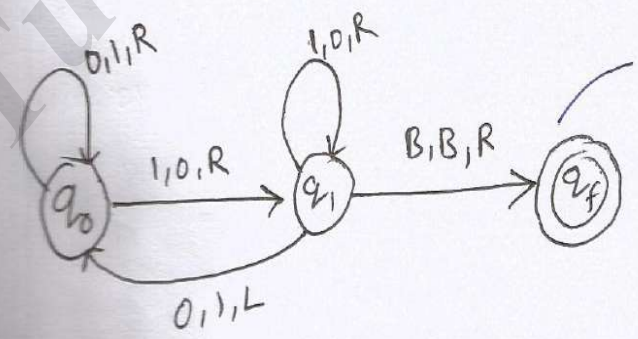
B Blank symbol

$\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}$

q_0 initial state

F final state

ex.



Transition diagram

$\delta(q_i, a) = (q_j, x, R)$
or
(q_j, x, L)

$\delta(q_0, 0) = (q_0, 1, R)$

Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

Representation of Turing Machine :-

it can be represented by two ways-

i) Transition table

(ii) Transaction diagram

Σ \ Q	q_0	q_1	q_f
0	q_0	q_0	
1	q_0	q_1	

Q \ Σ	0	1	B
q_0	$(q_0,1,R)$	$(q_1,0,R)$	halt
q_1	$(q_0,1,L)$	$(q_1,0,R)$	(q_f,B,R) ~~halt~~
q_f	halt	halt	halt

ID's

$\delta(q_0,0) = (q_0,1,R)$

$\delta(q_0,1) = (q_1,0,R)$

$\delta(q_1,0) = (q_0,1,L)$

$\delta(q_1,1) = (q_1,0,R)$

$\delta(q_1,B) = (q_f,B,R)$

TutorialsDuniya.com

Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

elimination of NULL Production :-

$$G = \begin{cases} S \to aS \mid AB \\ A \to \Lambda \\ B \to \Lambda \\ D \to b \end{cases}$$

$$G' = \begin{cases} S \to aS, \quad S \to AB \\ S \to B, \quad S \to A \\ S \to a \\ D \to b \end{cases}$$

Unit production :-

$S \to A$ (NON terminal), $A \to B$, $B \to C$

$C \to a$ → terminal

⇓ Result

$G' = \boxed{S \to a}$

$A = BC$ (NON terminal & start)

$S = BC$ दो 2 variable

Topics :-

i) Turing Machine recognisable Languages

ii) Construction of PDA

(iii) Pumping Lemma for CFL

(iv) Languages and their Automata

TutorialsDuniya.com

Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

$x = y^a$, $y = z^b$ $z = x^c$

find abc

$a = b = c = 1$

$x = y^{ab}$ $x = y^1$

$= z^{ab}$ $y = z^1$

$x = x^{abc}$ $z = x$

x

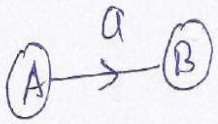
$|uv| \le |z|$

uv^iw

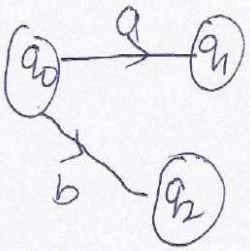
Imp.

Q diff^n b/w DFA and NFA

Path Unique in DFA



NFA



TutorialsDuniya.com

Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com

Download FREE Theory of Computation Notes PDF from TutorialsDuniya.com

TutorialsDuniya.com

Download FREE Computer Science Notes, Programs, Projects, Books PDF for any university student of BCA, MCA, B.Sc, B.Tech CSE, M.Sc, M.Tech at https://www.tutorialsduniya.com

- Algorithms Notes
- Artificial Intelligence
- Android Programming
- C & C++ Programming
- Combinatorial Optimization
- Computer Graphics
- Computer Networks
- Computer System Architecture
- DBMS & SQL Notes
- Data Analysis & Visualization
- Data Mining
- Data Science
- Data Structures
- Deep Learning
- Digital Image Processing
- Discrete Mathematics
- Information Security
- Internet Technologies
- Java Programming
- JavaScript & jQuery
- Machine Learning
- Microprocessor
- Operating System
- Operational Research
- PHP Notes
- Python Programming
- R Programming
- Software Engineering
- System Programming
- Theory of Computation
- Unix Network Programming
- Web Design & Development

Please Share these Notes with your Friends as well

facebook WhatsApp twitter Telegram

Download FREE Computer Science all Subjects Notes from TutorialsDuniya.com