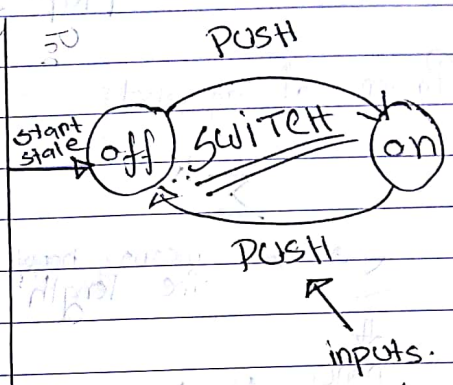
CHAP 1

Paper Source
Subject: CSE 331
Date: 14 Jan, 18 Time:

Automata: An abstract model of a computing machiene / system / software / device.

	type	Symbol
1	State	
2	transition	input
3	Start state	start state off
4	final state / accepting state	



inputs. without them states won't change.

Types of inputs.

Alphabets: A finite set of ~~finite~~ symbols. (Σ)

like, the english alphabel $\Sigma = \{a, b, c, \ldots z\}$

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

String / Word / w

a sequence of symbols choosen from an alphabet.

$\Sigma = \{0, 1\}$ = 01, 001, 11010 } these are all strings.

Scanned by CamScanner

Paper Source
Subject
Date: Time:

length of a string,

$|1010| = 4$

$|10| = 2$

Power of alphabet:

Σ^{K} Here K is an integer.

Σ^{0} ← means, how strings are there gonna be having the length 0.

⇓

none. so An empty string. Which is denoted by ε or ϵ

$\boxed{\Sigma^{0} = \epsilon}$ Fixed for all alphabets.

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

$\Sigma^{2} = \{\underbrace{00}_{\text{length 2}}, 01, 10, 11\}$

$\Sigma^{*} = \{\epsilon, 00, 01, 10, 11, 111, \text{------}\}$

STAR OF AN ALPHABET

In this case we don't know what the value of K is. so it will ALWAYS start from ZERO!!

Scanned by CamScanner

Paper Source
Subject
Date: Time:

Σ^+ = length starts from zero

= { 00, 01, 11, --- }

So, $\Sigma^* = \Sigma^0 \cup \Sigma^+$

combination of both

Language (L)

$\Sigma^* = \{\epsilon, a, b, c, ---\}$

These are all possible combinations. Among them those who are meaningful (they have a certain condition) are languages.

$L \subseteq \Sigma^*$

subset.

Language is a set of strings chosen from an Σ^* based on an condition.

Ex: All string starting with zero.

$L = \{01, 0, 001 --\}$

Scanned by CamScanner

Paper Source
Subject 331, chap2
Date: Time:

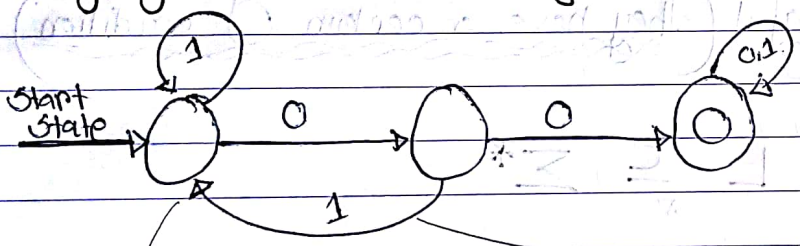
FA = finite Automata

DFA — deterministic FA

NFA — non-deterministic
can not be determined.

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

language = all strings that contain sub-string "00".



After finding 0 if found 1.

101 — not a part of (L)

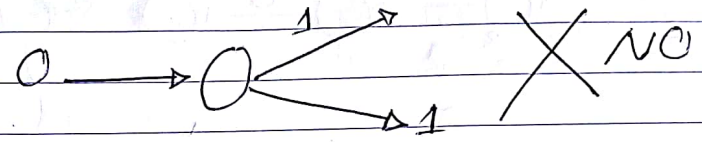
101001 — part of (L).

didn't find 0, then again look for 0.

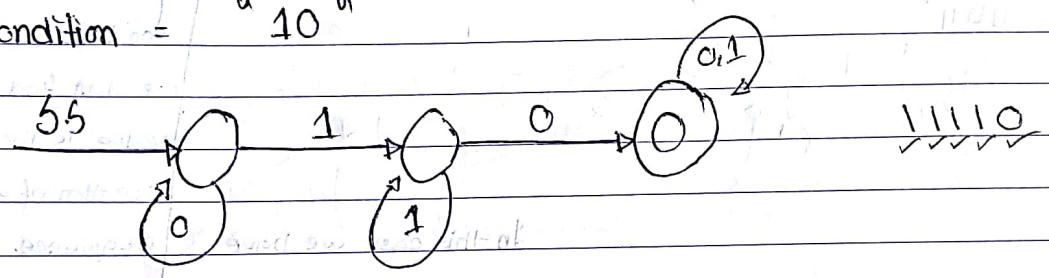
Paper Source
Subject
Date: Time:

*** Alphabet's symbol will always be the input ***

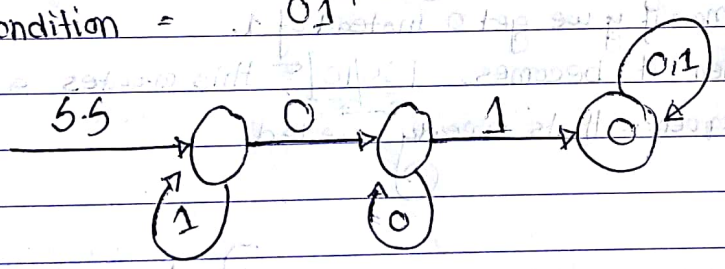
in DFA we have to show what will happen if we get either 0 or 1. & there can not be any double transitions.



condition = "10"



condition = "01"



For those which reach the final stage will be added.

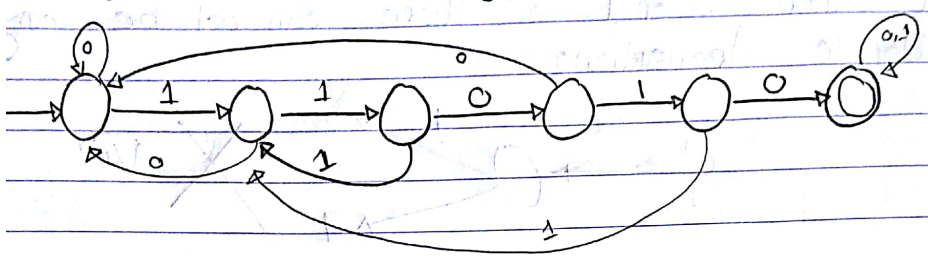
Scanned by CamScanner

Paper Source
Subject: CSE 331, Lec-2
Date: Jan, 17, 18 Time:

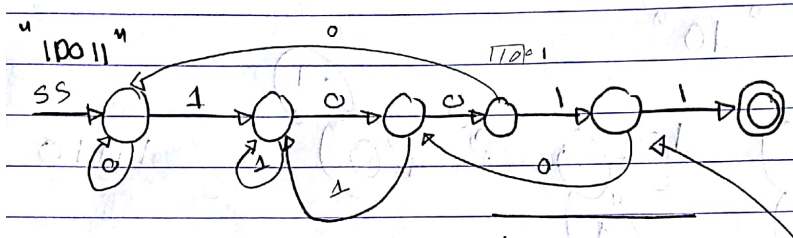
DFA

$\Sigma(0,1)$

language = all strings containing "11010"



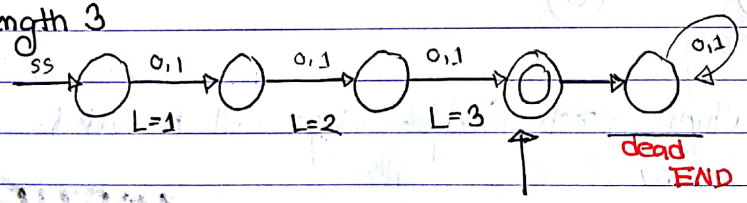
"10011"



You have to look, if we don't find the sequence value is there any creation of a new sequence.

In this case we have already gotten 1001.
now if we get 0 instead of 1.
then it becomes, 100$\boxed{10}$ this creates a new sequence thats already needed.

length 3



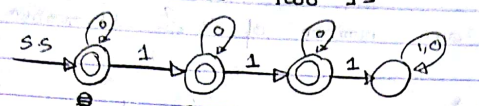
condition is satisfied.

If we have more then it will go on an infinite loop.

In terms of length there 3 types
→ at least
→

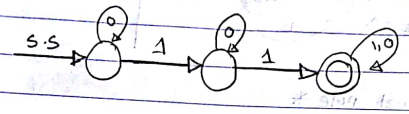
Paper Source
Subject
Date: Time:

- contains ~~atleast~~ almost two 1's



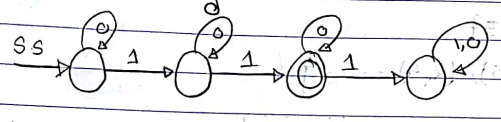
zero 1's One, 1 two 1's MORE than 2 thats why dead end.

- contains atleast two 1's.

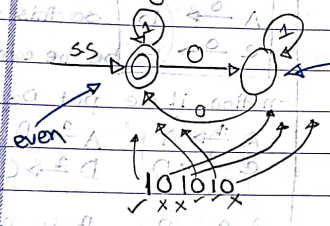


there can be conditions where they can tell you to find consecutive 1's.

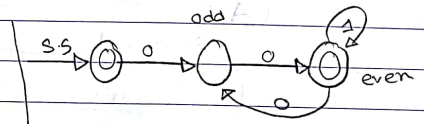
- contains exactly two 1's.



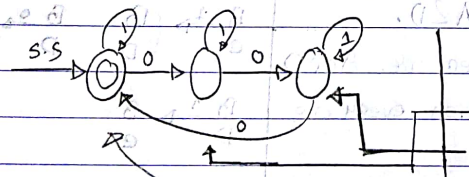
- All strings that contain even number of 0's.



OR



- whose number of 0's is a multiple of 3 (0, 3, 6, 9, 12, ...)



If we don't have any 0's then, ✓
for 1
for 2
for 3 → goes back to the final stage.

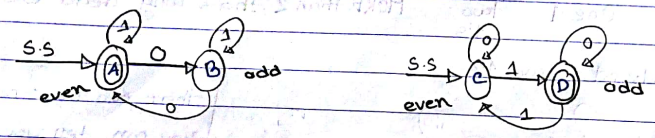
Scanned by CamScanner

Paper Source
Subject
Date: Time:

And, OR operators

→ all strings that contain even number of 0's & odd number of 1's.

* Create seperate DFA's *

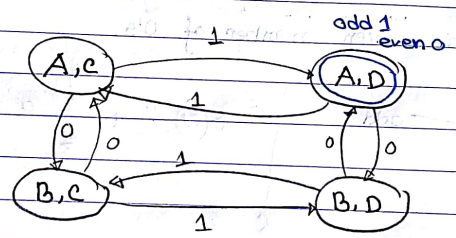


We have label each state.

* We have to follow cartesian product rule *

$S = \{1, 2, 3\}$ $T = \{4, 5\}$

$S \times T = \{(1,4), (1,5), (2,4) \ldots\}$



Now, if A,C both are zero

$\boxed{\begin{array}{l} A \xrightarrow{0} B \\ C \xrightarrow{0} C \end{array}}$ so this will be the combination if we put 0.

$A \xrightarrow{1} A$, $C \xrightarrow{1} D$ $\quad$ $A \xrightarrow{1} A$, $D \xrightarrow{1} C$

$A \xrightarrow{0} B$, $D \xrightarrow{0} D$ $\quad$ $B \xrightarrow{0} A$, $C \xrightarrow{0} C$

$B \xrightarrow{1} B$, $C \xrightarrow{1} D$ $\quad$ $B \xrightarrow{0} A$, $D \xrightarrow{0} D$

$B \xrightarrow{1} B$, $D \xrightarrow{1} C$

for determining the final state

1 fs ← AND consider both A & D.

3 fs OR " ~~ont~~ either one. A or D.

if the question was OR then every one will be final.

MAROF HOSSEN

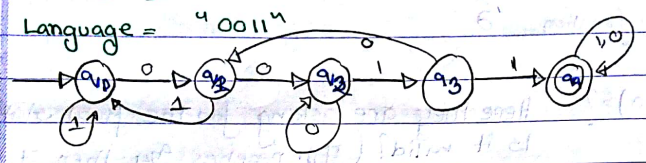
Paper Source
Subject CSE 331 Lec-3
Date: Jan 21, 18 Time:

DFA

(i) Transition diagram
(ii) " function
(iii) " table

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

Language = "0011"



This is the transition dia-gram.

(i) Transition table.

→ show all the states that occur in the diagram

States	0	1
→ * q_0	q_1	q_0
q_1	q_2	q_0
q_2	q_3	q_3
q_3	q_1	q_4
* q_4	q_4	q_4

→ means starting
* " end.

For every alphabet there will be one coloumn.

(ii) Transition function "δ"

$\delta(q_0, 0) = q_1$

current state
current symbol

~~do it for the next one.~~

Here we have to explicitly mention the starting state & end / final state.

q_0 = S.S
q_4 = F.S

Scanned by CamScanner

Paper Source
Subject
Date: Time:

$\partial(q_0, 1) = q_0$
$\partial(q_0, 0) = q_1$
$\partial(q_1, 0) = q_2$
$\partial(q_1, 1) = q_0$
-- so on

Extended transition function $\hat{\partial}$

$\hat{\partial}(q_0, 110101 00110) = ?$ Here they are asking for the final state. Is it valid? (If it reaches q_4 then it valid)

$\partial(q_0, 1) = q_0$
$\partial(q_0, 1) = q_0$
$\partial(q_0, 0) = q_1$
$\partial(q_1, 1) = q_0$
$\partial(q_0, 0) = q_1$
$\partial(q_1, 1) = q_0$
$\partial(q_0, 0) = q_1$
$\partial(q_1, 1) = q_0$

$\partial(q_0, 0) = q_1$
$\partial(q_1, 0) = q_2$
$\partial(q_2, 1) = q_3$
$\partial(q_3, 1) = q_4$
$\partial(q_4, 0) = q_4$

If it doesn't go to q_4 then its not Valid.
The ans can be written as,
$L = \{ w \mid w$ contain the substring "0011" $\}$

Hamal Pomem

Scanned by CamScanner

Paper Source
Subject
Date: Time:

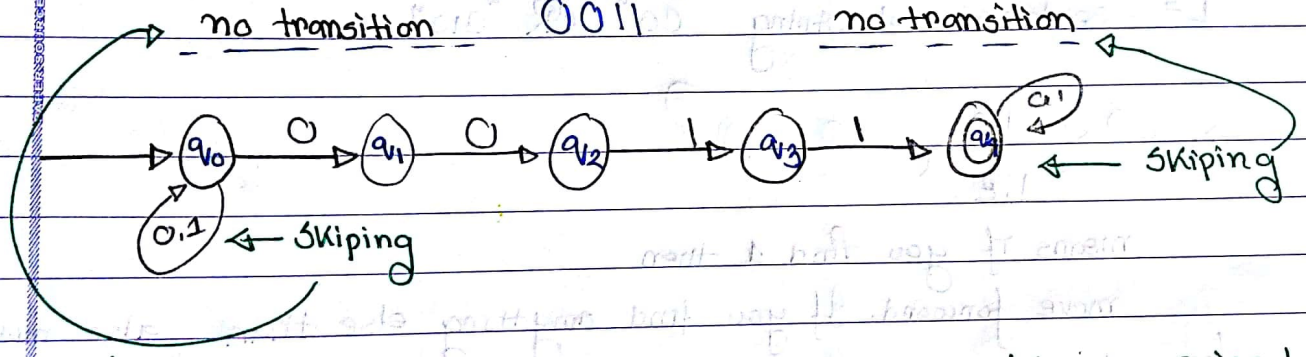
NFA

non-deterministic finite automata.

- → There are no rules
- → imaginary
- → parallel computation.

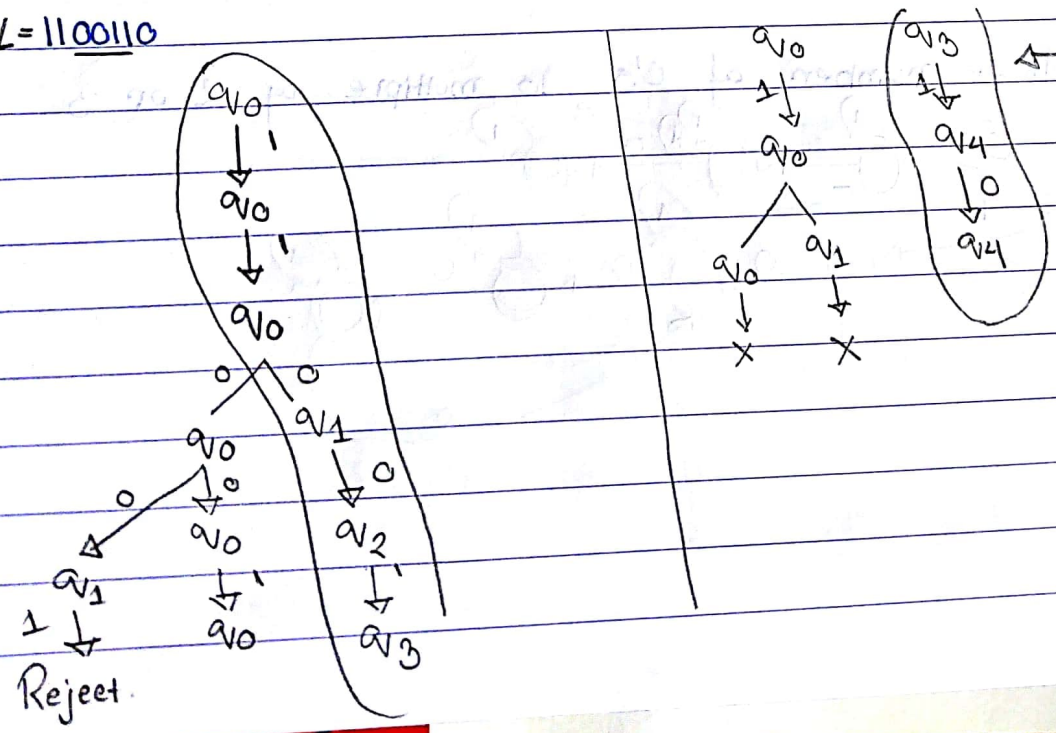
* all DFA's are NFA. However not all NFAs are DFAs. *

In the prev Question, string "0011". For NFA we don't think about the previous numbers & ~~after~~ later numbers that come after "0011."



In NFA there can be multiple branches of the same input.

L = 1100110

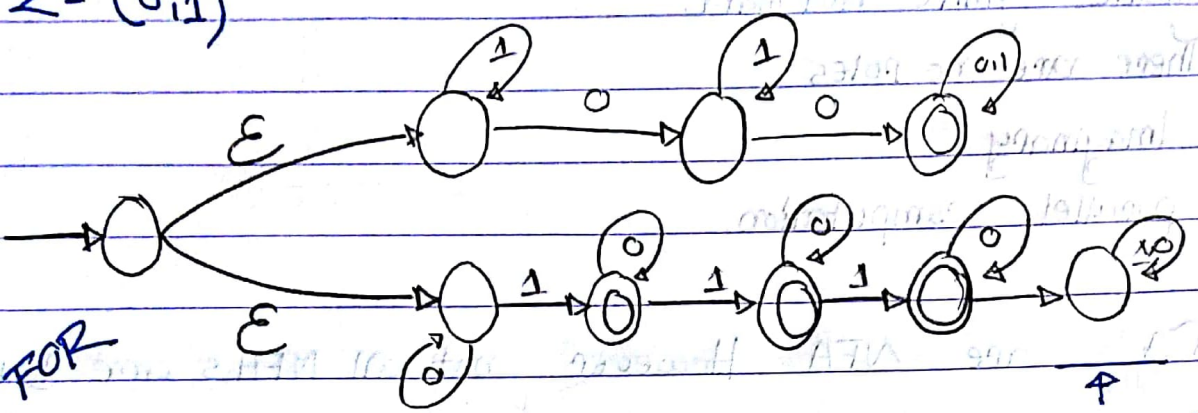


Scanned by CamScanner

Paper Source
Subject
Date: Time:

L= All strings that contain at least 2 0's or atmost 3 1's.

$\Sigma = (0,1)$

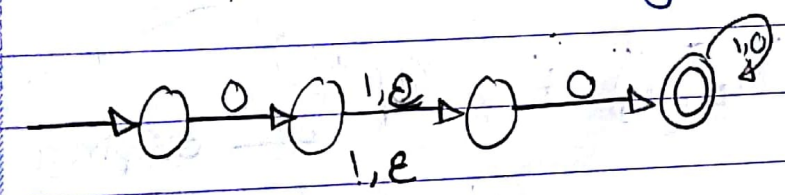


FOR 2^n

Start with empty easier to branch.

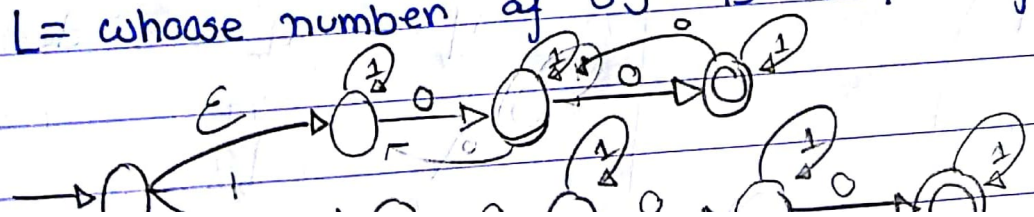
you can write the dead end or not. doesn't matter.

L= contains sub-string "00" OR "010"



means if you find 1 then move forward. If you find anything else than also move forward.

L= whose number of 0's is multiple of 2 or 3



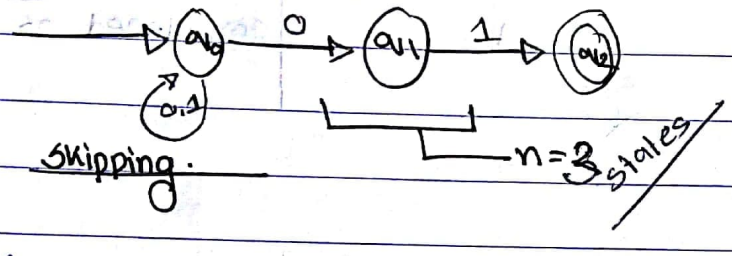
Scanned by CamScanner

NFA → DFA (transform)

subset construction 2^n. (all possible DFA combinations)

DFA ≥ NFA (NFA = n)

L = all strings that end with "01"



skipping.

Subset construction table

	State	0	1
empty	$\emptyset$	—	—
→	q_0	q_0, q_1	q_0
	q_1	$\emptyset$	q_2
*	q_2	$\emptyset$	$\emptyset$
	q_0, q_1	q_0, q_1	q_0, q_2
*	q_0, q_2	q_0, q_1	q_0
*	q_1, q_2	$\emptyset$	q_2
*	q_0, q_1, q_2	q_0, q_1	q_0

for multiples

Ex, q_0, q_1 for 0.

$q_0 \xrightarrow{0} q_0, q_1$

$q_1 \xrightarrow{0} \emptyset$

then union

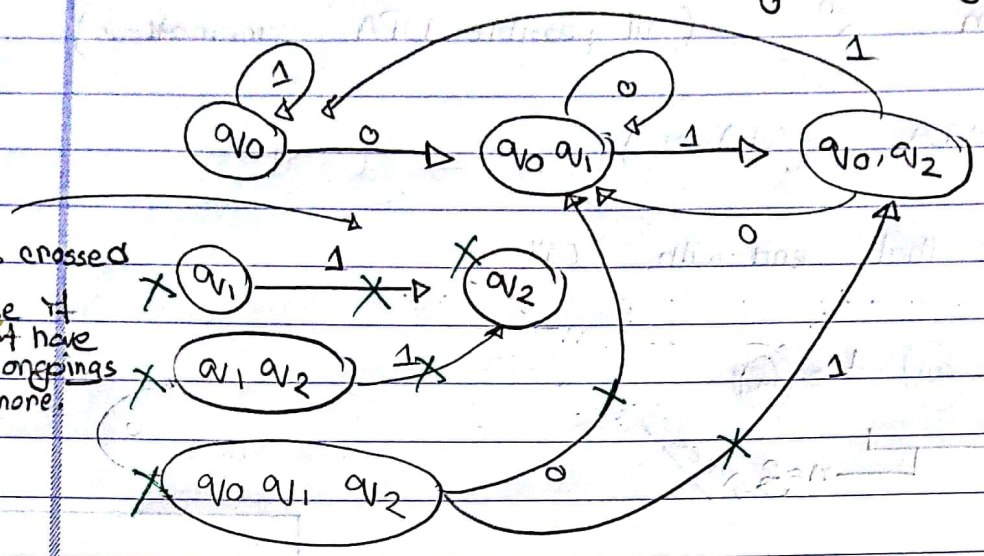
q_0, q_1

→ starting

* end (anything that contains the

Scanned by CamScanner

From the previous example, putting it into graph/diagram format

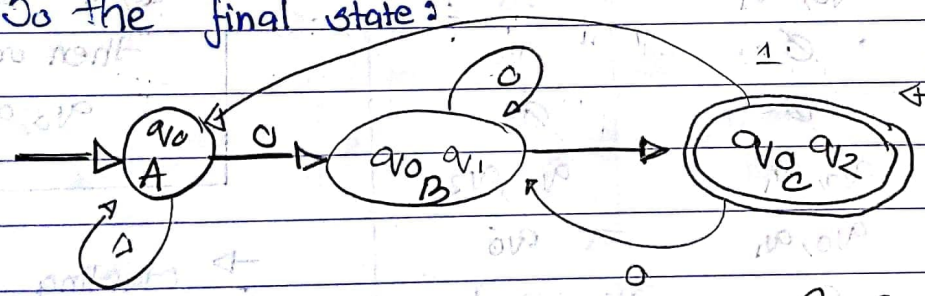


Here, for every new state that we find we have to create another state in the diagram. ($q_0 q_1$) are considered as one.

~~How~~ Eleminating inaccessible states:

The states that have no ongoing arrows are crossed with their corresponding path.

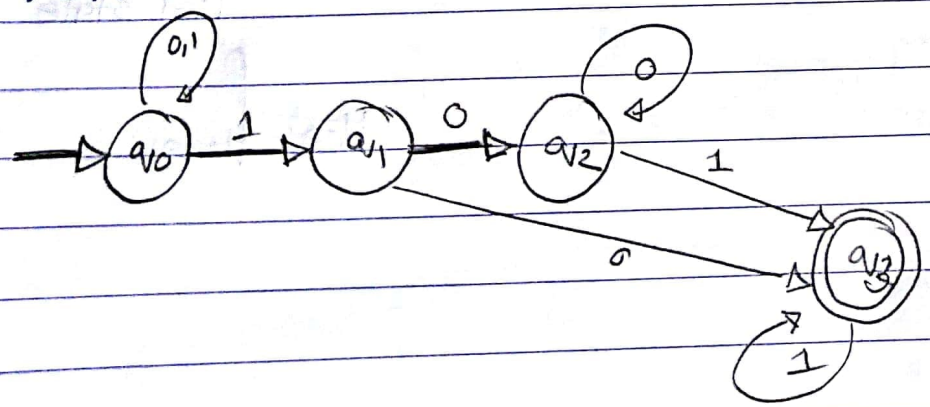
So the final state:



← final because contains q_2.

We can write A, B, & C.

NFA



Scanned by CamScanner

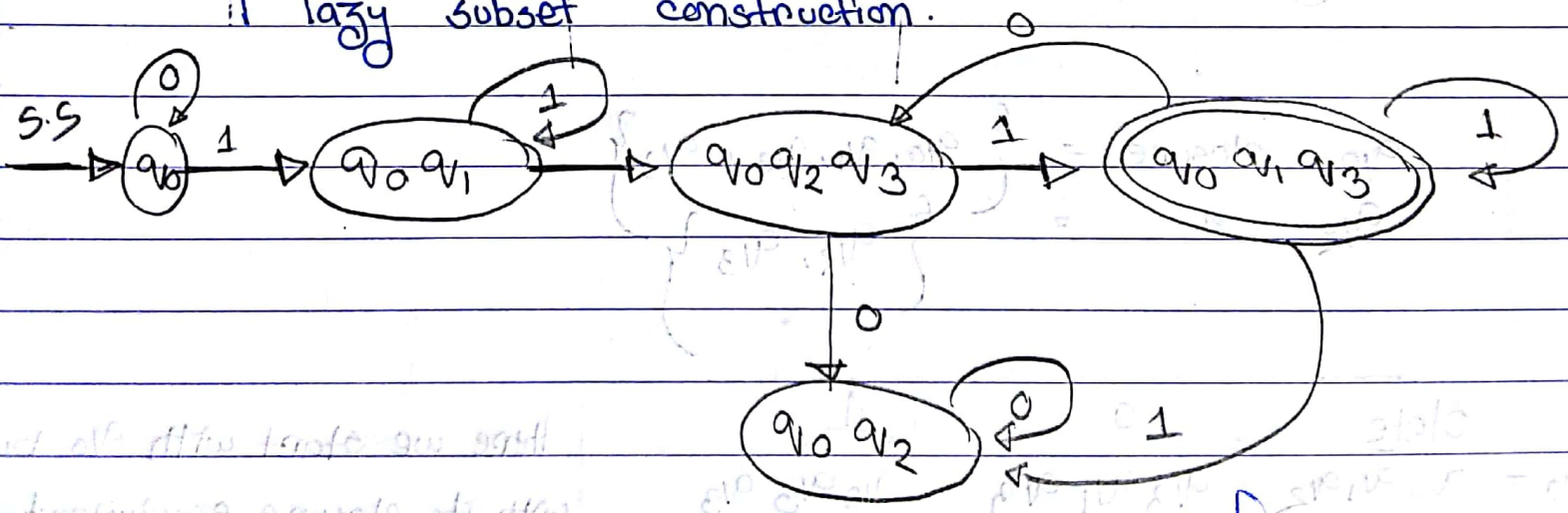
Another trick is that when we have multiple states in NFA then 2^n DFA becomes really big so we start with the starting node q_0 then if any new state appears in either 1 or 0 we use that to find the co-responding 1 & 0. So on -- untill there isn't any new to form.

State	0	1
→ q_0	q_0	$q_0 q_1$
$q_0 q_1$	$q_0 q_2 q_3$	$q_1 q_0$
* $q_0 q_2 q_3$	$q_0 q_2$	$q_0 q_1 q_3$
$q_0 q_2$	$q_0 q_2$	$q_0 q_1 q_3$
* $q_0 q_1 q_3$	$q_0 q_2 q_3$	$q_0 q_1 q_3$

new state.

Everything is Present.

↑ lazy subset construction.



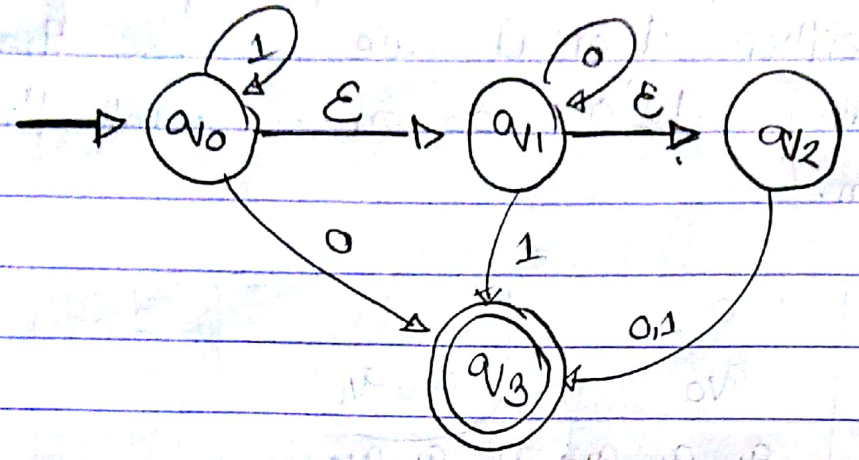
Hamd Finem

Scanned by CamScanner

Paper Source
Subject
Date: Time:

NFA = ε transition
DFA = no ε "

* for ε we don't have to add another column.

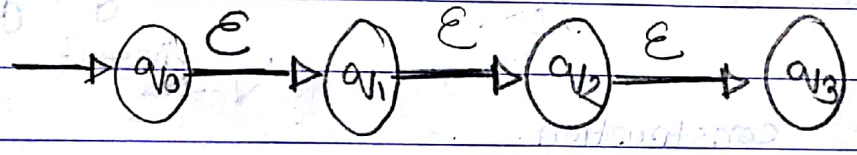


Closures
$q_0 = q_0 q_1 q_2$
$q_1 = q_1 q_2$
$q_2 = q_2$
$q_3 = q_3$

they don't have any so they will be unchanged

ε- closure

All states all- togather that can be reached by ε



q_0 closure = $\{q_0, q_1, q_2, q_3\}$
q_2 " = $\{q_2, q_3\}$

State	0	1
= $q_0 q_1 q_2$	$q_3 q_1 q_3$ = $q_1 q_2 q_3$	$q_0 q_3 q_3$ $q_0 q_1 q_2 q_3$
$q_1 q_2 q_3$	$q_1 q_2 q_3$	q_3
$q_0 q_1 q_2 q_3$	"	$q_0 q_1 q_2 q_3$
q_3	$\emptyset$	$\emptyset$

Here we start with q_0 but with its closure equivilent.
Same goes for the 0 & 1 values

$q_0 \rightarrow q_3 \xrightarrow{\text{closure}} q_3$
$q_1 \rightarrow q_1 \xrightarrow{"} q_1 q_2$
$q_2 \rightarrow q_3 \rightarrow q_3$

union them,
$q_1 q_2 q_3$

Scanned by CamScanner

Paper Source

Subject.....

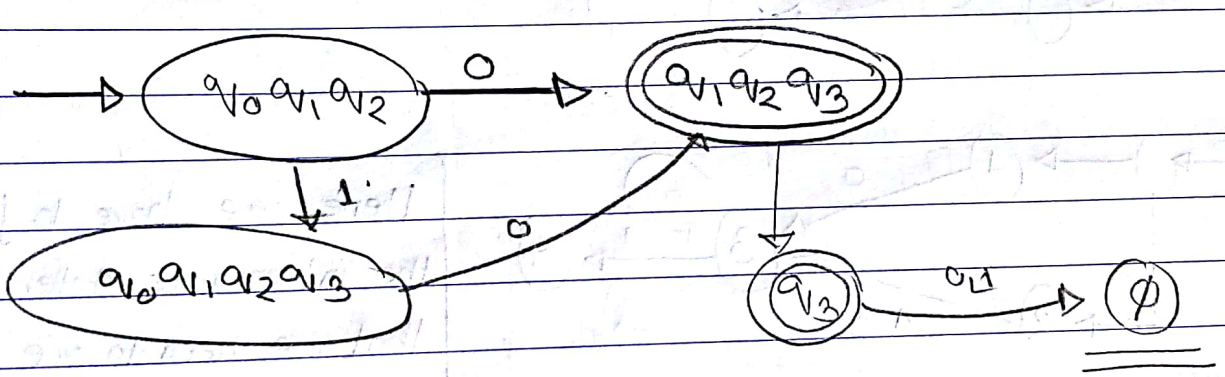
Date:.....Time:.....

$q_1 \xrightarrow{0} q_1 \xrightarrow{c} q_1 q_2$

$q_2 \xrightarrow{0} q_3 \xrightarrow{c} q_3$

$q_3 \xrightarrow{0} \emptyset \xrightarrow{c} \emptyset$

$\left.\begin{matrix} q_1 \\ q_2 \\ q_3 \end{matrix}\right\} \xrightarrow{1} \begin{matrix} q_3 \xrightarrow{c} q_3 \\ q_3 \quad " \quad q_3 \\ \emptyset \quad " \quad \emptyset \end{matrix}$

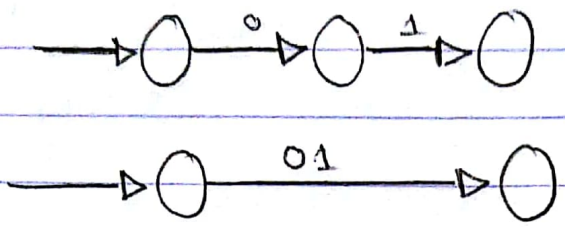


dead end.

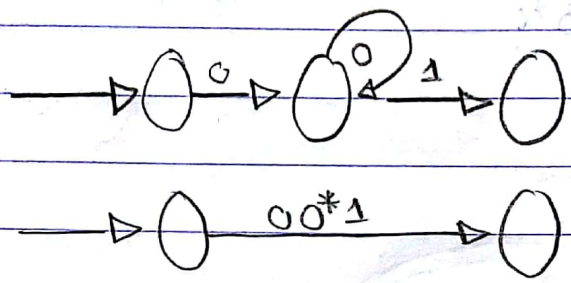
Mau Ponan

Scanned by CamScanner

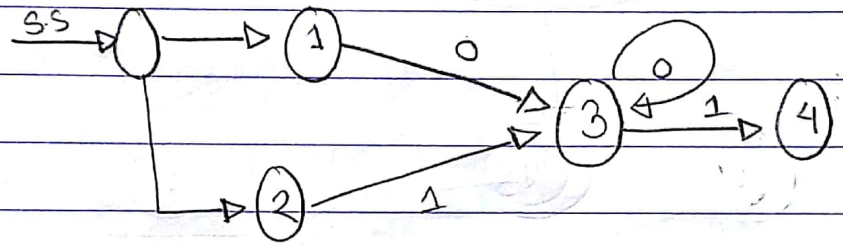
State elemination



if the states are in this pattern than we can eliminate the intermidiate state and put those values.



self loops can be written as 0^*



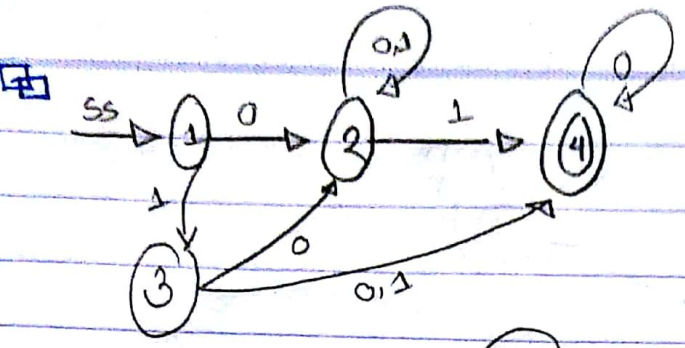
Here we have to find the intermidiate state. To identify that we need to see which node has the highest indegrees & outdegrees. Here (3)

Scanned by CamScanner

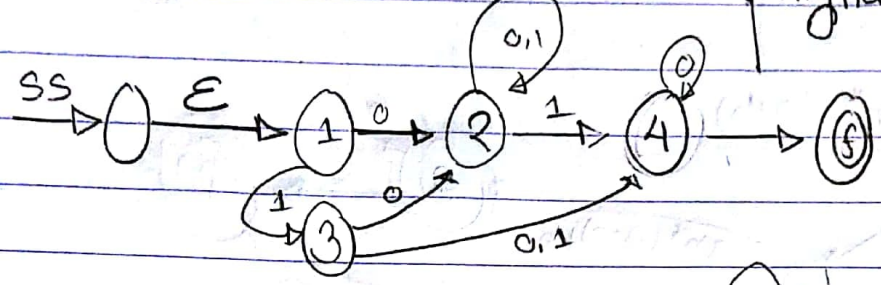
Paper Source

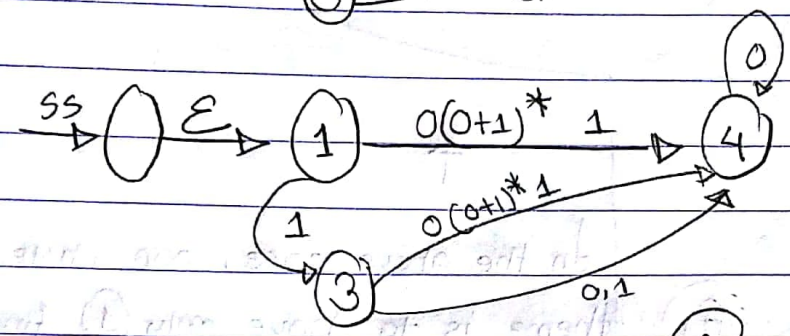
Subject

Date: Time:

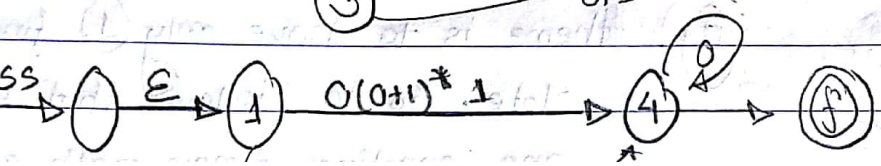


for these cases it's recommended to write a new start state & final state. Start with the node having the highest degrees. (self loops don't count)



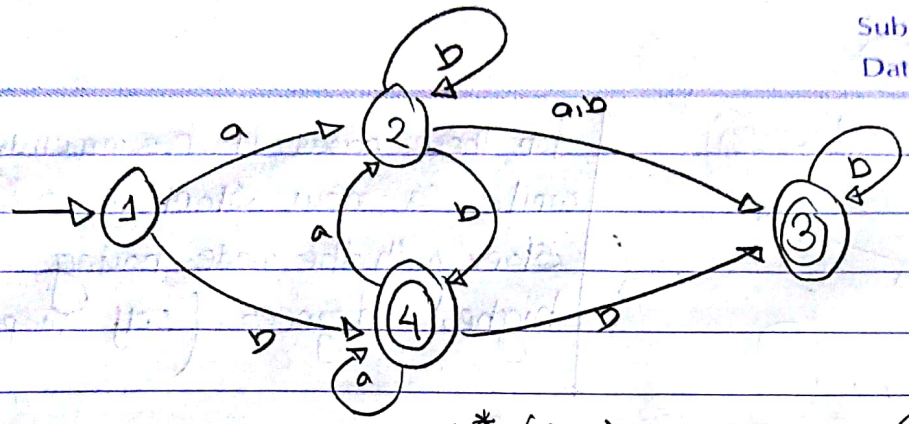


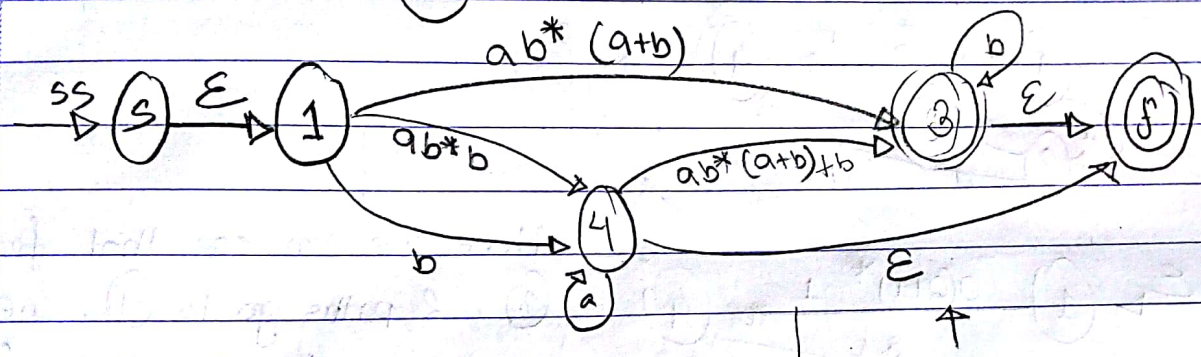
Here we can see that from (3), 2 paths go to (4). we can merge them together. (union)

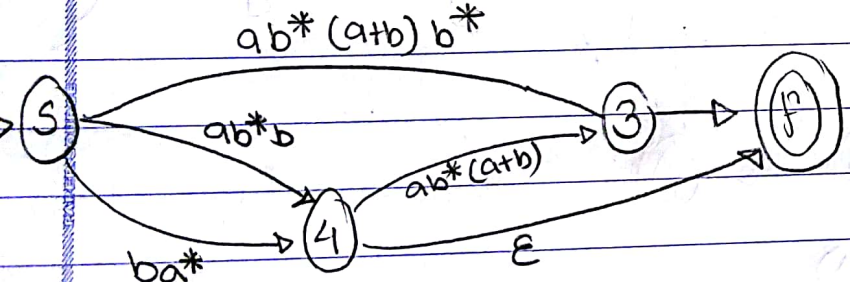


Scanned by CamScanner

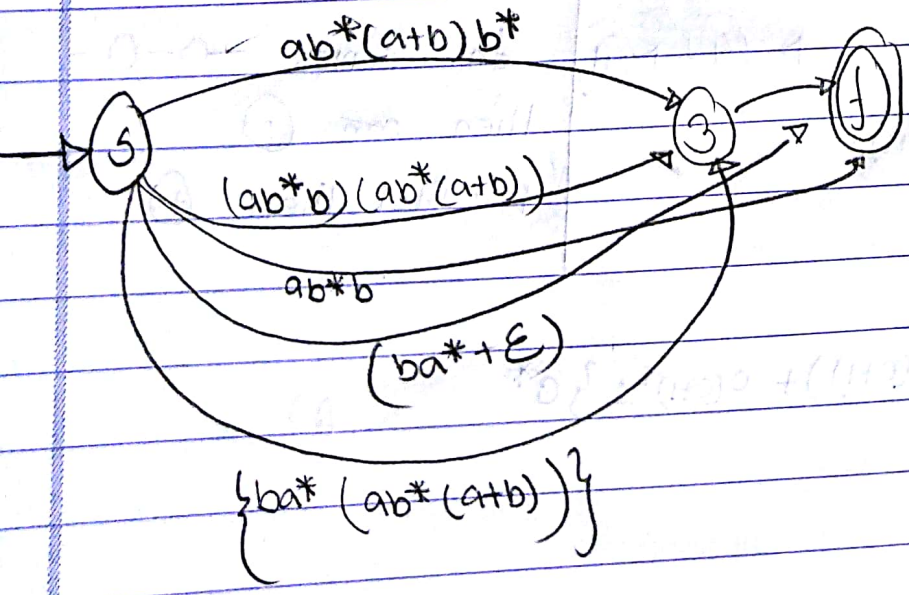
Paper Source
Subject
Date: Time:





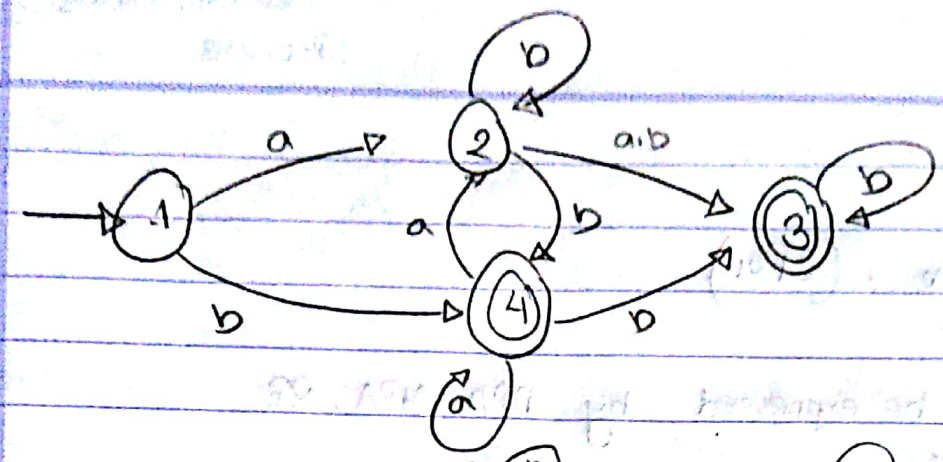


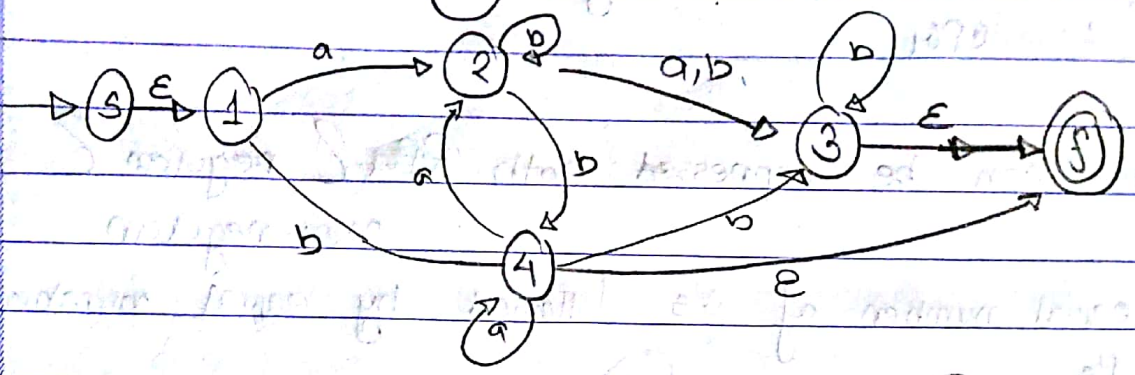
In the above case, our whole theme is to have only (1) final state. So to follow that we are creating a new path with (ε). which doesn't change anything.

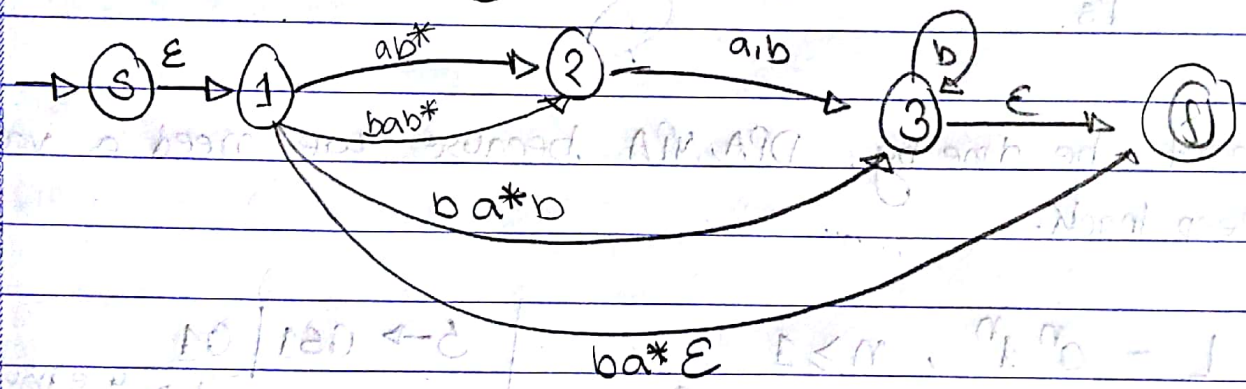


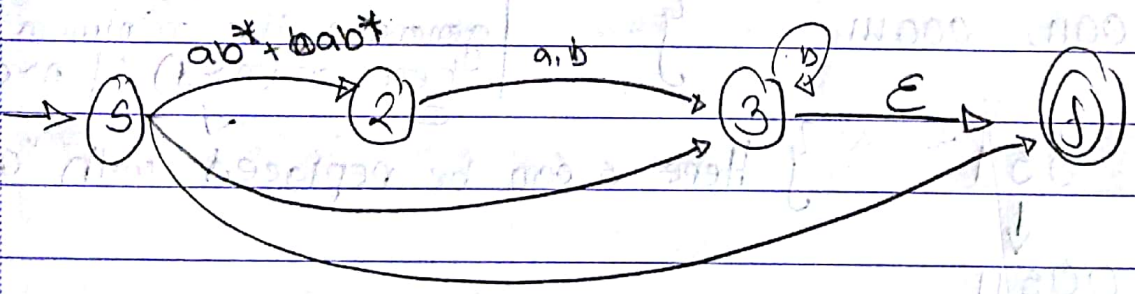
Scanned by CamScanner

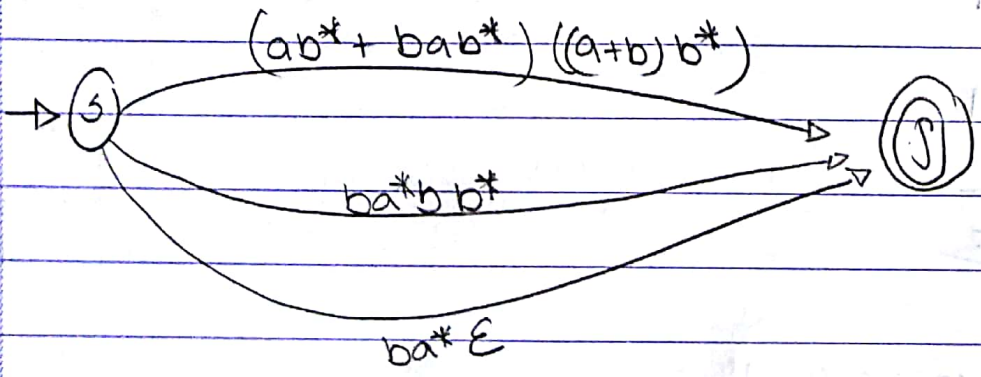
Paper Source
Subject
Date: Time:











(5) bb* ... *)((a+b)b*) + ba*bb*

Scanned by CamScanner

Paper Source
Subject: CSE 331, Lec-08
Date: 6 Feb, 2018 Time:

Chapter-5

Context free grammar (CFG)

Regular language : Can be expressed by DFA, NFA, RE
non-" " : CFG.

CFG can be expressed both ~~RE~~ & regular & non-regular.

language → equal number of 0's followed by equal number of 1's.

→ This can't be done by DFA, NFA because we need a variable to ~~cat~~ keep track.

Ans:

$L - 0^n 1^n, \; n \geq 1$

$\{01, 0011, 000111 \ldots\}$

$S \to 0S1 \mid \underline{01}$ → we have to generate the minimum value here. as $(n \geq 1)$ if $n \geq 0$ then ε.

$S \to 0S1$ } Here S can be replaced with "0S1".

$\downarrow$

$00S11$

$\downarrow$

$000S111$

$\downarrow$

until $S = \varepsilon$

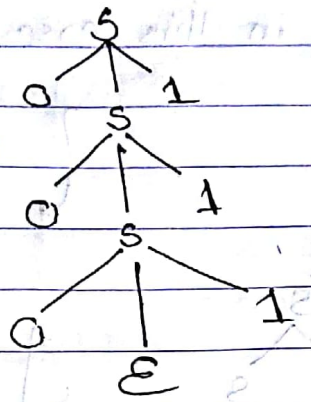
now if $L = 0^n 1^n, \; n \geq 0$ then

Scanned by CamScanner

Paper Source
Subject
Date: Time:

Parse tree →

input 000111



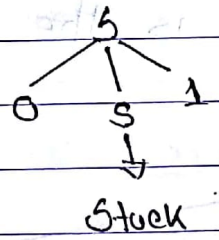
~~0~~ 00 11 ~~1~~

~~0~~ ~~0~~ 0 1 ~~1~~ ~~1~~

~~0~~ ~~0~~ ~~0~~ ~~1~~ ~~1~~ ~~1~~

parse will be valid when the leaves have either "ε" or the alphabet symbol: {0, 1}

input : 0101

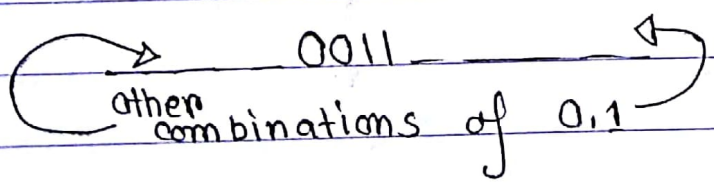


~~0~~ 1 0 ~~1~~

L- $0^n 1^{2n}$, $n \geq 0$

S → 0S11 | ε

L- contain substring "0011"



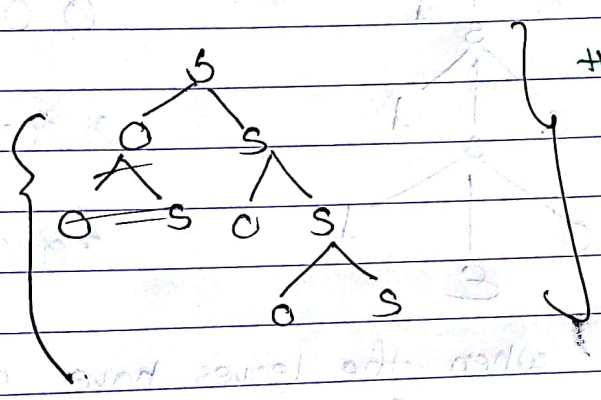
Scanned by CamScanner

Paper Source
Subject.....
Date:.....Time:

$S \to 0011S \mid 0S \mid 1S \mid \varepsilon$

We cant do this because in this case "000" is valid.

Works

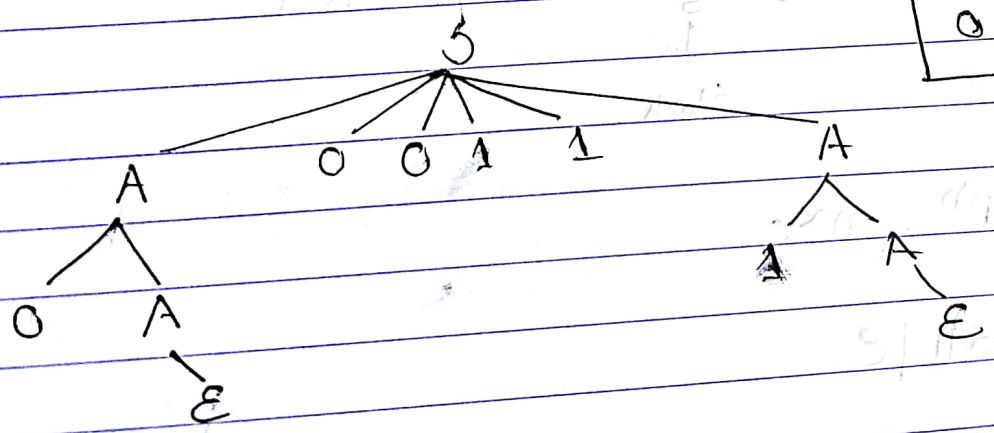


thats why this is wrong!

$S \to A\ 0011\ A$
$A \to 0A \mid 1A \mid \varepsilon$

← A substitute all possible combination with {0,1}.
This is the Ans. No parse tree.

$\boxed{0\ 0011\ 1}$



Scanned by CamScanner

Paper Source
Subject
Date: Time:

$\Sigma = \{0,1\}$, L = contains at least 3 1's

111 10101

$S \rightarrow A1A1A1A$
$A \rightarrow 0A \mid \epsilon \mid 1A$

exactly 3 1's

$S \rightarrow A1A1A1A$
$A \rightarrow 0A \mid \epsilon$

At most 3 1's

$S \rightarrow A1A \mid A1A1A \mid A1A1A1A$
$A \rightarrow 0A \mid \epsilon$

Length → even

$S \rightarrow$ ~~01A~~ | ~~10A~~ $01S \mid 10S \mid 00S \mid 11S \mid \epsilon$ ← all possible combo in front.

L- begin with "110"

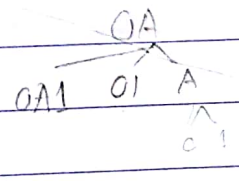
$S \rightarrow 110A$
$A \rightarrow 0A \mid 1A \mid \epsilon$

L- ends with "110"

$S \rightarrow A110$
$A \rightarrow 0A \mid 1A$

$L = 0^m 1^n, m > n$

001
0001 ✗



$S \rightarrow 0A$
$A \rightarrow 01 \mid \epsilon \mid 0A1$

Scanned by CamScanner

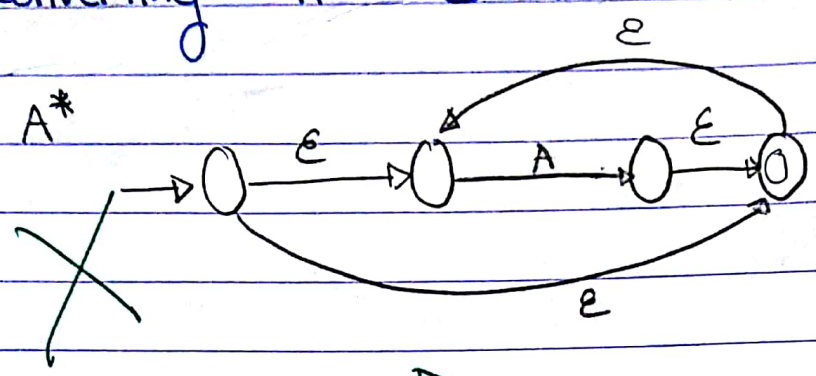
Paper Source

Subject.....

Date:..... Time:.....

Converting RE → ε-NFA

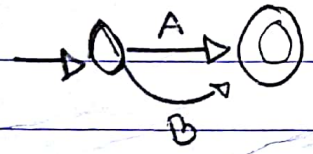
A^*



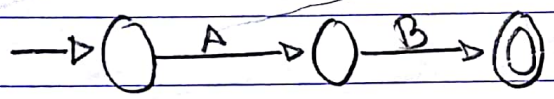
This is how to write for closure. There will always be 4 states. the middle one is going to represent the alpha bet. first & last one will have ε transitions.

→ next page

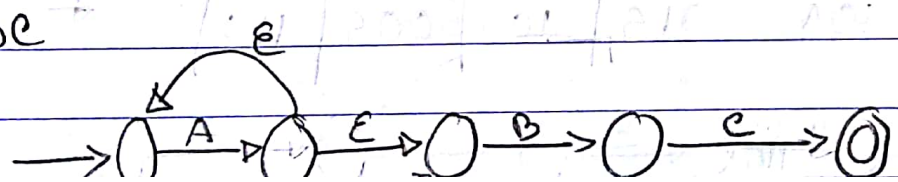
$(A+B)$



$A.B$



A^*BC



final Answer

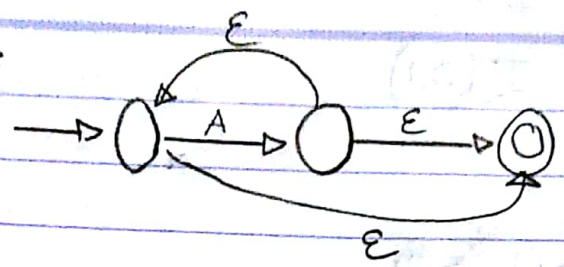
Scanned by CamScanner

Paper Source

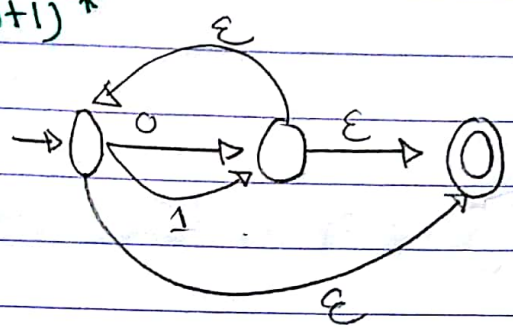
Subject

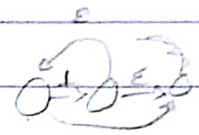
Date: Time:

A^*

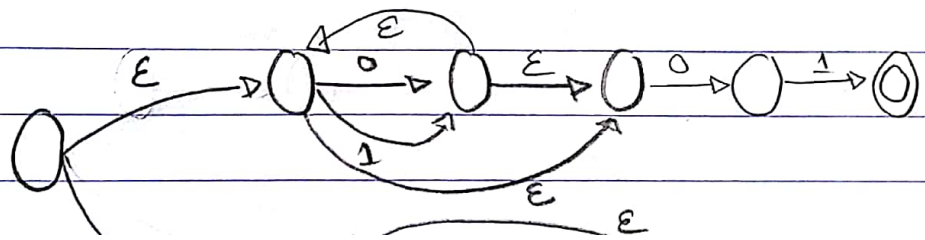


$(0+1)^*$





$(0+1)^*01 + (10+1)^*$



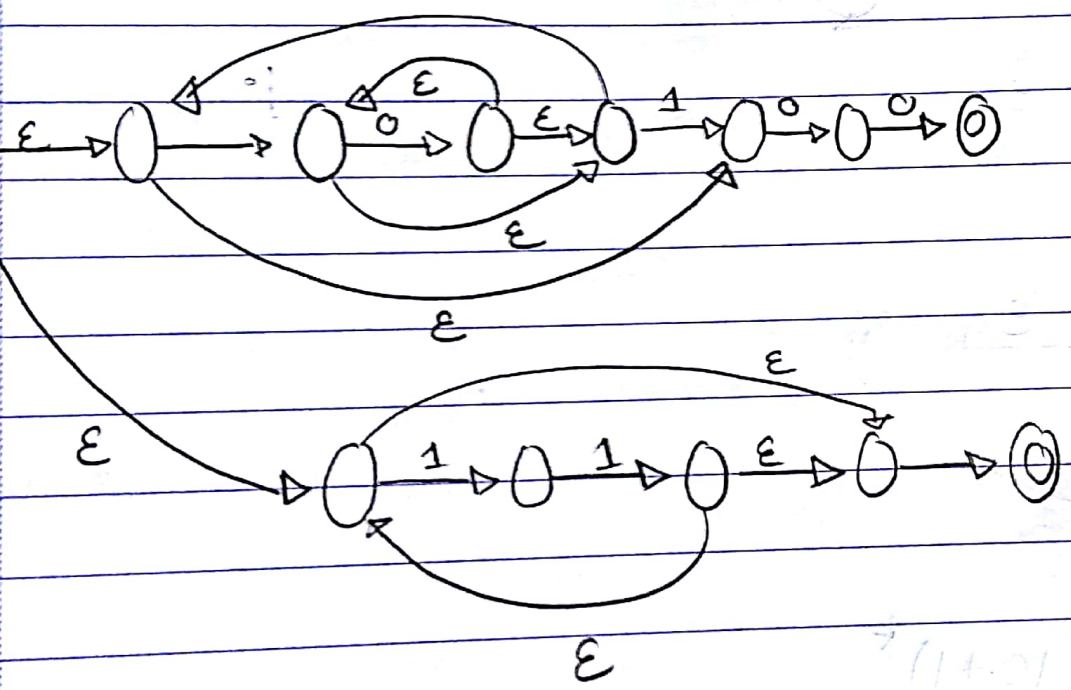
Scanned by CamScanner

Paper Source

Subject.....

Date:.....Time:

$(0^*1)^* 00 + (11)^*$ $\Sigma(0,1)$



Hanfonem

Paper Source
Subject: CSE 331 Lec-9
Date: 11 Feb, 2018 Time:

$L = 0^m 1^n \quad m < n$

$S = A1$

$A = 0A1 \mid \varepsilon \mid A1$

$L = 0^m 1^n \quad m \neq n$

$S = 0A$

$A = 0A \mid \varepsilon \mid 0A1$

$T = B1$

$B = 0B1 \mid \varepsilon \mid 0A1$

$R = S \mid T$

Scanned by CamScanner

Paper Source
Subject
Date: Time:

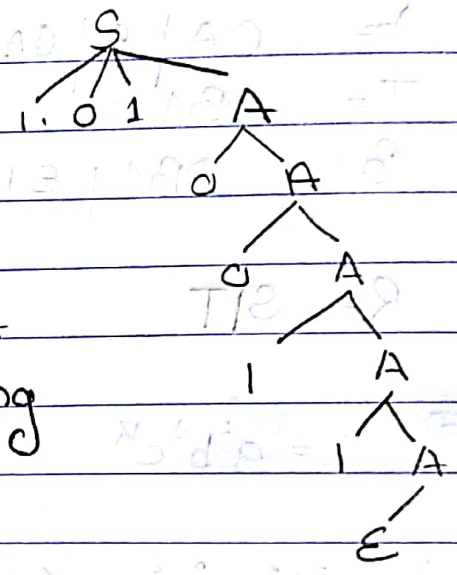
There are a few ways to do CFG

1. Parse tree
2. left most / ~~last~~ left deviation
3. Right " / Righ "

L - begin with "101"

$S = 101A$
$A = 0A \mid 1A \mid \varepsilon$

→ parse tree for 101001



$S \to 101A$
$\to 1010A$
$\to 10100A$
$\to 101001A$
$\to 1010011A$
$\to 1010011\varepsilon \to 1010011$

Keeping the left part & changing the right (A).

in this case $\boxed{101A}$ A is both left & Right. so left most / Right most is the same.

input 10101011

S → A 101 A
→ 1A 101 A

← working with left side first

Paper Source
Subject:
Date: Time:

Right derivation:

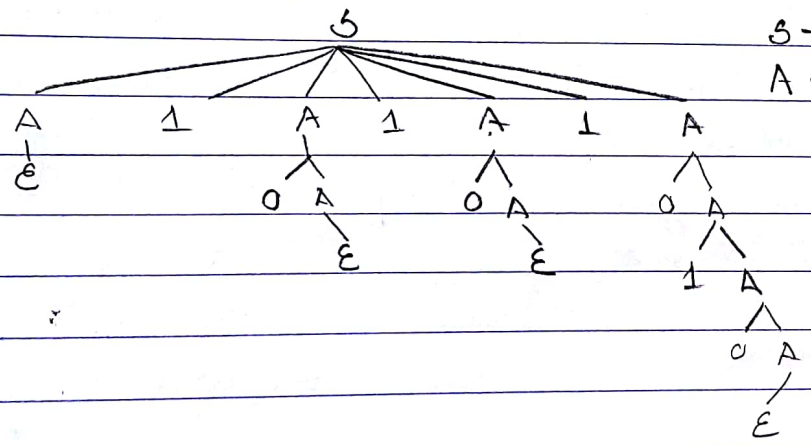
10101001 1

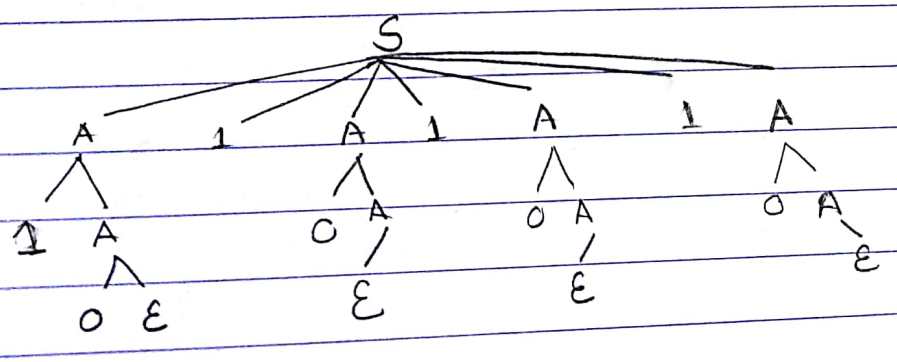
$S \to A101A$ ← Working with right
$\to A1010A$
$\to A10100A$
$\to A101001A$
$\to A1010011A$
$\to A10100110$
$\to \ldots \varepsilon 10101001 1$ ← only A. So no left & Right.
$\to 10101001 1$

Multiple parse tree is called, Ambigous grammar.
Example, input ~~10101010~~ (L contains atleast 3 1's)

$S \to A1A1A1A$
$A \to 0A \mid \varepsilon \mid 1A$

→ one type.





can be done using multiple parse tree so this language is Ambigous.

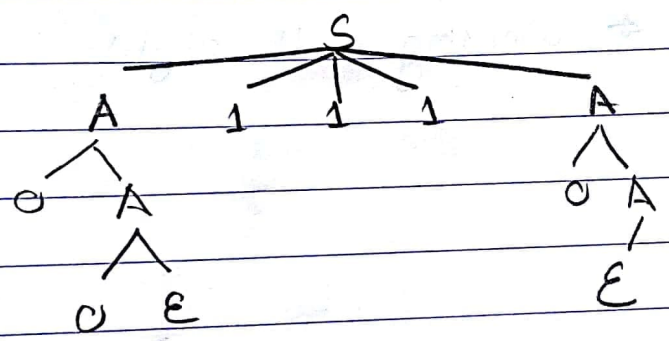
Scanned by CamScanner

Paper Source

Subject.....

Date:.....Time:.....

However, 001110 is not Ambigous.



This input doesn't have any other parse tree. So not ambigous.

Hem Honem

Scanned by CamScanner

Paper Source
Subject: CSE 331 Lec-10
Date: 20 Feb, 2018 Time:

Mid Question:

n occurances of a (n mod 4 is 3)

n = 3, 7, 11, 15 .... These are the values we would get.

~~more~~

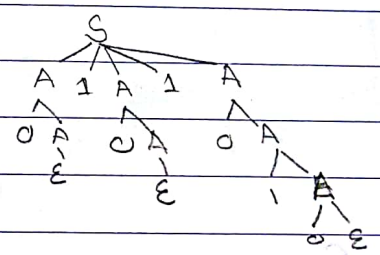
CFG

Ambigous grammar → We can create 2 parse tree for the same input.

If we get more than 2 parse tree showing 2 ~~one~~ will be enough.

$S \to A1A1A$ Input: 01010101

$A \to 0A \mid 1A \mid \varepsilon$



So ambigous!

this can be written an another way. Where the longer branch is on the left. (2 trees)

Removing ambuigity:

1. New variable
2. Variable position update
3. Variable remove

Scanned by CamScanner

Paper Source
Subject
Date: Time:

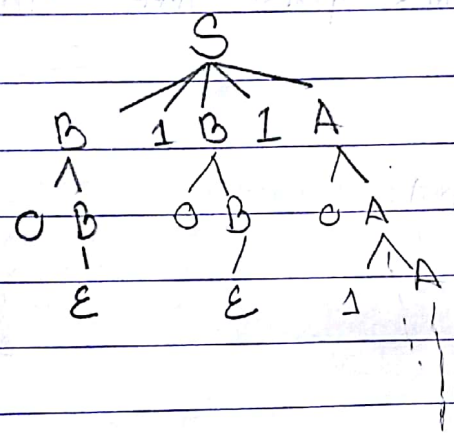
Introducing a new variable,

$S \to B1B1A$
$B \to 0B \mid \varepsilon$
$A \to 0A \mid 1A \mid \varepsilon$

Here we are doing left ordering / associativity
← Unambiguous Grammar.

$\underbrace{1}_{B}\underbrace{0}_{B}\underbrace{1\,0\,1\,0}_{A}$

Here there will always be one parse tree.



the right side will always have the taller branch.

Right ordering,

$S \to A1B1B$
$B \to 0B \mid \varepsilon$
$A \to 0A \mid 1A \mid \varepsilon$

~~Right~~ left side will always have a taller branch.

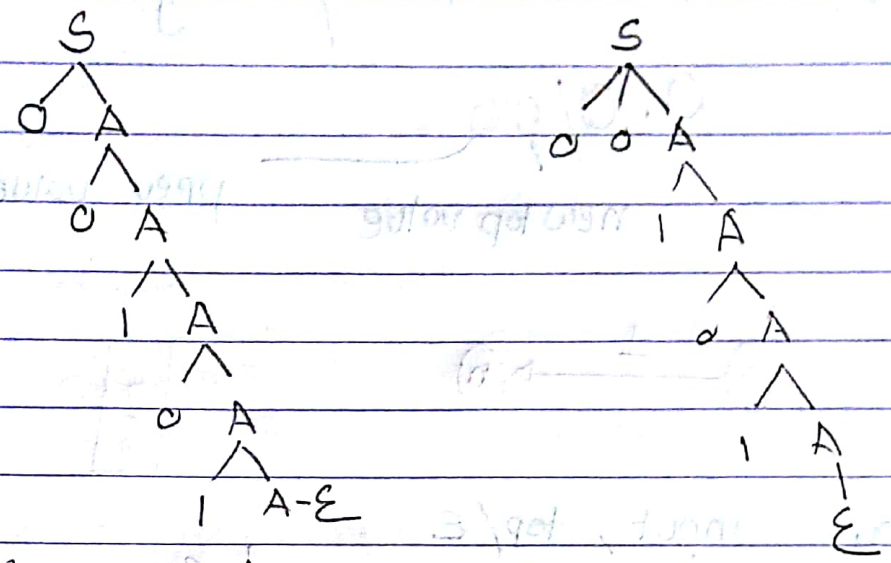
$\underbrace{1\,0\,1\,0}_{A}\underbrace{1\,0}_{B}\underbrace{1\,0}_{B}$

Scanned by CamScanner

Paper Source
Subject
Date: Time:

$S \to 0A \mid 00A \mid \varepsilon$
$A \to 1A \mid 0A \mid \varepsilon$

00101



Ambigous. To know if a grammar is ambigous or not we have to take an input satisfying the rules. Then check.

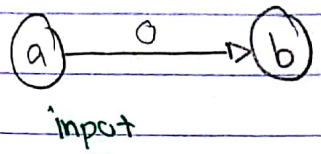
Because of, 0A & 00A, 00 can be expressed in 2 ways.

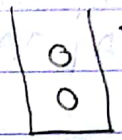
Chapter 6

Push down automata. (PDA)

Scanned by CamScanner

Push





already 2 in stack.

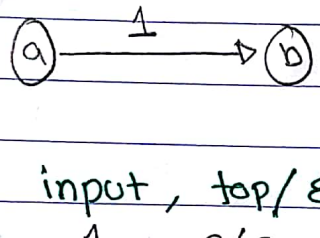
formula, ~~0~~, stack top value / change in the top value.

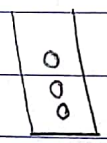
0, 0/00

new top value

prev value.

Pop:





formula, input, top/ε

1, 0/ε

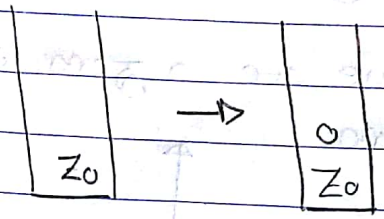
ε because when we pop something it is replaced with ε.

Scanned by CamScanner

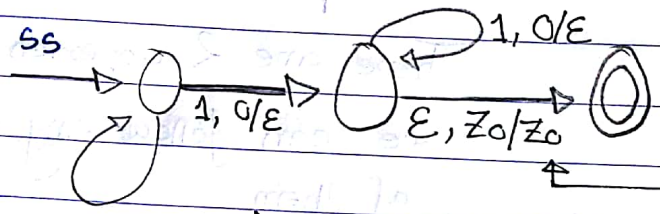
Paper Source
Subject.....
Date:.....Time:.....

* Always start from left *

$Z_0 \rightarrow$ bottom of the stack.



$0, Z_0 / 0, Z_0$



nothing to push/pop

$0, Z_0 / 0Z_0$
$0, 0/00$

we can give multiple condition.

Scanned by CamScanner

Paper Source
Subject CSE 331 Lec-11
Date: 25 feb, 2018 Time:

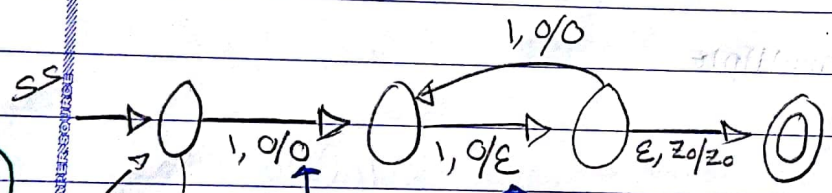
$L - 0^n 1^{2n}$ $\quad n \geq 1$

(1) we need 2 ones to pop one zero.
001111

(2) we can put 2 zeros when we see a zero, so each one with remove each zero.

There are 2 approach here. we can follow any one of them.

(1)

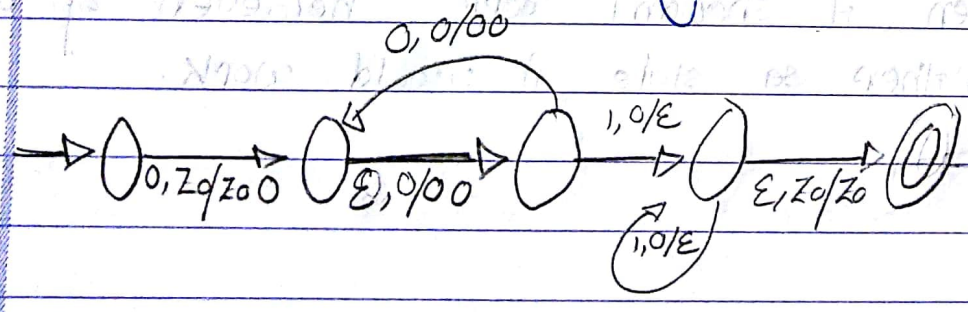


0, Z0/0 Z0
0, 0/00

entering all zeros

if we find another one then pop.

if we find one then no change



Scanned by CamScanner

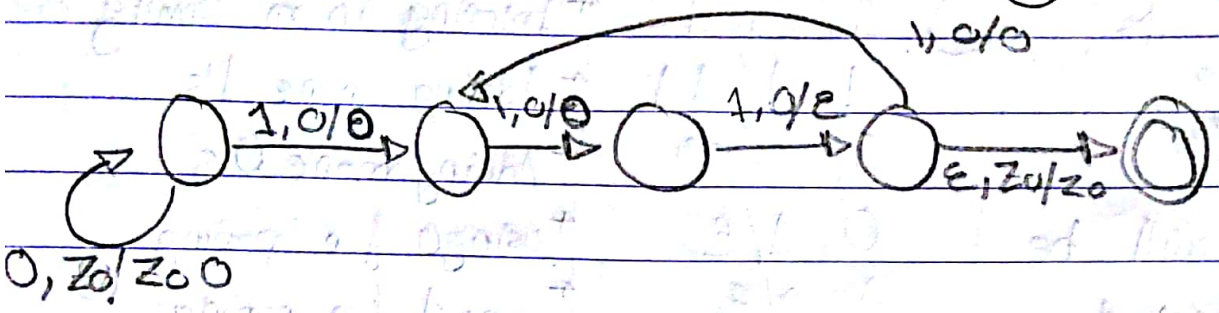
Paper Source

Subject.....

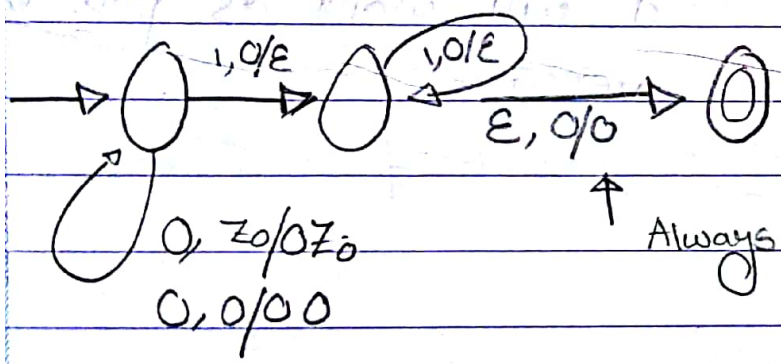
Date:..... Time:.....

$L - 0^{2n} 1^{3n}$

for 2 zeros we have to push 3 zeros.



$L - 0^{m} 1^{n}$ $m > n$ | there will always be '0's' left.



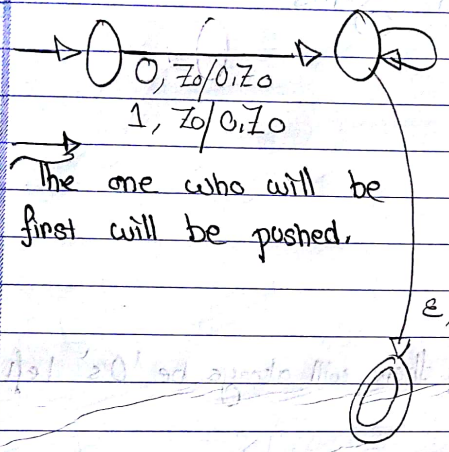
Always zero left.

we can't write, $L - a^n b^n c^n d^n$ because bcd all of them depend on a's value. However, $L - a^n b^m c^m d^n$ this

Scanned by CamScanner

Paper Source
Subject.....
Date:.....Time:.....

L- equal number of 0 & 1s. * there isn't sequence *



The one who will be first will be pushed.

0, Z0/0,Z0	Entering in an empty stack for 0
1, Z0/1Z0	Entering in an empty stack. a 1.
1, 1/11	Adding more 1's
0, 0/00	Adding more 0's
0, 1/ε	using 0 for poping
1, 0/ε	using 1 for poping

all of them are "OR"

If we enter 0 first then 1 will work as pop. vice versa.

Ex: 0110

here
0 push
1 pop
1 push
0 pop
ε, Z0/Z0

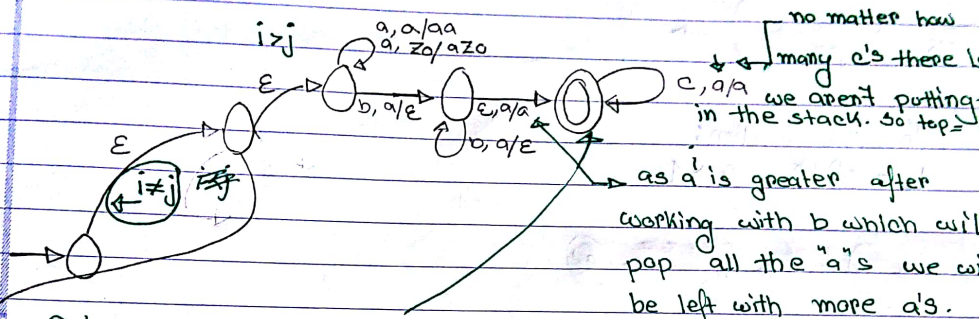
Paper Source
Subject: CSR 330 Lec-12
Date: 27 Feb, 2018 Time:

PDA

~~transition function~~:

$L = a^i b^j c^k$, $i \neq j$ or $j = k$

So, here we have to consider both situations,

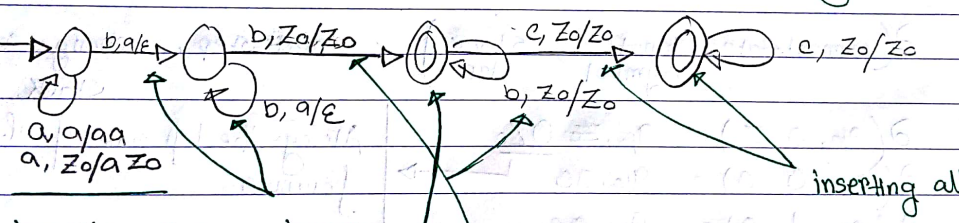


no matter how many c's there is we aren't putting them in the stack. So top = a.

as a's is greater after working with b which will pop all the "a"s we will be left with more a's.

This is the final state because there might be a situation where there isn't any c.

i<j



inserting all a

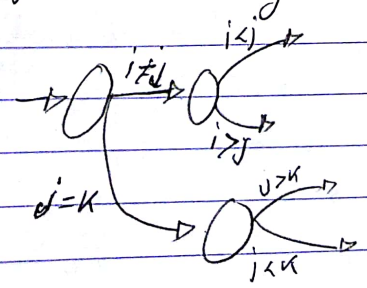
~~inserting~~ popping all a's

inserting all the b's

this could be final state because no "c" can occur.

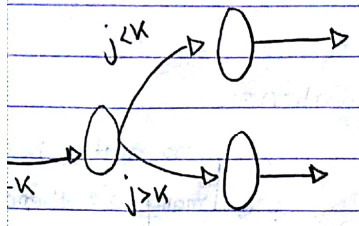
inserting all "c"s

Simplified (without writting) ↓



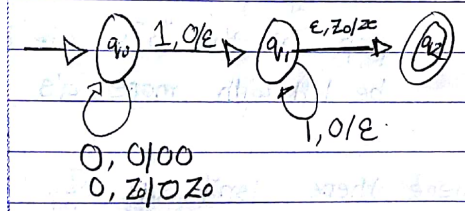
$a^i b^j c^k$

Paper Source
Subject
Date: Time:



Transition function:

$L = 0^n 1^n, \; n \geq 1$



∂(current state, input symbol, stack top) = next step, change in stack.

$\partial(q_0, 0, Z_0) = q_0, \cancel{Z_0}\ 0Z_0$
$\partial(q_0, 0, 0) = q_0, 00$
$\partial(q_0, 1, 0) = q_1, \cancel{0}\varepsilon$
$\partial(q_1, 1, 0) = q_1, \varepsilon$
$\partial(q_1, \varepsilon, Z_0) = q_2, Z_0$

Always the left side of the formula.

- If the PDA ε input value is given use instantaneous description or ID.

Scanned by CamScanner

Paper Source
Subject
Date: Time:

Input : 000111

ID:

State, input, everything inside the stack

q_0, 000111, Z_0	q_1, ε, Z_0
q_0, 00111, $0Z_0$	q_2, ε, Z_0
q_0, 0111, $00Z_0$	
q_0, 111, $000Z_0$	
q_1, 11, $00Z_0$	
q_1, 1, $0Z_0$	

This will also work for wrong inputs, (0101)

q_0, 0101, Z_0
q_0, 101, $0Z_0$
q_1, 01, Z_0
stuck ← Because there isn't any rule in q_1 that tells us what to do if we find 0 again.

Conversion: CFG → PDA

-Hamid Reza...

$E \to E+E \mid E*E \mid I$
$I \to Ia \mid Ib \mid a \mid b$

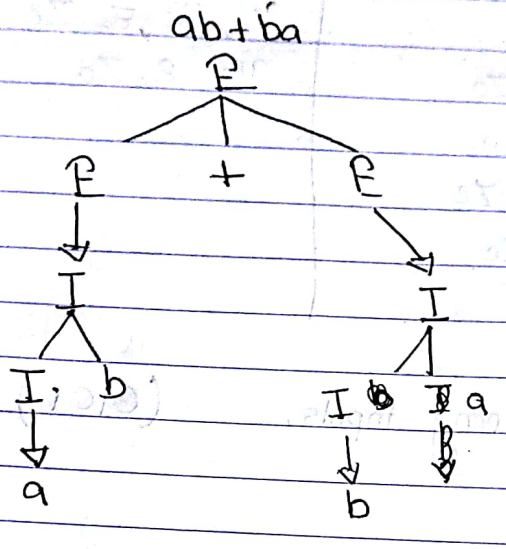
Start variable : E
$\Sigma = \{+, *, a, b\}$

Scanned by CamScanner

Paper Source
Subject
Date: Time:

input : ab + ba

for the conversion the stack has to be empty!



ID:

q_0, ab+ba, EZ_0
" , " , $E+E.Z_0$
" , " , $I+E, Z_0$
" , " , $Ib+E, Z_0$
" , ab+ba, $ab+E, Z_0$
" , b+ba, $b+E, Z_0$
" , +ba, $+EZ_0$
" , ba, EZ_0
" , ba, IZ_0
" , " , $Ia\ Z_0$
" , ba, $ba\ Z_0$
" , a, aZ_0
q_0, ε, Z_0

Used for checking

b
I
a
I
a
a I
b
I
E
+
E
E ← always
Z_0

Paper Source
Subject
Date: Time:

There are 2 rules,

1. If stack top ~~is~~ is a variabe A then pop A & push a procedure (ex: E+E) of ~~of~~ A.

2. If stack top is a terminal (+, *, a, b) then match with input symbol & pop e.

When converting use transition function ↓

$\partial(q_1, \varepsilon, E) = (q_1, E+E), q_1(E \times E), (q_1, I)$ | Any one of them.
$\partial(q_1, \varepsilon, I) = (q_1, Ia), (q_1, Ib), (q_1, a), (q_1, b)$
$\partial(q_1, a, a) = (q_1, \varepsilon)$ ← pop
$\partial(q_1, b, b) = (q_1, \varepsilon)$ "
$\partial(q_1, +, +) =$ " "
$\partial(q_1, *, *)$ " "

Scanned by CamScanner

Initial state $S \to A1B0 \mid 1B$ $\Sigma = \{0,1\}$ ← these are terminals

$A \to A11 \mid \varepsilon$

$B \to 0B \mid 0$

convert this CFG to PDA & verify your PDA for the input 111000.

$\partial(q_1, \varepsilon, S) = \{(q_1, A1B0), (q_1, B)\}$

$\partial(q_1, \varepsilon, A) = \{(q_1, A11), (q_1, \varepsilon)\}$

$\partial(q_1, \varepsilon, B) = \{(q_1, 0B), (q_1, 0)\}$

$\partial(q_1, 0, 0) = \{q_1, \varepsilon\}$

$\partial(q_1, 1, 1) = \{q_1, \varepsilon\}$

verification:

q_1, 111000, S

q_1, 111000, A1B0

q_1, 111000, A111B0 ← substitute the appropriate role.

q_1, ~~1~~11000, ~~1~~11B0

q_1, ~~1~~100, ~~1~~1B0

q_1, ~~1~~00, ~~1~~B0

q_1, 00, B0

q_1, ~~0~~0, ~~0~~00

q_1, ~~0~~, ~~0~~

q_1, ε, Z_0 ← empty stack.

Scanned by CamScanner

Paper Source
Subject
Date: Time:

Chapter 7

Chomsky normal form (CNF)

cnf → standard/normal form

Rule:

1. only 2 variables on the right side
2. " 1 terminal " " " "

CFG → CNF conversion rules

1. ε- production remove
2. unit " "
3. useless symbol "
4. convert to CNF.

ε- production remove

$S \to AB$
$A \to aAA \mid \varepsilon$
$B \to bBb \mid \varepsilon$

Here both A & B go to ε. and as S is related with A & B so S also goes to ε. so,

$S \to \varepsilon$
A "
B "

→ they are called nullable variable.

Scanned by CamScanner

Paper Source
Subject
Date: Time:

for S,

$S \to AB$

	A	B	
$\varepsilon \to$	✓	✓	AB
	✗	✓	εB
	✓	✗	Aε
	✗	✗	ε

so, $S \to AB \mid A \mid B$

for A,

$A \to aAA$

	A	A	
	✓	✓	A
	✗	✗	ε
	✓	✗	Aε
	✗	✓	εA

$A \to aAA \mid aA\varepsilon \mid a\varepsilon A \mid a\varepsilon\varepsilon$

$\to aAA \mid aA \mid a$

for B, $B \to bBb \mid \varepsilon$

~~$B \to$~~

$B \to bBb \mid bb$

B	
✓	B
✗	ε

Paper Source
Subject
Date: Time:

(2) unit production remove

Neso Academy

Ex:

$S \to A$
$E \to T$
$I \to A$

Keeping only the ~~a~~ variable

if $S \to A$ only one variable then chang

$S \to AB|A|B$

these are units

put the value of the units

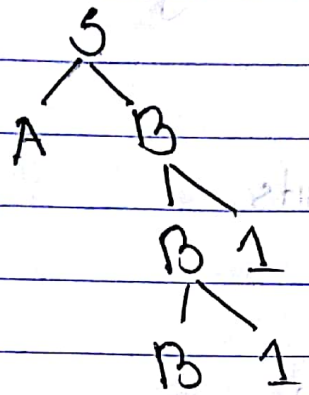
$S \to AB \mid aAA \mid aA \mid a \mid bBb \mid bb$
$A \to aAA \mid aA \mid a$
$B \to bBb \mid bb$

Hamd Ameen

Scanned by CamScanner

CNF → ③ Useless symbol remove

(i) non-generating
(ii) un-reachable



$S \to AB \mid a$
$A \to b$
$B \to B1$
$\Sigma = \{a, b, 1\}$

In the above case, we can't stop B it will never generate ε or any terminal that will stop B from occuring.
So, $B \to B1$ will be removed. & anything that contains B will also be removed.

$S \to a$
$A \to b$

Now we have to look if the variable is accessible from the top / SS.
Here A isn't accessible. So it will be removed.

$S \to a$.

Scanned by CamScanner

Paper Source
Subject
Date: Time:

(4) convert to CNF

$S \to A+C$ $\Sigma \{+, a, b\}$
$C \to a$
$A \to b$

to convert to cnf we have to convert,
variable - terminal
terminal - "
> Variable - variable

to convert to V-V, we have to bring in a new variable

$S \to APC$ ← in cnf we can't have ~~to~~ 3 vars.
$P \to +$
$C \to a$
$A \to b$

$\Downarrow$

$S \to QC$
$Q \to AP$
$P \to +$
$C \to a$
$A \to b$

} sequence doesn't matter.

Scanned by CamScanner

Subject.....
Date:.....Time:.....

* book excercise *

$S \to AB \mid CA$
$A \to a$
$B \to BC \mid AB$
$C \to aB \mid b$

① ε- production remove:
There isn't any nullable variable.

$S \to AB \mid CA$
$A \to a$
$B \to BC \mid AB$
$C \to aB \mid b$

} If there isn't any change we have write the same thing again.

② unit - production remove:
stay same.

③ Useless symbol remove:
B is non generating.

$S \to CA$
$A \to a$
$C \to b$

④ convert to CNF:
nothing changes

$S \to CA$
$A \to a$
$C \to b$

Scanned by CamScanner

Paper Source
Subject
Date: Time:

$S \to ASB \mid \varepsilon$
$A \to aAS \mid a$ $\Sigma = \{a, b\}$
$B \to SbS \mid A \mid bb$

SbS
✓ ✓ SbS
✗ ✓ bS
✓ ✗ Sb
✗ ✗ b

① ε - production remove.
nullable variable, S
$S \to ASB \mid AB$
$A \to aAS \mid aA \mid a$
$B \to b \mid Sb \mid bS \mid A \mid bb \mid SbS$

② unit production Remove,
S - - - -
A - - - -
$B \to b \mid Sb \mid bS \mid aAS \mid aA \mid a \mid bb$

③ useless symbol remove:
stays the same.

④ Converting to CNF

$S \to ASB \mid AB$
$A \to QAS \mid QA \mid a$
$Q \to a$
$B \to b \mid SP \mid PS \mid QAS \mid QA \mid a \mid PP$
$P \to b$

we can keep only b, a valid in CNF.

$\Downarrow$

$S \to XB \mid AB$
$X \to AS$
$A \to QX \mid QA \mid a$
$Q \to a$
$B \to b \mid SP \mid PS \mid QX \mid QA \mid a \mid PP$

Scanned by CamScanner

$S \to aAa \mid bBb \mid \varepsilon$

$A \to C \mid a$

$B \to C \mid b$

$C \to CDE \mid \varepsilon$

$D \to A \mid B \mid ab$

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

(1)

$S \to aAa \mid aa \mid bb \mid bBb$ nullable, S, A, B, C, D

$A \to C \mid a$

$B \to C \mid b$

$C \to CDE \mid DE \mid CE \mid E$

$D \to A \mid B \mid ab$

(2)

S — — —

$A = CDE \mid DE \mid CE \mid E \mid a$ marking because E has a problem

$B = u - u \quad u \mid b$

$C = u \quad u \quad u$

$D \to CDE \mid DE \mid CE \mid E \mid ab$

(3)

C non-generating because there isn't any terminal & E shouldn't exist

$S \to$ — —

$A \to a$

$B \to b$

$D \to ab$

(4)

$S \to QAQ \mid QQ \mid PP \mid QBQ$

$P \to b$

$Q \to a$

A — , B —

$D \to QP$

$S \to XQ \mid QQ \mid PP \mid YQ$

$X \to QA$

$Y \to QB$

P —

Q —

A —

D —

Scanned by CamScanner

Paper Source
Subject: CSE 331, Lec-15
Date: 13 March, 18 Time:

CYK algorithm

CFG

$S \to AB \mid BC$
$A \to BA \mid a$
$B \to CC \mid b$
$C \to AB \mid a$

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

← They must be in CNF.

input ω = baaba, is ω accepted by this grammar.

Ans,
1. Parse tree
2. left derivation
3. Right "

we can find the input works or not through these methods. These create the complexity $O(n^k)$

CYK complexity → $O(n^3)$

length of input |baaba| = 5

1 2 3 4 5
b a a b a

X_{15}, X_{17}, X_{23}, X_{35}

baaba X_{15} S,A,C				
baab X_{14} ∅	aaba X_{25} S,A,C			
baa X_{13} ∅	aab X_{24} B	aba X_{35} B		
ba X_{12} A,S	aa X_{23} B	ab X_{34} S,C	ba X_{45} A,S	
b X_{11} B	a X_{22} A,C	a X_{33} A,C	b X_{44} B	a X_{55} A,C
b	a	a	b	a

The trick is based on the subscript start from the first value then to the last value. then find the appropriate VARIABLE for that. X_{11} = b so, variable B

Scanned by CamScanner

Paper Source
Subject:
Date: Time:

$X_{22} = a$, A,C
$X_{33} = a$, A,C
$X_{44} = b$, B
$X_{55} = a$, A,C
$X_{12} = ba$
B A,C | Multiply each variable. BA, BC
A C ~~doesn't exist~~

$X_{13} = baa = b$
$|(1-3)| = 2$ so 2 situations, $ba.a \cup b.aa$
* Do left to right from each column! *

$X_{23} = aa$
A,C ∪ A,C = A,C

$X_{34} = ab$
A,C ∪ B = AB, CB
S,C

$X_{13} = baa$
ba, a ∪ b.aa
AS, AC ∪ B . B
AA, AC, SA, SC, BB

$X_{45} = ba$
B ∪ A,C = BA, BC
A S

$X_{24} = aab$
a.ab ∪ aa.b
A,C . S,C ∪ B B
AS, AC, CS, CC, BB
B

$X_{35} = aba$
a ba ∪ ab : a
(A,C) (A,S) ∪ (S,C) (A,C)
AA, AS, CA, CS, SA, SC, CC
B

Scanned by CamScanner

Paper Source
Subject
Date: Time:

X_{14} = baab

(3)

ba.ab ∪ b.aab ∪ baa.b

(A,S) (S,C) ∪ BB ∪ ∅

⇒ BB, AS, AC, SS, SC

X_{25} = aaba

a.aba ∪ aa.ba ∪ aab.a

(A,C).B ∪ B.(A,S) ∪ B (A,C)

$\underset{S,C}{AB}$, CB, $\underset{A}{BA}$, BS, BA, $\underset{S}{BC}$

X_{15} = baaba

⇒ b.aaba ∪ ba.aba ∪ baa.ba ∪ baab.a

⇒ B.(SAC) ∪ (A,S).B ∪ ∅ ∪ ∅

⇒ BS, $\underset{A}{BA}$, $\underset{S}{BC}$, AB, $\underset{S,C}{SB}$

When there is (5) in the top box (x_{15}) that means the input satisfys.

Abu Hanem

Scanned by CamScanner

Paper Source
Subject CSE 331 Lec-16
Date: 20 March, 18 Time:

Chap-8 : Turing Machine

in turing machine it doesn't matter how many dependents you have you can solve the problem. Where as in DFA, NFA, RE CFG, PDA you can't.

					B	B

tape head. Tape cell

Here we can put terminal, symbol
→ when the cell is empty there is, B.
→ Thead can go left or Right. * while moving it can change the value.

0011 → 0211
change

$L = 0^n 1^n 2^n$ $\Sigma = \{0,1,2\}$

we have to assume an input, 001122

→ size = ∞

B	B	0	0	1	1	2	2	B	B

Thead

→ we can put the value any where in the tape

X 0 1 1 2 2 B B

X 0 1 1 2 2 B B

X 0 y 1 2 2 B B

X 0 y 1 2 2 B B

X 0 y 1 Z 2 B B

After finding 0 we change the value to X and find corresponding 1 and then 2. & replace them with y, z. After find z it will go left.

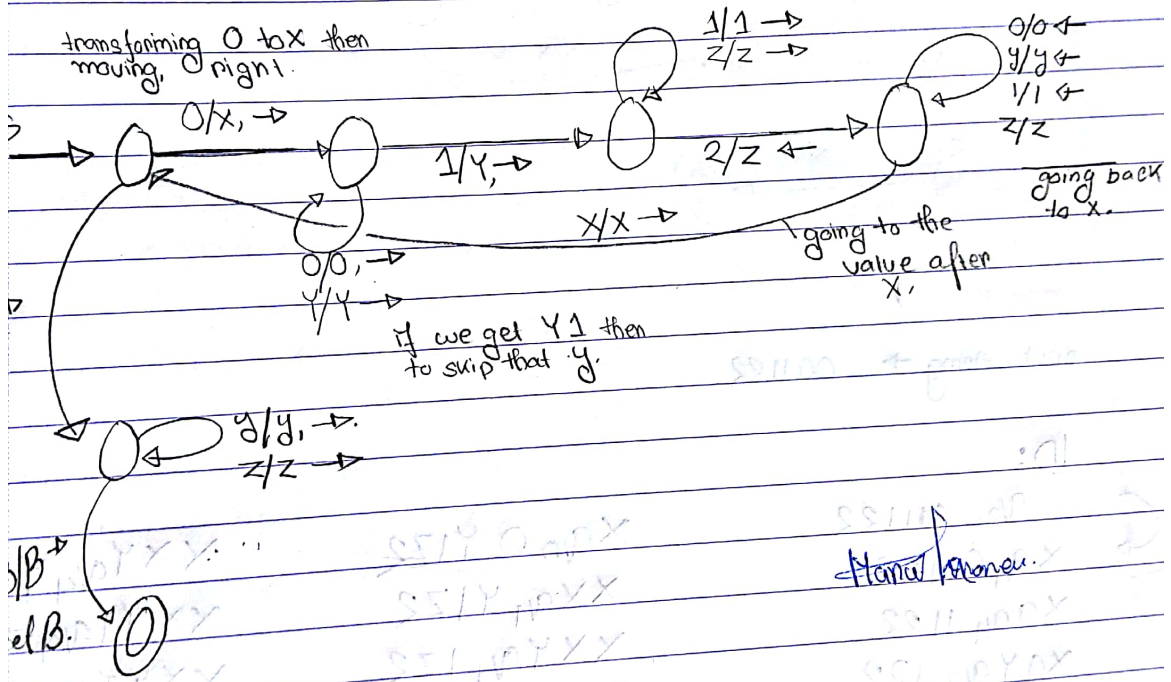
Scanned by CamScanner

Paper Source
Subject
Date: Time:

x x y y z z

Here no zero found then it ~~satisfy the input.~~ looks for B (going night). If B is found then satisfied.

Transition diagram:



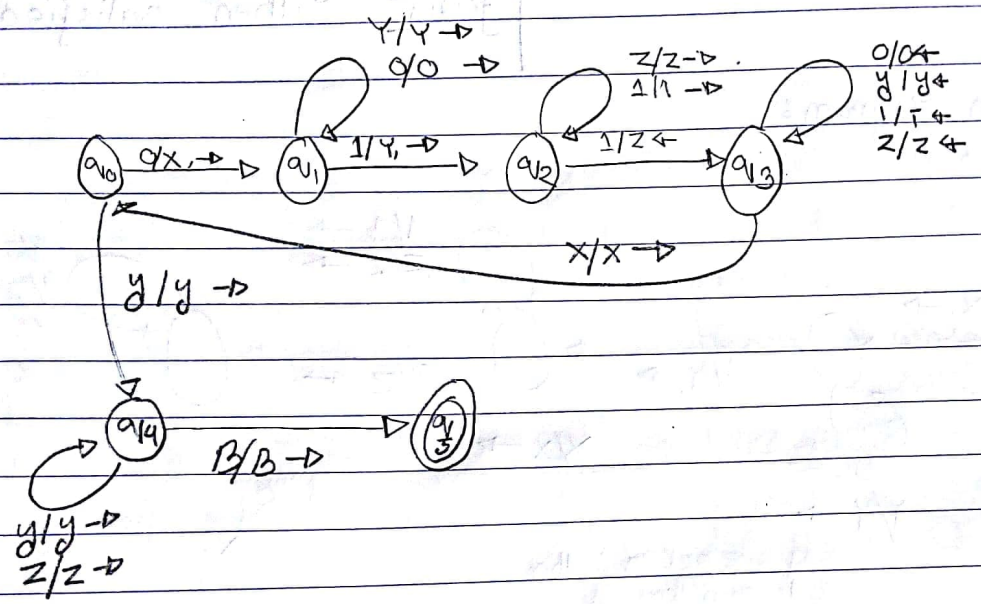
Manu Kumar.

Scanned by CamScanner

Paper Source
Subject: CSE 331, Lec-17
Date: 25 March 18 Time:

Turning machine

$L \Rightarrow 0^n 1^n 2^{2n}$



input string → 001122.

ID:

q_0, 001122
$X q_1 0 1122$
$X 0 q_1 1122$
$X 0 Y q_2 122$
$X 0 Y 1 q_2 22$
because after making it Z ~~$X 0 Y 1 q_3 Z 2$~~
$X 0 Y q_3 1 Z 2$
$X 0 q_3 Y 1 Z 2$
$X q_3 0 Y 1 Z 2$
$q_3 X 0 Y 1 Z 2$
$X q_0 0 Y 1 Z 2$
$X X q_1 Y 1 Z 2$
$X X Y q_1 1 Z 2$
$X X Y Y q_2 Z 2$
$X X Y Y Z q_2 2$
$X X Y Y q_3 Z Z$
$X X Y q_3 Y Z Z$
$X X q_3 Y Y Z Z$
$X q_3 X Y Y Z Z$
~~$X X Y Y Z Z$~~
$X X q_0 Y Y Z Z$
$X X Y q_4 Y Z Z$
$X X Y Y q_4 Z Z$
$X X Y Y Z q_4 Z$
$X X Y Y Z Z q_4 B$
$X X Y Y Z Z B q_4 B$

Scanned by CamScanner