SRN

PES UNIVERSITY, Bangalore

(Established under Karnataka Act No. 16 of 2013)

UE19CS205

Scheme & Solution

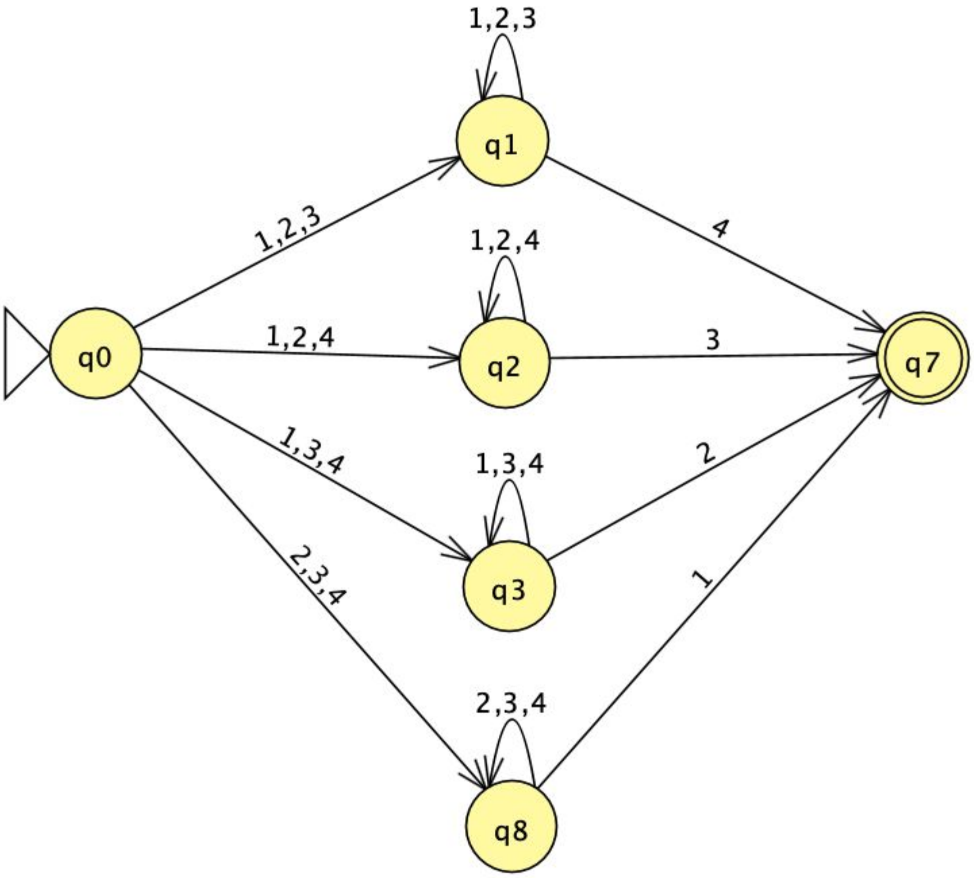
IN SEMESTER ASSESSMENT (ISA-1)- B.TECH III SEMESTER
October, 2020

Automata Formal Languages & Logic

Time: 2 Hrs **Answer All Questions** **Max Marks: 60**

Note:

- **Read all the Questions carefully before answering.**
- **The Question paper spans over 3 sheets and contains 6 Questions.**
- **Each Question carries 10 Marks and contains exactly 2 sub-parts (part a and part b).**

1	a	For the alphabet $\Sigma = \{1, 2, 3, 4\}$, construct an NFA for the language : $L = \{ w \in \Sigma^* \mid$ the last character of w appears nowhere else in the string, and $\|w\| > 1 \}$ **Solution:** 	5

b

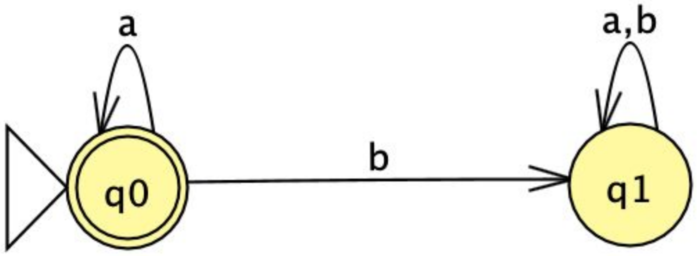
Let Σ = {a, b}. Consider the language L = {a*}.

Construct a DFA for the language L^c which accepts the complement of the Language L.

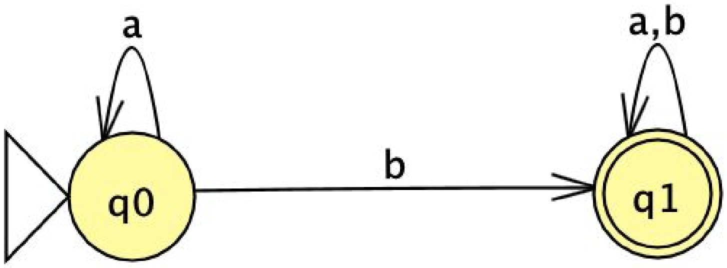
5

Solution:

DFA that accepts the language L ={a*} is given as:



DFA that accepts the complement of the Language L given as:

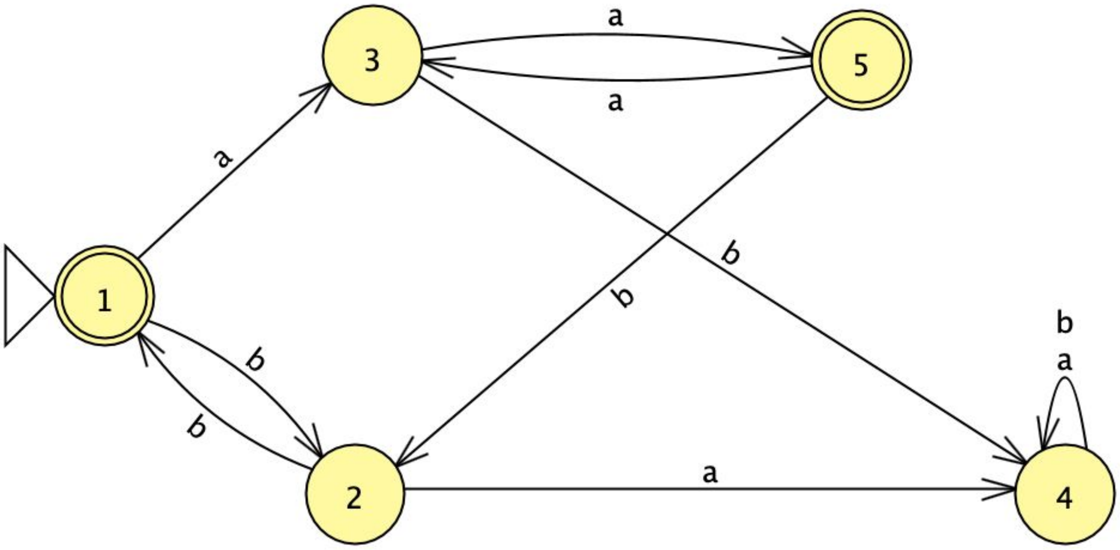


2

a

Minimize the following DFA of 5 States:

5

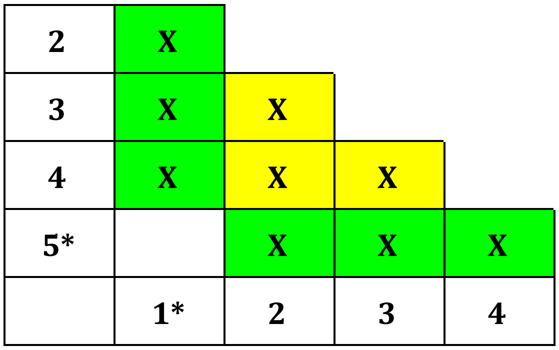


Solution:

Transition Table of the given DFA :

	a	b
→ 1*	3	2
2	4	1
3	5	4
4	4	4
5*	3	2

We minimize the DFA using the Table Filling algorithm:



2	X			
3	X	X		
4	X	X	X	
5*		X	X	X
	1*	2	3	4

Distinguishable pairs (Pair of Final and Non-Final State) are marked in green color.

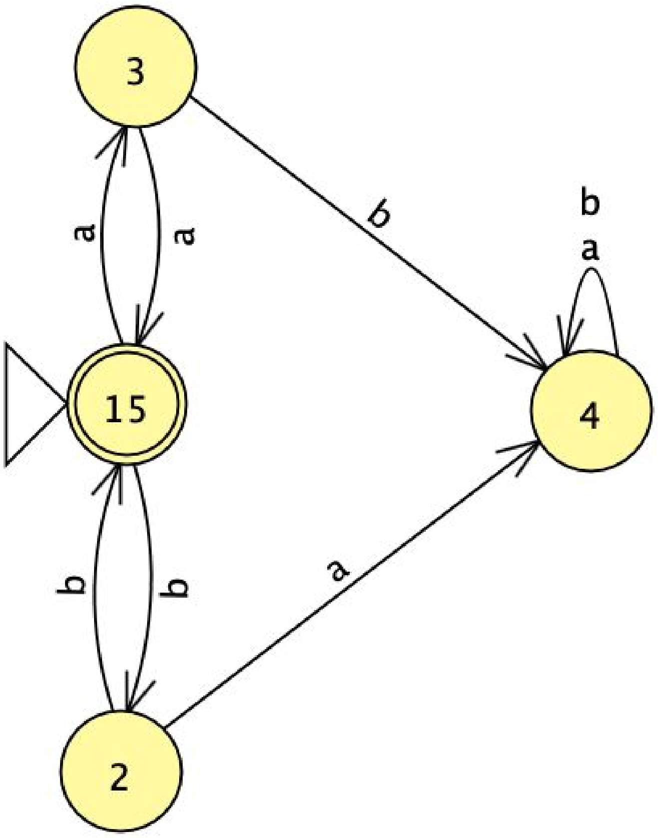
We must mark the following states as distinguishable due to the following reasons:

δ ((2,4) , b) = {1,4}
δ ((2,3) , a) = {4,5}
δ ((3,4) , a) = {4,5}

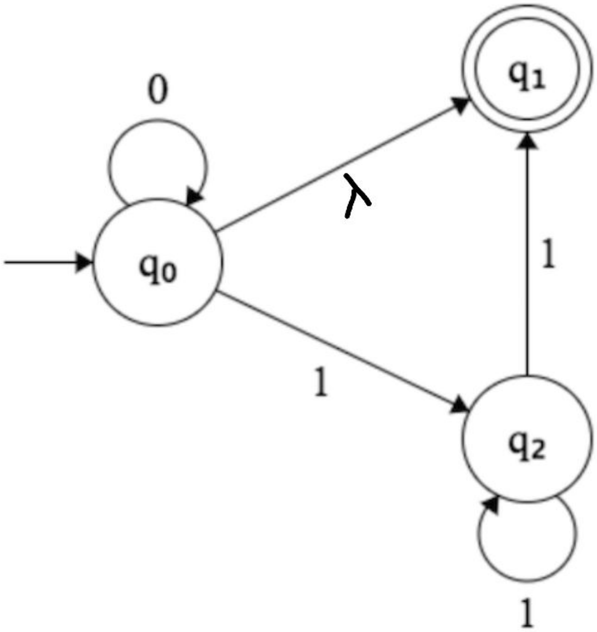
The two final states can be merged as:
δ ((1,5) , a) = (3} and δ ((1,5) , b) = (2}

Hence the minimized DFA is given as:

SRN



b Convert the following NFA to DFA: 5



Solution:

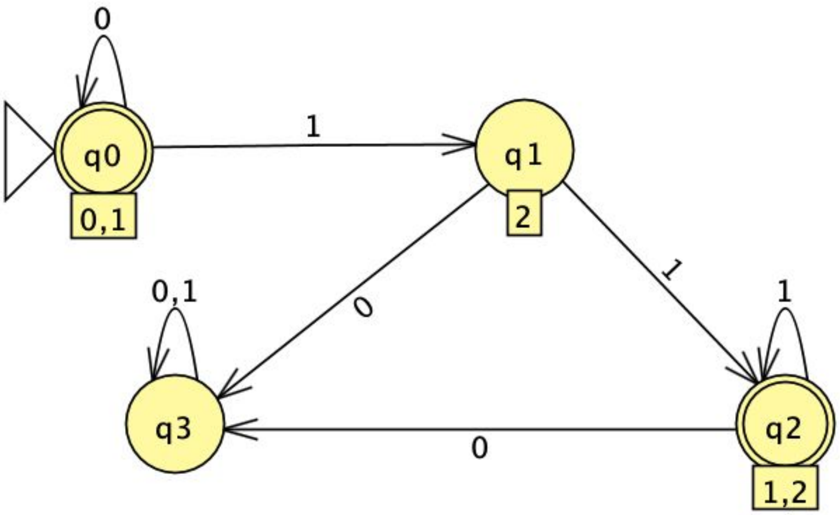
Transition Table of the given λ-NFA :

	0	1	λ-Closure
→ q0	q0	q2	q0, q1
q1*	Φ	Φ	q1
q2	Φ	q1, q2	q2

Transition Table of the given DFA :

	0	1
→ q0 q1	q0 q1	q2
q2	Φ	q1, q2
q1q2	Φ	q1, q2
Φ	Φ	Φ

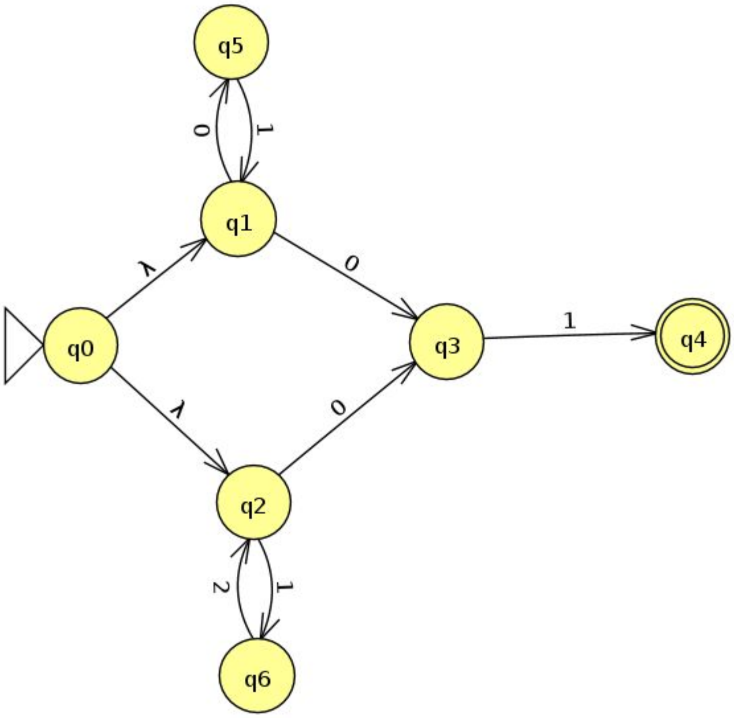
DFA Transition Diagram :



3 a Draw a NFA that accepts the language corresponding to the regular expression: 5

((01)* + (12)*)01

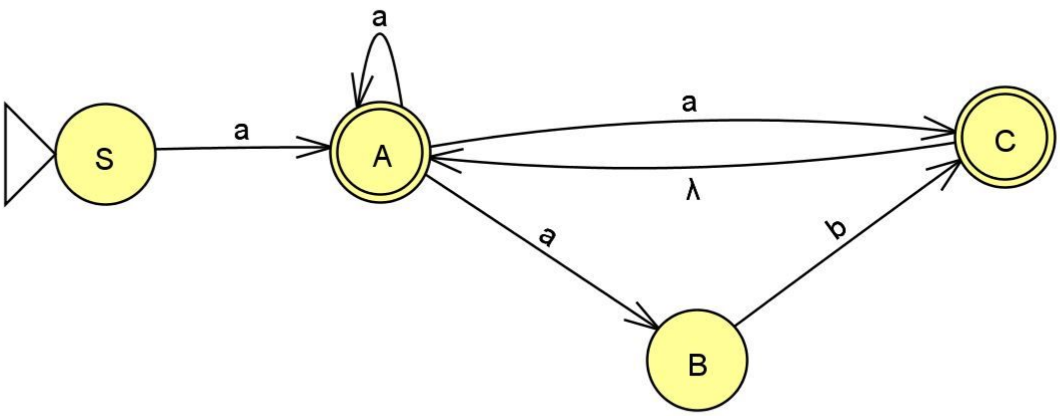
Solution:

			
	b	Let Σ = {1, 2, ≤} and let L be the language defined as follows: L = { w ∈ Σ * | w is a **valid** chain of inequalities relating the numbers 1 and 2 }. For example: The following strings belong to the language : 1≤2 , 1≤1≤2≤2, 2≤2≤2, 1≤1≤1≤1, 1≤1≤2 but, the following doesn't:2≤≤ , ≤2 , λ , 1, 12≤22 Note in particular that inequalities involving numbers like 12, 222, 121212, etc. whose digits are 1 and 2 aren't allowed i.e. 121 ≤ 112 (the inequality should only relate the numbers that is, single digit 1 and 2) and any individual number itself isn't allowed(i.e., 1 ∉ L and 2 ∉ L). Construct a regular expression for L. Solution: **(1 ≤ 1) (≤ 1)* + (1 ≤ 2) (≤ 2)* + (2 ≤ 2) (≤ 2)***	5
4	a	Using Pumping lemma, Determine whether the following language on Σ = {a,b } is regular or not. L = {$a^n b^n$: n >= 1} **Solution:** ● The opponent claims that the language L = {$a^n b^n$: n >= 1} is regular. ● Let the number of states in the opponent's hypothetical automata for language L is n (Pumping length).	5

- We choose a string w = a^nb^n such that,
  |w| > n (length of the string is greater than the number of states in the machine)
  and w ∈ L.
- ∀ w = xyz (for any break up of the string in 3 parts) such that,
  |xy| <= n (y- loop is within the n states)
  and |y| >= 1 (loop is made up of at least one symbol)
- In our string w= a^nb^n , the first n symbols are made up only of a's.
  Hence if we assume the loop is made up of single 'a', we can break the string as:
  $$a^{n-1} (a)^i b^n$$
- We see that, if we pump down the loop that is choose i=0, the resultant string does not belong to the language L, as the number of a's and b's become unequal i.e.,
  $$a^{n-1}b^n \notin L$$

Hence proved that the language L is not regular.

b Convert the following Finite Automata to Regular Grammar: 5



Solution:
S → aA
A → aA | aC | aB | λ
B → bC
C → A | λ

5 a Construct a Context free grammar to generate variable declaration statements in a C language. For example, your grammar should be able to generate strings of the following kind: 5

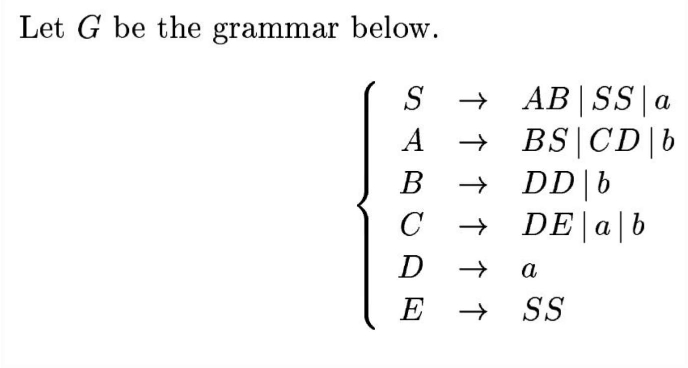
int a;
int a, b, c, d;
int a, b=2, c=5, d;
int d=8;

Assume you are handling only the basic types int and float.
The **Terminals** in your grammar are : {**int**, **float**, **id**, **num**, **=**, **,** , **;**}
Here id denotes an identifier (that is a variable name) and num denotes a number.
Assume the Start symbol is D.
Solution:

D → T L ;
T → int | float
L → L, X | X
X → id | id = num

b

5



Let G be the grammar below.

$$\begin{cases} S \rightarrow AB \mid SS \mid a \\ A \rightarrow BS \mid CD \mid b \\ B \rightarrow DD \mid b \\ C \rightarrow DE \mid a \mid b \\ D \rightarrow a \\ E \rightarrow SS \end{cases}$$

Use CYK algorithm to answer the following membership question:
Does the string w = “abaab” belong to L(G)?

5	S, E				
4	S, E	S, A			
3	Φ	S, A	S		
2	Φ	A	S, E, A, B	Φ	
1	S, C, D	A, B, C	S, C, D	S, C, D	A, B, C
	a	b	a	a	b

6 a

3

With an example, explain what is an Ambiguous grammar?

Solution:

A grammar G = (V, T, P, S) is called an Ambiguous grammar if there exists 2 different leftmost derivations or 2 different rightmost derivations (basically 2 different structures) for the same string w ∈ L(G).

For example :

S -> aS | a | λ is an ambiguous grammar as we can obtain string a in two different ways:

LMD 1 S ⇒ aS S ⇒ a (using S → λ)	LMD 2 S ⇒ a

SRN										

b

Let Σ = {(,), [,]}. Consider the language L as

L = {properly nested strings from Σ∗}.

So ([ ] ( )) is in L, but not ([ )) and not ([ )].

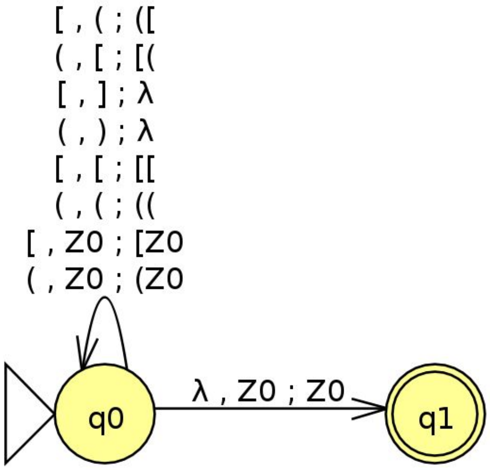
I. Construct a PDA to accept the language L. **[4 Marks]**

II. Do a short trace of the state sequence and sequence of stack contents as this machine recognizes the string "**[( ) ( )]**" . **[3 Marks]**

7

Solution:

I.



II.Trace of String : [( ) ( )]

δ (q0, **[( ) ( )]**, z0) **(Push [ )**

$\vdash$ δ (q0, **( ) ( )]**, [z0) **(Push ( )**

$\vdash$ δ (q0, **) ( )]**, ([z0) **(Match (and))**

$\vdash$ δ (q0, **( )]**, [z0) **(Push ( )**

$\vdash$ δ (q0, **)]**, ([z0) **(Match (and))**

$\vdash$ δ (q0, **]**, [z0) **(Match [and])**

$\vdash$ δ (q0, $\boldsymbol{\lambda}$, z0)

$\vdash$ (q1, $\boldsymbol{\lambda}$, z0)

Since q1 is a final state and input is completely processed, the string is accepted by the PDA.