Teacher Initial: MSZ

Daffodil International University

Fall 2020

Department of Computer Science and Engineering

Final Open Book Examination Answer Script

Full Marks: 25 Allowed Time: 4hrs (from: 02:00pm to: 06:00pm)

Date: Tuesday 3, November 2020

Submission Date: Tuesday 3 November, 2020 by 06:00pm

General Information (must be filled by the student)

COURSE CODE: CSE-331 SECTION: PC-C PROGRAM: DAY

STUDENT ID: 181-15-1815 TIME STARTED: 2:00 TIME ENDED: 6:00

[Student must either TYPE or HAND WRITE the answers in this template; In case needed just write your detail on the paper using hand]

Ans to the question no-1

Given equation:

$$E = m * g * h + \left(\frac{1}{2}\right) * m * v * v$$

(a)

Lexical Analyzer:

<id,1> <=> <id,2> <*> <id,3> <*> <id,4>
<+> <(> <1> </> <2> <)> <*> <id,5> <*>
<id,6> <*> <id,7>

(b) Ans to the question no-1

Intermediate Code Generation:

t_1 = intTofloat(1)
t_2 = intTofloat(2)

$t_3 = t_1 / t_2$

$t_4 = t_3 * id5$

$t_5 = t_4 * id6$

$t_6 = t_5 * id7$

$t_7 = id2 * id3$

$t_8 = t_7 * id4$

$id1 = t_6 + t_8$

(C) Ans to the ques no-1

Code optimization:

$t_1 = 1.0 / 2.0$

$t_2 = t_1 * id5$

$t_3 = t_2 * id6$

$t_4 = t_3 * id7$

$t_5 = id2 * id3$

$t_6 = t_5 * id4$

$id1 = t_4 + t_6$

(d) Ans to the ques no-1

Final Code Generation:

```
LDF   R1 , #1.0
DIVF  R1 , R1 , #2.0
LDF   R2 , id5
MULF  R1 , R1 , R2
LDF   R3 , id6
MULF  R1 , R1 , R3
LDF   R4 , id7
MULF  R1 , R1 , R4
LDF   R5 , id2
LDF   R6 , id3
MULF  R5 , R5 , R6
LDF   R7 , id4
MULF  R5 , R5 , R7
ADDF  R5 , R5 , R1
STF   id1 , R5
```

Ans to the question no-2

(a)

Formal defination of finite Automata of figure 1:

We know that, in finite automata there is 5 tuples and these are $(Q, \Sigma, \delta, q_0, F)$ where,

Q = finite set of state

Σ = finite set of symbol/alphabet

δ = transition function

q_0 = ~~inf~~ initial state

F = set of final state

From the figure formal defination will be,

$Q = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$

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

$q_0 = 0$

$F = \{10\}$

$\delta = Q \times \Sigma \to Q$ DFA or $Q \times \Sigma \to 2^Q$ NFA

Though it's NFA so δ(transition funtion will be) :-

	ε	a	b	c
→ 0	{1,10}	{ }	{ }	{ }
1	{2,5}	{ }	{ }	{ }
2	{ }	{3}	{ }	{ }
3	{ }	{ }	{4}	{ }
4	{7}	{ }	{ }	{ }
5	{6}	{ }	{ }	{ }
6	{7}	{ }	{ }	{ }
7	{ }	{ }	{8}	{ }
8	{ }	{ }	{ }	{9}
9	{1,10}	{ }	{ }	{ }
10	{ }	{ }	{ }	{ }

(b) Ans to the ques no-2

Yes, the finite Automata of Figure 1 is NFA, Because in NFA we know that, for DFA at least one can determine the state to which the machine will move and for NFA any combination of the state can move the machine. In the above figure 1. We see that for alphabet there is ϵ used that's mean it a NFA.

Though it is NFA now, I will convert this NFA to DFA using subset construction table.

NFA to DFA conversion for figure 1:.

For NFA states transition table:-

	ε	a	b	c
→ 0	{1,10}	{ }	{ }	{ }
1	{2,5}	{ }	{ }	{ }
2	{ }	{3}	{ }	{ }
3	{ }	{ }	{4}	{ }
4	{7}	{ }	{ }	{ }
5	{6}	{ }	{ }	{ }
6	{7}	{ }	{ }	{ }
7	{ }	{ }	{8}	{ }
8	{ }	{ }	{ }	{9}
9	{1,10}	{ }	{ }	{ }
10	{ }	{ }	{ }	{ }

DFA state transition table:

	ε	a	b	c
→ 0	[1,10]	[11]	[11]	[11]
[1,10]	[2,5]	[11]	[11]	[11]
[2,5]	[6]	[3]	[11]	[11]
[6]	[7]	[11]	[11]	[11]
[3]	[11]	[11]	[4]	[11]
[7]	[11]	[11]	[8]	[11]
[4]	[7]	[11]	[11]	[11]
[8]	[11]	[11]	[11]	[9]
[9]	[1,10]	[11]	[11]	[11]
[11]	[11]	[11]	[11]	[11]

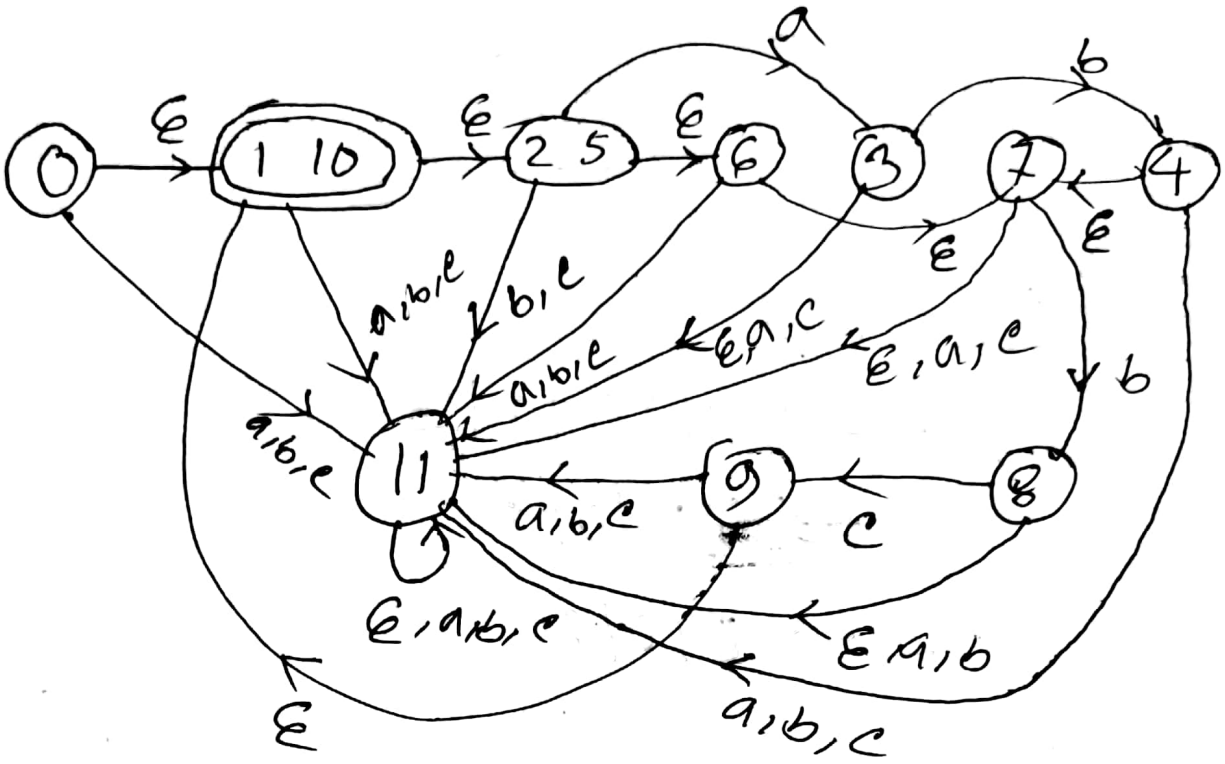


Fig: DFA

Ans to the ques no -3

(a)

Given that,

$X \Rightarrow XYX \mid 0 \mid 1 \mid Z$

$Y \Rightarrow ZX \mid 0 \mid 1$

$Z \rightarrow \epsilon \mid 0 \mid 1$

On,

$X \rightarrow XYX$

$X \rightarrow 0$

$X \rightarrow 1$

$X \rightarrow Z$

$Y \rightarrow ZX$

$Y \rightarrow 0$

$Y \rightarrow 1$

$Z \rightarrow \epsilon$

$Z \rightarrow 0$

$Z \rightarrow 1$

Given, String: 111110001011110

LMD:

$X \rightarrow XYX$
$\rightarrow 1YX$
$\rightarrow 1ZXX$
$\rightarrow 11XX$
$\rightarrow 11XYXX$
$\rightarrow 111YXX$
$\rightarrow 1111XXX$
$\rightarrow 1111XXX$
$\rightarrow 1111XYXXX$
$\rightarrow 1111YXXX$
$\rightarrow 1110XXX$
$\rightarrow 1110XYXXX$
$\rightarrow 1110YXXX$
$\rightarrow 1110XXXX$
$\rightarrow 1110XYXXXX$
$\rightarrow 1110YXXXX$
$\rightarrow 1110ZXXXXX$
$\rightarrow 11101XXXXX$
$\rightarrow 111011XXXX$
$\rightarrow 1110111XXX$
$\rightarrow 11101110XX$
$\rightarrow 111011100X$
$\rightarrow 1110111000$ (Ans)

Given String, 1111100010111 0

RMD:

$X \rightarrow XYX$

$\rightarrow XY0$

$\rightarrow XZX0$

$\rightarrow XZXYX0$

$\rightarrow XZXY10$

$\rightarrow XZXZX10$

$\rightarrow XZXZXY110$

$\rightarrow XZXZXZX110$

$\rightarrow XZXZXZXYX110$

$\rightarrow XZXZXZXYXY1110$

$\rightarrow XZXZXZXYXY1110$

$\rightarrow XZXZXZXY1011\underline{1}0$

$\rightarrow XZXZXZX01011\underline{1}0$

$\rightarrow XZXZXZ0010111\underline{1}0$

$\rightarrow XZXZX00101111\underline{1}0$

$\rightarrow XZXZ1000101110$

$\rightarrow XZX11000101110$

$\rightarrow XZ111000101110$

$\rightarrow X1111000101110$

$\rightarrow 1111000101110$

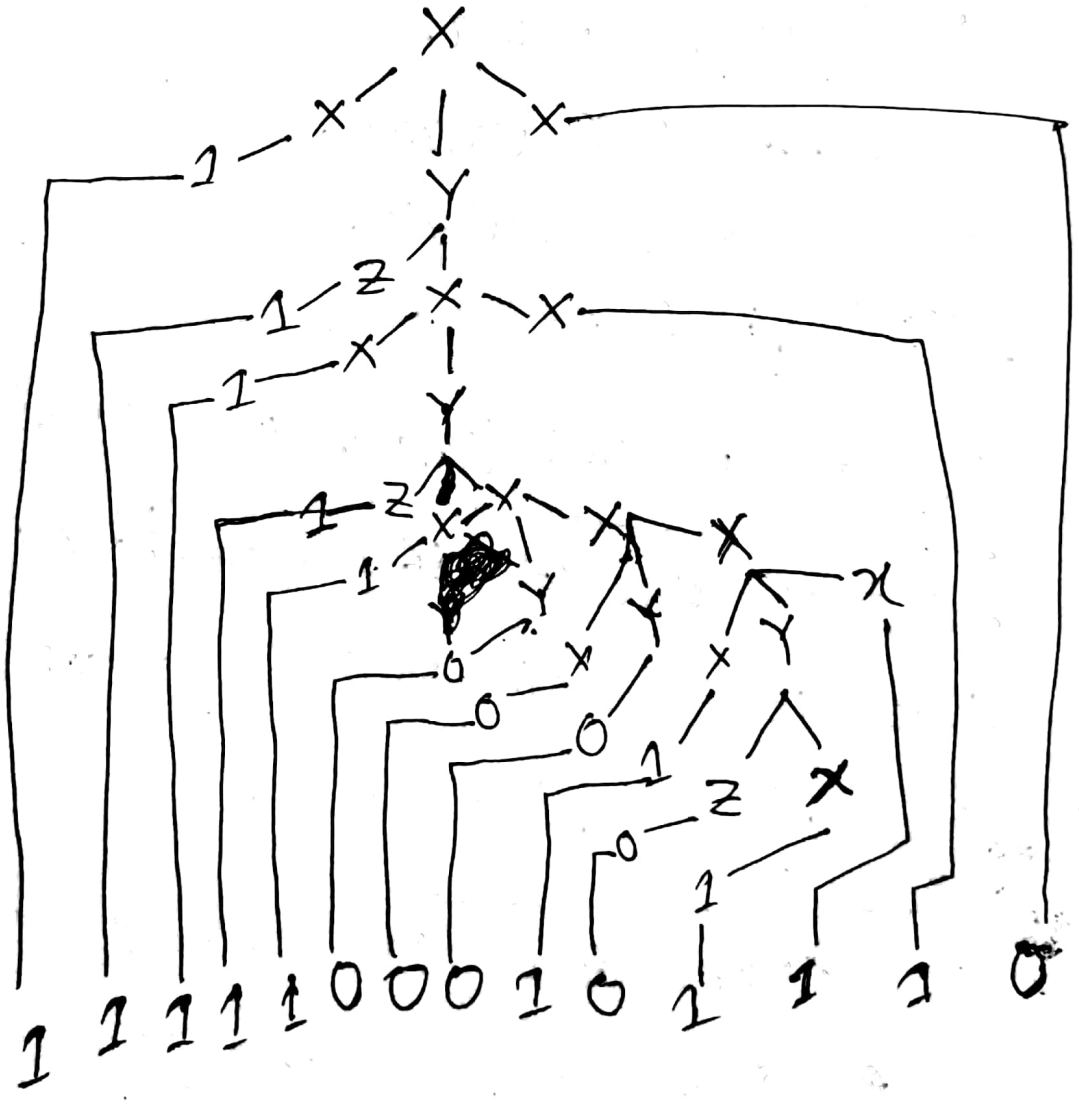
(Ans)

Ans to the question no-3

(b)

Drawing the parse tree for above ~~seq~~ scenario:

LMD:



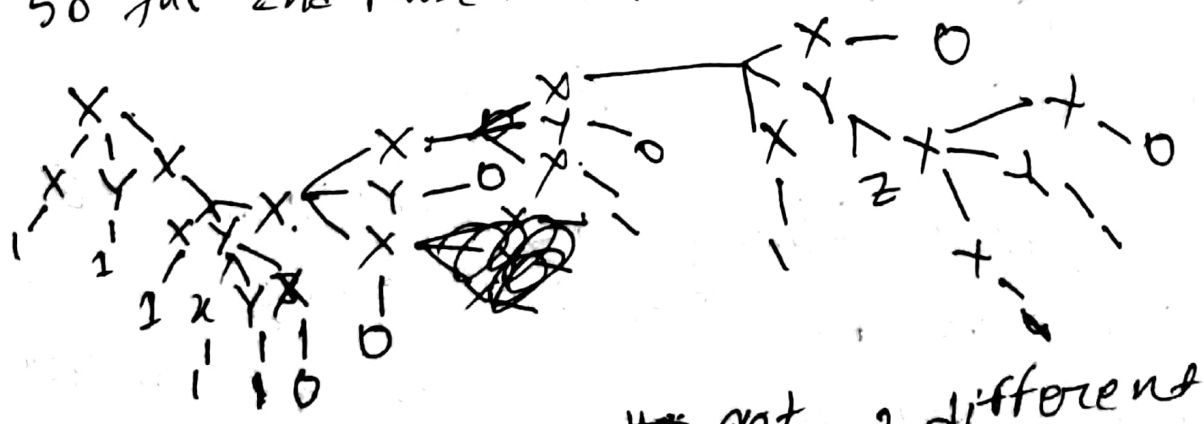
Ans to the question no-3

(e)

We already have found one LMD for this string.
So we are trying to find another LMD.

$X \rightarrow XYX$
$\rightarrow 1YX$
$\rightarrow 11X$
$\rightarrow 11XYX$
$\rightarrow 111YX$
$\rightarrow 111XYXX$
$\rightarrow 1111YXX$
$\rightarrow 11111XX$
$\rightarrow 11111 0X$
$\rightarrow 11111 0XYX$
$\rightarrow 111110 0YX$
$\rightarrow 1111100 0X$
$\rightarrow 1111100 0XYX$
$\rightarrow 1111100 01YX$
$\rightarrow 1111100 010X$
$\rightarrow 1111100 010XYX$
$\rightarrow 1111100 0101YX$
$\rightarrow 1111100 01010XX$
$\rightarrow 1111100 01010 1X$
$\rightarrow 1111100 01010 11$

So the 2nd parse tree.



So we ~~have gotten~~ got 2 different LMD
So, the grammar is ambiguous.

Ans to the question no-4

Ans: Indentifying the specific errors from given code:

We know, there can be 4 types error.

1. Lexical error.
2. Syntactical error.
3. Semantical error.
4. Logical error.

Lexical error: When we made some mistake on ~~id identifier~~ identifier, that is called lexical error.

Syntactical error: If we miss semicolon or unreadable or unbalanced parenthesis then it's called syntactical error.

Semantical error: If there has incompatible value assignment then it is semantical error.

Logical error: In the code output not reachable or there has infinite loop then it's called logical error.

So, In the given code, the errors are in 3, 4, 5 and 6 line.

In line 3: There has laxical error.

Because "int" can't define as "innt". And there also Semantical error, we can't able to state 3 value on 2 size array.

In line 4: There has two type error.

Like syntactical error because ";" is missing and a[6] it's made error and

that is semantical error.

In line 5: There has a lexical error. printf can't define as "printf". and ";" is missing so there also has syntactical error.

In line 6: There will be return 0, but given "return b". So this is error. but the code compilation will not impacted by this return type.

So, correct code. is:

```
# include <stdio.h>
  int main()
  {
     int a[3] = {2,4,6}, b=1:
      sum = a[0]+b;
      printf("Result is %d", sum);
      return 0;
      } (Ans).
```