BRACT's

Vishwakarma Institute of Information Technology, kondhwa(BK), Pune-48

Department of Computer Engineering

THEROY OF COMPUTATION

SCE

SY BTech Computer Engineering

Year: 2020-21

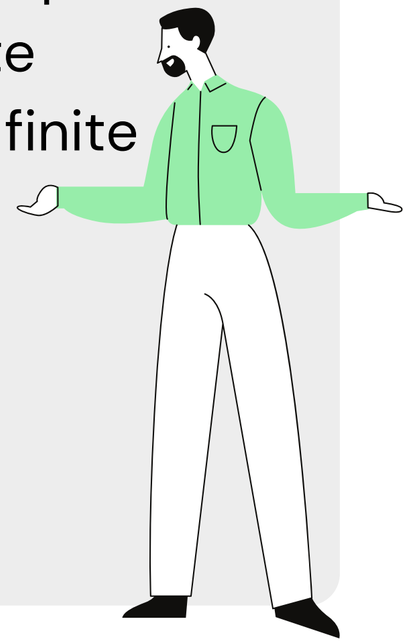
Group Members

74	Rohit Kudale
78	Shekhar Chaughule
79	Devesh Pathak
81	Masood Ismail Tamboli

Let's Start!

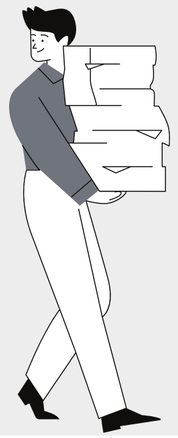
Finite State Automata

- Finite Automata(FA) is the simplest machine to recognize patterns.
- The finite automata or finite state machine is machine which have five elements or tuple(Q, Σ, q, F, δ)
- It has a set of states and rules for moving from one state to another but it depends upon the applied input symbol. Basically it is an abstract model of digital computer.
- There are two types of finite state machines (FSMs): deterministic finite state machines, often called deterministic finite automata, and non-deterministic finite state machines, often called non-deterministic finite automata.

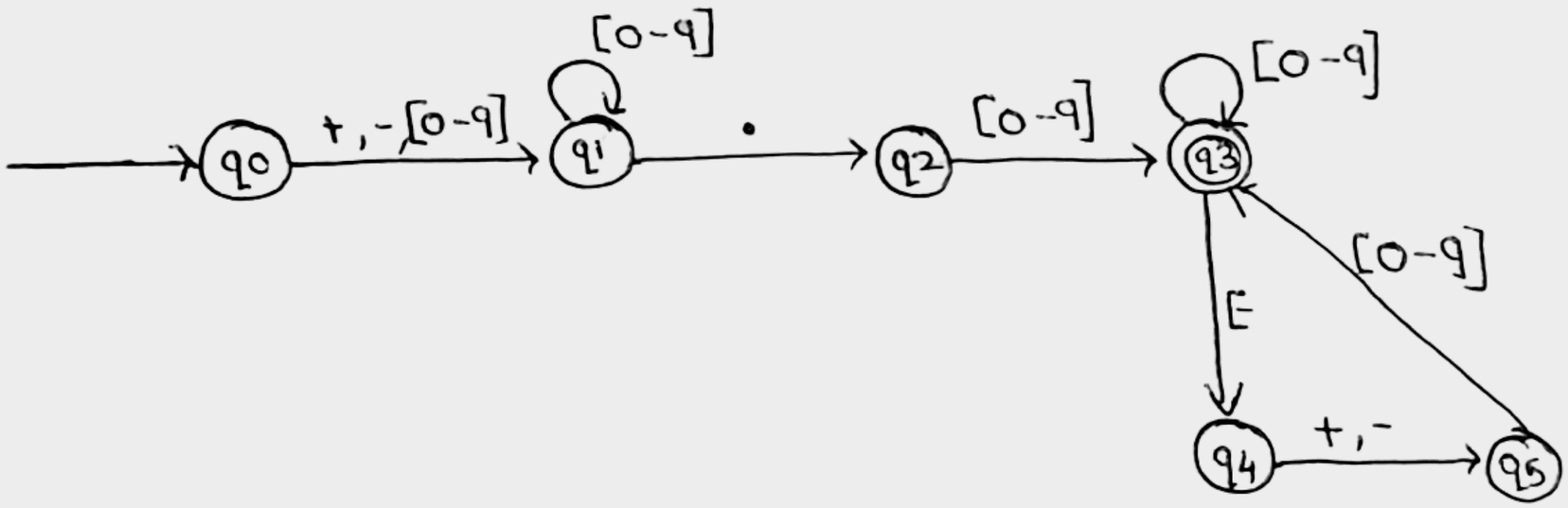


Question 1

- A real number is written as string of digits containing a decimal point.
- There must be atleast one digit before and after the decimal point.
- A real data item written in scientific notation consists of a sign followed by a real number, follwed by the letter E, another +/- sign and an integer
- For ex. .9 and 9. are invalid and 0.9, 9.0, +1.2, -22.34, 22.3E+23, -2.3E2 are all valid.

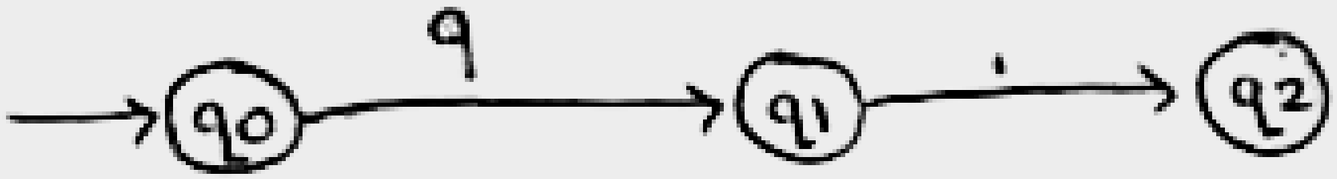


Solution:



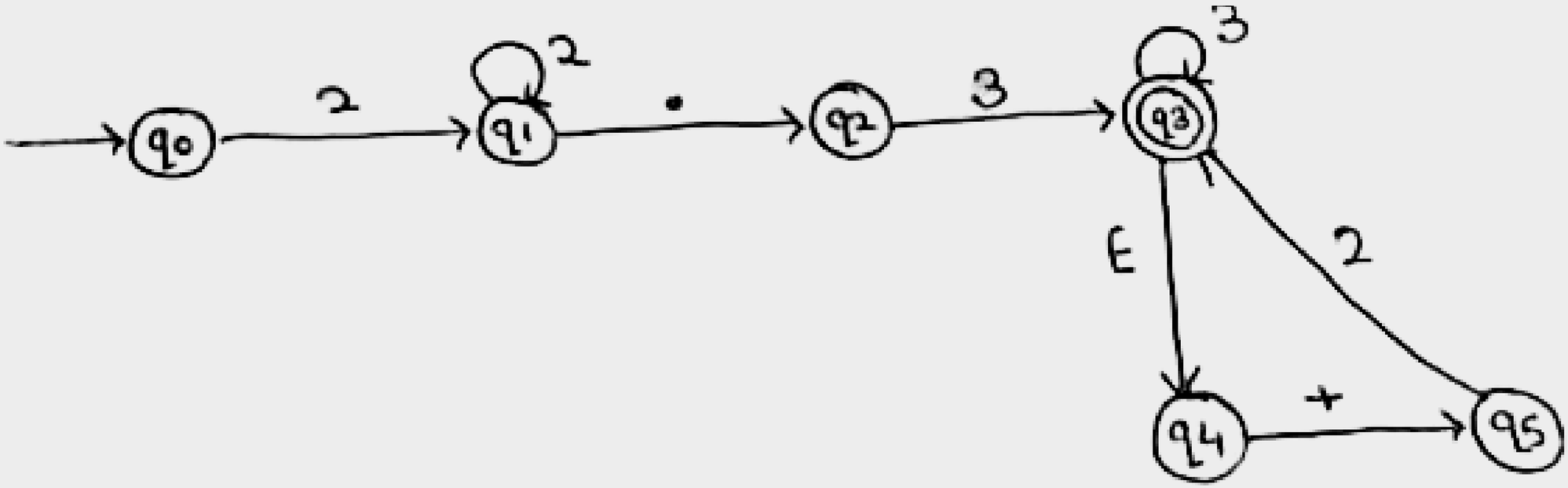
Ex. 1

Input = 9.



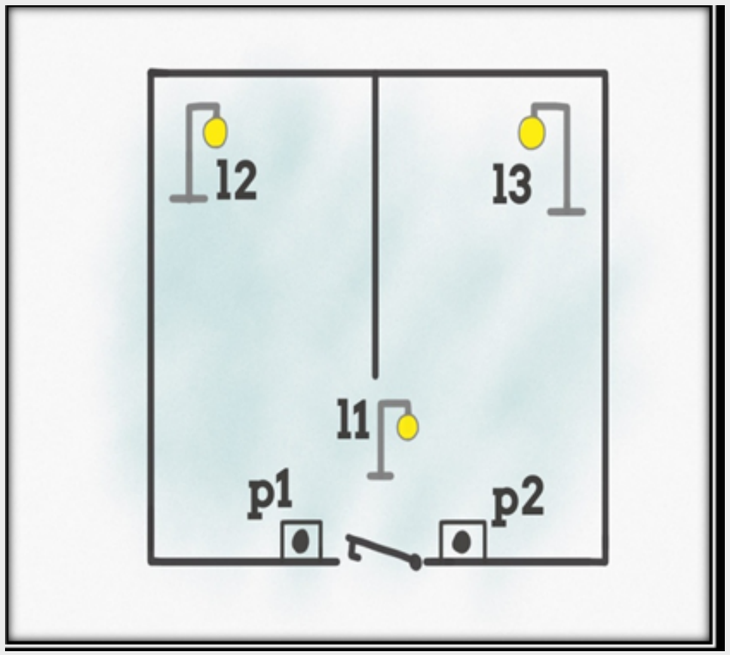
Ex. 2

Input = 22.3E+23



Question 2

- We have a house, with one door, 2 buttons and 3 lights. At the default state the lights are all turned off.
- When you enter the house, you can press one of the 2 push buttons you have, P1 or P2. When you press any of those buttons, the l1 light turns on.
- As we know that we are talking about the real life scenario over here.
- This Finite State Machine is slightly more complex, because this time we have multiple routes depending on the input given by user.



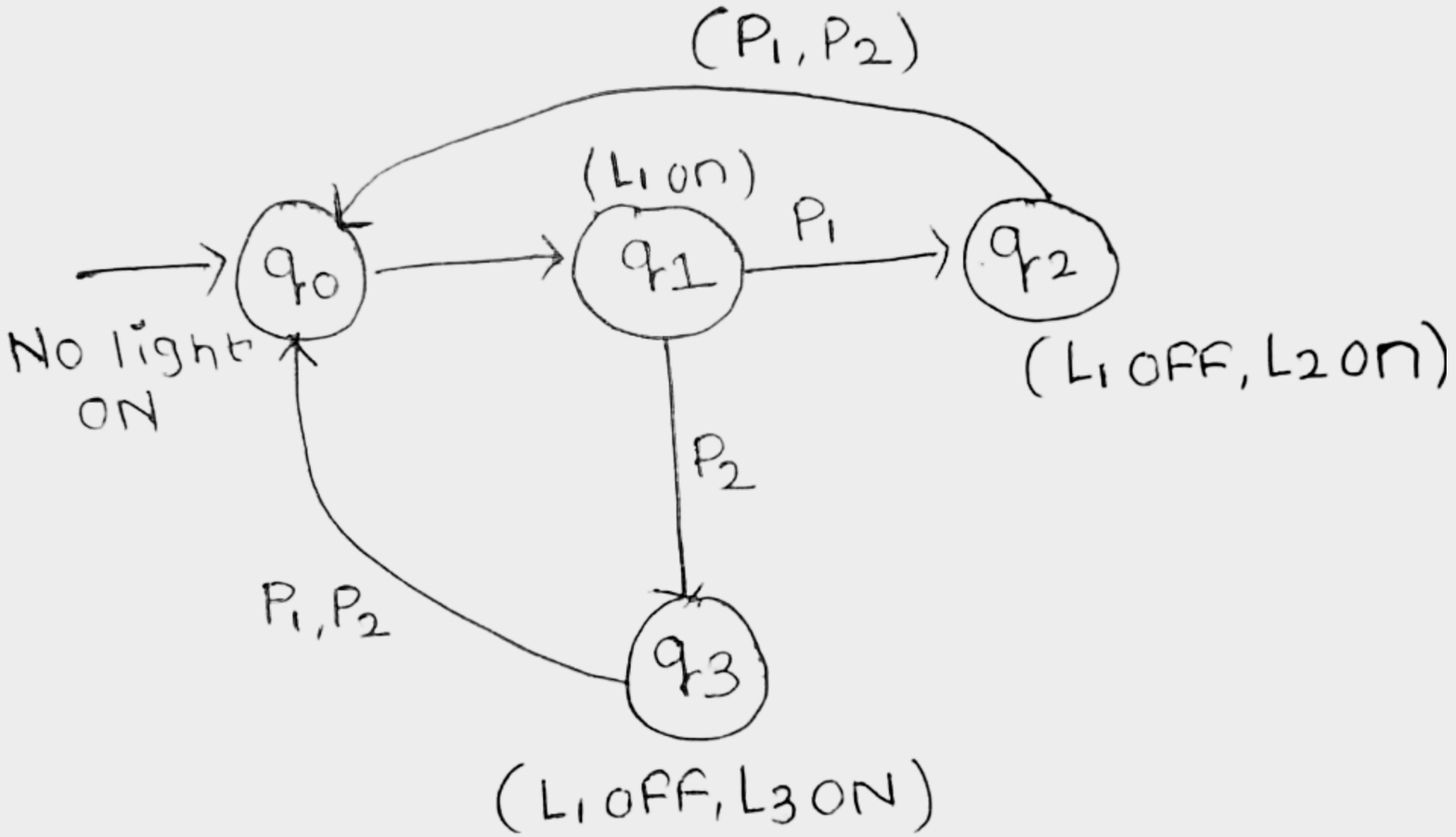
Solution

So, we are having 4 states :-

1. No lights on
2. L1 on
3. L2 on
4. L3 on

we are having 2 inputs as well :-

1.P1
2.P2



Question 3

Develop, Implement and Execute a program using YACC tool to recognize all strings ending with ‘bbb’ preceded by n a’s using the grammar a^ n bbb (note: input n value)

*assgn5.l | assgn5.y | *sce.l | sce.y

```
1 %{
2         #include"y.tab.h"
3
4 %}
5 %%
6 a       {return A;}
7 b       {return B;}
8 [ \n]   return '\n';
9
10 %%
11
```

Lex | Tab Width: 8 | Ln 11, Col 1 | INS

*assgn5.l | assgn5.y | sce.l | sce.y

```
%{
#include<stdio.h>
%}

%token A B
%%
//stmt : A A A A A B B B
//{
//      printf("valid string \n");return 0;
//}
input:n '\n' {printf(" valid grammmer\n"); return 0;}
n: A s1 B B B |B B B
s1:; |A s1
%%
main()
{

        printf("enter the str\n");
        yyparse();
}
int yyerror()
{
        printf("Error");
        return 0;
}
```

Yacc | Tab Width: 8 | Ln 21, Col 14 | INS

```
shekhar@shekhar-VirtualBox:~$ lex sce.l
shekhar@shekhar-VirtualBox:~$ yacc -d sce.y
shekhar@shekhar-VirtualBox:~$ gcc lex.yy.c y.tab.c -ll
sce.l:12:1: warning: return type defaults to ‘int’ [-Wimplicit-int]
   12 | yywrap()
      | ^~~~~~
y.tab.c: In function ‘yyparse’:
y.tab.c:1216:16: warning: implicit declaration of function ‘yylex’ [-Wimplicit-function-declaration]
 1216 |       yychar = yylex ();
      |                ^~~~~
y.tab.c:1349:7: warning: implicit declaration of function ‘yyerror’; did you mean ‘yyerrok’? [-Wimplicit-function-declar
ation]
 1349 |       yyerror (YY_("syntax error"));
      |       ^~~~~~~
      |       yyerrok
sce.y: At top level:
sce.y:15:1: warning: return type defaults to ‘int’ [-Wimplicit-int]
   15 | main()
      | ^~~~
shekhar@shekhar-VirtualBox:~$ ./a.out
enter the str
aab
Errorshekhar@shekhar-VirtualBox:~$ ./a.out
enter the str
aaaabbb
 valid grammmer
shekhar@shekhar-VirtualBox:~$ ./a.out
enter the str
aaaaaaaaaaaaaaaaaaaaaaaabbb
 valid grammmer
shekhar@shekhar-VirtualBox:~$
```