THEORY OF COMPUTATION

PROJECT

VENDING MACHINE

B-10

SUBMITTED BY:

Sakshi Dubey(13103716)

Puja Kandel(13103729)

Anupreksha Sinha(13103745)

Pranjal Yadav(9913103674)

IMPLEMENTATION OF VENDING MACHINE

In this project a state diagram is constructed for the proposed machine which can vend four products that is coffee, cold drink, candies and snacks. Four select (select1, select2, select3, select4) inputs are taken for selection of products. Select1 is used for the selection of snacks. Similarly select2, select3, select4 are used for coffee, cold drink and candies respectively. Rs_10 and rs_20 inputs represents rupees 10/- and 20/- notes respectively. A cancel input is also used when the user wants to withdraw his request and also the money will be returned through the return output. Return, product and change are the outputs. Return and change vectors are seven bits wide. Money is an in/out signal which can be updated with the total money of all products delivered at a time. Money signal is seven bits wide. Money_count is an internal signal which can be updated at every transition. This signal is also seven bits wide. If the inserted money is more than the total money of products then the change will be returned through the change output signal. The products with their prices are shown by table 1. The machine will work on the positive edge of clock and will return to its initial state when reset button is pressed.

DESIGN METHODOLOGY

The state diagram mainly consists of four states (User Selection, Waiting for the money insertion, product delivery and servicing (when product_not_available='1')). Initially when the reset button is pressed, the machine will be ready for the users to select the product. This state is the initial state of the design. After this the user will select the product to be dispensed. This state can be one of the select1, select2, select3 and select 4. The machine can accept only two types of notes i.e. rupees 10/- and 20/-. Let us suppose that the user selects sel1 input. The machine will firstly check that whether the products are available in the machine or not. After this the control unit will move to the waiting state, where it will wait for the money to be inserted. Then if rupees 10/- note is inserted then the machine will go to state_1 and wait until the desired money is inserted. And if rupees 20/- note is inserted the machine will move to state_2 and then wait until 30/- rupees are inserted to the machine. When the desired amount is inserted the machine will go to the snacks state and snacks will be delivered at the product output. If products are not available in the machine then the control unit will demand for servicing and after service the machine will get reset. This methodology is explained using a flow diagram shown in figure 3.

There is also an additional feature of withdrawing the request if the user doesn't want to take the product. When cancel button is pressed then the money inserted will be returned to the user Yes Yes No No No Yes Yes Yes No No Product out with change returns if any Insert Desired Money for selected product If right Amount? Start Select 1? Select 2? Select 3? Select 4? If Product available? Select Product Cancel the request Inserted Money Returned End. A money_count signal is used for calculating the total money inserted in the machine. And if the money inserted is more than the money of the product then the extra change will be returned to the user. The total amount of the product taken at a time is shown by the money signal. Similarly the user can select and get the other products following the above procedure.

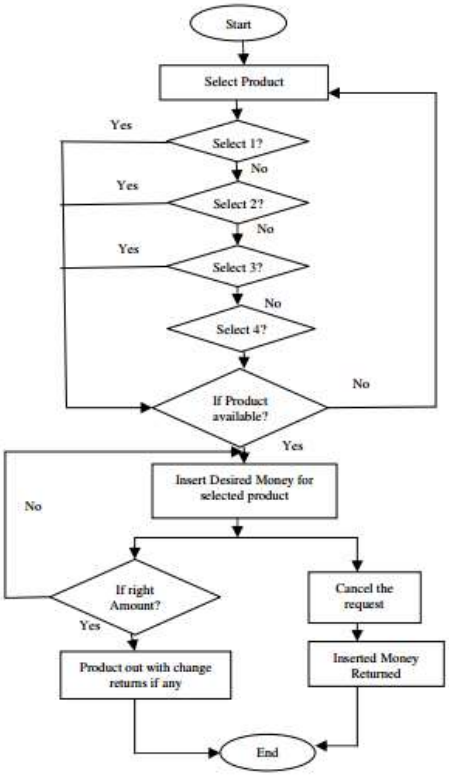


Figure 3: Flow Chart for Proposed Vending Machine

Description of states

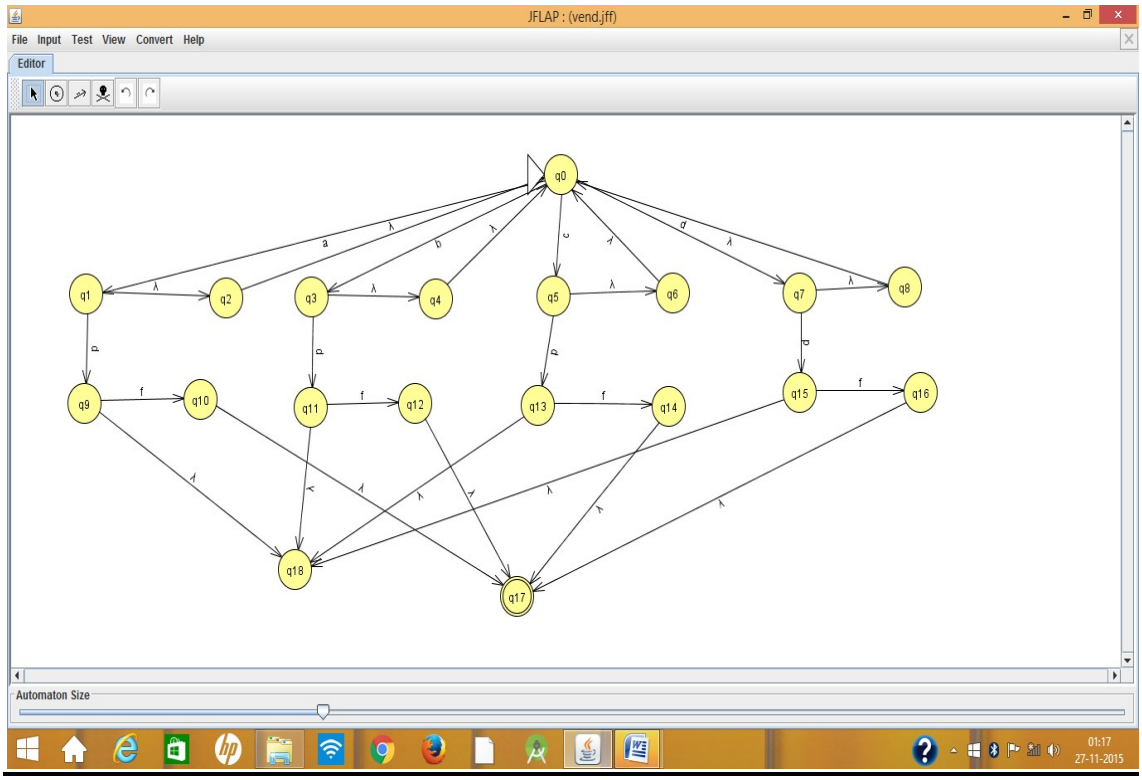
The selection of products and all the states are shown below in figure 4.

- When initialize=>
  - money_count=0;
  - Change=0;
  - Product=0;
- When select1=>
  - Sel1&!sel2&!sel3&!sel4
  - When product_available=1 => nx_st1<= waiting1;
  - When product_available=0 => nx_st1<= service1;
- When waiting1=>
  - When rs_10&!rs_20 => nx_st1<=state_1;
  - When !rs_10&rs_20=> nx_st1<= state_2;
  - Change=0; product=0;
  - When money_count>=30 nx_st1<= snacks;
- When state_1=>
  - Rs_10=1 & rs_20=0;
  - Change=0; Product=0;
  - Money_count=money_count+10;
- When state_2=>
  - Rs_10=0 & rs_20=1;
  - Change=0; Product=1;
  - Money_count=money_count+20;
- When snacks=>
  - Money_count>=30;
  - Product=1;
  - Change=money_count-30;
  - Snack_count=snack_count-1;
- When service=>
  - snack_count= 4
  - product<=0;
  - next_state<=resett;
- When cancel1=>
  - cancel=1;
  - return<=money_count;

Similarly we can select other products (coffee, Cold drink and candies).

Screenshots of Automata on J-Flap Environmemt

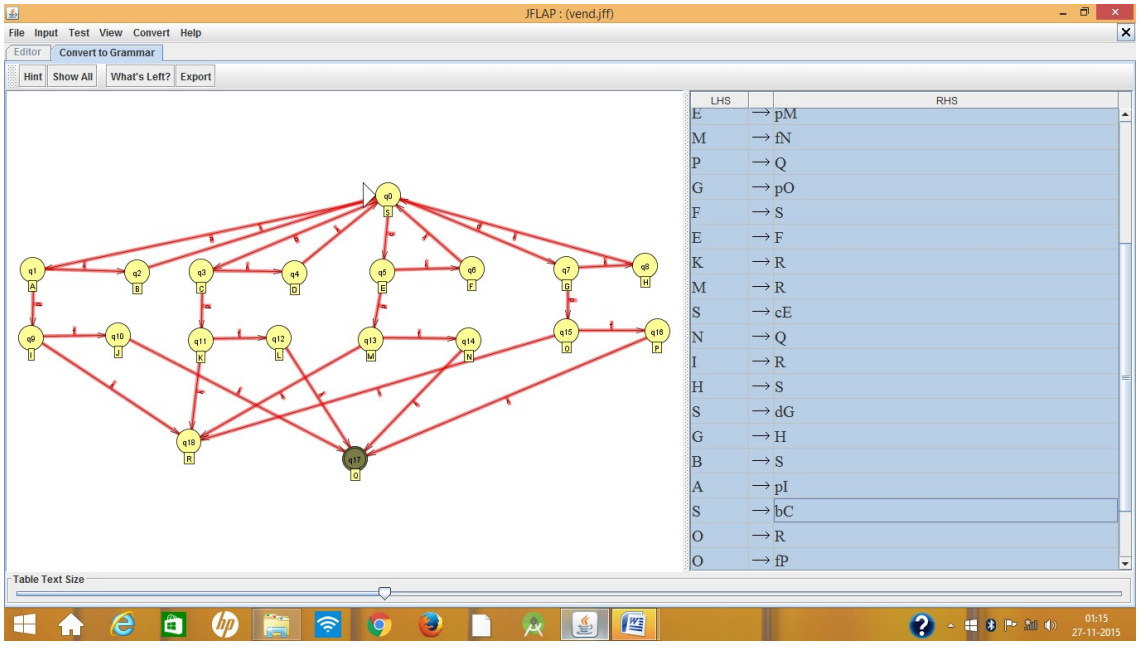
- **Screnshot of selecting & ordering an item**



States & Its meanings

State	Meaning
Q0	Initialize
Q1	Select 1
Q2	Service 1
Q3	Select 2
Q4	Service 2
Q5	Select3
Q6	Service 3
Q7	Select 4
Q8	Service 4
Q9	Waiting 1
Q10	Snacks
Q11	Waiting 2
Q12	Coffee
Q13	Waiting 3
Q14	Cold Drink
Q15	Waiting 4
Q16	Candie
Q17	Final State
Q18	cancel

➢ Regular Grammmar



Regular Expressions

S -> aA
Q -> ƛ
I -> fJ
C -> D
L -> Q
A -> B
D -> S
K -> fL
E -> pM
M -> fN
P -> Q
G -> pO
F -> S
E -> F
K -> R
M -> R
S -> cE
N -> Q
I -> R
H -> S

S -> dG

G -> H

B -> S

A -> pI

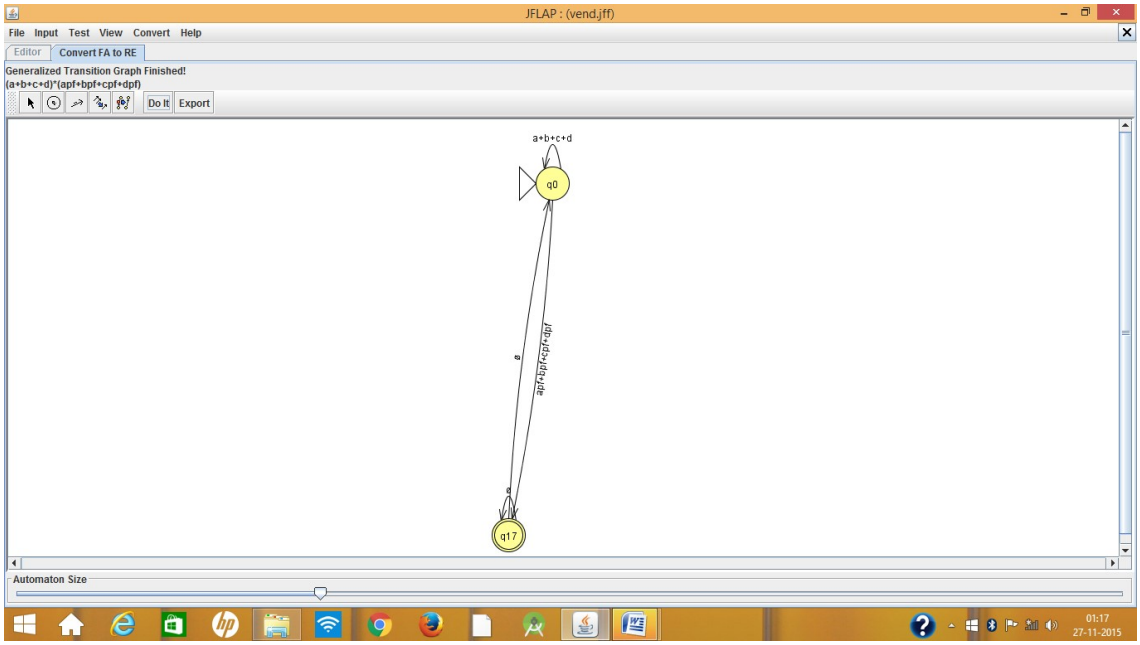
S -> bC

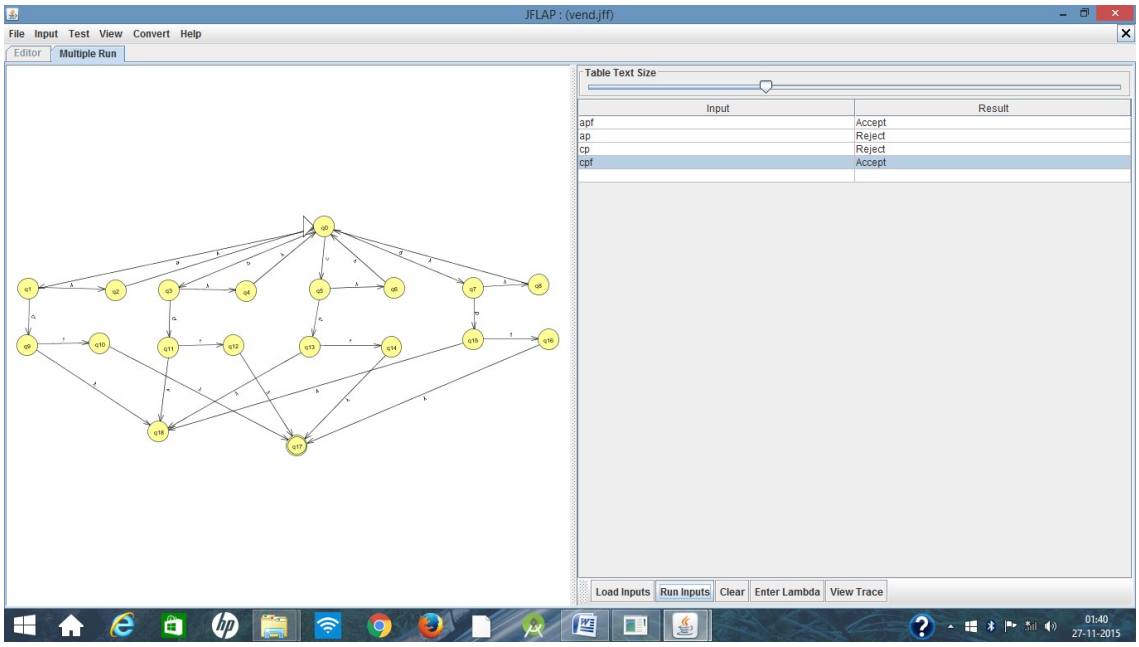
O -> R

O -> fP

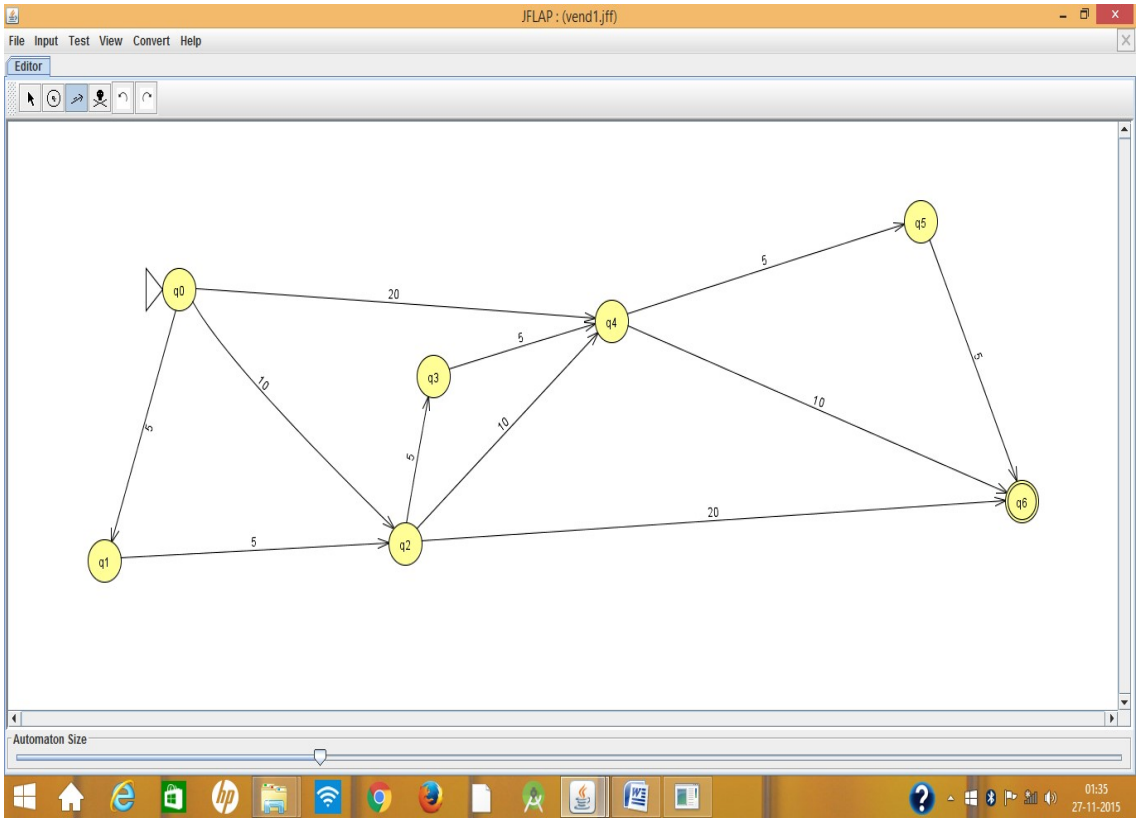
J -> Q

C -> pK

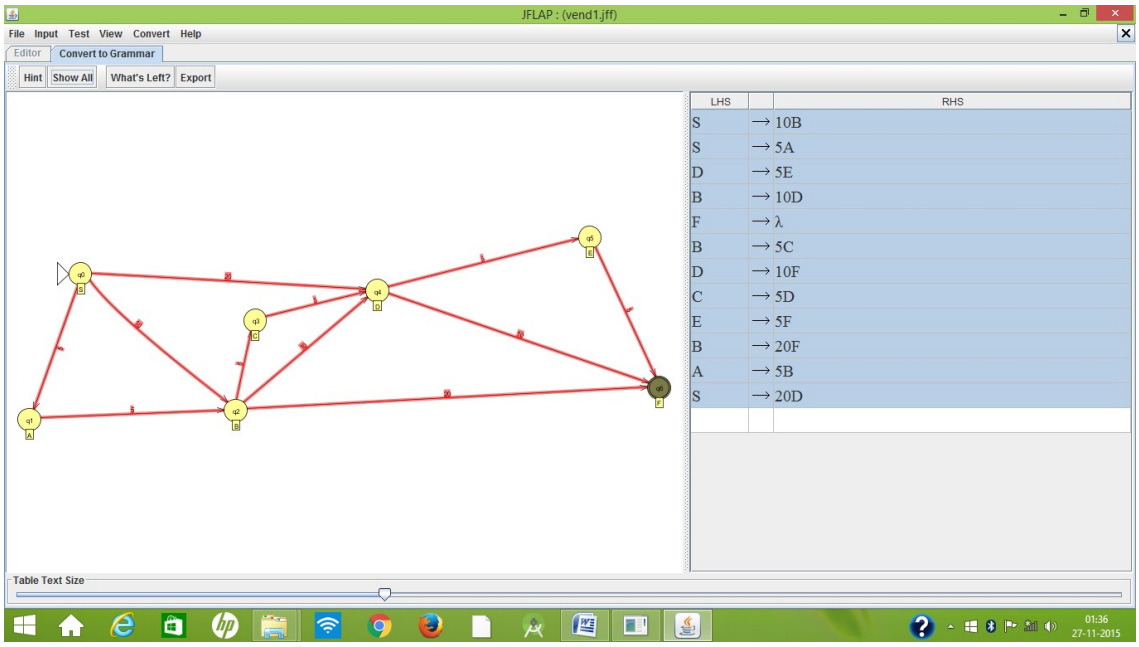




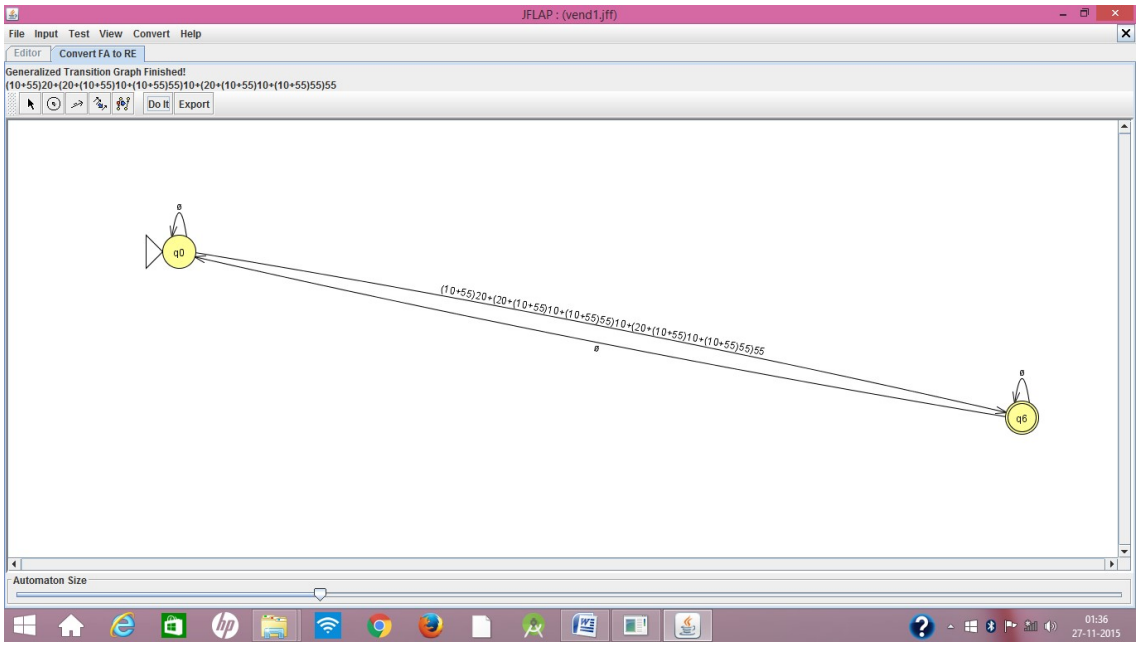
- **Screenshot of vending machine accepting money**

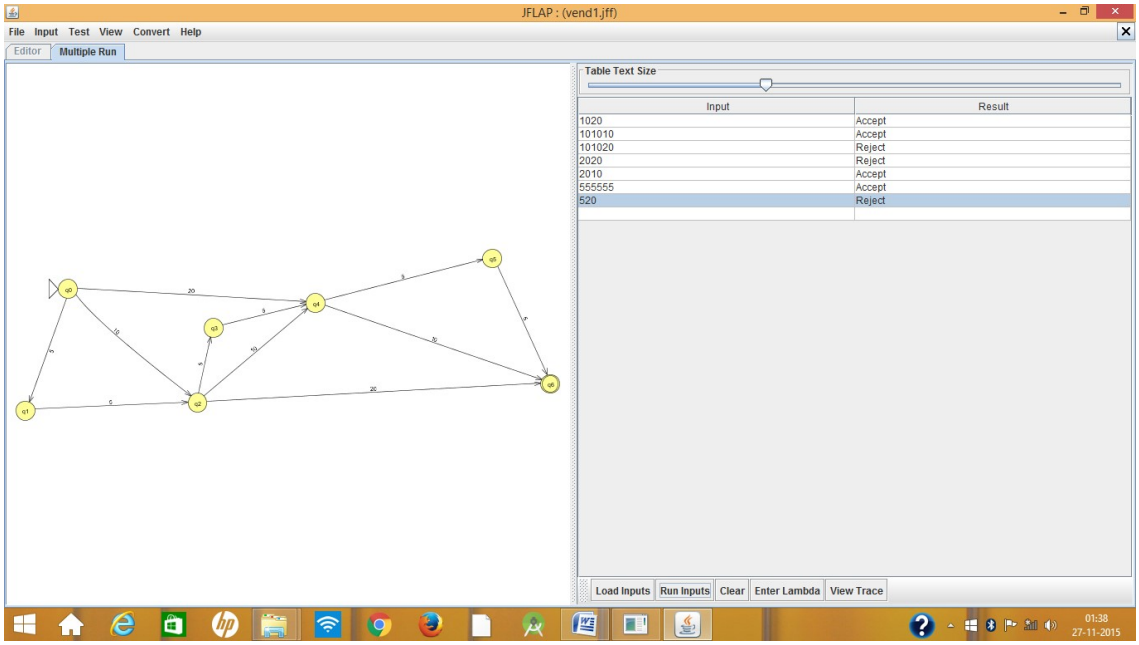


This automata accepts only Rs 30 that might be in 1go or may be in several combinations.



S -> 10B
S -> 5A
D -> 5E
B -> 10D
F -> λ
B -> 5C
D -> 10F
C -> 5D
E -> 5F
B -> 20F
A -> 5B
S -> 20D


JFLAP : (vend1.jff)
File Input Test View Convert Help
Editor Convert FA to RE
Generalized Transition Graph Finished!
(10+55)20+(20+(10+55)10+(10+55)55)10+(20+(10+55)10+(10+55)55)55
Do It Export
q0
q6
(10+55)20+(20+(10+55)10+(10+55)55)10+(20+(10+55)10+(10+55)55)55
ε
Automaton Size
01:36
27-11-2015


JFLAP : (vend1.jff)
File Input Test View Convert Help
Editor Multiple Run
Table Text Size
Input
Result
1020
Accept
101010
Accept
101020
Reject
2020
Reject
2010
Accept
555555
Accept
520
Reject
Load Inputs
Run Inputs
Clear
Enter Lambda
View Trace
01:38
27-11-2015