Automata CW2

Melisa-Andreea Bogdan

ID:190030423

Q1

a) L(G) over Σ ={0,1} Y0->

1Y0 | Y1

Y1-> 0 Y0 | 1 Y1 | Y2

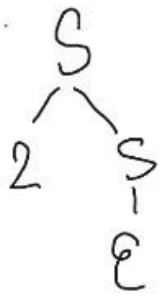
Y2 -> 0 Y2 |1 Y1| ε

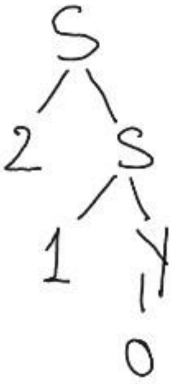
b) i) For 2: For 210 For 210010

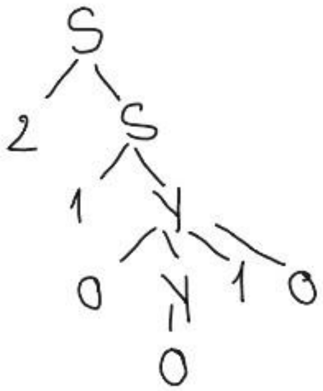
S-> 2S->2 ε -> 2

S-> 1Y-> 10

S-> 2S-> 21Y -> 210Y10 -> 210010







ii) The grammar is not in Chomsky normal form because the rules of this automaton do not respect this rule: CNF grammar must not contain it's start variable in the right hand side of any grammar rule.

Thus, S-> 2S is not respecting the rule and as a consequence the grammar loses the Chomsky properly.

iii)

Since S appears in a RHS rule, we first apply the 'start' routine:

S0 -> S , S0 = start variable

S -> 1Y| 2S | ε

Y -> S 1|0Y10|0

There are several rules on the RHS with length > 2, so we apply the BIN routine:

S0 -> S

S -> 1 Y | 2 S | ε

Y -> S 1| 0 X| 0

X -> Y X1

X1 -> 1 0

We apply the DEL routine to remove the S -> ε rule:

S0 -> S| ε

S -> 1 Y| 2 S | 2

Y -> S 1|0 X| 0 |1

X -> Y X1

X1-> 1 0

Next we apply the UNIT routine to remove the unit rules: S

S0 -> 1 Y| 2 S | 2| ε

S -> 1 Y| 2 S | 2

Y -> S1|0 X| 0 |1

X -> Y X1

X1-> 1 0

Then we apply TERM to remove terminals that appear in binary rules: U1-> 1, U0-> 0, U2-> 2

S0 -> U1 Y| U2 S | 2| ε

S -> U1 Y| U2 S | 2

Y -> S U1|U0 X| 0 |1

X -> Y X1

X1-> U1 U0

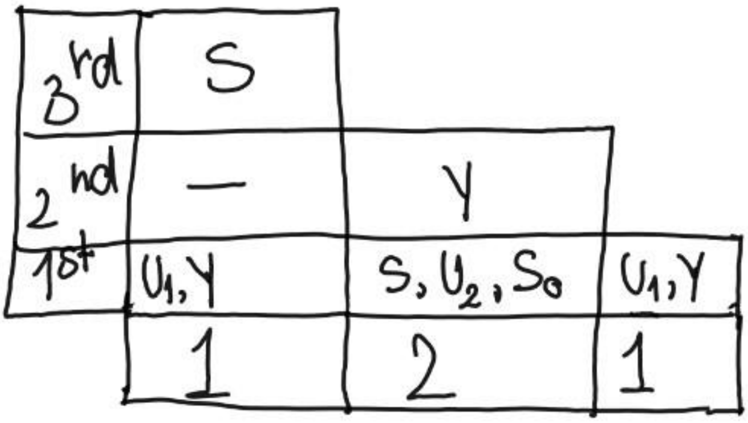
U1-> 1

U0-> 0

U2-> 2

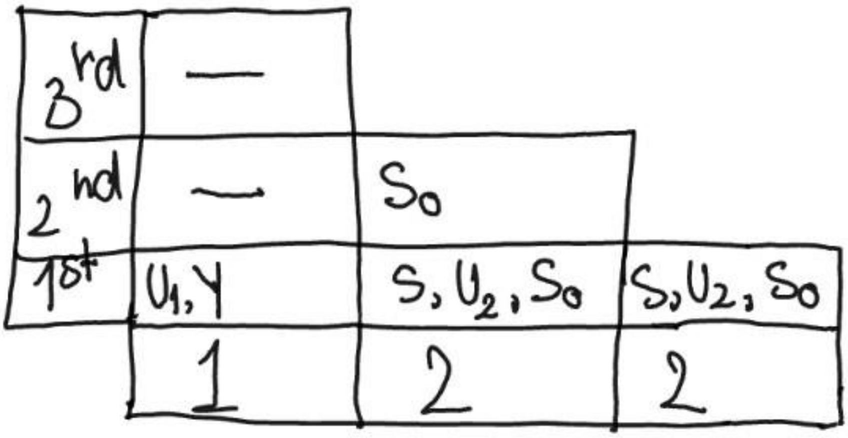
IV) G: 121, 122

For 121: S0-> U1 Y ->U1 S U1->U1 2 U1-> 121



3rd	S		
2nd	—	Y	
1st	U_1, Y	S, U_2, S_0	U_1, Y
	1	2	1

For 122: S0 -> U1 Y , AND THE REST DOESN'T WORK because Y doesn't lead to any word containing '2'.



3rd	—		
2nd	—	S_0	
1st	U_1, Y	S, U_2, S_0	S, U_2, S_0
	1	2	2

c) i)

Σ ={0,1,2,3}

YES	NO (Σ*)
132031	132033
31202013	ε
13120131	12032031

ii) for 132031

S -> 1 S 1-> 1 3 S 3 1 -> 1 3 Y 3 1 -> 1 3 2 0 3 1

For 31202013

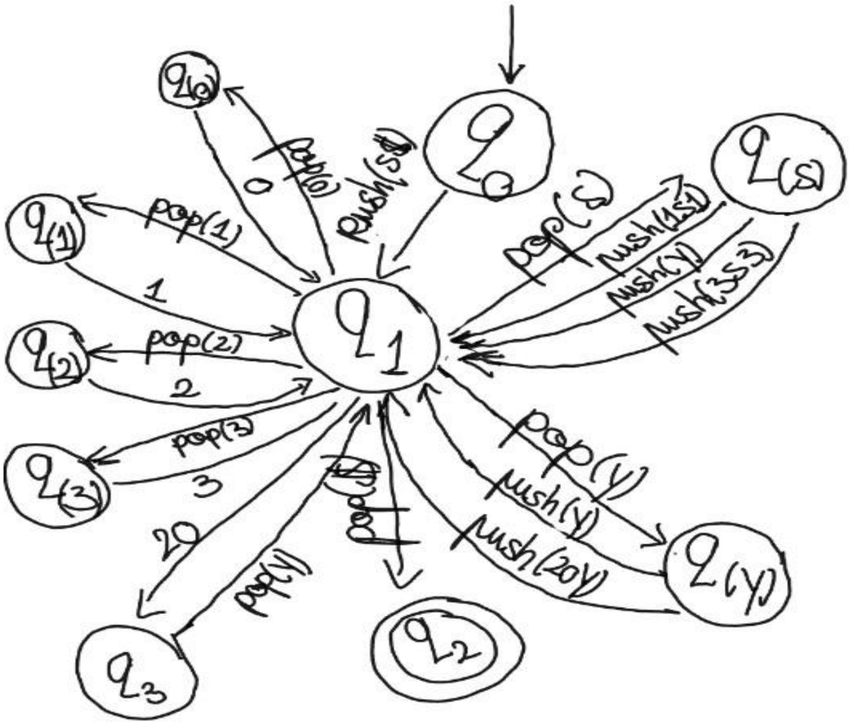
S -> 3 S 3-> 3 1 S 1 3 -> 3 1 Y 1 3 -> 3 1 2 0 Y 1 3

for 13120131

S -> 1 S 1-> 1 3 S 3 1 -> 1 3 1 S 1 3 1 -> 1 3 1 2 0 1 3 1

iii) L(G) is the language of all words (of even length) following this rule: the second half is the first half in a reverse order (it’s like looking at the first half in a mirror) but the only problem would be that the 2’s will be transformed into 0’s and 2’s will be replaced by 0’s.

iv)



The language of the PDA is: A=({1,2,3}, {$,1,2,3}, {q0,q1,q2,q(1),q(2),q(3),q(s)}, δ, q0,q2}

With δ =({q0, push(S$),q1}, {q1, pop(1), q(1)}, {q(1), 1, q1}, {q1, pop(2), q(2)}, {q(2), 2, q1}, {q1, pop(3), q(3)}, {q(3), 3, q1}, {q1, pop($), q2}, {q1, pop(S), q(S)}, {q(S), push(1S1), q1} , {q(S), push(3S3), q1}, {q(S), push(Y), q1}, , {q(S), ε , q1}, {q1, pop(Y), q(Y)}, {q(Y), push(20), q1}, {q(Y), push(20Y), q1}, {q(Y), ε, q1}, {q1, pop($), q2}).

(v) For 132031:

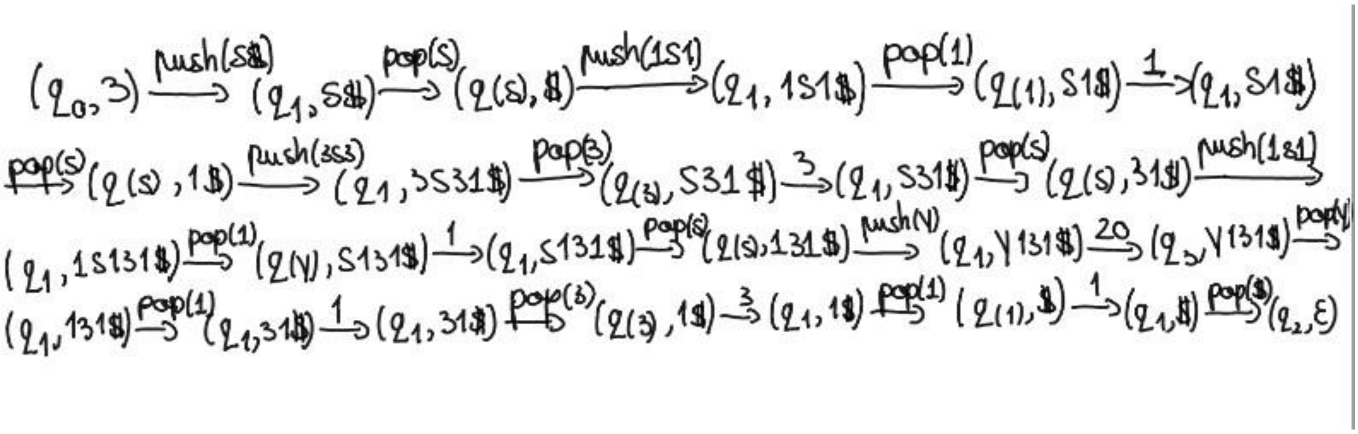
$$(q_0,3) \xrightarrow{push(S\$)} (q_1, S\$) \xrightarrow{pop(S)} (q(S), \$) \xrightarrow{push(1S1)} (q_1, 1S1\$) \xrightarrow{pop(1)} (q(1), S1\$) \xrightarrow{1} (q_1, S1\$)$$

$$\xrightarrow{pop(S)} (q(S), 1\$) \xrightarrow{push(3S3)} (q_1, 3S31\$) \xrightarrow{pop(3)} (q(3), S31\$) \xrightarrow{3} (q_1, S31\$) \xrightarrow{pop(S)} (q(S), 31\$) \xrightarrow{push(Y)}$$

$$(q_1, Y31\$) \xrightarrow{20} (q_3, Y31\$) \xrightarrow{pop(Y)} (q_1, 31\$) \xrightarrow{pop(3)} (q(3), 1\$) \xrightarrow{3} (q_1, 1\$) \xrightarrow{pop(1)} (q(1), \$) \xrightarrow{1} (q_1, \$)$$

$$\xrightarrow{pop(\$)} (q_2, \varepsilon)$$

For 13120131:



$$(q_0, 3) \xrightarrow{push(S\$)} (q_1, S\$) \xrightarrow{pop(S)} (q(S), \$) \xrightarrow{push(1S1)} (q_1, 1S1\$) \xrightarrow{pop(1)} (q(1), S1\$) \xrightarrow{1} (q_1, S1\$)$$

$$\xrightarrow{pop(S)} (q(S), 1\$) \xrightarrow{push(3S3)} (q_1, 3S31\$) \xrightarrow{pop(3)} (q(3), S31\$) \xrightarrow{3} (q_1, S31\$) \xrightarrow{pop(S)} (q(S), 31\$) \xrightarrow{push(1S1)}$$

$$(q_1, 1S131\$) \xrightarrow{pop(1)} (q(Y), S131\$) \xrightarrow{1} (q_1, S131\$) \xrightarrow{pop(S)} (q(S), 131\$) \xrightarrow{push(Y)} (q_1, Y131\$) \xrightarrow{20} (q_3, Y131\$) \xrightarrow{pop(Y)}$$

$$(q_1, 131\$) \xrightarrow{pop(1)} (q_1, 31\$) \xrightarrow{1} (q_1, 31\$) \xrightarrow{pop(3)} (q(3), 1\$) \xrightarrow{3} (q_1, 1\$) \xrightarrow{pop(1)} (q(1), \$) \xrightarrow{1} (q_1, \$) \xrightarrow{pop(\$)} (q_2, \varepsilon)$$

Q2

(a) (i)

```
public static CFG generateCFG1() {
       String[] alphabet = new String[]{ "0", "1", "2" };
       String[] vars = new String[]{ "S", "Y" };        Rule[]
R = new Rule[] {            new Rule("S",new
String[]{"1","Y"}),            new Rule("S",new
String[]{"2","S"}),            new Rule("S",new
String[]{""}),            new Rule("Y",new
String[]{"S","1"}),            new Rule("Y",new
String[]{"0","0","Y","1","0"}),            new Rule("Y",new
String[]{"0"})
       };
       return new CFG(alphabet,vars,R,"S");
   }
```

(ii)

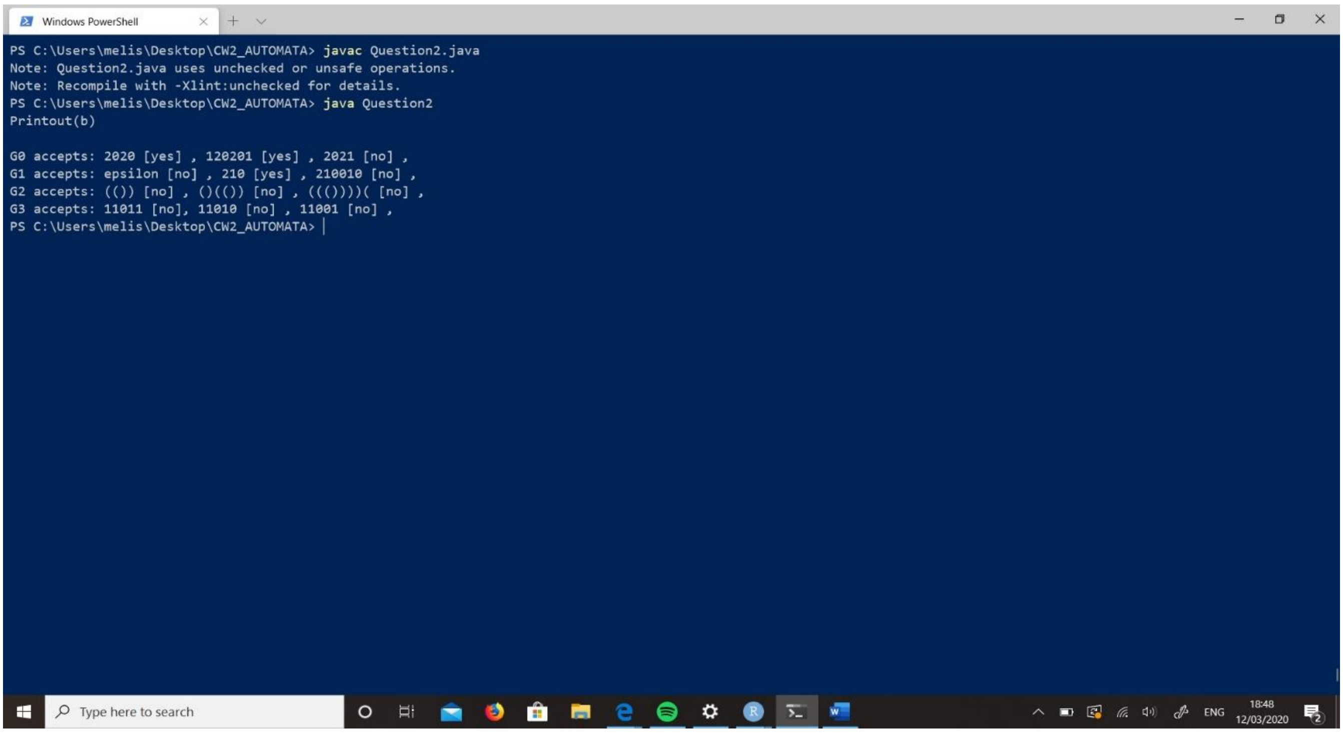
```
public static CFG generateCFG2() {
       String[] alphabet = new String[]{ "(", ")" };
       String[] vars = new String[]{ "S" };
       Rule[] R = new Rule[] {
          new Rule("S",new String[]{""),
new Rule("S",new String[]{"(","S",")"}),
new Rule("Y",new String[]{"S","S"})
       };
```

```
        return new CFG(alphabet,vars,R,"S");
    }
```

(iii)

```
 public static CFG generateCFG3() {
        String[] alphabet = new String[]{ "0", "1", "2", "3" };
        String[] vars = new String[]{ "S", "Y" };
Rule[] R = new Rule[] {            new
Rule("S",new String[]{"3","X"}),
new Rule("S",new String[]{"1","X"}),
new Rule("X",new String[]{""}),
new Rule("X",new String[]{"3","S"}),
new Rule("X",new String[]{"1","Y"}),
new Rule("Y",new String[]{"0","X"}),
new Rule("Y",new String[]{"3","X"}),
new Rule("Y",new String[]{""}),
        };
        return new CFG(alphabet,vars,R,"S");
    }
```

(b) The snapshot of the printout of *runThem* is:



```
PS C:\Users\melis\Desktop\CW2_AUTOMATA> javac Question2.java
Note: Question2.java uses unchecked or unsafe operations.
Note: Recompile with -Xlint:unchecked for details.
PS C:\Users\melis\Desktop\CW2_AUTOMATA> java Question2
Printout(b)

G0 accepts: 2020 [yes] , 120201 [yes] , 2021 [no] ,
G1 accepts: epsilon [no] , 210 [yes] , 210010 [no] ,
G2 accepts: (()) [no] , ()(()) [no] , ((())))( [no] ,
G3 accepts: 11011 [no], 11010 [no] , 11001 [no] ,
PS C:\Users\melis\Desktop\CW2_AUTOMATA> |
```

The code for this implementation is:

```
public static void runThem(){
        CFG G0=generateCFG0();
        CFG G1=generateCFG1();
        CFG G2=generateCFG2();
        CFG G3=generateCFG3();
        // first CFG: G0
        String [] alphabet1=new String []{"2","0", "2","0"};
        String [] alphabet2=new String []{"1","2","0", "2","0","1"};
        String [] alphabet3=new String[]{"2","0", "2","1"};

          System.out.println("");
          System.out.print("G0"+" accepts: "+ "2020 [");
if( isAccepted(G0,alphabet1) == true)
            {
                System.out.print("yes");
            } else
            {
                System.out.print("no");
            }
             System.out.print("] ,");

             System.out.print(" 120201 [" );
if( isAccepted(G0,alphabet2) == true)
            {
                System.out.print("yes");
            } else
            {
                System.out.print("no");
            }
             System.out.print("] ,");
             System.out.print(" 2021 [" );
if( isAccepted(G0,alphabet3) == true)
          {
             System.out.print("yes");
          } else
          {
             System.out.print("no");
          }
           System.out.print("] ,");

       // second CFG: G1
       String [] alphabet11=new String []{ };
       String [] alphabet22=new String []{"2","1","0"};
       String [] alphabet33=new String[]{"2","1","0", "0","1","0"};

       System.out.println("");
         System.out.print("G1"+" accepts: "+ "epsilon [");
          if( isAccepted(G1,alphabet11) == true)
          {
             System.out.print("yes");
          } else
          {
             System.out.print("no");
          }
           System.out.print("] ,");

           System.out.print(" 210 [" );
 if( isAccepted(G1,alphabet22) == true)
          {
             System.out.print("yes");
          } else
          {
             System.out.print("no");
          }
           System.out.print("] ,");

             System.out.print(" 210010 [" );
if( isAccepted(G1,alphabet33) == true)
          {
             System.out.print("yes");
          } else
          {
             System.out.print("no");
          }
           System.out.print("] ,");

           // 3rd CFG :G2
      System.out.println("");
         System.out.print("G2"+" accepts: "+ "(()) [");
          if( isAccepted(G2,new String[]{"(","(", ")",")"}) == true)
          {
             System.out.print("yes");
          } else
          {
             System.out.print("no");
          }
           System.out.print("] ,");

           System.out.print(" ()(()) [" );
          if( isAccepted(G2,new String[]{"(",")","(", "(",")",")"}) == true)
          {
             System.out.print("yes");
          } else
          {
             System.out.print("no");
          }
           System.out.print("] ,");


             System.out.print(" ((()))( [" );
          if( isAccepted(G2,new String[]{"(","(", "(",")", ")",")",")","("}) == true)
          {
             System.out.print("yes");
          } else
          {
             System.out.print("no");
          }
           System.out.print("] ,");


           //4th CFG:  G3
      String [] alphabet1111=new String[]{"1","1","0","1","1"};
      String [] alphabet2222=new String[]{"1","1","0","1","1"};
      String [] alphabet3333=new String[]{"1","1","0", "0","1"};


       System.out.println("");
         System.out.print("G3"+" accepts: "+ "11011 [");
if( isAccepted(G1,alphabet1111) == true)
          {
             System.out.print("yes");
          } else
          {
             System.out.print("no");
          }
           System.out.print("],");


           System.out.print(" 11010 [" );
   if( isAccepted(G1,alphabet2222) == true)
          {
             System.out.print("yes");
          } else
          {
             System.out.print("no");
          }
           System.out.print("] ,");


            System.out.print(" 11001 [" );
 if( isAccepted(G1,alphabet3333) == true)
          {
             System.out.print("yes");
          } else
          {
             System.out.print("no");
          }
           System.out.print("] ,");
   }
```

(c) I implemented my code following these rules:

If its rules have one of these forms then the CFG is not in CFN and the method returns false immediately after it finds one of these rules applied:

- a rule X → ε where X is not the start variable.
- a rule X → w where w has length greater than 1 and contains some x from the alphabet.
- a rule X → Y where Y is a variable
- a rule X → w where w has length greater than 2
- a rule X →(var) S (var) where S is the start variable

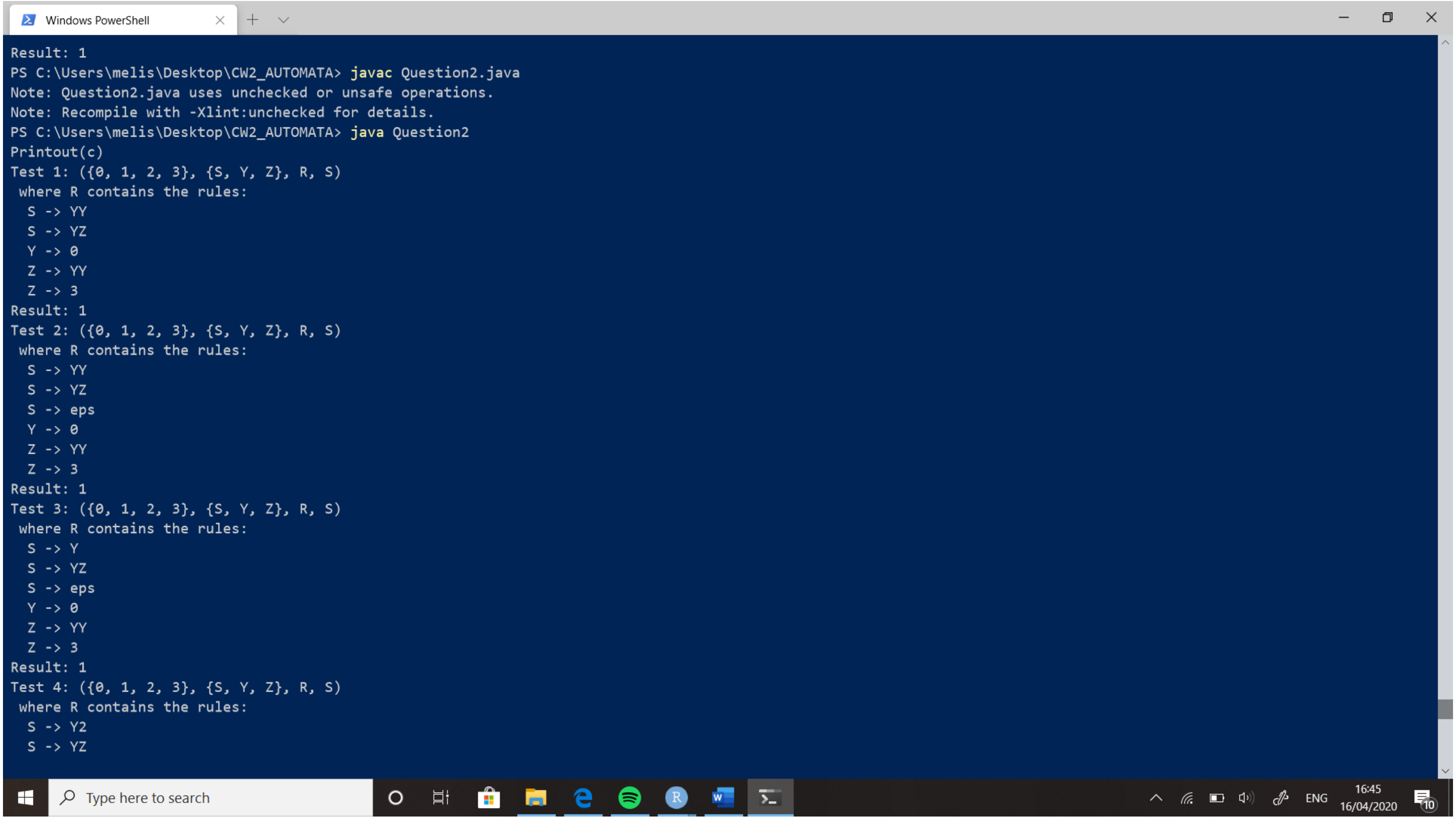
Otherwise, it returns true at the end of the program which means the CFG is in CFN.

The code of my implementation is:

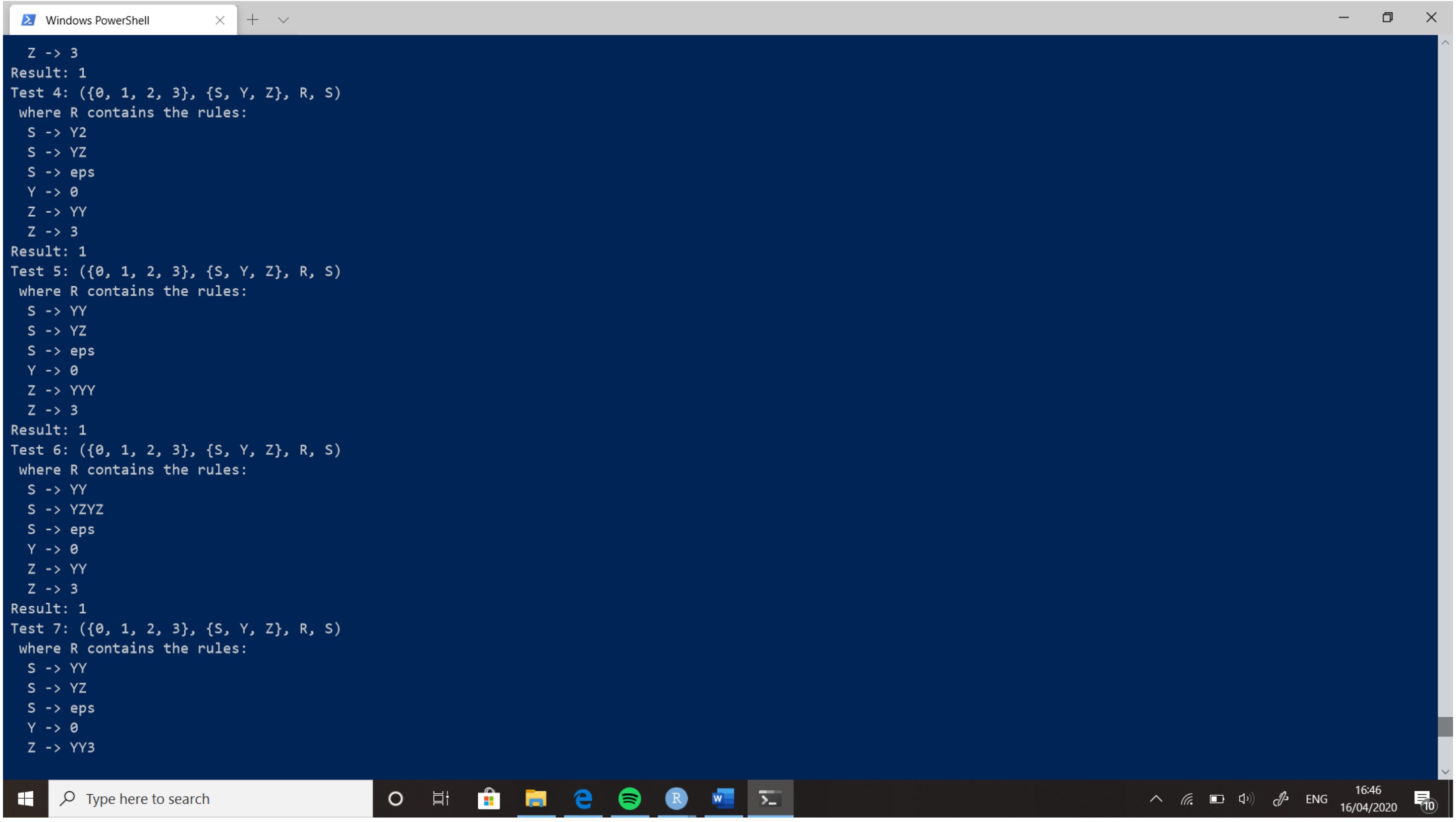
```
public static Boolean isInCNF(CFG G)  {
      // TODO return true if G is in Chomsky-Normal Form
      // and false otherwise



      for(int j = 0; j < G.rules.length; j++) {
         // a rule X → ε where X is not the start variable
```

```
       if(!G.rules[j].var.equals(G.startVar) && G.rules[j].rhs.length==0)  {
               return false;
          }
     for(int i = 0; i < G.rules[j].rhs.length; i++)
    {
     //a rule  X → … S …  where S is the start variable
     String  s = G.rules[j].rhs[i];
          if(s.equals(G.startVar)) {
             return false;
          }

         if(G.rules[j].rhs.length > 2) {
         //a rule X → w  where w has length greater than 2
             return false;
         }

        if(G.rules[j].rhs.length > 1 && G.checkLetter(s)) {
         //  a rule X → w where w has length greater than 1 and contains some x from Σ.
                    return false;
         }

          if(G.rules[j].rhs.length == 1 && G.checkVar(s)) {
         // a rule X → Y where Y is a variable
              return false;
             }
          }
     }
   return true;
}
```

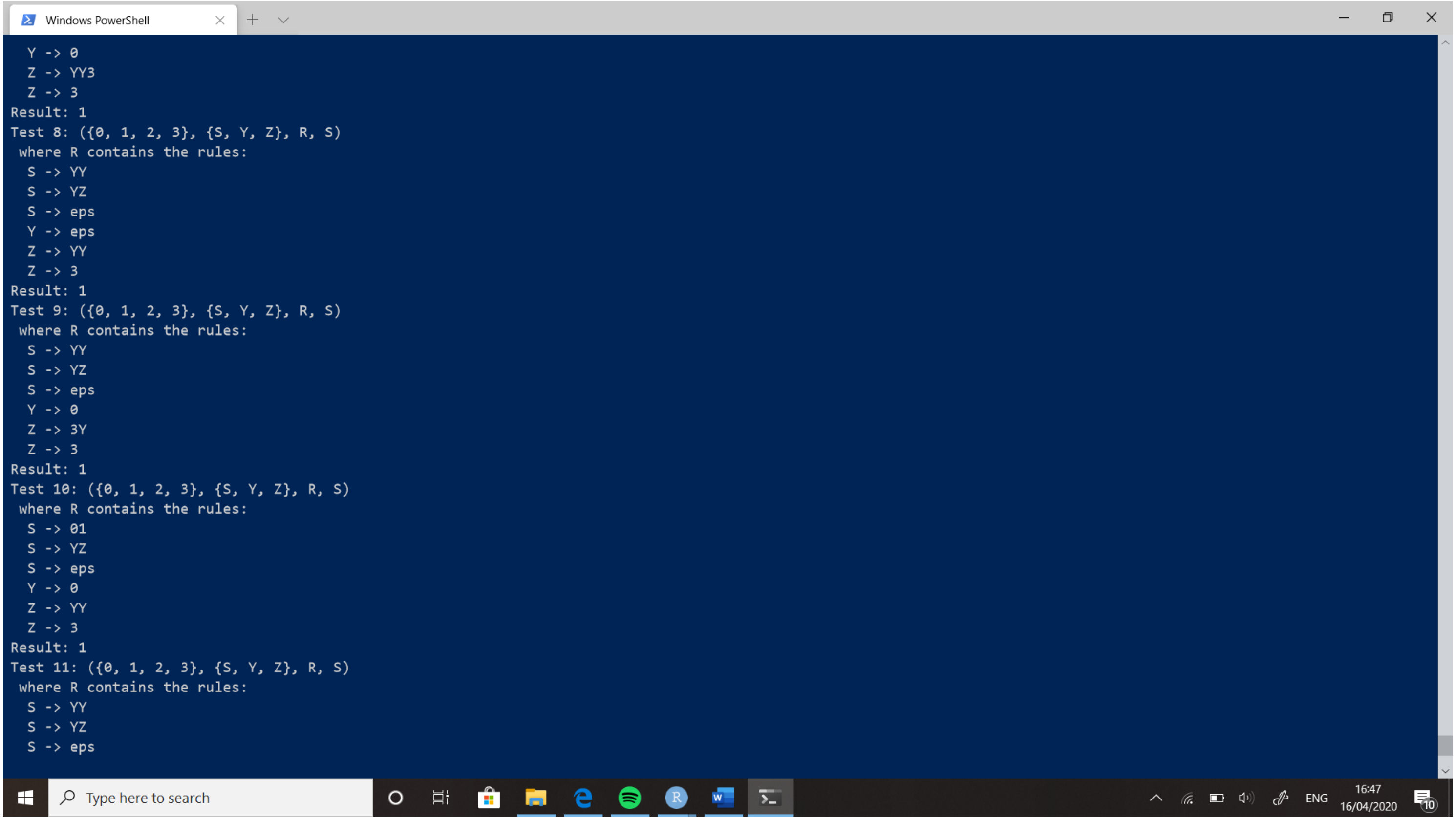
The printout of my code can be found in the photos above:



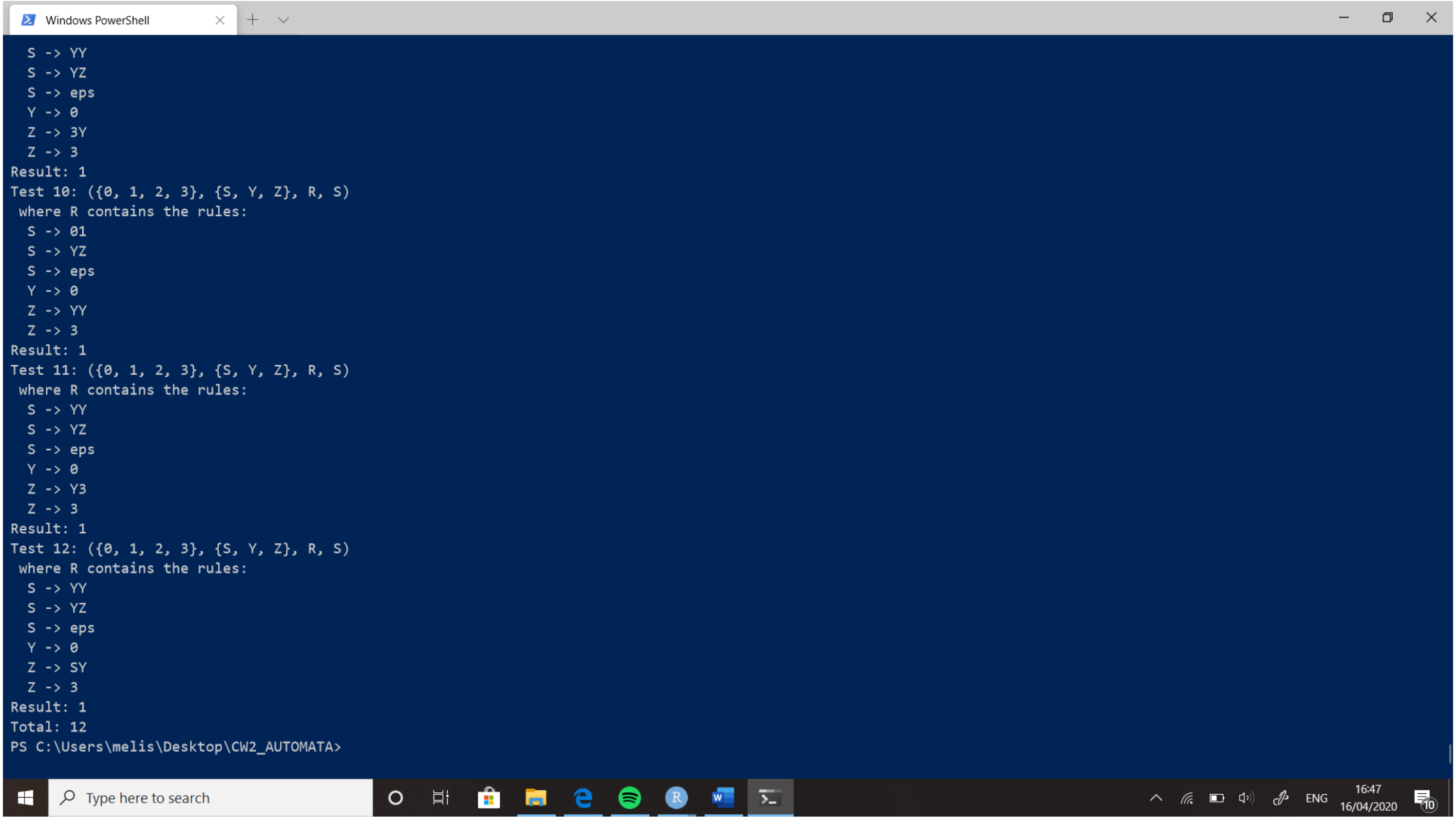
```
Result: 1
PS C:\Users\melis\Desktop\CW2_AUTOMATA> javac Question2.java
Note: Question2.java uses unchecked or unsafe operations.
Note: Recompile with -Xlint:unchecked for details.
PS C:\Users\melis\Desktop\CW2_AUTOMATA> java Question2
Printout(c)
Test 1: ({0, 1, 2, 3}, {S, Y, Z}, R, S)
 where R contains the rules:
  S -> YY
  S -> YZ
  Y -> 0
  Z -> YY
  Z -> 3
Result: 1
Test 2: ({0, 1, 2, 3}, {S, Y, Z}, R, S)
 where R contains the rules:
  S -> YY
  S -> YZ
  S -> eps
  Y -> 0
  Z -> YY
  Z -> 3
Result: 1
Test 3: ({0, 1, 2, 3}, {S, Y, Z}, R, S)
 where R contains the rules:
  S -> Y
  S -> YZ
  S -> eps
  Y -> 0
  Z -> YY
  Z -> 3
Result: 1
Test 4: ({0, 1, 2, 3}, {S, Y, Z}, R, S)
 where R contains the rules:
  S -> Y2
  S -> YZ
```



```
  Z -> 3
Result: 1
Test 4: ({0, 1, 2, 3}, {S, Y, Z}, R, S)
 where R contains the rules:
  S -> Y2
  S -> YZ
  S -> eps
  Y -> 0
  Z -> YY
  Z -> 3
Result: 1
Test 5: ({0, 1, 2, 3}, {S, Y, Z}, R, S)
 where R contains the rules:
  S -> YY
  S -> YZ
  S -> eps
  Y -> 0
  Z -> YYY
  Z -> 3
Result: 1
Test 6: ({0, 1, 2, 3}, {S, Y, Z}, R, S)
 where R contains the rules:
  S -> YY
  S -> YZYZ
  S -> eps
  Y -> 0
  Z -> YY
  Z -> 3
Result: 1
Test 7: ({0, 1, 2, 3}, {S, Y, Z}, R, S)
 where R contains the rules:
  S -> YY
  S -> YZ
  S -> eps
  Y -> 0
  Z -> YY3
```



```
  Y -> 0
  Z -> YY3
  Z -> 3
Result: 1
Test 8: ({0, 1, 2, 3}, {S, Y, Z}, R, S)
 where R contains the rules:
  S -> YY
  S -> YZ
  S -> eps
  Y -> eps
  Z -> YY
  Z -> 3
Result: 1
Test 9: ({0, 1, 2, 3}, {S, Y, Z}, R, S)
 where R contains the rules:
  S -> YY
  S -> YZ
  S -> eps
  Y -> 0
  Z -> 3Y
  Z -> 3
Result: 1
Test 10: ({0, 1, 2, 3}, {S, Y, Z}, R, S)
 where R contains the rules:
  S -> 01
  S -> YZ
  S -> eps
  Y -> 0
  Z -> YY
  Z -> 3
Result: 1
Test 11: ({0, 1, 2, 3}, {S, Y, Z}, R, S)
 where R contains the rules:
  S -> YY
  S -> YZ
  S -> eps
```



```
  S -> YY
  S -> YZ
  S -> eps
  Y -> 0
  Z -> 3Y
  Z -> 3
Result: 1
Test 10: ({0, 1, 2, 3}, {S, Y, Z}, R, S)
 where R contains the rules:
  S -> 01
  S -> YZ
  S -> eps
  Y -> 0
  Z -> YY
  Z -> 3
Result: 1
Test 11: ({0, 1, 2, 3}, {S, Y, Z}, R, S)
 where R contains the rules:
  S -> YY
  S -> YZ
  S -> eps
  Y -> 0
  Z -> Y3
  Z -> 3
Result: 1
Test 12: ({0, 1, 2, 3}, {S, Y, Z}, R, S)
 where R contains the rules:
  S -> YY
  S -> YZ
  S -> eps
  Y -> 0
  Z -> SY
  Z -> 3
Result: 1
Total: 12
PS C:\Users\melis\Desktop\CW2_AUTOMATA>
```