public static Boolean isComplete(FSA A) {  
 int FSA\_Length = A.delta.length;  
 int alph\_Length = A.alphabet.length;  
 System.*out*.println(alph\_Length+"test");  
 Boolean success;  
 boolean label\_check\_0=false;  
 boolean label\_check\_1=false;  
 boolean label\_check\_2=false;  
 boolean q0\_success=false;  
 boolean q1\_success=false;  
 boolean q2\_success=false;  
 for(int k=0; k<FSA\_Length;k++)  
 {  
 if(A.delta[k].from==0)  
 {  
 for(int j=0; j<FSA\_Length; j++)  
 {  
 if(A.delta[j].from==0&&A.delta[j].label.equals("0"))  
 {  
 label\_check\_0=true;  
 }  
 else if(A.delta[j].from==0&&A.delta[j].label.equals("1"))  
 {  
 label\_check\_1=true;  
 }  
 else if(A.delta[j].from==0&&A.delta[j].label.equals("2")) {  
 label\_check\_2 = true;  
 }  
 }  
 if(label\_check\_1&&label\_check\_0&&label\_check\_2)  
 {  
 q0\_success=true;  
 }  
 else  
 {  
 q0\_success=false;  
 }  
 }  
 else if(A.delta[k].from==1)  
 {  
 for(int j=0; j<FSA\_Length; j++)  
 {  
 if(A.delta[j].from==1&&A.delta[j].label.equals("0"))  
 {  
 label\_check\_0=true;  
 }  
 else if(A.delta[j].from==1&&A.delta[j].label.equals("1"))  
 {  
 label\_check\_1=true;  
 }  
 else if(A.delta[j].from==1&&A.delta[j].label.equals("2")) {  
 label\_check\_2 = true;  
 }  
 }  
 if(label\_check\_1&&label\_check\_0&&label\_check\_2)  
 {  
 q1\_success=true;  
 }  
 else  
 {  
 q1\_success=false;  
 }  
 }  
  
 else if(A.delta[k].from==2)  
 {  
 for(int j=0; j<FSA\_Length; j++)  
 {  
 if(A.delta[j].from==2&&A.delta[j].label.equals("0"))  
 {  
 label\_check\_0=true;  
 }  
 else if(A.delta[j].from==2&&A.delta[j].label.equals("1"))  
 {  
 label\_check\_1=true;  
 }  
 else if(A.delta[j].from==2&&A.delta[j].label.equals("2")) {  
 label\_check\_2 = true;  
 }  
 }  
 if(label\_check\_1&&label\_check\_0&&label\_check\_2)  
 {  
 q2\_success=true;  
 }  
 else  
 {  
 q2\_success=false;  
 }  
 }  
  
  
 }  
  
 if(q0\_success&&q1\_success&&q2\_success)  
 {  
 success=true;  
 }  
 else  
 {  
 success=false;  
 }  
  
  
 return success; *// TODO remove this*}

public static FSA complement(FSA A) {  
  
 int alph\_Length = A.alphabet.length;  
 int FSA\_Length = A.delta.length;  
 int final\_length = A.finalStates.length;  
 int[] state\_array = new int[A.numStates];  
 if(A.numStates==2)  
 {  
 for(int k=0; k<A.numStates-1; k++)  
 {  
 A.finalStates[k]=A.delta[k].from;  
 }  
 }  
 else  
 {  
 for(int j=0; j<A.numStates; j++)  
 {  
 for(int k=0; k<FSA\_Length; k++) {  
 if (k != FSA\_Length - 1) {  
 if (A.delta[k].from != A.delta[k + 1].from) {  
 state\_array[j] = A.delta[k].from;  
 }  
 }  
 }  
 }  
 System.*out*.println(state\_array[0]);  
 System.*out*.println(state\_array[1]);  
 System.*out*.println(state\_array[2]);  
 for(int i=0; i< A.numStates; i++)  
 {  
  
 if(state\_array[i]!=A.finalStates[i])  
 {  
 A.finalStates[i]=state\_array[i];  
 }  
 }  
 }

Graphical user interface, text, application

Description automatically generated

public static FSA generateFSA0() {  
 String[] alphabet = new String[]{ "0", "1", "2" };  
 Transition[] delta = new Transition[] {   
 new Transition(0,"0",0),   
 new Transition(0,"0",1),  
 new Transition(0,"2",2),  
 new Transition(1,"1",2),  
 new Transition(2,"0",2),  
 new Transition(2,"1",2)  
 };  
 int[] finals = new int[] { 2 };  
 FSA A = new FSA(3,alphabet,delta,finals);  
 return A;  
}  
  
*// -----------------------------------  
// TODO CODE (REQUIRES IMPLEMENTATION)  
// -----------------------------------  
  
// TODO construct FSA of Question 13 and return it*public static FSA generateFSA1()  
{  
 String[] alphabet = new String[]{ "0", "1", "2" };  
 Transition[] delta = new Transition[] {  
 new Transition(0,"3",2),  
 new Transition(0,"3",5),  
 new Transition(1,"2",2),  
 new Transition(2,"2",1),  
 new Transition(2,"2",3),  
 new Transition(3,"3",4),  
 new Transition(4,"2",2),  
 new Transition(5,"2",3),  
  
  
 };  
 int[] finals = new int[] { 3,4 };  
 FSA A = new FSA(6,alphabet,delta,finals);  
 return A;  
  
}  
  
*// TODO construct FSA of Question 14 and return it*public static FSA generateFSA2() {  
 String[] alphabet = new String[]{ "0", "1", "2" };  
 Transition[] delta = new Transition[] {  
 new Transition(0,"3",2),  
 new Transition(0,"5",2),  
 new Transition(2,"3",2),  
 new Transition(2,"1",1),  
 new Transition(1,"1",1),  
 new Transition(1,"5",0),  
  
  
  
 };  
 int[] finals = new int[] { 2};  
 FSA A = new FSA(3,alphabet,delta,finals);  
 return A;  
}  
  
*// TODO implement this for Question 15*public static void runFSA(FSA A, String name, Word input) {  
  
 if(A.isAccepted(input))  
 {  
 System.*out*.println(name+" accepts: "+input+" [yes]");  
 }  
 else  
 {  
 System.*out*.println(name+" accepts: "+input+" [no]");  
 }  
}

Graphical user interface, text, application, email

Description automatically generated