TEORÍA DE AUTÓMATAS Y LENGUAJES FORMALES

ALEJANDRO SERRANO FERNÁNDEZ

LAB SESSION 4

STATEMENT

A metallurgical company needs your help! There has been an incident and the database files have been mixed up.
They need the steel's sheet ASAP! And they know you have the necessary capacities for it!

The company has given you a comma separated values file with all the data mixed in it, they say the data they need have the following structure:
Element1,Element2,Tolerance
An explanation of every field is detailed next:

1. Element 1: is a real number in the range between 0 and 2 both included.
2. Element 2: is a real number in the range between -1 and 2, both included.
3. Tolerance: is an integer number between 1 and 4.

The company needs you to perform the following task:

Task 1: Design a Regular Expression (as defined in theory lectures) to check the entries of their databases, in order to discover the right ones. Define the alphabet of the language represented by the RE.

E1 = 0+1;
E2 = 2
E3 = -1 + 2
E4 = 1 + 2 + 3 + 4 (Tolerance)
E5 = 1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9
E6 = 0 + 1 + -0

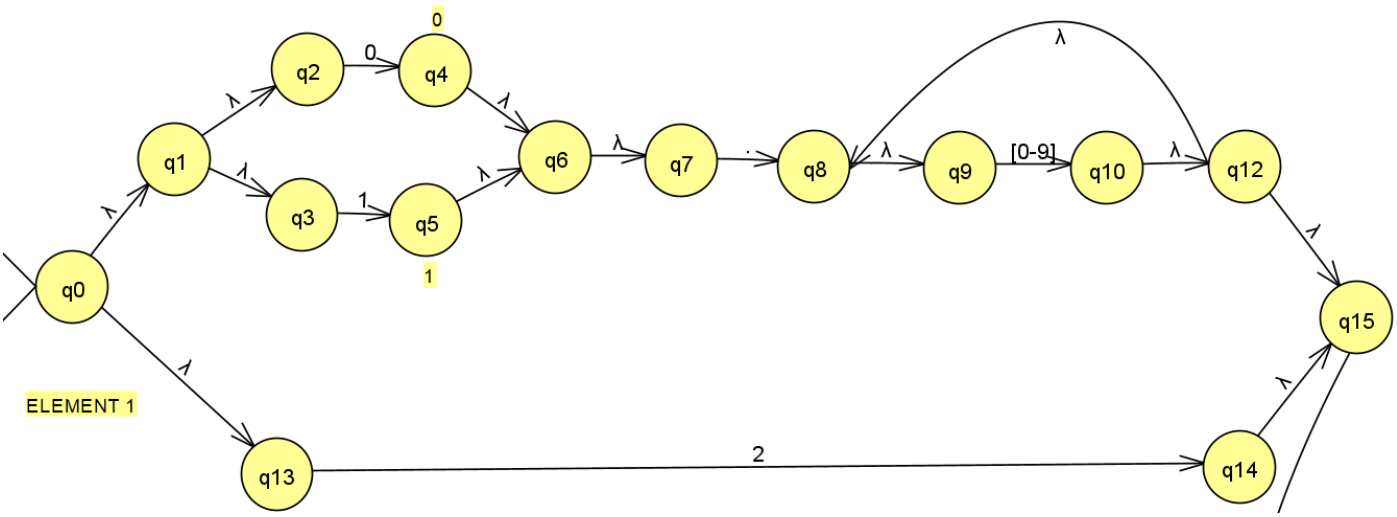
ELEMENT 1 = $(E1, E5^{+}) + (E2)$
ELEMENT 2 = $(E6,E5^{+}) + (E3)$
TOLERANCE = E4

RESULTING REGULAR EXPRESION:

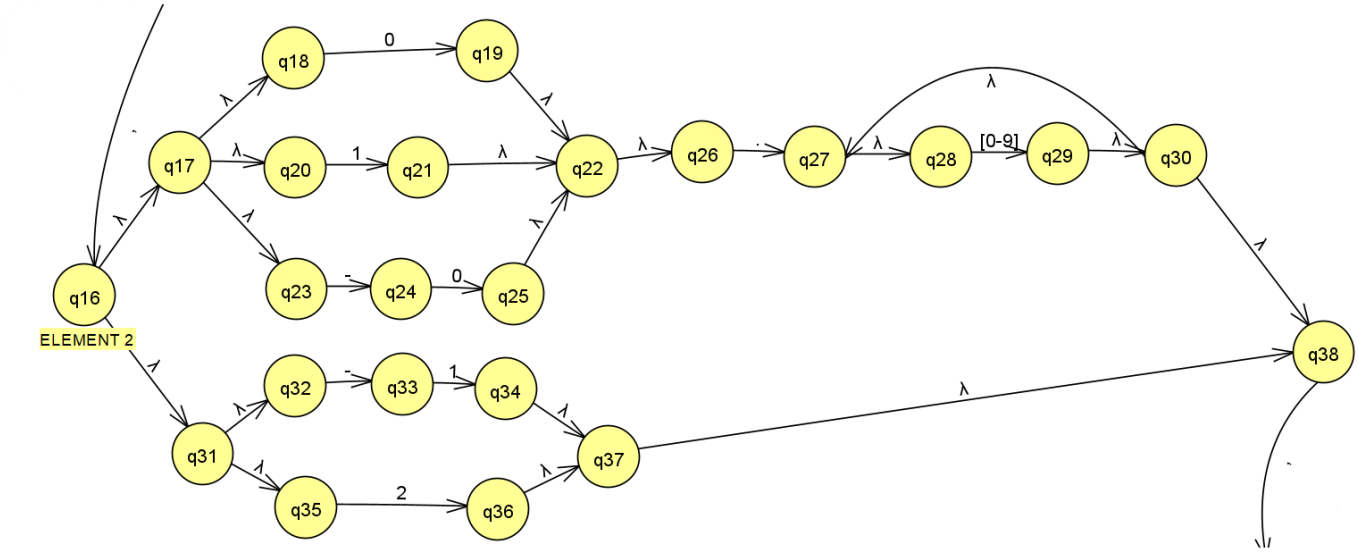
$(((E1, E5^{+}) + (E2)) , ((E6,E5^{+}) + (E3)) , E4)$

Task 2: Convert the obtained RE to an Ɛ-NFA following the Thompson construction method.

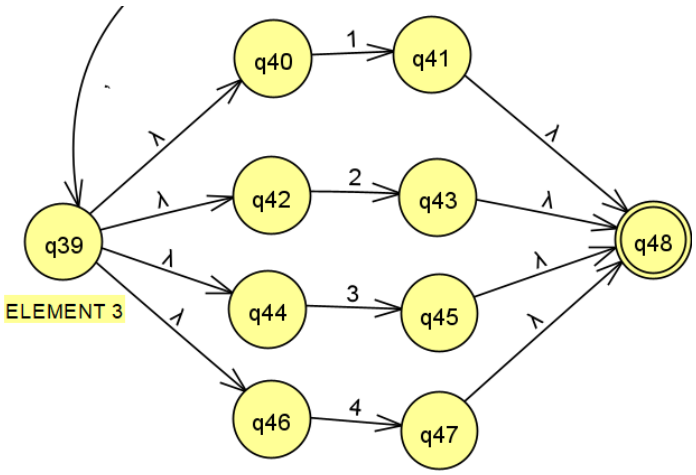
FIRST ELEMENT ε-NFA:



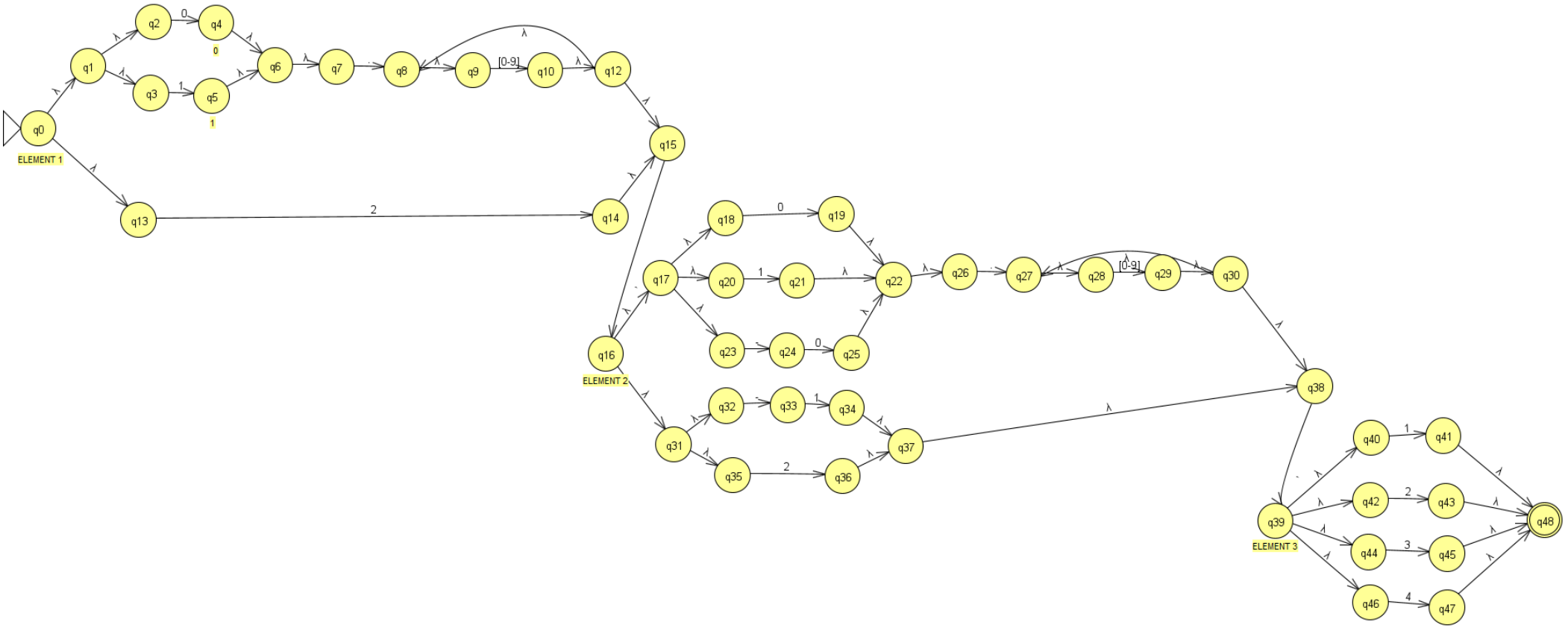
SECOND ELEMENT ε-NFA:



THIRD ELEMENT ε-NFA:



COMPLETE ε-NFA:



Task 3: Get the minimum DFA equivalent to the designed ε-NFA

First we copy the ε-NFA in the table:

Note: All white spaces are Ø states

	0	1	2	3	4	5	6	7	8	9	,	.	-	e
>Q0														Q1,q13
Q1														Q2,q3
Q2	Q4													
Q3		Q5												
Q4														Q6
Q5														Q6
Q6														Q7
Q7												Q8		
Q8														Q9
Q9	Q10	Q10	Q10	Q10	Q10	Q10	Q10	Q10	Q10	Q10				
Q10														Q12
Q12														Q8,q15
Q13			Q14											
Q14														Q15
Q15											Q16			
Q16														Q17,q31
Q17														Q18 q20,q23
Q18	Q19													
Q19														Q22
Q20		Q21												
Q21														Q22
Q22														Q26
Q23													Q24	
Q24	Q25													
Q25														Q22
Q26												Q27		
Q27														Q28
Q28	Q29	Q29	Q29	Q29	Q29	Q29	Q29	Q29	Q29	Q29				
Q29														Q30
Q30														Q38,q27
Q31														Q32,q35
Q32													Q33	
Q33		Q34												
Q34														Q37
Q35			Q36											
Q36														Q37
Q37														Q38
Q38											Q39			
Q39														Q40,q42 Q44,q46

Q40		Q41												
Q41														Q48
Q42			Q43											
Q43														Q48
Q44				Q45										
Q45														Q48
Q46					Q47									
Q47														Q48
*Q48														

Second, we compute all CL(Q)

>Q0	Q0,Q1,q2,q3,q13
CL(Q1)	Q1,q2,q3
CL(Q2)	Q2
CL(Q3)	Q3
CL(Q4)	Q4,q6,q7
CL(Q5)	Q5,q6,q7
CL(Q6)	Q6,q7
CL(Q7)	Q7
CL(Q8)	Q8,q9
CL(Q9)	Q9
CL(Q10)	Q10,q12,q8,q9,q15
CL(Q12)	Q12,q8,q9,q15
CL(Q13)	Q13
CL(Q14)	Q14,q15
CL(Q15)	Q15
CL(Q16)	Q16,q17,q18,q20,q23,q31,q32,q35
CL(Q17)	Q17,q18,q20,q23
CL(Q18)	Q18
CL(Q19)	Q19,q22,q26
CL(Q20)	Q20
CL(Q21)	Q21,q22,q26
CL(Q22)	Q22,q26
CL(Q23)	Q23
CL(Q24)	Q24
CL(Q25)	Q25,q22,q26
CL(Q26)	Q26
CL(Q27)	Q27,q28
CL(Q28)	Q28
CL(Q29)	Q29,q30,q38,q27,q28
CL(Q30)	Q30,q38,q27,q28
CL(Q31)	Q31,q32,q35
CL(Q32)	Q32
CL(Q33)	Q33
CL(Q34)	Q34,q37,q38,
CL(Q35)	Q35
CL(Q36)	Q36,q37,q38
CL(Q37)	Q37,q38
CL(Q38)	Q38
CL(Q39)	Q39,q40,q42,q44,q46
CL(Q40)	Q40
*CL(Q41)	Q41,q48
CL(Q42)	Q42
*CL(Q43)	Q43,q48

CL(Q44)	Q44
*CL(Q45)	Q45,q48
CL(Q46)	Q46
*CL(Q47)	Q47,q48
*CL(Q48)	Q48

Then, calculate $\delta_N(q, a)$ ($\delta_N(q, a)$ is the union over all $p \in S$ of $\delta_E(p, a)$ is the union over all $p \in S$ of $\delta_E(p, a)$)

	0	1	2	3	4	5	6	7	8	9	,	.	-
>Q0	Q4	Q5	Q14										
Q1	Q4	Q5											
Q2	Q4												
Q3		Q5											
Q4												Q8	
Q5												Q8	
Q6												Q8	
Q7												Q8	
Q8	Q10	Q10	Q10	Q10	Q10	Q10	Q10	Q10	Q10	Q10			
Q9	Q10	Q10	Q10	Q10	Q10	Q10	Q10	Q10	Q10	Q10			
Q10	Q10	Q10	Q10	Q10	Q10	Q10	Q10	Q10	Q10	Q10	Q16		
Q12	Q10	Q10	Q10	Q10	Q10	Q10	Q10	Q10	Q10	Q10	Q16		
Q13			Q14										
Q14											Q16		
Q15											Q16		
Q16	Q19	Q21	Q36										Q24,Q33
Q17	Q19	Q21											Q24
Q18	Q19												
Q19												Q27	
Q20		Q21											
Q21												Q27	
Q22												Q27	
Q23													Q24
Q24	Q25												
Q25												Q27	
Q26												Q27	
Q27	Q29	Q29	Q29	Q29	Q29	Q29	Q29	Q29	Q29	Q29			
Q28	Q29	Q29	Q29	Q29	Q29	Q29	Q29	Q29	Q29	Q29			
Q29	Q29	Q29	Q29	Q29	Q29	Q29	Q29	Q29	Q29	Q29	Q39		
Q30	Q29	Q29	Q29	Q29	Q29	Q29	Q29	Q29	Q29	Q29	Q39		
Q31			Q36										Q33
Q32													Q33
Q33		Q34											
Q34											Q39		
Q35			Q36										
Q36											Q39		
Q37											Q39		
Q38											Q39		
Q39		Q41	Q43	Q45	Q47								
Q40		Q41											
*Q41													-

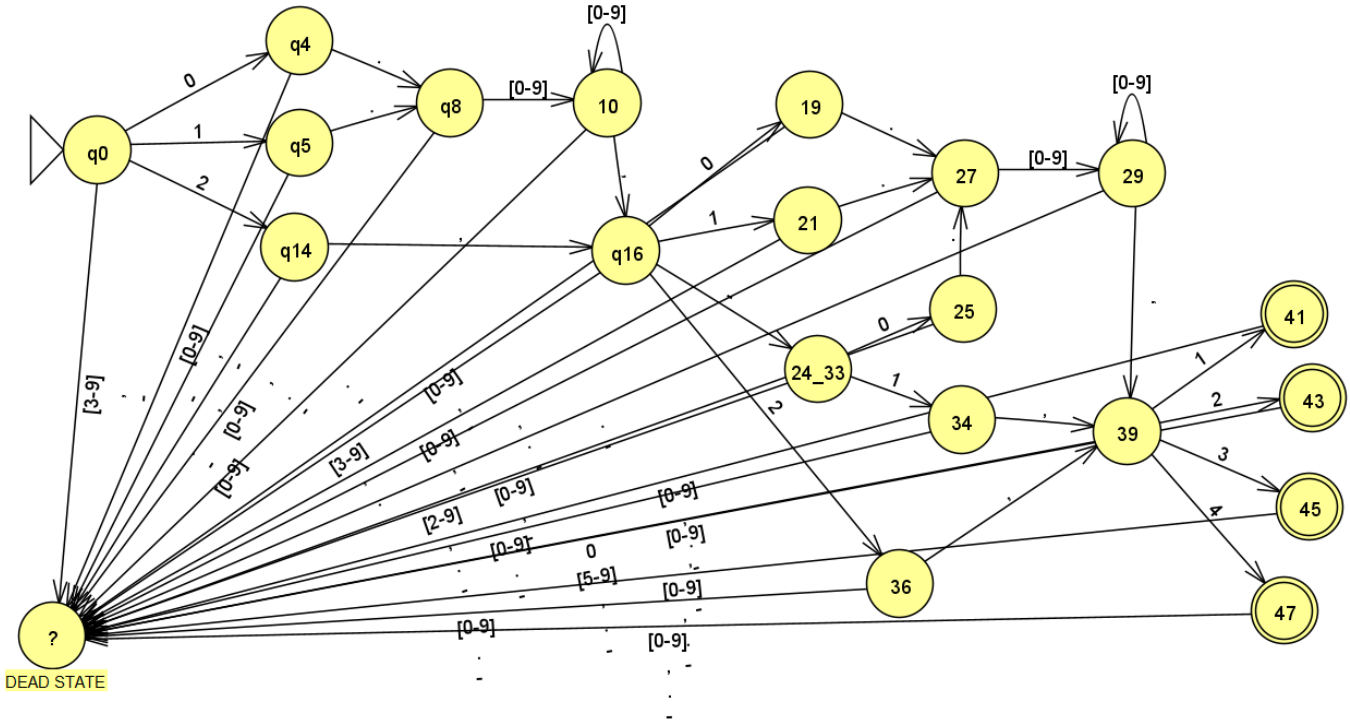
Q42			Q43										
*Q43													
Q44				Q45									
*Q45													
Q46					Q47								
*Q47													
*Q48													

Note: All white spaces are Ø states

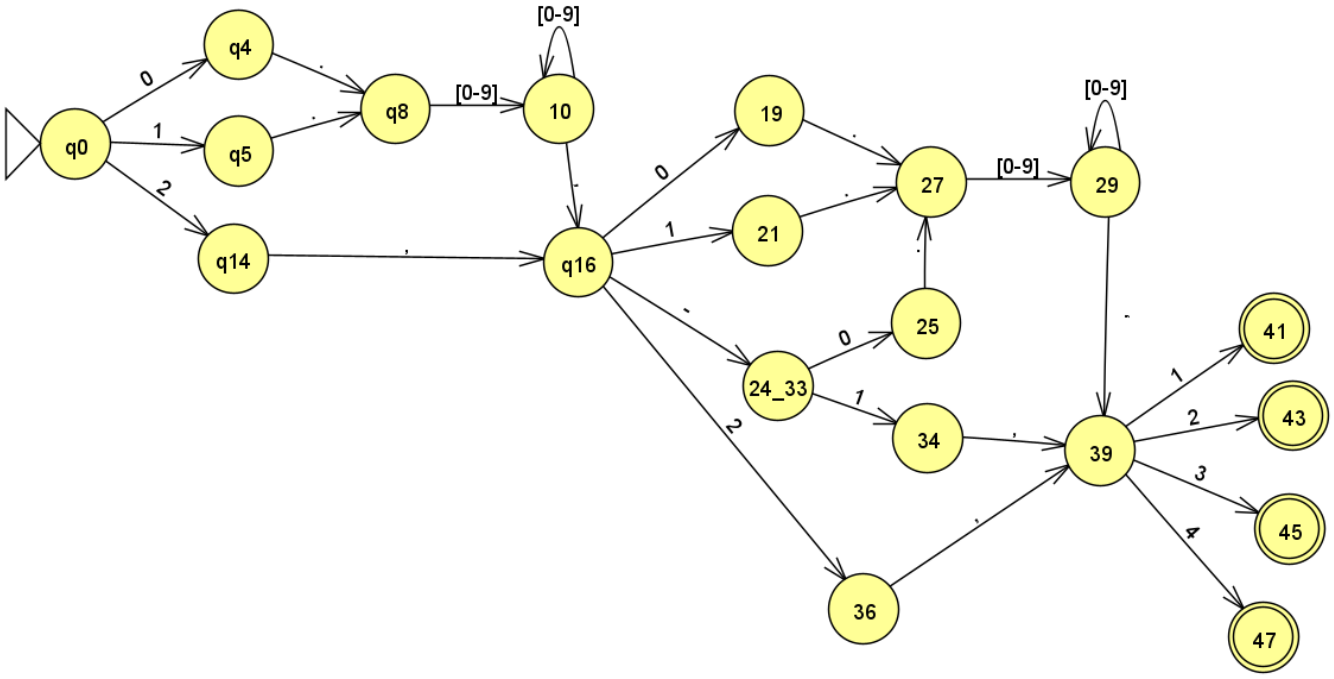
Once we obtain the NFA table, let´s obtain the DFA table transition:
Note: All white spaces(∅) are replace with a dead state called ?

	0	1	2	3	4	5	6	7	8	9	,	.	-
>Q0	Q4	Q5	Q14	?	?	?	?	?	?	?	?	?	?
Q4	?	?	?	?	?	?	?	?	?	?	?	Q8	?
Q5	?	?	?	?	?	?	?	?	?	?	?	Q8	?
Q14	?	?	?	?	?	?	?	?	?	?	Q16	?	?
Q8	Q10	Q10	Q10	Q10	Q10	Q10	Q10	Q10	Q10	Q10	?	?	?
Q16	Q19	Q21	Q36	?	?	?	?	?	?	?	?	?	Q24,Q33
Q10	Q10	Q10	Q10	Q10	Q10	Q10	Q10	Q10	Q10	Q10	Q16	?	?
Q19	?	?	?	?	?	?	?	?	?	?	?	Q27	?
Q21	?	?	?	?	?	?	?	?	?	?	?	Q27	?
Q36	?	?	?	?	?	?	?	?	?	?	Q39	?	?
Q24_Q33	Q25	Q34	?	?	?	?	?	?	?	?	?	?	?
Q27	Q29	Q29	Q29	Q29	Q29	Q29	Q29	Q29	Q29	Q29	?	?	?
Q39	?	Q41	Q43	Q45	Q47	?	?	?	?	?	?	?	?
Q25	?	?	?	?	?	?	?	?	?	?	?	Q27	?
Q34	?	?	?	?	?	?	?	?	?	?	Q39	?	?
Q29	Q29	Q29	Q29	Q29	Q29	Q29	Q29	Q29	Q29	Q29	Q39	?	?
*Q41	?	?	?	?	?	?	?	?	?	?	?	?	?
*Q43	?	?	?	?	?	?	?	?	?	?	?	?	?
*Q45	?	?	?	?	?	?	?	?	?	?	?	?	?
*Q47	?	?	?	?	?	?	?	?	?	?	?	?	?
?	?	?	?	?	?	?	?	?	?	?	?	?	?

Now let's construct the obtained DFA in JFLAP:



For a better understanding, i will eliminate those transitions to dead states.



Finally we will minimize the DFA, with the following algorithm:

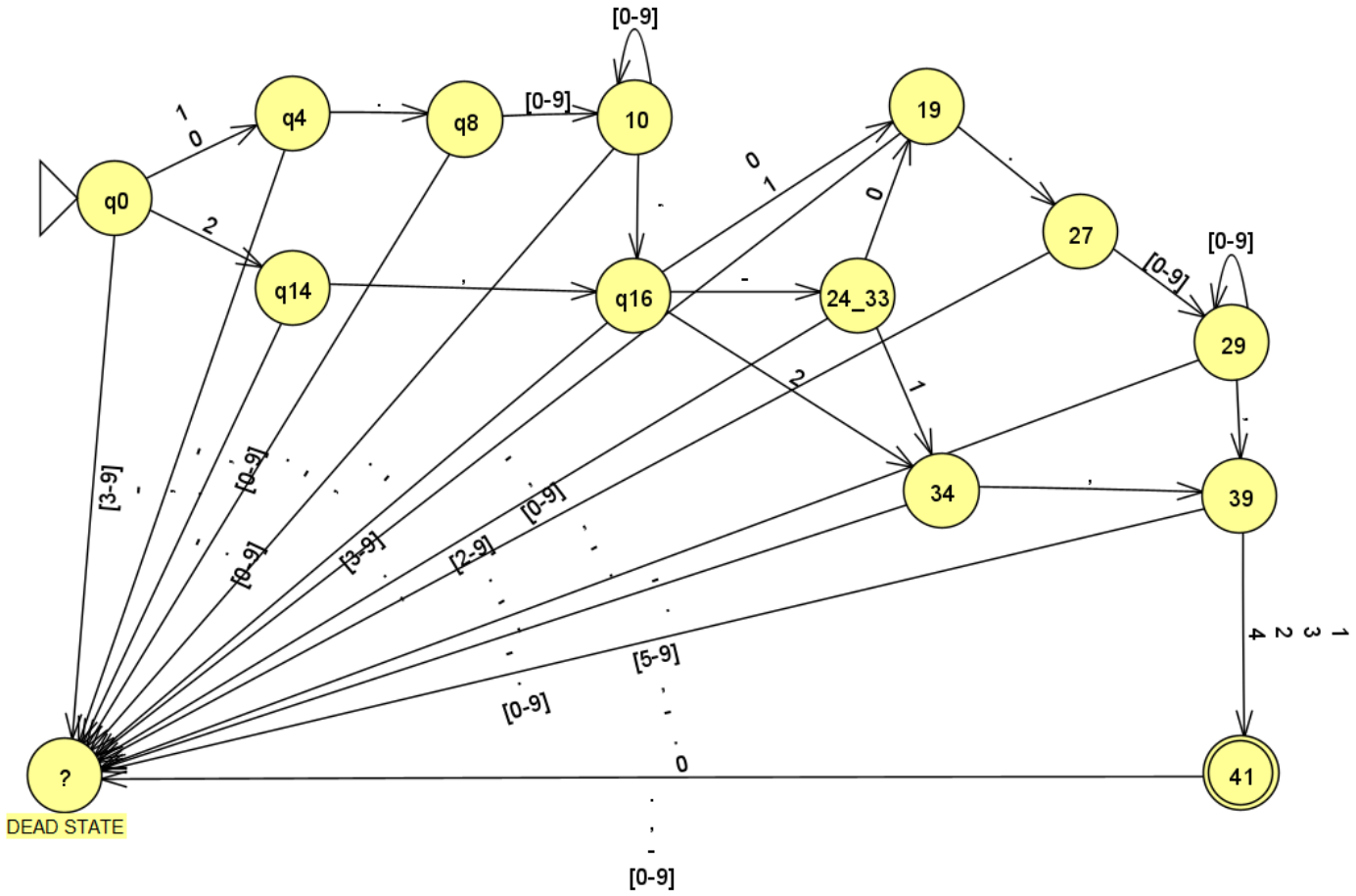
1. Build a table with all pairs of states
2. Mark all pairs of distinguishable states
    1. Mark those pairs with
        - One final state
        - One non-final state
    2. Mark those pairs [q, r] if
        - $\exists a$ / $[\delta(q,a), \delta(r,a)]$ is marked
    3. Go to 2 until no more marks are possible
3. All unmarked pairs represent equivalent states – Remove equivalent states

	?	Q47	Q45	Q43	Q41	29	34	25	39	27	24_33	36	21	19	10	16	8	14	5	4
>Q0	X	X	X	X	x	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Q4	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X		-
Q5	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	-	
Q14	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	-		
Q8	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	-			
Q16	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	-				
Q10	X	X	X	X	X	X	X	X	X	X	X	X	X	X	-					
Q19	X	X	X	X	X	X	X		X	X	X	X		-						
Q21	X	X	X	X	X	X	X		X	X	X	X	-							
Q36	X	X	X	X	X	X		X	X	X	X	-								
Q24_Q33	X	X	X	X	X	X	X	X	X	X	-									
Q27	X	X	X	X	X	X	X	X	X	-										
Q39	X	X	X	X	X	X	X	X	-											
Q25	X	X	X	X	X	X	X	-												
Q34	X	X	X	X	X	X	-													
Q29	X	X	X	X	x	-														
*Q41	X				-															
*Q43	X			-																
*Q45	X		-																	
*Q47	X	-																		

Equivalent States:
Q4 and Q5
Q19, Q25 and Q21
Q36 and Q34
Q41, Q47, Q45 and Q43

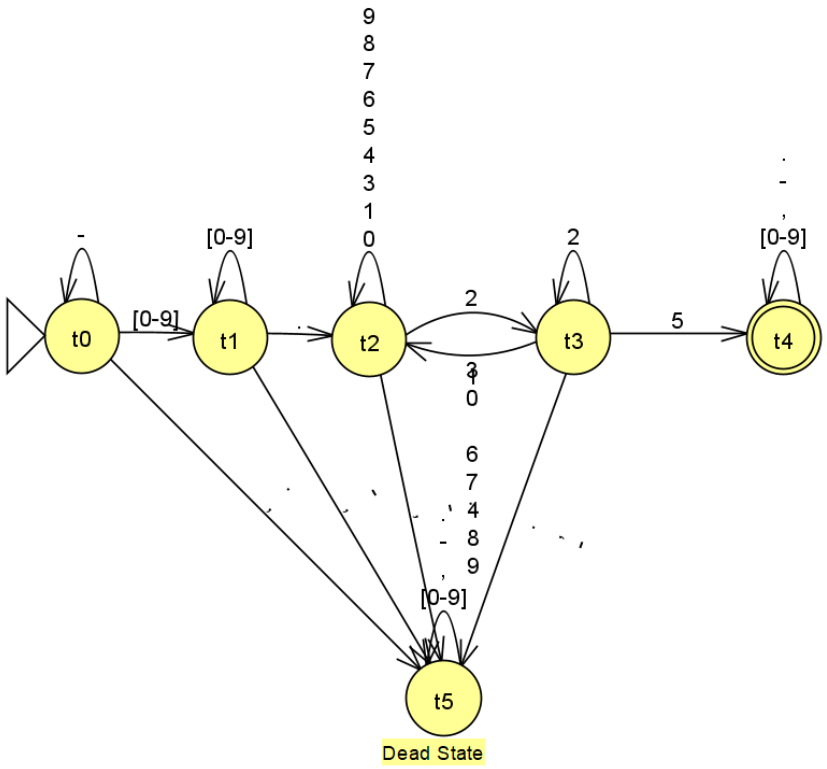
All aparitions of Q5 is called Q4
All aparitions of Q25 and Q21 is called Q9
All aparitions of Q36 is called Q34
All aparitions of Q47 and Q45 is called Q41

Minimized DFA:



Also the company has detected that some steels that include number 25 in their decimal numbers can cause some danger so they need to identify these entries.

Task 4: Design a minimum DFA that accepts database entries, correct or not, where the decimal part of the element 1 has the number 25 in it.



Task 5: Combine de DFAs designed in tasks 3 and 4 with the product automaton method to find a DFA that accepts only correct entries in which the element1 has number 25 in its decimals.

For the automata product, I will use Excel to be able to combine them more easily.

Once the table is obtained, I will start with the initial state, in our case Q0t0, since Q0 is the initial state of our first automaton and t0 is the initial state of our second automaton.

Well, in order to minimize the automated product as much as possible, I will only put those states that I find along the way (starting by Q0t0, our initial state). To minimize it even more, I will take those states that contains the dead state t5 or ? as a single dead state ?

For example, Q0t0 has transitions to Q4, Q14 and ?, So we will put those transitions and so on until no new state is found, just like the algortithm that converts NFA to DFA.

	0	1	2	3	4	5	6	7	8	9	,	.	-
>Q0t0	Q4t1	Q4t1	Q14t1	?t1	?t1	?t1	?t1	?t1	?t1	?t1	?t5	?t5	?t0
Qot1	Q4t1	Q4t1	Q14t1	?t1	?t1	?t1	?t1	?t1	?t1	?t1	?t5	?t2	?t5
Qot2	Q4t2	Q4t2	Q14t3	?t2	?t2	?t2	?t2	?t2	?t2	?t2	?t5	?t5	?t5
Qot3	Q4t2	Q4t2	Q14t3	?t2	?t2	?t4	?t2	?t2	?t2	?t2	?t5	?t5	?t5
Qot4	Q4t4	Q4t4	Q14t4	?t4	?t4	?t4	?t4	?t4	?t4	?t4	?t4	?t4	?t4
Qot5	Q4t5	Q4t5	Q14t5	?t5	?t5	?t5	?t5	?t5	?t5	?t5	?t5	?t5	?t5
Q4t0	?t1	?t1	?t1	?t1	?t1	?t1	?t1	?t1	?t1	?t1	?t5	Q8t5	?t0
Q4t1	?t1	?t1	?t1	?t1	?t1	?t1	?t1	?t1	?t1	?t1	?t5	Q8t2	?t5
Q4t2	?t2	?t2	?t3	?t2	?t2	?t2	?t2	?t2	?t2	?t2	?t5	Q8t5	?t5
Q4t3	?t2	?t2	?t3	?t2	?t2	?t4	?t2	?t2	?t2	?t2	?t5	Q8t5	?t5
Q4t4	?t4	?t4	?t4	?t4	?t4	?t4	?t4	?t4	?t4	?t4	?t4	Q8t4	?t4
Q4t5	?t5	?t5	?t5	?t5	?t5	?t5	?t5	?t5	?t5	?t5	?t5	Q8t5	?t5
Q14t0	?t1	?t1	?t1	?t1	?t1	?t1	?t1	?t1	?t1	?t1	Q16t5	?t5	?t0
Q14t1	?t1	?t1	?t1	?t1	?t1	?t1	?t1	?t1	?t1	?t1	Q16t5	?t2	?t5
Q14t2	?t2	?t2	?t3	?t2	?t2	?t2	?t2	?t2	?t2	?t2	Q16t5	?t5	?t5
Q14t3	?t2	?t2	?t3	?t2	?t2	?t4	?t2	?t2	?t2	?t2	Q16t5	?t5	?t5
Q14t4	?t4	?t4	?t4	?t4	?t4	?t4	?t4	?t4	?t4	?t4	Q16t4	?t4	?t4
Q14t5	?t5	?t5	?t5	?t5	?t5	?t5	?t5	?t5	?t5	?t5	Q16t5	?t5	?t5
Q8t0	Q10t1	Q10t1	Q10t1	Q10t1	Q10t1	Q10t1	Q10t1	Q10t1	Q10t1	Q10t1	?t5	?t5	?t0
Q8t1	Q10t1	Q10t1	Q10t1	Q10t1	Q10t1	Q10t1	Q10t1	Q10t1	Q10t1	Q10t1	?t5	?t2	?t5
Q8t2	Q10t2	Q10t2	Q10t3	Q10t2	Q10t2	Q10t2	Q10t2	Q10t2	Q10t2	Q10t2	?t5	?t5	?t5
Q8t3	Q10t2	Q10t2	Q10t3	Q10t2	Q10t2	Q10t4	Q10t2	Q10t2	Q10t2	Q10t2	?t5	?t5	?t5
Q8t4	Q10t4	Q10t4	Q10t4	Q10t4	Q10t4	Q10t4	Q10t4	Q10t4	Q10t4	Q10t4	?t4	?t4	?t4
Q8t5	Q10t5	Q10t5	Q10t5	Q10t5	Q10t5	Q10t5	Q10t5	Q10t5	Q10t5	Q10t5	?t5	?t5	?t5
Q16t0	Q19t1	Q19t1	Q34t1	?t1	?t1	?t1	?t1	?t1	?t1	?t1	?t5	?t5	Q24_Q33t0
Q16t1	Q19t1	Q19t1	Q34t1	?t1	?t1	?t1	?t1	?t1	?t1	?t1	?t5	?t2	Q24_Q33t5
Q16t2	Q19t2	Q19t2	Q34t3	?t2	?t2	?t2	?t2	?t2	?t2	?t2	?t5	?t5	Q24_Q33t5
Q16t3	Q19t2	Q19t2	Q34t3	?t2	?t2	?t4	?t2	?t2	?t2	?t2	?t5	?t5	Q24_Q33t5
Q16t4	Q19t4	Q19t4	Q34t4	?t4	?t4	?t4	?t4	?t4	?t4	?t4	?t4	?t4	Q24_Q33t4
Q16t5	Q19t5	Q19t5	Q34t5	?t5	?t5	?t5	?t5	?t5	?t5	?t5	?t5	?t5	Q24_Q33t5
Q10t0	Q10t1	Q10t1	Q10t1	Q10t1	Q10t1	Q10t1	Q10t1	Q10t1	Q10t1	Q10t1	Q16t5	?t5	?t0

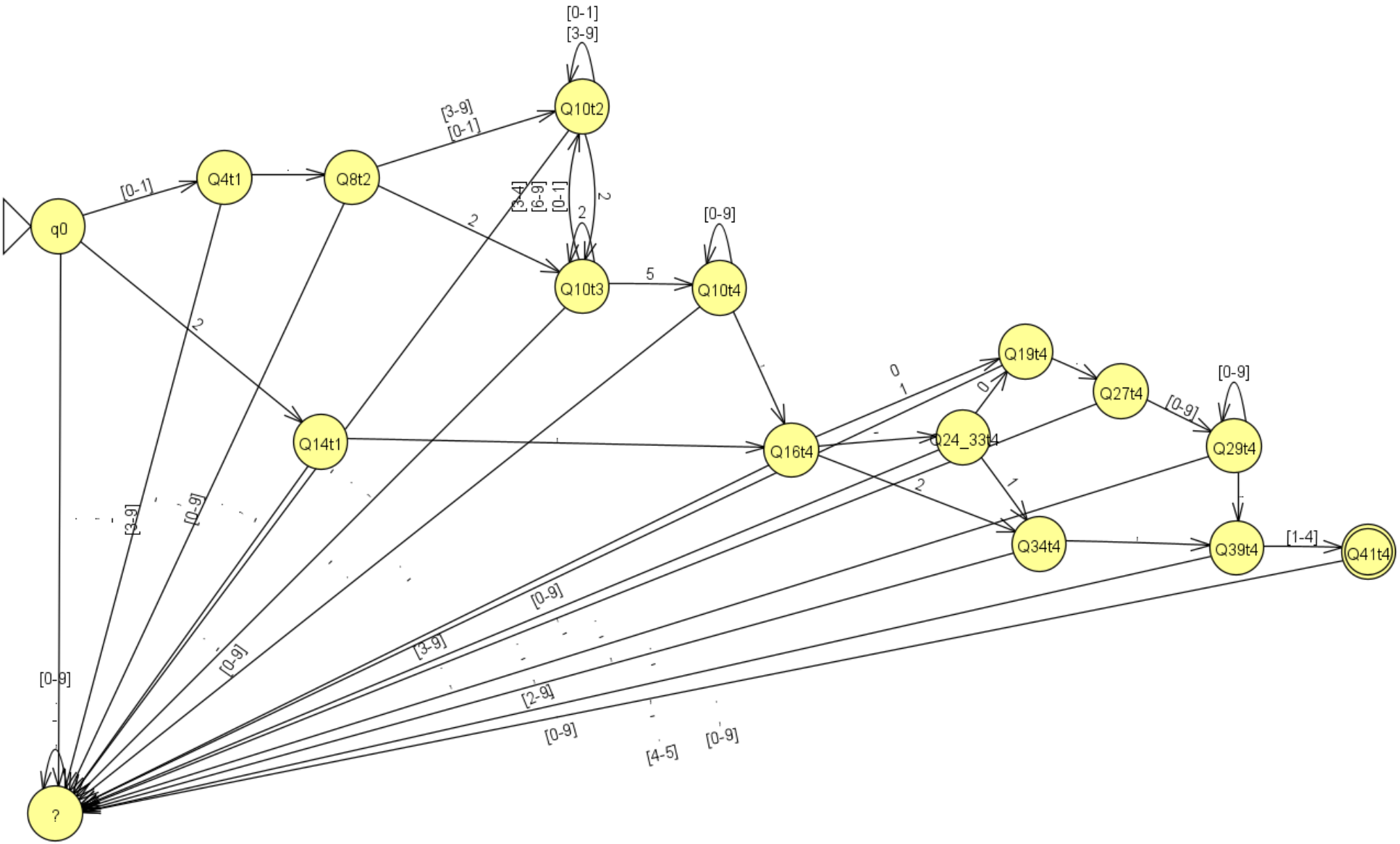
Q10t1	Q10t1	Q10t1	Q10t1	Q10t1	Q10t1	Q10t1	Q10t1	Q10t1	Q10t1	Q10t1	Q16t5	?t2	?t5
Q10t2	Q10t2	Q10t2	Q10t3	Q10t2	Q10t2	Q10t2	Q10t2	Q10t2	Q10t2	Q10t2	Q16t5	?t5	?t5
Q10t3	Q10t2	Q10t2	Q10t3	Q10t2	Q10t2	Q10t4	Q10t2	Q10t2	Q10t2	Q10t2	Q16t5	?t5	?t5
Q10t4	Q10t4	Q10t4	Q10t4	Q10t4	Q10t4	Q10t4	Q10t4	Q10t4	Q10t4	Q10t4	Q16t4	?t4	?t4
Q10t5	Q10t5	Q10t5	Q10t5	Q10t5	Q10t5	Q10t5	Q10t5	Q10t5	Q10t5	Q10t5	Q16t5	?t5	?t5
Q19t0	?t1	?t1	?t1	?t1	?t1	?t1	?t1	?t1	?t1	?t1	?t5	Q27t5	?t0
Q19t1	?t1	?t1	?t1	?t1	?t1	?t1	?t1	?t1	?t1	?t1	?t5	Q27t2	?t5
Q19t2	?t2	?t2	?t3	?t2	?t2	?t2	?t2	?t2	?t2	?t2	?t5	Q27t5	?t5
Q19t3	?t2	?t2	?t3	?t2	?t2	?t4	?t2	?t2	?t2	?t2	?t5	Q27t5	?t5
Q19t4	?t4	?t4	?t4	?t4	?t4	?t4	?t4	?t4	?t4	?t4	?t4	Q27t4	?t4
Q19t5	?t5	?t5	?t5	?t5	?t5	?t5	?t5	?t5	?t5	?t5	?t5	Q27t5	?t5
Q24_33t0	Q19t1	Q34t1	?t1	?t1	?t1	?t1	?t1	?t1	?t1	?t1	?t5	?t5	?t0
Q24_33t1	Q19t1	Q34t1	?t1	?t1	?t1	?t1	?t1	?t1	?t1	?t1	?t5	?t2	?t5
Q24_33t2	Q19t2	Q34t2	?t3	?t2	?t2	?t2	?t2	?t2	?t2	?t2	?t5	?t5	?t5
Q24_33t3	Q19t2	Q34t2	?t3	?t2	?t2	?t4	?t2	?t2	?t2	?t2	?t5	?t5	?t5
Q24_33t4	Q19t4	Q34t4	?t4	?t4	?t4	?t4	?t4	?t4	?t4	?t4	?t4	?t4	?t4
Q24_33t5	Q19t5	Q34t5	?t5	?t5	?t5	?t5	?t5	?t5	?t5	?t5	?t5	?t5	?t5
Q27t0	Q29t1	Q29t1	Q29t1	Q29t1	Q29t1	Q29t1	Q29t1	Q29t1	Q29t1	Q29t1	?t5	?t5	?t0
Q27t1	Q29t1	Q29t1	Q29t1	Q29t1	Q29t1	Q29t1	Q29t1	Q29t1	Q29t1	Q29t1	?t5	?t2	?t5
Q27t2	Q29t2	Q29t2	Q29t3	Q29t2	Q29t2	Q29t2	Q29t2	Q29t2	Q29t2	Q29t2	?t5	?t5	?t5
Q27t3	Q29t2	Q29t2	Q29t3	Q29t2	Q29t2	Q29t4	Q29t2	Q29t2	Q29t2	Q29t2	?t5	?t5	?t5
Q27t4	Q29t4	Q29t4	Q29t4	Q29t4	Q29t4	Q29t4	Q29t4	Q29t4	Q29t4	Q29t4	?t4	?t4	?t4
Q27t5	Q29t5	Q29t5	Q29t5	Q29t5	Q29t5	Q29t5	Q29t5	Q29t5	Q29t5	Q29t5	?t5	?t5	?t5
Q39t0	?t1	Q41t1	Q41t1	Q41t1	Q41t1	?t1	?t1	?t1	?t1	?t1	?t5	?t5	?t0
Q39t1	?t1	Q41t1	Q41t1	Q41t1	Q41t1	?t1	?t1	?t1	?t1	?t1	?t5	?t2	?t5
Q39t2	?t2	Q41t2	Q41t3	Q41t2	Q41t2	?t2	?t2	?t2	?t2	?t2	?t5	?t5	?t5
Q39t3	?t2	Q41t2	Q41t3	Q41t2	Q41t2	?t4	?t2	?t2	?t2	?t2	?t5	?t5	?t5
Q39t4	?t4	Q41t4	Q41t4	Q41t4	Q41t4	?t4	?t4	?t4	?t4	?t4	?t4	?t4	?t4
Q39t5	?t5	Q41t5	Q41t5	Q41t5	Q41t5	?t5	?t5	?t5	?t5	?t5	?t5	?t5	?t5
Q34t0	?t1	?t1	?t1	?t1	?t1	?t1	?t1	?t1	?t1	?t1	Q39t5	?t5	?t0
Q34t1	?t1	?t1	?t1	?t1	?t1	?t1	?t1	?t1	?t1	?t1	Q39t5	?t2	?t5
Q34t2	?t2	?t2	?t3	?t2	?t2	?t2	?t2	?t2	?t2	?t2	Q39t5	?t5	?t5

Q34t3	?t2	?t2	?t3	?t2	?t2	?t4	?t2	?t2	?t2	?t2	Q39t5	?t5	?t5
Q34t4	?t4	?t4	?t4	?t4	?t4	?t4	?t4	?t4	?t4	?t4	Q39t4	?t4	?t4
Q34t5	?t5	?t5	?t5	?t5	?t5	?t5	?t5	?t5	?t5	?t5	Q39t5	?t5	?t5
Q29t0	Q29t1	Q29t1	Q29t1	Q29t1	Q29t1	Q29t1	Q29t1	Q29t1	Q29t1	Q29t1	Q39t5	?t5	?t0
Q29t1	Q29t1	Q29t1	Q29t1	Q29t1	Q29t1	Q29t1	Q29t1	Q29t1	Q29t1	Q29t1	Q39t5	?t2	?t5
Q29t2	Q29t2	Q29t2	Q29t3	Q29t2	Q29t2	Q29t2	Q29t2	Q29t2	Q29t2	Q29t2	Q39t5	?t5	?t5
Q29t3	Q29t2	Q29t2	Q29t3	Q29t2	Q29t2	Q29t4	Q29t2	Q29t2	Q29t2	Q29t2	Q39t5	?t5	?t5
Q29t4	Q29t4	Q29t4	Q29t4	Q29t4	Q29t4	Q29t4	Q29t4	Q29t4	Q29t4	Q29t4	Q39t4	?t4	?t4
Q29t5	Q29t5	Q29t5	Q29t5	Q29t5	Q29t5	Q29t5	Q29t5	Q29t5	Q29t5	Q29t5	Q39t5	?t5	?t5
Q41t0	?t1	?t1	?t1	?t1	?t1	?t1	?t1	?t1	?t1	?t1	?t5	?t5	?t0
Q41t1	?t1	?t1	?t1	?t1	?t1	?t1	?t1	?t1	?t1	?t1	?t5	?t2	?t5
Q41t2	?t2	?t2	?t3	?t2	?t2	?t2	?t2	?t2	?t2	?t2	?t5	?t5	?t5
Q41t3	?t2	?t2	?t3	?t2	?t2	?t4	?t2	?t2	?t2	?t2	?t5	?t5	?t5
*Q41t4	?t4	?t4	?t4	?t4	?t4	?t4	?t4	?t4	?t4	?t4	?t4	?t4	?t4
Q41t5	?t5	?t5	?t5	?t5	?t5	?t5	?t5	?t5	?t5	?t5	?t5	?t5	?t5
?t0	?t1	?t1	?t1	?t1	?t1	?t1	?t1	?t1	?t1	?t1	?t5	?t5	?t0
?t1	?t1	?t1	?t1	?t1	?t1	?t1	?t1	?t1	?t1	?t1	?t5	?t2	?t5
?t2	?t2	?t2	?t3	?t2	?t2	?t2	?t2	?t2	?t2	?t2	?t5	?t5	?t5
?t3	?t2	?t2	?t3	?t2	?t2	?t4	?t2	?t2	?t2	?t2	?t5	?t5	?t5
?t4	?t4	?t4	?t4	?t4	?t4	?t4	?t4	?t4	?t4	?t4	?t4	?t4	?t4
?t5	?t5	?t5	?t5	?t5	?t5	?t5	?t5	?t5	?t5	?t5	?t5	?t5	?t5

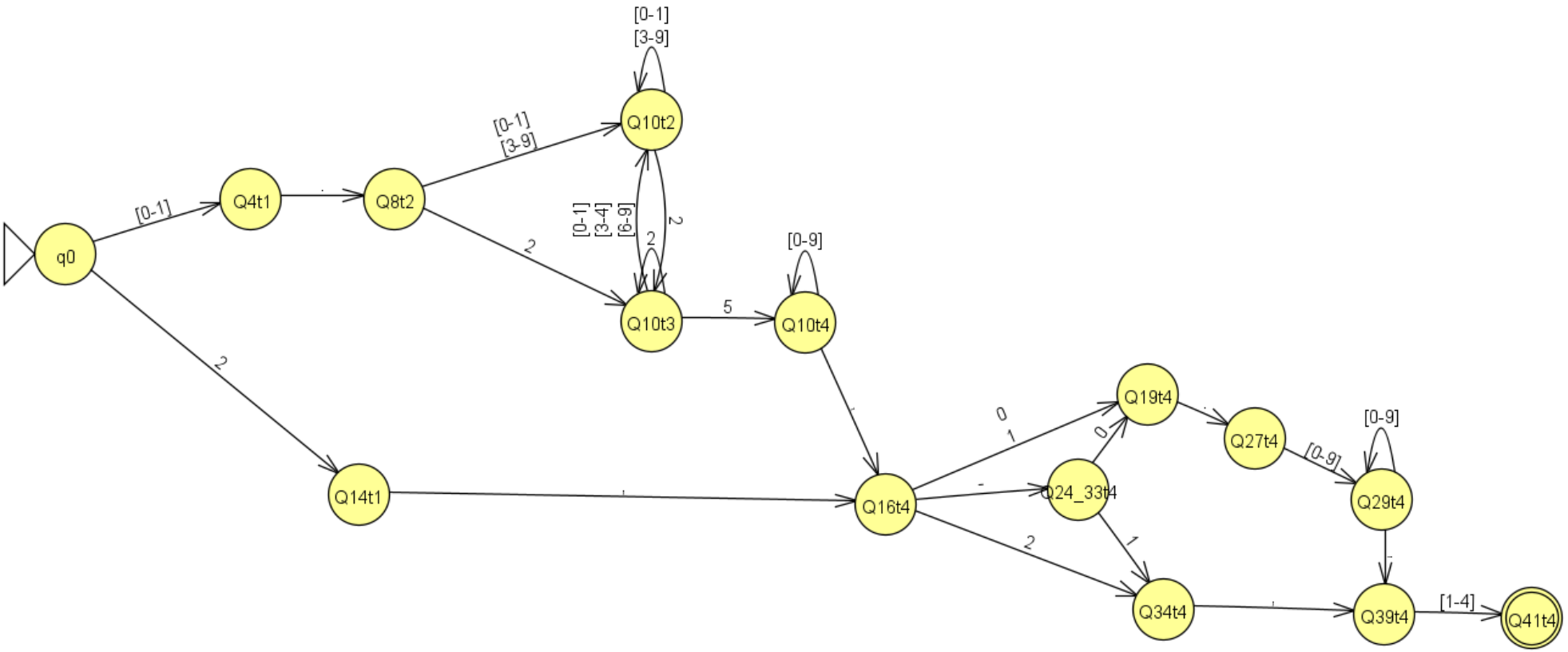
Once the states we are crossing have been obtained, we obtain the following table, which corresponds to the minimized product automata.

	0	1	2	3	4	5	6	7	8	9	,	.	-
>Q0t0	Q4t1	Q4t1	Q14t1	?	?	?	?	?	?	?	?	?	?
Q4t1	?	?	?	?	?	?	?	?	?	?	?	Q8t2	?
Q14t1	?	?	?	?	?	?	?	?	?	?	?	?	?
Q8t2	Q10t2	Q10t2	Q10t3	Q10t2	Q10t2	Q10t2	Q10t2	Q10t2	Q10t2	Q10t2	?	?	?
Q10t2	Q10t2	Q10t2	Q10t3	Q10t2	Q10t2	Q10t2	Q10t2	Q10t2	Q10t2	Q10t2	?	?	?
Q10t3	Q10t2	Q10t2	Q10t3	Q10t2	Q10t2	Q10t4	Q10t2	Q10t2	Q10t2	Q10t2	?	?	?
Q10t4	Q10t4	Q10t4	Q10t4	Q10t4	Q10t4	Q10t4	Q10t4	Q10t4	Q10t4	Q10t4	Q16t4	?	?
Q16t4	Q19t4	Q19t4	Q34t4	?	?	?	?	?	?	?	?	?	Q24_Q33t4
Q24_33t4	Q19t4	Q34t4	?	?	?	?	?	?	?	?	?	?	?
Q19t4	?	?	?	?	?	?	?	?	?	?	?	Q27t4	?
Q34t4	?	?	?	?	?	?	?	?	?	?	Q39t4	?	?
Q27t4	Q29t4	Q29t4	Q29t4	Q29t4	Q29t4	Q29t4	Q29t4	Q29t4	Q29t4	Q29t4	?	?	?
Q29t4	Q29t4	Q29t4	Q29t4	Q29t4	Q29t4	Q29t4	Q29t4	Q29t4	Q29t4	Q29t4	Q39t4	?	?
Q39t4	?	Q41t4	Q41t4	Q41t4	Q41t4	?	?	?	?	?	?	?	?
*Q41t4	?	?	?	?	?	?	?	?	?	?	?	?	?
?	?	?	?	?	?	?	?	?	?	?	?	?	?

Product DFA:

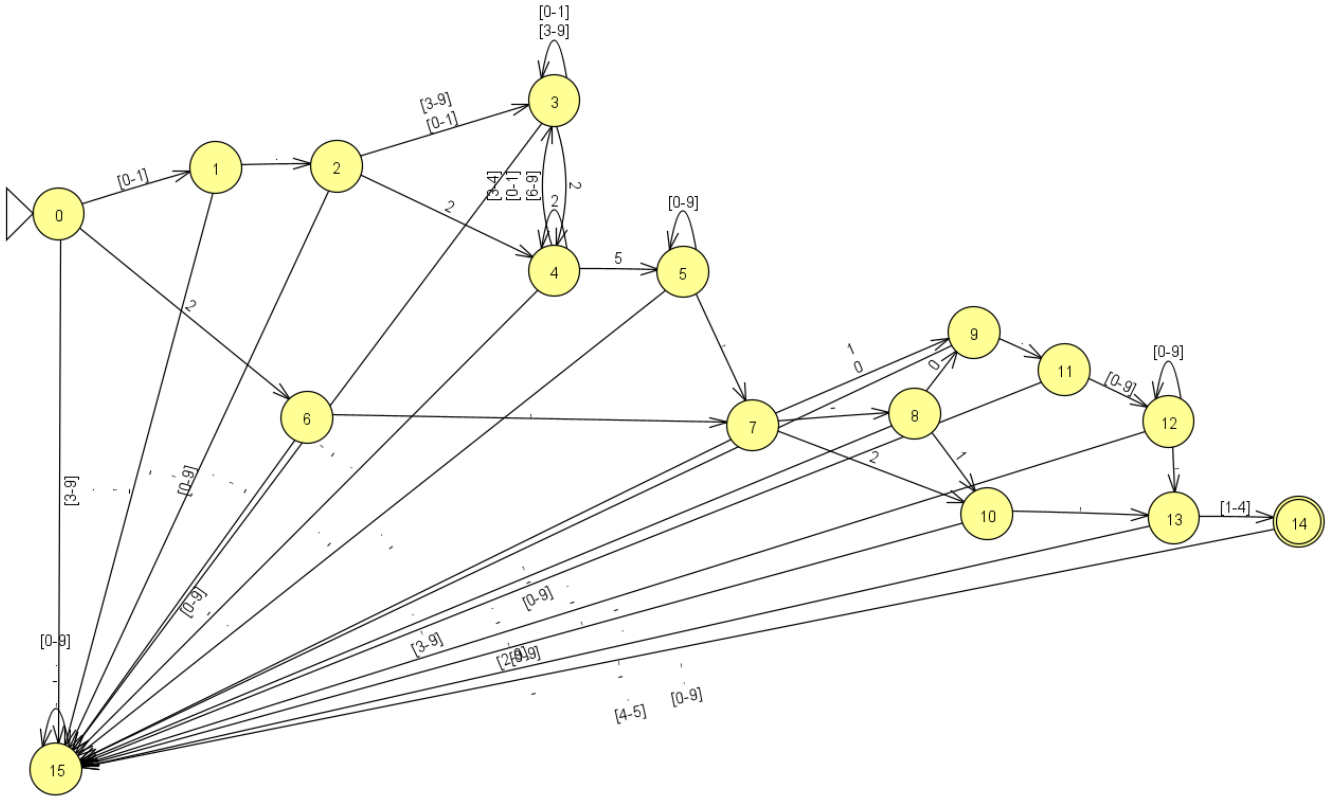

q0
Q4t1
Q8t2
Q10t2
Q10t3
Q10t4
Q14t1
Q16t4
Q19t4
Q24_33t4
Q27t4
Q29t4
Q34t4
Q39t4
Q41t4
?
[0-1]
[3-9]
[0-9]
[1-4]
[2-9]
[4-5]
[3-4]
[6-9]
2
5
0
1

For a better understanding, i will eliminate those transitions to dead states.



Task 6: Implement in your favorite language the obtained DFA using the table method. The program must read a file with the database and print the correct entries where the element1 has the number 25 in its decimal part.

Note: The states have been renamed, as follows to better code it. The code is written in java language.




```
private int[][] G ={
        // 0  1  2  3  4  5  6  7  8  9  ,  .  -
    /*0*/ {1,1,6,15,15,15,15,15,15,15,15,15,15},
    /*1*/ {15,15,15,15,15,15,15,15,15,15,15,2,15},
    /*2*/ {3,3,4,3,3,3,3,3,3,3,15,15,15},
    /*3*/ {3,3,4,3,3,3,3,3,3,3,15,15,15},
    /*4*/ {3,3,4,3,3,5,3,3,3,3,15,15,15},
    /*5*/ {5,5,5,5,5,5,5,5,5,5,7,15,15},
    /*6*/ {15,15,15,15,15,15,15,15,15,15,7,15,15},
    /*7*/ {9,9,10,15,15,15,15,15,15,15,15,15,8},
    /*8*/ {9,10,15,15,15,15,15,15,15,15,15,15,15},
    /*9*/ {15,15,15,15,15,15,15,15,15,15,15,11,15},
    /*10*/{15,15,15,15,15,15,15,15,15,15,13,15,15},
    /*11*/{12,12,12,12,12,12,12,12,12,12,15,15,15},
    /*12*/{12,12,12,12,12,12,12,12,12,13,15,15},
    /*13*/{15,14,14,14,14,15,15,15,15,15,15,15,15},
    /*14*/{15,15,15,15,15,15,15,15,15,15,15,15,15},
    /*15*/{15,15,15,15,15,15,15,15,15,15,15,15,15},
};
```

```java
import java.io.*;

public class DFA
{
    private int[][] G ={
            // 0  1  2  3  4  5  6  7  8  9  ,  .  -
        /*0*/ {1,1,6,15,15,15,15,15,15,15,15,15,15},
        /*1*/ {15,15,15,15,15,15,15,15,15,15,15,2,15},
        /*2*/ {3,3,4,3,3,3,3,3,3,3,15,15,15},
        /*3*/ {3,3,4,3,3,3,3,3,3,3,15,15,15},
        /*4*/ {3,3,4,3,3,5,3,3,3,3,15,15,15},
        /*5*/ {5,5,5,5,5,5,5,5,5,5,7,15,15},
        /*6*/ {15,15,15,15,15,15,15,15,15,15,7,15,15},
        /*7*/ {9,9,10,15,15,15,15,15,15,15,15,15,8},
        /*8*/ {9,10,15,15,15,15,15,15,15,15,15,15,15},
        /*9*/ {15,15,15,15,15,15,15,15,15,15,15,11,15},
        /*10*/{15,15,15,15,15,15,15,15,15,15,13,15,15},
        /*11*/{12,12,12,12,12,12,12,12,12,12,15,15,15},
        /*12*/{12,12,12,12,12,12,12,12,12,12,13,15,15},
        /*13*/{15,14,14,14,14,15,15,15,15,15,15,15,15},
        /*14*/{15,15,15,15,15,15,15,15,15,15,15,15,15},
        /*15*/{15,15,15,15,15,15,15,15,15,15,15,15,15},
    };

    private FileReader fr;
    private BufferedReader bf;
    public static final String ANSI_RESET = "\u001B[0m";
    public static final String ANSI_RED = "\u001B[31m";
    public static final String ANSI_GREEN = "\u001B[32m";

    public DFA()
    {
        try {
            fr = new FileReader("data.txt");
            bf = new BufferedReader(fr);
        } catch (Exception e) {
            //TODO: handle exception
        }

        this.comprobar();
    }

    public int column(char c)
    {
        if(c == ',')
        {
            return 10;
        }

        else if(c == '.')
        {
            return 11;
        }
        else if(c == '-')
        {
            return 12;
        }
        else if(Character.getNumericValue(c) >= 0 && Character.getNumericValue(c) <= 9)
        {
            return Character.getNumericValue(c);
        }
        else return -1;
    }

    public void comprobar()
    {
        String sCadena;
        int state = 0;
        try{
            while ((sCadena = bf.readLine())!=null) {
                state = 0;
                Thread.currentThread().sleep(100);
                for(int i = 0; i < sCadena.length(); i++)
                {
                    state = G[state][this.column(sCadena.charAt(i))];
                }

                if(state == 14)
                {
                    System.out.println(sCadena+ANSI_GREEN+" ....ACEPTADO"+ANSI_RESET);
                }
                else{
                    System.out.println(sCadena+ANSI_RED+" ....RECHAZADO"+ANSI_RESET);
                }
             }
        }
        catch(Exception e){}

    }

    public static void main(String[] args) {
        new DFA();
    }
}
```

Some test cases:

```
0.25,0.14,4 ....ACEPTADO
0.285,1.25,4 ....RECHAZADO
0.25,8,4 ....RECHAZADO
0.25,2,4 ....ACEPTADO
1.47766418058765,574,2 ....RECHAZADO
1627617.9,509,4 ....RECHAZADO
101,1.22887401609811,3 ....RECHAZADO
975,-0.631187651139485,2 ....RECHAZADO
9940170.6,-0.92562429502453,3 ....RECHAZADO
1.05457795149631,-0.773560013378283,2 ....RECHAZADO
1.48657652494399,1.79218674165367,2 ....RECHAZADO
998,-0.566708692881646,3 ....RECHAZADO
0.267044145275387,-0.383354726215553,3 ....RECHAZADO
0.236724799442856,276,1 ....RECHAZADO
12100154.6,1.69205936630503,4 ....RECHAZADO
0.860829518223393,502,3 ....RECHAZADO
1.68986327664323,0.921287719665168,1 ....RECHAZADO
0.767407858809478,999,3 ....RECHAZADO
1.40026907505898,0.0637300541489991,4 ....RECHAZADO
17140097,416,2 ....RECHAZADO
17300104.2,1.52930709818764,1 ....RECHAZADO
462,1.67885823800142,3 ....RECHAZADO
0.786203151954699,0.643807333547477,2 ....RECHAZADO
982,1.66976946576406,4 ....RECHAZADO
1.88523317679645,1.10121820674057,4 ....RECHAZADO
689,40,2 ....RECHAZADO
1.5584515745125,0.684575679,2 ....ACEPTADO
2,0.56432184,4 ....ACEPTADO
1.2525252525258825,0.326543216854,1 ....ACEPTADO
```