UNIVERSIDAD CARLOS III DE MADRID
FORMAL LANGUAGES AND AUTOMATA THEORY.
COMPUTER SCIENCE DEGREE.
Second name(s):______________________________
First name:________________________________
NIA:__________________ **Signature:**_______________

Duration of the exam: 45 minutes TEST + 2 hours PROBLEMS
Maximum mark: 5 POINTS

PROBLEM 1: Maximum mark 1.25 points.

Obtain formally, applying the method of the derivates of regular expressions, the right-linear Type-3 grammar (G3RL) that generates the language described by the regular expression R_0.

$$\mathbf{R_0 = (a + b)^* \, b \, (\lambda + a)}$$

Explain, using examples of words with length 1, 2 and 3, how these words are represented by the RE and generated by the equivalent G3RL.

SOLUTION PROBLEM 1:

$R_0 = (a + b)^* \, b \, (\lambda + a)$

It is necessary to derive R_0 and the REs that we obtain with regard a and b until no new regular expressions are obtained. It is also required to calculate the value of δ for R0 and the REs that we obtain.

- $\underline{Da(R_0)} = Da(a+b)^*b(\lambda+a) + \delta(a+b)^* \, Da(b\,(\lambda+a)) =$
  $= (Da(a)+Da(b))\,(a+b)^*\,b\,(\lambda+a) + \lambda\, Da(b)\,(\lambda+a) + \delta(b)\, Da((\lambda+a)) =$
  $= (\lambda + \Phi)\,(a+b)^*\,b\,(\lambda+a) + \Phi + \Phi =$
  $= \boxed{(a+b)^*\,b\,(\lambda+a) = R_0}$

- $\underline{Db(R_0)} = Db(a+b)^*b(\lambda+a) + \delta(a+b)^* \, Db(b\,(\lambda+a)) =$
  $= (Db(a)+Db(b))\,(a+b)^*\,b\,(\lambda+a) + \lambda\, Db(b)\,(\lambda+a) + \delta(b)\, Db((\lambda+a)) =$
  $= (\Phi + \lambda)\,(a+b)^*\,b\,(\lambda+a) + (\lambda+a) + \Phi =$
  $= (a+b)^*\,b\,(\lambda+a) + (\lambda+a)\ \boxed{= R_0+(\lambda+a) = R_1}$

- $\underline{Da(R_1)} = Da(R_0) + Da((\lambda+a)) =$
  $= R_0 + Da(\lambda) + Da(a) =$
  $= R_0 + \Phi + \lambda =$
  $= \boxed{R_0 + \lambda = R_2}$

- $\underline{Db(R_1)} = Db(R_0) + Db((\lambda+a)) =$
  $= R_1 + Db(\lambda) + Db(a) =$
  $= R_1 + \Phi + \Phi =$
  $= \boxed{R_1}$

- $\underline{Da(R_2)} = Da(R_0) + Da(\lambda) =$
  $= \boxed{R_0 + \Phi = R_0}$

- $\underline{Db(R_2)} = Db(R_0) + Db(\lambda) =$
  $= \boxed{R_1 + \Phi = R_1}$

Now we calculate the value of the δ function:

- $\delta(R_0) = \Phi$
- $\delta(R_1) = \lambda$
- $\delta(R_2) = \lambda$

With the derivates and the values of δ, we can write the corresponding G3RL: $(\{a, b\}, \{R_0, R_1, R_2\}, R_0, P)$
where P:
P = {

$R_0 ::= aR_0 \ / \ bR_1 \ / \ b$
$R_1 ::= aR_2 \ / \ bR_1 \ / \ a \ / \ b$
$R_2 ::= aR_0 \ / \ bR_1 \ /b$

}

PROBLEM 2: Maximum mark 1.25 points.

We want to develop a system for detecting errors in the transmission of words over the alphabet S= {a, b, c} and included in the language $L_1 = S^+$. To do this, the L_2 language is generated by adding a sequence of bits to the words in L_1 (so many bits as transmitted symbols). The added bits indicate whether the current symbol is equal to previous one or not in the words of L_1. 0 is used to indicate that the current symbol is equal to the previous one, and 1 to indicate that there is a change. The transmission of the first symbol is always corresponded with a 0. The sequence of bits will be interpreted reading it from right to left, in opposite order to the sequence of corresponding letters of the message.

Example: caabcc011010

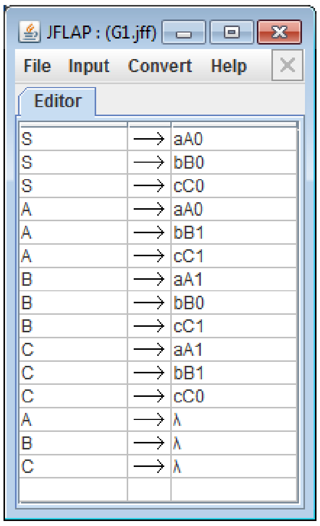
	Word in L_2											
	transmitted symbol						Control bits					
	c	a	a	b	c	c	0	1	1	0	1	0
Order in the transmission	1	2	3	4	5	6	7	8	9	10	11	12

The symbol c (number 1) is not interpreted as a change; therefore, the bit 12 is a 0
The symbol a (number 2) supposes a change with respect to the previous one; then, the bit 11 is a 1
The symbol a (number 3) is not a change with respect to the previous one; then, the bit 10 is a 0
The symbol b (number 4) supposes a change with respect to the previous one; then, the bit 9 is a 1
The symbol c (number 5) supposes a change with respect to the previous one; then, the bit 8 is a 1
The symbol c (number 6) does not change with respect to the previous one; then, the bit 7 is a 0.

Design, from the L2 grammar, an automaton to recognize this language.

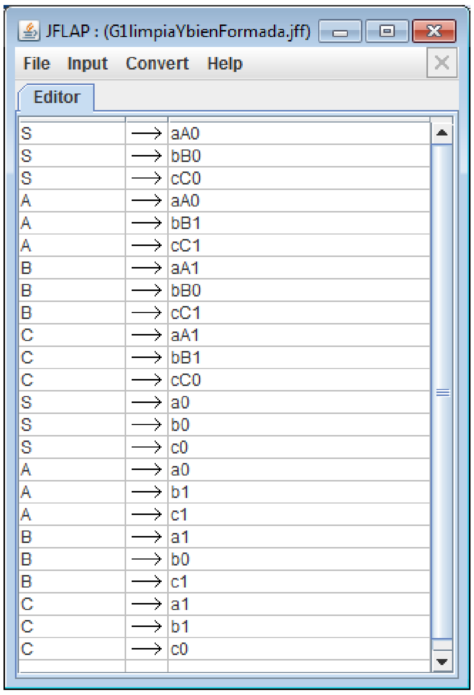
SOLUTION PROBLEM 2:

Initial grammar:



S	$\longrightarrow$	aA0
S	$\longrightarrow$	bB0
S	$\longrightarrow$	cC0
A	$\longrightarrow$	aA0
A	$\longrightarrow$	bB1
A	$\longrightarrow$	cC1
B	$\longrightarrow$	aA1
B	$\longrightarrow$	bB0
B	$\longrightarrow$	cC1
C	$\longrightarrow$	aA1
C	$\longrightarrow$	bB1
C	$\longrightarrow$	cC0
A	$\longrightarrow$	λ
B	$\longrightarrow$	λ
C	$\longrightarrow$	λ

Corresponding well-formed grammar:



S	$\longrightarrow$	aA0
S	$\longrightarrow$	bB0
S	$\longrightarrow$	cC0
A	$\longrightarrow$	aA0
A	$\longrightarrow$	bB1
A	$\longrightarrow$	cC1
B	$\longrightarrow$	aA1
B	$\longrightarrow$	bB0
B	$\longrightarrow$	cC1
C	$\longrightarrow$	aA1
C	$\longrightarrow$	bB1
C	$\longrightarrow$	cC0
S	$\longrightarrow$	a0
S	$\longrightarrow$	b0
S	$\longrightarrow$	c0
A	$\longrightarrow$	a0
A	$\longrightarrow$	b1
A	$\longrightarrow$	c1
B	$\longrightarrow$	a1
B	$\longrightarrow$	b0
B	$\longrightarrow$	c1
C	$\longrightarrow$	a1
C	$\longrightarrow$	b1
C	$\longrightarrow$	c0

From this grammar, we can easily obtain the corresponding grammar in GNF by replacing 0 by X, 1 by Y, and adding the rules X →0, Y→1. Once obtained the GNF grammar, the PDA is created with the algorithm G2 → PDA_E that we have studied.

PROBLEM 3: Maximum mark 1.25 points.

Design a Turing Machine to order, from lowest to highest, two natural numbers in unary code separated by the symbol #. The tape initially contains *number1#number2*, and it will provide *LowestNumber#HighestNumber*. It is mandatory to place the header in the first digit of the smallest number once the result is provided.

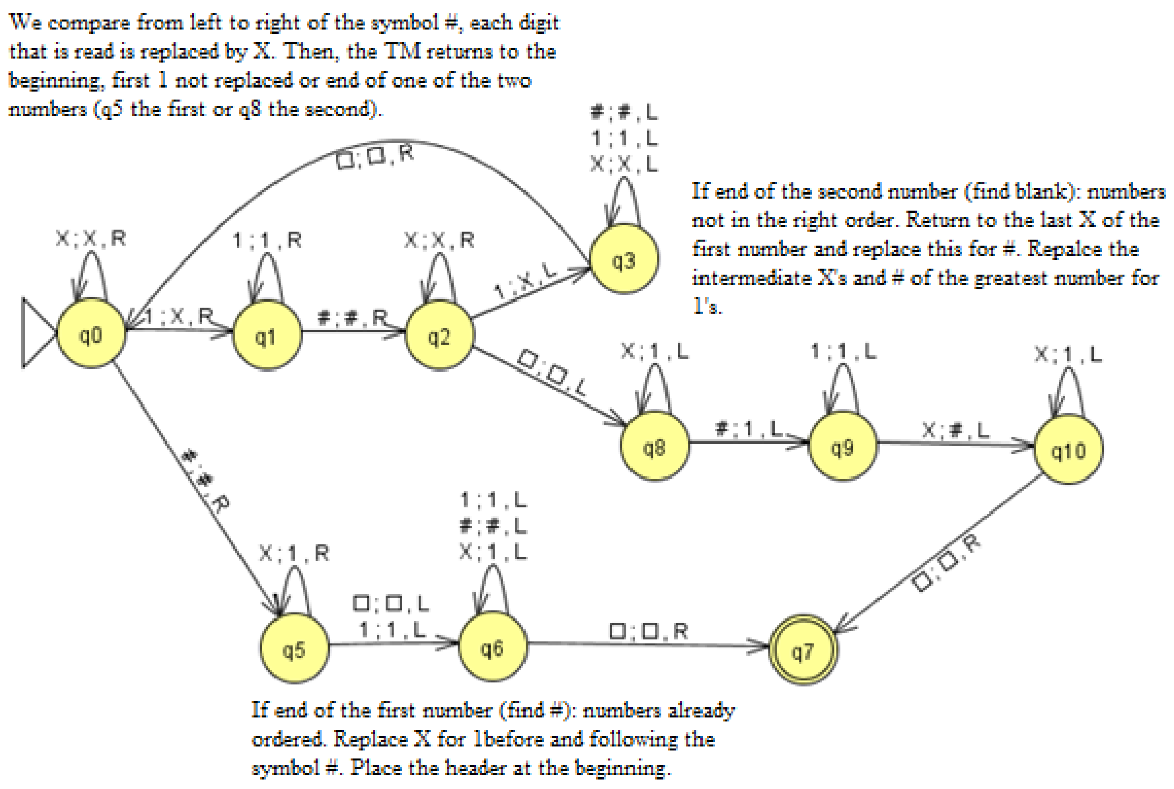
For example, the input 1111#11 generates the output 11#1111; and the input 11#1111 generates the output 11#1111. It is required:

a) Detailed description of the algorithm implemented by the Turing Machine.
b) Formal definition with the seven elements of the Turing Machine. Include the transition diagram (not the list, nor the table of the transition function).
c) Explain the meaning of:
   - each symbol of the alphabet of the tape not defined in this wording,
   - each one of the states and transitions, or group of states and transitions.

SOLUCIÓN PROBLEM 3:

a) Formal definition:

MT=({1,#}, {1,#,□,X}, □, {q0,q1,q2,q3,q4,q5,q6,q7,q8,q9,q10 }, q0, f, {q7}), where f (with the header initially placed at the beginning of the first word) is given by the following transition diagram:


We compare from left to right of the symbol #, each digit that is read is replaced by X. Then, the TM returns to the beginning, first 1 not replaced or end of one of the two numbers (q5 the first or q8 the second).

If end of the second number (find blank): numbers not in the right order. Return to the last X of the first number and replace this for #. Repalce the intermediate X's and # of the greatest number for 1's.

If end of the first number (find #): numbers already ordered. Replace X for 1before and following the symbol #. Place the header at the beginning.

b) Algorithm that is implemented by the Turing Machine.

General idea: Compare digit by digit of both numbers, marking the digits that are read from left to right. Repeat this until the last digit of one of the numbers, which will correspond with the smallest one. If this is the first one, they are already ordered. If it is the second one, they are disordered, and it is required to write the second number ahead. To do this, the TM returns until the last mark of the first number (what it is ahead is equal to the second number), this symbol is replaced by # and the rest of the word is replaced by 1' s (including the original #).

Detailed algorithm:

1. Read each digit of the first number (marking it with an X), do the same with the digits of the second number, until reaching the end of one of the numbers. Go to step 2 or 3 depending on the number that is firstly read.
2. If the number that is firstly read is the first one (identified by the symbol "#" in the tape), then the numbers are already ordered. The marks X preceding and following the symbol # are replaced by 1's. Go to the step 4.
3. If the number that is firstly read is the second one (identified by the blank symbol in the tape), then the numbers are disordered. To order them, it is enough with deleting the first digit in the point where the TM has verified that it is equal to the second number (by replacing last mark X by the symbol #), and the rest will be the greatest number (after replacing the old # by 1). All the marks of both numbers are then replaced by 1's.
4. Return to the beginning to place the header on the first digit of the smallest number.

c) Explain the meaning of:

- **each symbol of the alphabet of the tape not defined in this wording:**

  □: blank symbol, which indicates that the cell is empty.

  X: mark that indicates that the digits have been already read and compared among them.

  (#: symbol between both numbers already defined in the wording.)

- **each one of the states and transitions, or group of states and transitions.**

  i. **cycle q0-q3** [step 1 of the algorithm]: in each cycle, for each digit of the second number to be marked with a X, a digit of the first number is marked with a X. From q0 to q1, the first number is marked. Between q1 and q2 one advances until first digit of not dialed number 2. Between q2 to q3, the TM goes to the first digit of the second number not marked. In q3 the TM goes to the first position of the input and, using the recursive transition in q1, the TM goes to the first digit not marked of the first number.

  ii. transition q0 to q5: detection of the end of the first number once the symbol "#" is detected.

iii. **q5** [step 2 of the algorithm, first part]: unmark every digit of the second number (changing X by 1), going from left to right.

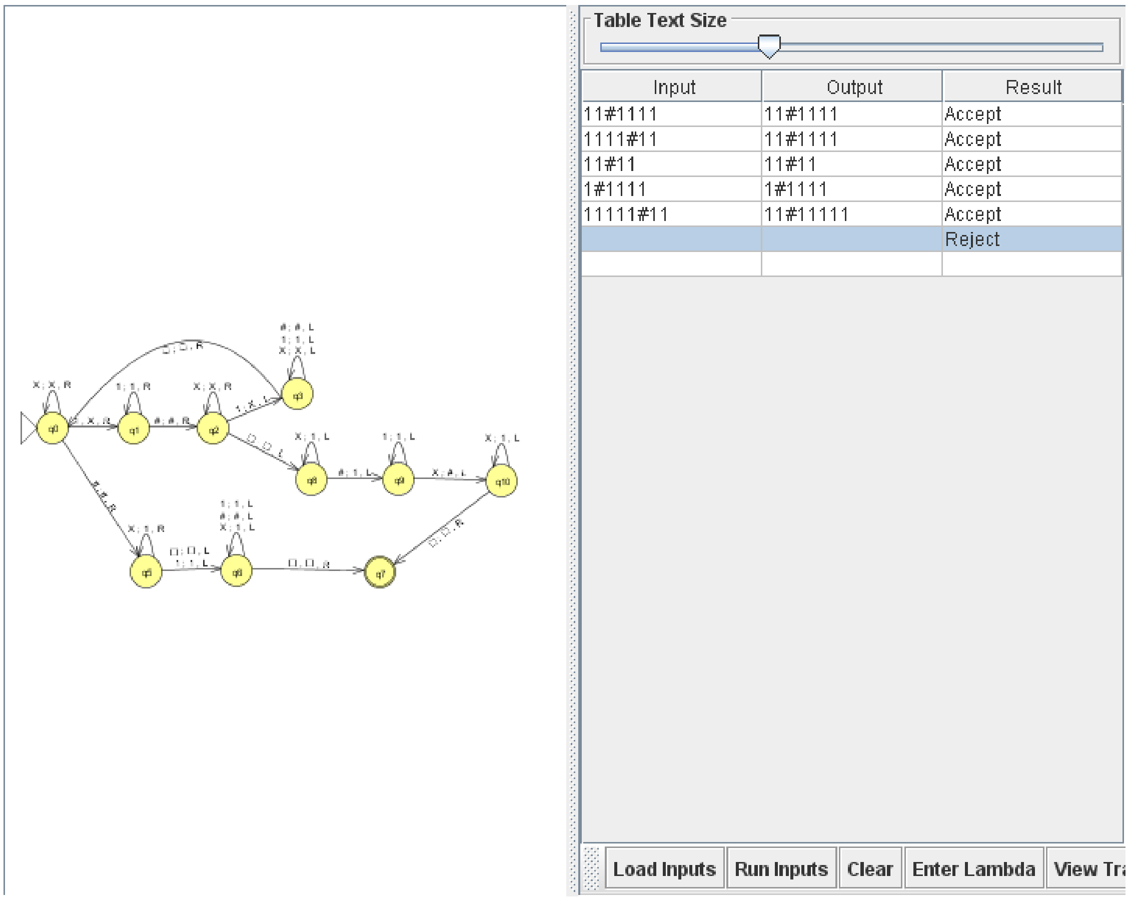
iv. **q6, q7** [step 2 of the algorithm, second part]: back to the first position of the input, changing every X by 1, (transition of q6 to q7), as it required by the exercise.

v. **transition q2 to q8**: detection of the end of the second number, when reaching the first blank going from left to right..

vi. **q8, q9** [step 3 of the algorithm, first part]: unmark all the symbols of the highest number (changing X by 1, and # by 1 going from right to left.

vii. q9, q10 [step 3 of the algorithm, second part]: when reaching the end of the first number (recursive transition in q9), go from right to left until finding a X. Replace this X by a # (transition q9 to q10) to separate both numbers. Recursive transition in q10 to unmark all the symbols of the smallest number (changing X by 1), go from right to left until reaching the first position of the input (transition of q10 to q7), as it required in the exercise.

Recognition of words:



Input	Output	Result
11#1111	11#1111	Accept
1111#11	11#1111	Accept
11#11	11#11	Accept
1#1111	1#1111	Accept
11111#11	11#11111	Accept
		Reject