LAST NAME(s) (Capital letters)			
FIRST NAME (Capital letters)			
NIA		DNI	

INSTRUCTIONS FOR THE EXAM.

- Read these instructions carefully before starting the exam.
- Do not forget to write your name, NIA and DNI in every answer sheet.
- Pay attention to what it is asked in each question and/or problem, given that it is not the same: to explain, to list, to describe, to define, etc., always, sometimes, at least.
- If the sheets are stapled, you cannot remove the staple.
- ANSWER EACH PROBLEM IN DIFFERENT SOLUTION SHEETS, WHETHER IN WHITE.
- THE SHEETS WITH THE FORMULATION OF THE PROBLEMS HAVE TO BE GIVEN AT THE END OF THE EXAM. YOU MUST PRESENT AT LEAST ONE SHEET FOR EACH PROBLEM.
- **The duration of the exam (Test + problems) is 3 hours.**

Problem 1

Recommendations:

- Read the formulations of the problem in detail before answering.
- You must include a detailed justification of your answers.

Given the following grammar:

$G = (\Sigma_T, \Sigma_N, S, P)$ with $\Sigma_T = \{0, 1\}$; $\Sigma_N = \{S, A, B, C, D, E, F\}$
P = {
S ::= λ | 0S | 1A
A ::= 1F | 0B
B ::= 0S | 1F | 0C
C ::= 0C | 1D
D ::= 0D | 1E
E ::= 0DD | 11D
F ::= 1A| 0B
}

1. Apply the necessary minimum changes to the grammar (i.e. to clean, to well-form, to transform into a right-linear G3) to obtain an equivalent Deterministic Finite Automaton (this is required in the following section).

> We can obtain a DFA from the given grammar by just eliminating the production rules with no generative nonterminal symbols. This way, we obtain a right-linear G3 grammar. This is the only aspect that we have to transform, given that questions like inducted axiom or recursion have not influence in this grammar.

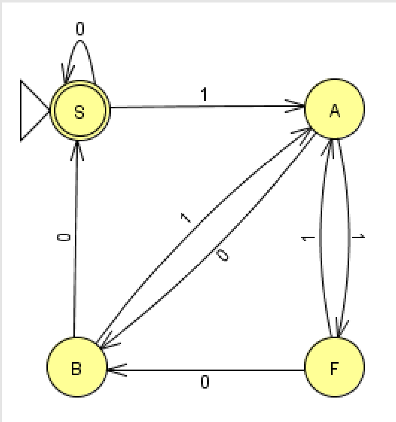
2. Obtain this Deterministic Finite Automaton based on the grammar obtained in the previous section.

   Note: The DFA must be specified by means of the transitions diagram (i.e., a graph, and not through the transition function or table). In addition, the diagram must include all the states and transitions. If not, the answer will be considered wrong.

1. There are not nondeterministic productions, given that the only derivation to the empty word is only for the axiom.
2. We have to generate a node for each nonterminal symbol, an additional final state (X), and a drain state (Q).

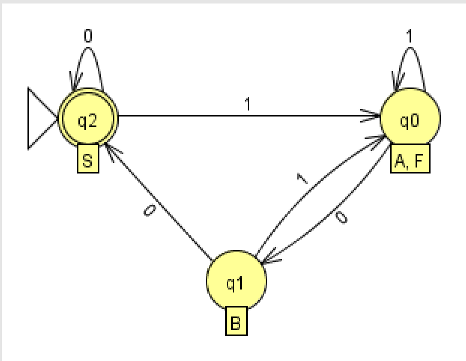
$$P = \{S, A, B, C, X, Q\}$$

3. The initial state (S) is also a final state, given that S ::= λ.
4. Transitions are created according to the production rules in the grammar.
5. Transitions to the drain state are also incorporated.
6. The state X is not connected with the rest of states in the automaton, given that the only generative derivation to lambda is from the axiom.



3. Minimice the DFA obtained.

The minimiced DFA is the following:



4. Indicate the set of words of length less or equal to 4 that is recognized by the DFA.

{λ, 0, 00, 000, 100, 0000, 0100, 1000, 1100}

These words are generated by the grammar, given that:

$$L(G1) = L\ (DFA1) = L(DFAm)$$

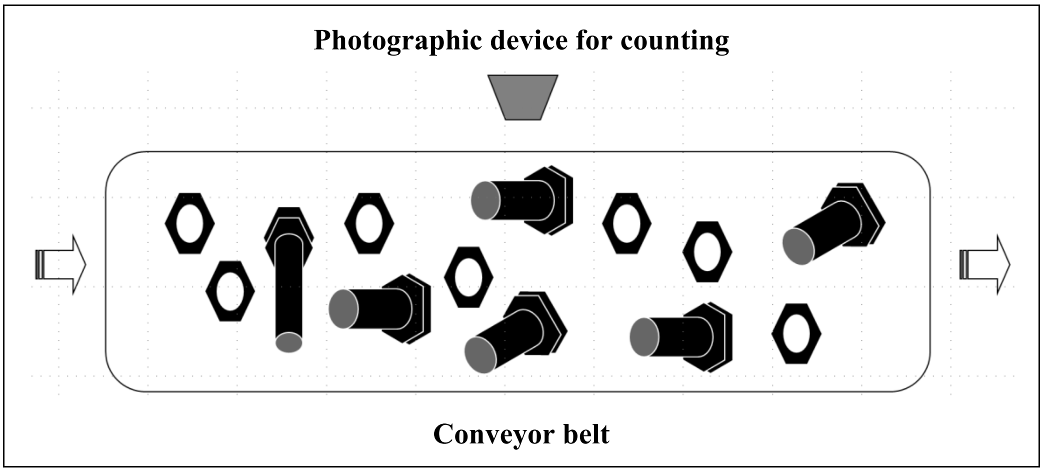
5. Describe the language generated by the grammar of this problem. Justify your answer.

This grammar generates the languages of binary numbers that are multiple of 4 (including nonsignificative ceros on the left side and any possible representation of the values 0 and λ).

Problem 2

An industrial factory has received an order to produce a kit for mounting bookshelves that contains screws and each screw has its corresponding nut.

This factory has decided to develop a photographic device that ensures that there is an **excess** of nuts (with regard the number of screws) before the packaging process starts.



1. Formalize and explain the problem, setting it in the Chomsky Hierarchy, from the point of view of the language and the corresponding automaton. Indicate the alphabet, describe the language, indicate the type of grammar (of the most restricted level) that would generate the language, and what type of Automata/Machine would recognize this language.

A Word can be defined as a sequence of the screws and nuts that are detected, in any order, on the conveyor belt. Therefore, the alphabet of the language is:

$$\Sigma = \{\text{screws, nut}\} = \{s, n\}.$$

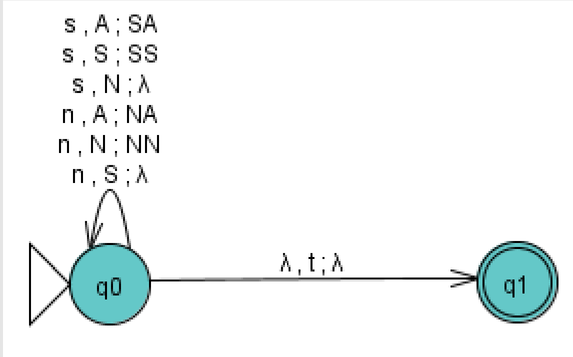
Given that the system has to notify that the process is correct when there is an excess of nuts, it is required a structure/automaton/machine that measures how many crews or nuts are present. The simplest structure than can be designed to calculate this operation is a Pushdown Automaton.

The language consists of the words containing at least one nut more than the number of screws (eg. *s n n*). The grammar that generated this language is a Type-2 grammar.

2. Design the Automaton/Machine, corresponding with the most restricted level of the Chomsky Hierarchy, which provides its approval when the number of nuts is in excess.

Pushdown Automata = (Σ={n,s}, Γ= {N,S,A}, {p,q}, A, p, f, q) which recognizes by means of a final state:

n= nut; *s* = screws.


s , A ; SA
s , S ; SS
s , N ; λ
n , A ; NA
n , N ; NN
n , S ; λ
q0
λ , t ; λ
q1

Problem 3

a. Given the following grammar G= ({a,b},{S,A,B}, S, P) with P:

S ::= Aba

A ::= a

Ab ::= AAbA | ABb | AbB

B ::= A | AB

a.1. Which is the type of grammar? Explain in detail.

It is a Type-1 grammar, given that there are productions rule with more than one symbol on the left side and the length of the left side in these productions is always less than or equal to the length off the right side.

a.2. Define the language generated by means of a regular expression.

The language generated by this grammar can be denoted by the following regular expression aa*ba*a.

a.3. Prove that it is an ambiguous grammar.

To prove that this is an ambiguous grammar it is necessary to find at least one string in the language that could be generated by means of two or more possible derivations (i.e. syntax trees). One of these words is aaabaaaa:

S → Aba → AAbAa → AAAbAAa → AAAbBAAa → AAAbAAAa → aaabaaaa
S → Aba → AbBa → AAbABa → AAAbAaBA →AAAbAaAA → aaabaaaa

a.4. Explain in detail if it is possible to represent this language by means of the following formal machines and structures: a Pushdown Automaton, a Deterministic Finite Automaton, a Turing Machine, a Type-3 grammar.

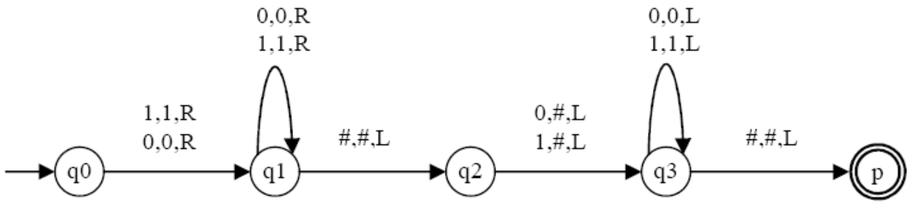
Given that we have found a regular expression to denote the language, this is also a regular language. A Type-3 grammar that generates the same language is:

A → aB
B → aB | bC
C → aC | a

This way, the language can be recognized by means of a deterministic Finite Automaton. As Type-1 languages include Type-2 languages, this language can also be represented by a Pushdown Automaton.

A Turing Machine can recognize the language generated by this grammar, given that these machines can recognize every Type-0, Type-1, Type-2, and Type-3 languages.

b. Given the following Turing Machine



$TM = (\{q_0,q_1,q_2,q_3\},\{0,1\},\{0,1,\#\},\delta,q_0,\{p\})$

b.1. Write the transition table.

The transition table is the following:

Q \ Σ	0	1	#
q_0	$(q_1, 0, R)$	$(q_1, 1, R)$	∅
q_1	$(q_1, 0, R)$	$(q_1, 1, R)$	$(q_2, \#, L)$
q_2	$(q_3, \#, L)$	$(q_3, \#, L)$	∅
q_3	$(q_3, 0, L)$	$(q_3, 1, L)$	$(p, \#, L)$

b.2. Which is the function that the TM carries out? Explain in detail.

This Turing machine reads strings in the language given by the expression (0,1)* and replaces the right-most symbol by a blank (#).

b.3. Show the sequence of movements to process the input string "1110".

This Turing machine processes the input string "1110" as follows:

$\#q_01110\# \rightarrow \#1q_1110\# \rightarrow \#11q_110\# \rightarrow \#111q_10\# \rightarrow \#1110q_1\# \rightarrow \#111q_20\# \rightarrow \#11q_31\#\# \rightarrow$

$\rightarrow \#1q_311\#\# \rightarrow \#q_3111\#\# \rightarrow q_3\#111\#\# \rightarrow p\#\#111\#\#$