uc3m | Universidad **Carlos III** de Madrid

DEPARTMENT OF COMPUTER SCIENCE
CARLOS III UNIVERSITY OF MADRID

Computer Science
Language Processors

Rules

- The duration of the test is **30 minutes**
- Questions will not be answered during the test
- One cannot re-enter the classroom after leaving it
- The answers must be written using a pen (not a pencil)

1.- Write which is the phase and module of the compiling process that would carry out the following functions:

a) Assign a variable to the register 5.

Code generation

b) Identify loop as a label.

Lexical analysis

c) Detect that in an assignment sentence, types are not compatible.

Semantic analysis

d) Change ind + 4*4 by ind+16.

Code optimization

e) Create the syntax tree.

Syntax analysis

f) Detect that a variable has not been declared.

Semantic analysis

David Griol Barres, Antonio Berlanga de Jesús, Jesús García Herrero, Juan Manuel Alonso Weber

2.- Describe in detail which are the steps for developing a lexical analyzer given a regular expression.

The main method that we have studied for implementing a scanner is using regular expressions and finite automata. We can use regular expressions to define the tokens in a programming language. Once we have all our tokens defined using regular expressions, we can create a finite automaton for recognizing them.

First step: From regular expressions to NFA
It is quite easy to take a regular expression and convert it to an equivalent NFA or ʎ-NFA, thanks to the simple rules of Thompson's construction.

Second step: From NFA to DFA
Once we have built an NFA from a regular expression, we can then employ subset construction to convert the NFA to a DFA. Subset construction is an algorithm for constructing the deterministic FA that recognizes the same language as the original nondeterministic FA. Each state in the new DFA is made up of a set of states from the original NFA. The start state of the DFA will be the start state of NFA. The alphabet for both automata is the same.

Third step: Minimizing the number of states
The objective is to apply the specific algorithm to construct a DFA M' accepting the same language as the original M and having as few states as possible.

Fourth step: Implementation of the scanner
We have studied different methodologies to develop the final program: specific purpose programs and programs using transition tables.

We have also to take into account possible methods to carry out error handling, priority of tokens and construction of the symbol table.