uc3m | Universidad Carlos III de Madrid

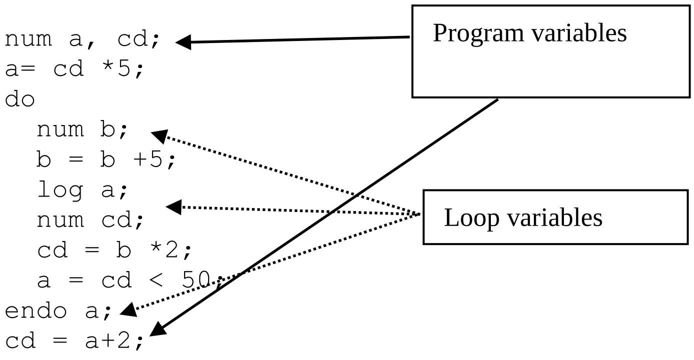
UNITS 2, 3 AND 4: LEXICAL ANALYSIS AND GRAMMAR DESIGN FOR THE SYNTAX ANALYSIS

Build a compiler for the following programming language. There are three types of statements: declaration, arithmetic / logical expression and loop and two types of data: numeric and logical. The sentences are described as follows:

- Declaration
  - **type name_variable [, name_variable]* ;**
  - where type can be **num** or **log** and **name_variable** is a char string with a máximum number of 8 characters
- Arithmetic/Logic expressions
  - **name_variable = expression_arithmetic ;**
  - **name_variable = expression_logic;**
  - Arithmetic expressions can contain variables of type **num** and numbers with the operators: +, -, *, /
  - The logical expression relates variables of type num with numbers through logical operators: <,>, =, #. The possible results of the evaluation of the logical expression is V or F. The variable to which it is assigned must be type log.
- Loop
  - **do [sentence]+ endo [expression_logic | variable_logic] ;**
  - where sentences can be declaration, expression or loop.

The sentences of the loop are executed at least once, and the loop is repeated while the logical expression in endo takes the value V. The variables are local to the loop where they are declared, if variables are used in the logical expression of the endo then there must be been declared inside the loop. The variables declared in the main body of the program are considered local to it.

Example of a program without errors:



```
num a, cd;
a= cd *5;
do
  num b;
  b = b +5;
  log a;
  num cd;
  cd = b *2;
  a = cd < 50;
endo a;
cd = a+2;
```

It is required:

1. Define the G grammar that would generate valid sentences of this programming language.

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