

A grammar example

Daniela da Cruz Pedro Rangel Henriques

{danieladacruz,prh}@di.uminho.pt
Departamento de Informática, CCTC
Universidade do Minho

Language description

In a programming language, named *nLPD*, there are no pre-defined types; it is up to the programmer (the one that writes programs in *nLPD* language) to declare the name of each type that he intends to use in his program. For each declared type, he must establish the length (number of bytes) that should be allocated in memory to the variables of that type. All variables that are declared must have one of the earlier defined types. Those variables will be allocated, by the compiler, in memory continuously, starting at address 0.

The context free grammar *G* below defines just the part of *nLPD* language for **type and variable declarations**.

```
p1: nLPD      --> Types Vars
p2: TYPES     --> TYPES Ts
p3: Ts        --> Type
p4:           | Ts Tipo
p5: Type      --> IdT Len
p6: Vars      --> &
p7:           | VARIABLES Vs
p8: Vs        --> Ids ':' IdT
p9:           | Vs Ids ':' IdT
p10: Ids      --> IdV RIds
p11: RIds     --> &
p12:          | ',' Ids
p13: IdT      --> id
p14: Len      --> num
p15: IdV      --> id
```

Note: Initial Symbol is *nLPD*; Terminal symbols are written in lowercase (pseudo-terminals), or uppercase (reserved-words), or in inverted commas (like `'`, `'` or `'(` or `...)`); empty string is denoted by `&` and character `$` represents EOF (end of input file).

An example of a sentence, derived from *G*, that belongs to *nLPD* language is:

```
TYPES idType 12
VARIABLES a, b : idType
```