

Casos de conflictos por LL (left languages)

var x;
var x = 10; ?

shift conflict

(nombr (exp rag) gccion)

Especificación léxica

(define especificacion-lexica

→ '(
 (espacio-blanco (whitespace) skip)
 (comentario ("% (arbno (not #\newline))) skip)
 (identificador (letter (arbno (or letter digit "?" "\$"))) symbol)
 (numero (digit (arbno digit)) number)
 (numero ("-" digit (arbno digit)) number)
 (numero (digit (arbno digit) "." digit (arbno digit)) number)
 (numero ("-" digit (arbno digit) "." digit (arbno digit)) number)
)
)

(dgtypr (exp rag) varrente)

(define especificacion-gramatical

→ (
 (declaracion ("{" declaracion ";" declaracion "}")) decl-1)
 (declaracion ("while" expresion "do" declaracion) decl-2)
 (declaracion (identificador "!=" expresion) decl-3)
 (expresion (identificador) exp-1)
 (expresion (numero) exp-2)
 (expresion ("(" expresion "+" expresion ")") exp-3))
)

Lenguaje Univalle pro max

Comentarios comienzan con # y terminan en final de línea

- Espacios en blanco son ignorado
- Numeros 1, -1, 1.1, -1.2 --> number
- Identificadores letras seguidas letras o digitos o \$ o ? o ! -> sy

Valores denotados: Numeros

Valores expresados: Numero

Gramatica

<programa> ::= <expresion>

<expresion> ::=

<identificador> Busqueda de variable var-exp

<lit-exp> numero

"(" <expresion> <prim> <expresion> ")"

<prim> := +, *, **, -, /, %

1. Hacer especificación lexica y gramatical
2. Hacer el interprete
3. Funcionalidad