
PRÁCTICA 4:

ANALIZADOR SEMÁNTICO

Alejandro Cárdenas Barranco

Jesús García León

Manuel Moya Martín-Castaño

Juan Manuel Rodríguez Gómez

Procesadores de Lenguajes (Grupo A2, Equipo 7)

Curso 2023 – 2024



UNIVERSIDAD
DE GRANADA

Lenguaje asignado: BBAAD

Nuestro lenguaje asignado tiene las siguientes características:

- Sintaxis inspirada en el lenguaje de programación **C**.
- Palabras reservadas en **inglés**.
- Estructura de datos considerada como tipo elemental: **Listas**.
- Subprogramas: **Funciones**.
- Estructura de control adicional: **do-until**.

Descripción formal de la sintaxis del lenguaje usando BNF (Gramática abstracta)

```
# Programa principal
<programa> ::= <MAIN> <bloque>

# Bloque
<bloque> ::= <INIBLOQUE>
            <declaracion_variables_locales>
            <declaracion_subprogramas>
            <sentencias>
            <FINBLOQUE>

# Variables locales
<declaracion_variables_locales> ::= <LOCAL> <INIBLOQUE>
                                   <variables_locales>
                                   <FINBLOQUE>
                                   |
<variables_locales> ::= <variables_locales> <cuerpo_variables_locales>
                       | <cuerpo_variables_locales>

<cuerpo_variables_locales> ::= <TIPO> <lista_variables> <PYC>

<lista_variables> ::= <lista_variables> <COMA> <ID>
                    | <ID>

# Subprograma (funciones)
<declaracion_subprogramas> ::= <declaracion_subprogramas> <declaracion_subprog>
                              |

<declaracion_subprog> ::= <cabecera_subprog> <bloque>

<cabecera_subprog> ::= <TIPO> <ID> <LPAR> <cabecera_argumentos> <RPAR>

<cabecera_argumentos> ::= <parametros>
                        |

<parametros> ::= <parametros> <COMA> <parametro>
                | <parametro>
```

```

<parametro> ::= <TIPO> <ID>

# Sentencias
<sentencias> ::= <sentencias> <sentencia>
                |

<sentencia> ::= <bloque>
                | <expresion> <PYC>
                | <sentencia_asignacion> <PYC>
                | <sentencia_lista> <PYC>
                | <sentencia_if>
                | <sentencia_while>
                | <sentencia_entrada> <PYC>
                | <sentencia_salida> <PYC>
                | <sentencia_do_until> <PYC>
                | <sentencia_return> <PYC>

# Asignación de valores
<sentencia_asignacion> ::= <ID> <ASIGN> <expresion>

# Listas
<sentencia_lista> ::= <expresion> <SHIFT>
                    | <OPDOLLAR> <expresion>

# Condicional
<sentencia_if> ::= <IF> <LPAR> <expresion> <RPAR> <sentencia> <sentencia_else>

<sentencia_else> ::= <ELSE> <sentencia>
                  |

# Bucle while
<sentencia_while> ::= <WHILE> <LPAR> <expresion> <RPAR> <sentencia>

# Entrada
<sentencia_entrada> ::= <CIN> <lista_id>

<lista_id> ::= <lista_id> <COMA> <ID>
            | <ID>

# Salida
<sentencia_salida> ::= <COUT> <lista_expresiones_cadenas>

<lista_expresiones_cadenas> ::= <lista_expresiones_cadenas> <COMA>
<expresion_cadena>
                            | <expresion_cadena>

<expresion_cadena> ::= <expresion>
                    | <CADENA>

# Do-until
<sentencia_do_until> ::= <DO> <sentencia> <UNTIL> <LPAR> <expresion> <RPAR>

# Return
<sentencia_return> ::= <RETURN> <expresion>

```

```

# Expresion
<expresion> ::= <LPAR> <expresion> <RPAR>
              | <OPSUMRES> <expresion>
              | <OPNOT> <expresion>
              | <OPINTHASH> <expresion>
              | <expresion> OPAT <expresion>
              | <expresion> OPAND <expresion>
              | <expresion> OPOR <expresion>
              | <expresion> OPEQN <expresion>
              | <expresion> OPSUMRES <expresion>
              | <expresion> OPMULDIV <expresion>
              | <expresion> OPPORPOR <expresion>
              | <expresion> OPBORRLIST <expresion>
              | <expresion> OPCOMP <expresion>
              | <expresion> <OPMASMAS> <expresion> <OPAT> <expresion>
              | <llamada_funcion>
              | <ID>
              | <constante>

<llamada_funcion> ::= <ID> <LPAR> <argumentos> <RPAR>

<argumentos> ::= <lista_argumentos>
                |

<lista_argumentos> ::= <lista_argumentos> <COMA> <expresion>
                      | <expresion>

<lista_expresiones> ::= <lista_expresiones> <COMA> <expresion>
                      | <expresion>

# Constantes
<constante> ::= <CONSTANTS>
              | <lista>

<lista> ::= <LCOR> <lista_expresiones> <RCOR>

```

Tabla de Tokens

Nombre	Expresión regular	Código del token	Atributos
ID	[a-z A-Z][a-z A-Z 0-9 _]*	257	
ASIGN	"="	258	
IF	"if"	259	
ELSE	"else"	260	
LPAR	"("	261	
RPAR)"	262	

Nombre	Expresión regular	Código del token	Atributos
OPSUMRES	"+" "-"	263	0: + 1: -
OPMULDIV	"*" "/"	264	0: * 1: /
OPPORPOR	"**"	265	
OPEQN	"==" "!="	266	0: == 1: !=
OPAND	"&&"	267	
OPOR	" "	268	
OPCOMP	"<" ">" "<=" ">="	269	0: < 1: > 2: <= 3: >=
OPBORRLIST	"--" "%"	270	0: -- 1: %
OPNOT	"!"	271	
OPINTHASH	"?" "#"	272	0: ? 1: #
INIBLOQUE	"{"	273	
FINBLOQUE	"}"	274	
LOCAL	"local"	275	
TIPO	"int" "float" "char" "bool" "list_of" "+"int" "list_of" "+"float" "list_of" "+"char" "list_of" "+"bool"	276	0: int 1: float 2: char 3: bool 4: list_of int 5: list_of float 6: list_of char 7: list_of bool
PYC	","	277	
CIN	"cin"	278	
COUT	"cout"	279	
CADENA	"\[^\"]*"	280	
RETURN	"return"	281	
OPAT	"@"	282	
OPMASMAS	"++"	283	
CONSTANTS	[0-9]+ [0-9]+\.[0-9]+ \"[^\"]\" ("true" "false")	284	0: int 1: float 2: char 3: bool
COMA	","	285	
MAIN	"main"	286	
DO	"do"	287	
UNTIL	"until"	288	

Nombre	Expresión regular	Código del token	Atributos
WHILE	"while"	289	
LCOR	"["	290	
RCOR	"]"	291	
OPDOLLAR	"\$"	292	
SHIFT	"<<" ">>"	293	0: << 1: >>