



UNIVERSIDAD DE GRANADA

PRÁCTICA 1

Diseño del Lenguaje

Grupo 4 Viernes

Víctor Manuel Arroyo Martín
Sergio Cabezas González de Lara
Pedro Ramos Suárez
David Villar Martos

Quinto curso del Doble Grado de Ingeniería Informática y Matemáticas:
Procesadores de Lenguajes

13 de enero de 2021

Índice

1. Introducción	1
2. Especificaciones del lenguaje BABAA	1
3. Descripción formal de la sintaxis del lenguaje usando BNF	1
4. Definición de la semántica en lenguaje natural	4
5. Identificación de los tokens con el máximo nivel de abstracción	5

1. Introducción

Nuestro lenguaje será un subconjunto de un lenguaje de programación estructurado en el que los identificadores deben ser declarados antes de ser usados. Nos basaremos en una sintaxis inspirada en el lenguaje C donde las palabras reservadas van a estar escritas en castellano de la forma más natural posible además, como estructura de datos, usaremos arrays de 1 o 2 dimensiones (vectores o matrices). Como estructura de control usaremos “case-switch” a parte de las comunes “if-then”, “if-else” y “while”.

2. Especificaciones del lenguaje BABAA

1. **Sintaxis inspirada en un lenguaje de programación:** B, Lenguaje C.
2. **Palabras reservadas en:** A, Castellano.
3. **Estructura de datos considerada como tipo elemental:** B, Array 1D y 2D.
4. **Subprogramas:** A, Funciones.
5. **Estructuras de control adicional:** A, Case/switch.

3. Descripción formal de la sintaxis del lenguaje usando BNF

$$\begin{aligned} \langle \textit{programa} \rangle &::= \langle \textit{cabecera_programa} \rangle \langle \textit{bloque} \rangle \\ \langle \textit{bloque} \rangle &::= \langle \textit{inicio_bloque} \rangle \\ &\quad \langle \textit{declarar_variables_locales} \rangle \\ &\quad \langle \textit{declarar_subprogs} \rangle \\ &\quad \langle \textit{sentencias} \rangle \\ &\quad \langle \textit{fin_bloque} \rangle \\ \langle \textit{declarar_subprogs} \rangle &::= \langle \textit{declarar_subprogs} \rangle \langle \textit{declarar_subprog} \rangle \\ &\quad | \\ \langle \textit{declarar_subprog} \rangle &::= \langle \textit{cabecera_subprograma} \rangle \langle \textit{bloque} \rangle \\ \langle \textit{declarar_variables_locales} \rangle &::= \langle \textit{marca_ini_declarar_variables} \rangle \\ &\quad \langle \textit{variables_locales} \rangle \\ &\quad \langle \textit{marca_fin_declarar_variables} \rangle \\ &\quad | \end{aligned}$$

$\langle \text{marca_ini_declarar_variables} \rangle ::= \text{"INIVAR"}$
 $\langle \text{marca_fin_declarar_variables} \rangle ::= \text{"FINVAR"}$
 $\langle \text{cabecera_programa} \rangle ::= \text{"PRINCIPAL"}$
 $\langle \text{inicio_bloque} \rangle ::= \text{"\{"}$
 $\langle \text{fin_bloque} \rangle ::= \text{"\}"}$
 $\langle \text{variables_locales} \rangle ::= \langle \text{variables_locales} \rangle \langle \text{cuerpo_declarar_variables} \rangle$
 $\quad |$
 $\langle \text{cuerpo_declarar_variables} \rangle ::= \langle \text{tipo} \rangle \langle \text{lista_identificadores} \rangle \text{";"}$
 $\langle \text{lista_identificadores} \rangle ::= \langle \text{lista_identificadores} \rangle \text{","} \langle \text{identificador} \rangle$
 $\quad [\text{"="} \langle \text{expresion} \rangle]$
 $\quad | \langle \text{identificador} \rangle [\text{"="} \langle \text{expresion} \rangle]$
 $\langle \text{cabecera_subprograma} \rangle ::= \langle \text{tipo} \rangle \langle \text{identificador} \rangle$
 $\quad \text{"("} \langle \text{lista_parametros} \rangle \text{")"}$
 $\quad | \langle \text{tipo} \rangle \langle \text{identificador} \rangle \text{"(" "}"$
 $\langle \text{sentencias} \rangle ::= \langle \text{sentencias} \rangle \langle \text{sentencia} \rangle$
 $\quad | \langle \text{sentencia} \rangle$
 $\langle \text{sentencia} \rangle ::= \langle \text{bloque} \rangle$
 $\quad | \langle \text{sentencia_asignacion} \rangle$
 $\quad | \langle \text{sentencia_if} \rangle$
 $\quad | \langle \text{sentencia_while} \rangle$
 $\quad | \langle \text{sentencia_entrada} \rangle$
 $\quad | \langle \text{sentencia_salida} \rangle$
 $\quad | \langle \text{sentencia_return} \rangle$
 $\quad | \langle \text{sentencia_switch} \rangle$
 $\langle \text{sentencia_asignacion} \rangle ::= \langle \text{identificador} \rangle \text{"="} \langle \text{expresion} \rangle \text{";"}$
 $\quad | \langle \text{identificador} \rangle [\text{"["} \langle \text{expresion} \rangle \text{"]"} \text{"="} \langle \text{expresion} \rangle \text{";"}$
 $\quad | \langle \text{identificador} \rangle [\text{"["} \langle \text{expresion} \rangle \text{","} \langle \text{expresion} \rangle \text{"]"} \text{"="} \langle \text{expresion} \rangle \text{";"}$
 $\langle \text{sentencia_if} \rangle ::= \text{"SI"} \text{"("} \langle \text{expresion} \rangle \text{")"} \langle \text{sentencia} \rangle$
 $\quad [\text{"SINO"} \langle \text{sentencia} \rangle]$
 $\langle \text{sentencia_while} \rangle ::= \text{"MIENTRAS"} \text{"("} \langle \text{expresion} \rangle \text{")"} \langle \text{sentencia} \rangle$
 $\langle \text{sentencia_entrada} \rangle ::= \text{"LEER"} \langle \text{lista_variables} \rangle \text{";"}$
 $\langle \text{sentencia_salida} \rangle ::= \text{"IMPRIMIR"} \langle \text{lista_expresiones_o_cadena} \rangle \text{";"}$
 $\langle \text{sentencia_return} \rangle ::= \text{"DEVOLVER"} \langle \text{expresion} \rangle \text{";"}$
 $\langle \text{sentencia_switch} \rangle ::= \text{"SEGUN"} \langle \text{identificador} \rangle \text{"CASO"}$
 $\quad \langle \text{constante} \rangle \text{":"} \langle \text{sentencia} \rangle \text{"FINCASO"} \text{";"}$
 $\quad \langle \text{mas_casos} \rangle$
 $\quad \text{"PORDEFECTO"} \text{":"} \langle \text{sentencia} \rangle \text{"FINCASO"} \text{";"}$
 $\langle \text{mas_casos} \rangle ::= \langle \text{mas_casos} \rangle \text{"CASO"} \langle \text{constante} \rangle \text{":"} \langle \text{sentencia} \rangle$
 $\quad \text{"FINCASO"} \text{";"}$
 $\quad |$
 $\langle \text{lista_parametros} \rangle ::= \langle \text{lista_parametros} \rangle \text{","} \langle \text{tipo} \rangle \langle \text{identificador} \rangle$
 $\quad | \langle \text{tipo} \rangle \langle \text{identificador} \rangle$

$$\begin{aligned}
< lista_variables > &::= < lista_variables > \text{“,”} < identificador > \\
&| < identificador > \\
< lista_expresiones_o_cadena > &::= < lista_expresiones_o_cadena > \text{“,”} < cadena > \\
&| < lista_expresiones_o_cadena > \text{“,”} < expresion > \\
&| < cadena > \\
&| < expresion > \\
< cadena > &::= \text{“ ”} < lista_car > \text{“ ”} \\
< lista_car > &::= < caracter > < lista_car > \\
&| \\
< expresion > &::= \text{“ (”} < expresion > \text{“) ”} \\
&| < operador_un > < expresion > \\
&| < expresion > < operador_sig > \\
&| < expresion > < operador_bi > < expresion > \\
&| < expresion > \text{“ [”} < expresion > \text{“] ”} \\
&| < expresion > \text{“ [”} < expresion > \text{“ , ”} < expresion > \text{“] ”} \\
&| < constante > \\
&| < identificador > \\
&| < funcion > \\
< operador_un > &::= \text{“ - ”} | \text{“ ! ”} | < operador_inc > \\
< operador_sig > &::= \text{“ + + ”} | \text{“ - - ”} | \text{“ [”} < expresion > \text{“] ”} \\
< operador_bin > &::= \text{“ + ”} | \text{“ - ”} | \text{“ * ”} | \text{“ / ”} | \text{“ < ”} | \text{“ < = ”} | \text{“ > ”} | \text{“ > = ”} | \text{“ = = ”} \\
&| \text{“ ! = ”} | \text{“ & & ”} | \text{“ | ”} | \text{“ * * ”} \\
< constante > &::= < entero > \\
&| < real > \\
&| \text{“ ' ”} < caracter > \text{“ ' ”} \\
&| < booleano > \\
&| < array1d > \\
&| < array2d > \\
< entero > &::= < digito > < entero > \\
&| < digito > \\
< real > &::= < entero > \\
&| < entero > \text{“ . ”} < entero > \\
&| < entero > \text{“ . ”} \\
< booleano > &::= \text{“ VERDADERO ”} | \text{“ FALSO ”} \\
< array1d > &::= \text{“ { ”} < expresion > < mas_elementos > \text{“ } ”} \\
&| \text{“ { ”} \text{“ } ”} \\
< mas_elementos > &::= < mas_elementos > \text{“ , ”} < expresion > \\
&| \\
< array2d > &::= \text{“ [”} < array1d > < mas_arrays > \text{“] ”} \\
< mas_arrays > &::= < mas_arrays > < array1d > \\
&|
\end{aligned}$$

$$\begin{aligned}
< \text{identificador} > &::= < \text{letra} > < \text{ident_aux} > \\
&\quad | \text{"_"} < \text{ident_aux} > \\
< \text{ident_aux} > &::= < \text{letra} > < \text{ident_aux} > \\
&\quad | < \text{digito} > < \text{ident_aux} > \\
&\quad | \text{"_"} < \text{ident_aux} > \\
&\quad | \\
< \text{funcion} > &::= < \text{identificador} > \text{"("} < \text{lista_expresiones} > \text{"}")} \\
< \text{lista_expresiones} > &::= < \text{lista_expresiones_aux} > < \text{expresion} > \\
&\quad | \\
< \text{lista_expresiones_aux} > &::= < \text{lista_expresiones_aux} > < \text{expresion} > \text{" ,"} \\
&\quad | \\
< \text{tipo} > &::= \text{"ARRAY1D"} \text{"["} < \text{entero} > \text{"]"} < \text{tipo_base} > \\
&\quad | \text{"ARRAY2D"} \text{"["} < \text{entero} > \text{" ,"} < \text{entero} > \text{"]"} < \text{tipo_base} > \\
&\quad | < \text{tipo_base} > \\
< \text{tipo_base} > &::= \text{"ENTERO"} \\
&\quad | \text{"REAL"} \\
&\quad | \text{"CARACTER"} \\
&\quad | \text{"BOOLEANO"} \\
< \text{caracter} > &::= \text{Cualquier carácter en formato ASCII.} \\
< \text{digito} > &::= \text{"0"} | \dots | \text{"9"} \\
< \text{letra} > &::= \text{"a"} | \dots | \text{"z"} | \text{"A"} | \dots | \text{"Z"} | \text{"ñ"} | \text{"Ñ"} | \text{"á"} | \text{"é"} | \text{"í"} \\
&\quad | \text{"ó"} | \text{"ú"} | \text{"Á"} | \text{"É"} | \text{"Í"} | \text{"Ó"} | \text{"Ú"} | \text{"ü"} | \text{"Û"}
\end{aligned}$$

4. Definición de la semántica en lenguaje natural

Nuestro lenguaje se basa en la sintaxis de C traducido al español. A continuación explicamos el significado de determinados aspectos estructurales del lenguaje, así como de distintos tipos de órdenes que podrán implementarse:

- Utilizamos dos marcas para indicar el inicio y el fin de la declaración de variables locales en un bloque. Dichas marcas son INIVAR y FINVAR. Fuera de éstas no se podrán declarar nuevas variables, pero sí se podrán manipular las variables existentes.
- Sentencia if: En nuestro lenguaje es la sentencia SI y funciona como el if del lenguaje C. Añadiendo un SINO obtenemos un funcionamiento similar al else tradicional.
- Sentencia while: Se usa la palabra MIENTRAS, y su funcionamiento es el de un bucle while convencional.
- Sentencia de lectura: LEER, desde una entrada lee un valor y lo almacena en una o más variables.
- Sentencia de salida: IMPRIMIR, imprime el valor de una lista de expresiones o una cadena de caracteres por la salida estándar.

- Sentencia switch: Utilizando las expresiones SEGUN(switch), CASO(case), FINCASO(break) y PORDEFECTO(default) ejecuta una serie de sentencias según el valor de la variable que aparezca tras SEGUN.
- Funciones: Al igual que en C, se declaran con una cabecera, el contenido como un bloque y una sentencia return si fuera.
- Como tipo de dato adicional, implementamos arrays 1D y 2D. Para declararlos, especificaremos el tipo: ARRAY1D o ARRAY2D, seguido de su dimensión con la misma sintaxis que se usa en C. Tras ello, especificaremos el tipo base de los elementos del array.
- Las operaciones que definiremos sobre el array son las especificadas en la página 29 del guión, utilizando la misma sintaxis allí indicada.

5. Identificación de los tokens con el máximo nivel de abstracción

TOKENS	ID	ATRIBUTOS	PATRÓN
CABECPROG	257		PRINCIPAL
INILLAVE	258		{
FINLLAVE	259		}
MARCAINI	260		INIVAR
MARCAFIN	261		FINVAR
TIPOVAR	262	0:ENTERO, 1:REAL 2:CARACTER, 3:BOOLEANO	ENTERO, REAL CARACTER, BOOLEANO
FINLINEA	263		;
COMA	264		,
IGUAL	265		=
INICOR	266		[
FINCOR	267]
INIPAR	268		(
FINPAR	269)
IDENTIFICADOR	270		[a-ZáéíóúñÁÉÍÓÚüŮÑ_]+ [a-ZáéíóúñÁÉÍÓÚüŮÑ_0-9]*
CONDIF	271		SI
CONDELSE	272		SINO
CONDWHILE	273		MIENTRAS
ENTRADA	274		LEER
SALIDA	275		IMPRIMIR
RETORNO	276		DEVOLVER
CONDSWITCH	277		SEGUN
CASOSWITCH	278		CASO
DEFECTOSWITCH	279		PORDEFECTO
FINSWITCH	280		FINCASO
SENTSWITCH	281		:
CADENA	282		\ "[\x00-\xFF]* \ "

TOKENS	ID	ATRIBUTOS	PATRÓN
CARACTER	283		\ ' [\x00-\xFF]* \'
OPSIG	284	0:++, 1:- -	++, - -
OPUN	285		!
OPBIN	286	0:+,1:*, 2:/, 3:<, 4:<=, 5:>, 6:>=, 7:==, 8:!=, 9:&&, 10: , 11:**	+,*, /, <, <=, >, >=, ==, !=, &&, , **
OPMENOS	287		-
ENTERO	288		[0-9]+
DECIMAL	289		.
REAL	290		[0-9]+.[0-9]*
VALBOOLEANO	291	0:VERDADERO, 1: FALSO	VERDADERO, FALSO
TIPOARRAY1D	292		ARRAY1D
TIPOARRAY2D	293		ARRAY2D
DELARRAY2D	294		