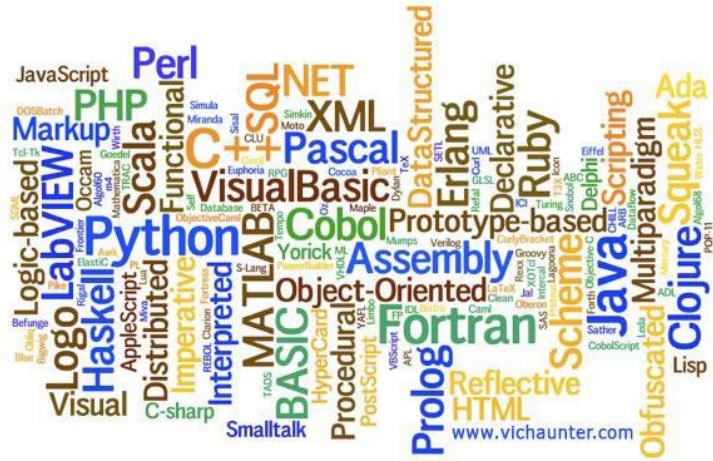


Tecnológico de Costa Rica
Escuela de Ingeniería en Computación



Tarea I “Gramática BNF”

Compiladores e intérpretes

Profesor:
Rodríguez Dávila, Allan

Presentado por:
Bonilla Espinoza, Alina [2016248502]
Lara Osés, Froylan [2018276191]

Marzo, 2023

Descripción del problema

Un grupo de desarrolladores desea crear un nuevo lenguaje imperativo, ligero, que le permita realizar operaciones básicas para la configuración de chips, ya que esta es una industria que sigue creciendo constantemente, y cada vez estos chips necesitan ser configurados por lenguajes más ligeros y potentes.

Es por esto que este grupo de desarrolladores requiere desarrollar su propio lenguaje para el desarrollo de sistemas empotrados, y como primer paso necesitan desarrollar una gramática simple y poderosa.

Diseño del programa

A continuación, se detalla la lista de terminales, no terminales y la gramática BNF del lenguaje en producción:

1. Lista de terminales:

Terminales			
int	+	*	==
float	-	/	!=
char	=	~	<
string	\$	**	>
bool	true	--	<=
arregloEstatico	false	++	>=
^	#	main	if
elif	else	do while	for
while	print	input	\s
()	{	}
[]	\d	\f
\n	,		

2. Lista de no terminales:

No terminales		
<signos>	<tiposDatos>	<elementosBloque>
<equal>	<identificador>	<operandoLogicos>
<endLine>	<operador>	<operacionLogica>
<retorno>	<operadorUnario>	<condicionales>
<numFloat>	<expresionesLogicasBool>	<ifEstructura>
<sentencias>	<expresionesLogicasNum>	<whileEstructura>
<tipoBool>	<expresionesLogicas>	<condicionFor>
<tipoChar>	<operadorRelacional>	<forEstructura>
<tipoString>	<tiposOperandos>	<doWhile>
<break>	<comentarioSimple>	<estructurasControl>
<input>	<comentarioMultiple>	<operadorEntero>
<print>	<operadorFloat>	<expresionBoolean>
<varString>	<varBoolean>	<operandorArreglos>
<varChar>	<operadorChar>	<operadorString>
<funcion>	<asignacionArreglo>	<obtenerValorArreglo>
<bloque>	<operacionUnitaria>	<formatoDato>
<sentencia>	<expresionAritmetica>	<arregloEstatico>
<main>	<varFloatCreaAsigOp>	<llamadaFuncion>
<numEntero>	<variableCrea>	<varEnteroCreaAsigOp>
<parametro>	<varArregloEstatico>	

3. Símbolo inicial:

<programa> ::=
 <sentencia>*<funcion>*<sentencia>*<main><sentencia>*<funcion><sentencia>*

4. Producciones:

A continuación, se detalla la gramática BNF del lenguaje en producción:

<signos> :: = +|-

<equal> :: = \=

<endLine> :: = \\$

```
<numEntero> ::= <signos>?[1-9][\d]*|0
<numFloat> ::= <signos>?[\d]+.\.[\d]+
<arregloEstatico> ::= int[*]*|char[*]*
<tipoBool> ::= true|false
<tipoChar> ::= .
<tipoString> ::= .*
<formatoDato> ::= "%d"|"%f"
<tiposDatos> ::= int|float|bool|char|string|<arregloEstatico>
<identificador> ::= [a-zA-Z][a-zA-Z0-9\_]*
<operador> ::= \+|-|\*|\V|\~|\**|
<operadorUnario> ::= --|++
<expresionesLogicasBool> ::= ==|!=
<expresionesLogicasNum> ::= <>|<=|=|
<expresionesLogicas> ::= <expresionesLogicasNum>|<expresionesLogicasBool>
<operadorRelacional> ::= ^|#
<tiposOperandos> ::=
[<identificador>|<numEntero>|<numFloat>|<identificador>\[numEntero\]]]
<parametro> ::= <tiposDatos><identificador>
<operandosLogicos> ::=
<identificador>||<numEntero>|<numFloat>|<identificador>\[numEntero\]|<llamada
Funcion>
<operacionLogica> ::=
\!<operandosLogicos><expresionesLogicas><operandosLogicos>
<condicionales> ::=
[!<identificador>][<operacionLogica>[<operadorRelacional><operacionLogica>
]*]
```

```

<ifEstructura> ::=

“if”\(<condicionales>\)\{<bloque>\} [“elif”\(<condicionales>\)\{bloque\}]*[“else”\{bloque\}]?<endLine>

<whileEstructura> ::= while\(<condicionales>\)\{bloque\}<endLine>

<condicionFor> ::=

[int|float]<identificador><equal><operandoLogico><endLine><condicionales><endLine><operacionUnitaria>

<forEstructura> ::= “for”\(<condicionFor>\)\{bloque\}<endLine>

<doWhile> ::= “do”\{<bloque>\}“while”\(<condicionales>\)<endLine>

<estructurasControl> ::=

<ifEstructura>|<whileEstructura>|<forEstructura>|<doWhile>

<elementosBloque> ::= <sentencias>|<estructurasControl>

<bloque> ::= <elementosBloque>*

<retorno> ::=

“return”[<operandoLogico>|<tipoBool>|<tipoChar>|<tipoString>]<endLine>

<break> ::= “break”<endLine>

```

- Expresion Aritmetica

```

<expresionAritmetica> ::=

[<numEnteros>|<numFloat>|<identificador>|<identificador>\[<numEntero>\]]<operador>[<numEnteros>|<numFloat>|<identificador>|<identificador>\[<numEntero>\]]<endLine>

```

- Creacion y asignacion de variables de tipo entero

```

<operadorEntero> ::=

<identificador>|<numEntero>|<identificador>\[<numEntero>\]|<llamadaFuncion>

```

- Sentencias:

```

<sentencia> ::=

<operacionUnitaria>|<varEnteroCreaAsigOp>|<varChar>|<print>|<input>|<varString>|<varArregloEstatico>|<asignacionArreglo>|<obtenerValorArreglo>

<sentencias> ::= <sentencia>|<retorno>|<break>

<operacionUnitaria> ::=

[<identificador>|<identificador>\[numEntero\]]<operadorUnario><endLine>

<variableCrea> ::= <tiposDatos><identificador>*<endLine>

<varEnteroCreaAsigOp> ::=

"int"<identificador><equal><operadorEntero>[<operador><operadorEntero>]*<endLine>

<varFloatCreaAsigOp> ::=

"float"<identificador><equal><operadorFloat>[<operador><operadorFloat>]*<endLine>

<print> ::= "print"\("."")<endLine>

<input> ::= "input"\(<formatoDato>\,<identificador>\)<endLine>

<comentarioSimple> ::= "@".*"\n"

<comentarioMultiple> ::= "V_".*\s.* "_V"

```

- Creacion y asignacion de variables de tipo float

```

<operadorFloat> ::=

<identificador>|<numFloat>|<identificador>\[<numEntero>\]|<llamadaFuncion>

```

- Creacion y asignacion de variables de tipo booleano

```

<varBoolean> ::= "bool"<identificador><equals><tipoBool><endLine>

<expresionBoolean> ::=

<tiposOperandos>[<expresionesLogicas><tiposOperandos>]*<endLine>

```

- Funcion:

```

<funcion> ::=

<tiposDatos><identificador>\([<parametro>\[,<parametro>]*]?)\{<bloque>\}<endLine>

<main> ::= "int main"\([<parametro>\[,<parametro>]*]?)\{<bloque>\}<endLine>

<llamadaFuncion> ::= <identificador>\([<parametro>\[,<parametro>]*]?)
```

- Creación y asignación de variables de tipo char

```

<operadorChar> ::=

<tipoChar>|<identificador>|<identificador>\[<numEntero>\]|<llamadaFuncion>

<varChar> ::= "char"<identificador><equals><operadorChar><endLine>
```

- Creación y asignación de variables de tipo string

```

<operadorString> ::= <tipoString>|<identificador>|<llamadaFuncion>

<varString> ::=

"string"<identificador><equals><operadorString>[+<operadorString>]*<endLine>
```

- Creación y asignación de variables de arreglo int o char

```

<varArregloEstatico> ::= ["int"]|"char"<identificador>\[<numEntero>\]<endLine>

<operandorArreglos> ::=

<numEntero>|<tipoChar>|<identificador>\[<numEntero>]|<identificador>|<expresionAritmetica>|<llamadaFuncion>

<asignacionArreglo> ::=

<identificador>\[<numEntero>\]<equals><operadorArreglos><endLine>

<obtenerValorArreglo> ::= <identificador>\[<numEntero>\]<endLine>
```

Análisis de resultados

Lecciones aprendidas

Gracias a la realización de esta tarea, los responsables pusieron en práctica su conocimiento obteniendo con respecto a los temas de: expresiones regulares, terminales y no terminales. Así mismo se perfeccionaron habilidades de trabajo en equipo como: comunicación asertiva, trabajar de forma ordenada, la mayoría de los avances se realizaron de forma sincrónica para facilitar la comunicación, ya que muchas partes del trabajo dependen de otras.

Objetivos alcanzados

- Todos los objetivos solicitados en la tarea fueron desarrollados (del inciso a hasta el q), se realizaron con éxito.

Objetivos no logrados

- Todos los objetivos solicitados fueron alcanzados

Bitácora

Seguidamente, se muestra la bitácora recaudada del archivo compartido en la plataforma de google Drive de los integrantes: Froylan Lara y Alina Bonilla (tatiana espinoza).

<p>▶ 18 de marzo, 0:58 :</p> <p><i>Versión actual</i></p> <p>● tatiana espinoza ● Froylan Lara</p>	<p>▶ 16 de marzo, 20:20 :</p> <p>● tatiana espinoza</p>
<p>AYER</p> <hr/>	<p>▶ 16 de marzo, 19:09</p> <p>● tatiana espinoza</p>
<p>▶ 17 de marzo, 19:38</p> <p>● tatiana espinoza</p>	<p>16 de marzo, 14:49</p> <p>● tatiana espinoza</p>
<p>▶ 17 de marzo, 18:40</p> <p>● tatiana espinoza</p>	<p>MARTES</p>
<p>▶ 17 de marzo, 10:46</p> <p>● Froylan Lara</p>	<p>▶ 14 de marzo, 23:47</p> <p>● Froylan Lara</p>
<p>JUEVES</p> <hr/>	<p>LUNES</p>
<p>▶ 16 de marzo, 22:57</p> <p>● Froylan Lara ● tatiana espinoza</p>	<p>13 de marzo, 11:24</p> <p>● Froylan Lara</p> <p>Archivo .docx importado - Ver el original</p>
	<p><input checked="" type="checkbox"/> Mostrar cambios</p>