

Práctica 4 - gramática libre

Objetivo: Diseñar la sintaxis de cada sentencia posible en el lenguaje propio con el fin de preparar una gramática LL que permita notificar fallos durante el análisis.

Instrucciones:

1. Tomando en cuenta el programa que contiene el código de su propuesta de lenguaje, analice las variaciones que podrían tener al menos 3 tipos de sentencias y elabore una gramática siguiendo las reglas de generación de producciones, siga el ejemplo:

gramatica \square *encabezado listaInstr*

encabezado \square **add** *file next*

next \square *, file next* | *, file* | ϵ

listaInstr \square **begin** *listaInstr* **end** | *instr listaInstr* | *instr* | ϵ

instr \square *id = factor* | **display** *opcion* | **keyboard** *id*

factor \square *factor+factor* | *opcion*

opcion \square *id* | *num*

file \square **common** | **graphic** | **advance** | *io*

```
VARIABLE ENTERO num1 = 1;
VARIABLE ENTERO num2 = 2;
```

```
SI? (num1 == num2){
    IMPRIME('HOLA')
}
DE_LO_CONTRARIO {
    num1=num2;
}
```



DECLARACION -> **VARIABLE** TIPO ID = VALOR;

TIPO -> **ENTERO** | **DECIMAL** | **TEXTO** | **BOOLEANO**

VALOR -> ENTERO | DECIMAL | TEXTO | VERDADERO | FALSO

IF -> **SI?**(EVALUACION){INSTRUCCION} ELSE

EVALUACION -> ID OPERADOR_COMP ID | ID OPERADOR_COMP VALOR | VALOR OPERADOR_COMP VALOR | VALOR OPERADOR_COMP ID

OPERADOR_COMP -> != | == | <= | >= | < | >

INSTRUCCION -> IMPRIMIR; INSTRUCCION | ASIGNACION; INSTRUCCION | EPSILON

ASIGNACION -> ID = ID; | ID = VALOR; | ID = ID O; | ID = VALOR O;

O -> + ID | - ID | + VALOR | - VALOR | O

ELSE -> **DE_LO_CONTRARIO**{INSTRUCCION} | EPSILON

2. Una vez definida su gramática inicial, obtenga en secuencia los lexemas que pueden formar una sentencia y las derivaciones, por ejemplo:

add common, io	>>	add file, file
add graphic	>>	add file
begin	>>	begin
display 3	>>	display num
end	>>	end

VARIABLE ENTERO num1 = 1;

VARIABLE ENTERO num2 = 2;

DECLARACION-> VARIABLE TIPO ID = VALOR;

DECLARACION-> VARIABLE ENTERO ID = ENTERO;



```
SI? (num1 == num2){  
    IMPRIME('HOLA')  
}  
DE_LO_CONTRARIO {  
    num1=num2;  
}
```

```
IF-> SI?( ID OPERADOR_COMP ID){  
    INSTRUCCION  
}  
ELSE-> DE_LO_CONTARIO{  
    ASIGNACION  
}
```

3. Encuentre el equivalente apropiado generando una lista de alternativas y muestre la línea donde hubo un error.

VARIABLE num1 = 2; -> Falta especificar el tipo de dato. Se diría error en la línea correspondiente

variable DECIMAL x == 44.3 -> Las palabras reservadas deben ir en mayúscula, y las asignaciones se hacen con un solo =. Se marcaría múltiples errores en la línea correspondiente.

```
SI (fact1 > 3){  
    IMPRIME(fact1);  
}
```

-> El bloque de la sentencia necesita el símbolo ?. Se notifica de error en la línea correspondiente.

```
DE_LO_CONTRARIO{  
    id32 == id3324  
}
```

-> El error se produce por tratar de hacer una comparación de identificadores en un bloque de sentencia. Se notifica del error en la línea correspondiente.

