

Proyecto Fase: Parser

Programación de Sistemas

M.C. Orlando Muñoz Texzocotetla
Universidad Autónoma de la Ciudad de México

Actividades

Modificar la gramática dada en laboratorio para que reconozca:

- Los otros tipos de datos en las declaraciones: carácter, flotante.
carácter variable;
flotante variable;
entero variable;
- Más de una declaración de variables. Hasta el momento solo acepta una.
- Más de una variable por declaración.
carácter letra, clase, tipo,;
flotante temperatura, velocidad, ;
entero numero_dias, edad,;
- La instrucción de asignación:
variable <- valor;
donde: valor puede ser de tipo flotante, entero o carácter.
- En las declaraciones de variables debe poderse inicializar cada variable:
flotante temperatura <- 35.5, altura, volumen, saldo <- 0.0;
- Los caracteres en las expresiones

```
1 caracter letra;  
2 :::  
3 mientras( letra == 'a'){  
4     :::  
5 }
```

Observación: una “expresión” no solo se incluye en el ciclo “mientras” sino en otras estructuras como “para”, “haz-mientras”, “si-entonces”, etc.

- Los flotantes en las expresiones. Igual que el inciso anterior solo que para flotantes.
- Completar las expresiones para el resto de operadores (dados en clase), incluyendo paréntesis para poder agrupar subexpresiones. Hay que tomar en cuenta la precedencia entre los operadores usados y el tipo de asociatividad.
- El ciclo *para* con la siguiente sintaxis:

```
1 para(expresion; expresion; expresion){  
2     instrucciones  
3 }
```

- j. El ciclo *haz-mientras* con la siguiente sintaxis:

```
1 haz{
2     instrucciones
3 }mientras (expresion) ;
```

- k. Agregar la estructura selectiva si-entonces con las siguientes sintaxis:

```
7 si (expresion) {
8     instrucciones
9 }
11 si (expresion) {
12     instruccion
13 }
16 si (expresion)
17     instruccion
```

- l. Agregar la estructura selectiva si-entonces-otro incluyendo la parte de “otro” con las siguientes sintaxis:

```
7 otro (expresion) {
8     instrucciones
9 }
11 otro (expresion) {
12     instruccion
13 }
16 otro (expresion)
17     instruccion
```

Nota: Para la adición de si-entonces y si-entonces-otro debe usarse la solución investigada para evitar el problema “dangling if else” investigada anteriormente.

Es importante hacer la gramática de manera incremental, es decir, agregar un elemento menor y probar, y al probar que ya esté bien entonces continuar con el siguiente elemento.

Entregables:

1. Reporte en pdf (por class room)
2. Archivos flex y bison (por class room). Solo estos archivos, los generados al crear el ejecutable NO.
3. Reporte impreso en clase.

Estructura del reporte: portada, introducción, explicación de gramáticas libres de contexto, descripción de la gramática del lenguaje que estamos creando (en notación BNF), desarrollo y explicación de la gramática hecha en Bison, y conclusiones.