UNAHUR

UNIVERSIDAD NACIONAL DE HURLINGHAM

Parseo y Generación de Código
Trabajo Práctico

AÑO 2022

Prof: Mag. Ing. Pablo Pandolfo

Alumnos:

Joaquin Pettinari

Yamil Amarillo

Adrián Fazio

Índice

Compilador Gobstone reducido

Objetivo

Diseñar y desarrollar un traductor de una versión reducida del lenguaje de programación Gobstone, mediante los metacompiladores JFLEX y CUP.

Descripción

Gobstones es un lenguaje pensado y diseñado para la enseñanza introductoria a la programación. Que permite a los estudiantes aprender conceptos más avanzados utilizando un lenguaje formal.

La versión reducida consiste en la implementación de los movimientos en un tablero, mediante acciones (Mover, Sacar, Poner colores Rojos y/o Azules).

Ejemplo

```
Tablero(5,5);
START
Poner(Rojo);
END
```

Figura 1

```
Rojos: 0; Azules: 0 | Rojos: 0; Azules: 0 | Rojos: 0; Azules: 0 | Rojos: 0; Azules: 0 |
Rojos: 0; Azules: 0 | Rojos: 0; Azules: 0 | Rojos: 0; Azules: 0 | Rojos: 0; Azules: 0 |
Rojos: 0; Azules: 0 | Rojos: 0; Azules: 0 | Rojos: 0; Azules: 0 | Rojos: 0; Azules: 0 |
Rojos: 0; Azules: 0 | Rojos: 0; Azules: 0 | Rojos: 0; Azules: 0 | Rojos: 0; Azules: 0 |
[ Rojos: 1; Azules: 0 ] | Rojos: 0; Azules: 0 | Rojos: 0; Azules: 0 | Rojos: 0; Azules: 0 |
```

Figura 2

En la Figura 1 se visualiza el programa mínimo, escrito en el lenguaje Gobstone reducido
En la Figura 2 se visualiza el tablero inicial con la bola roja puesta en la posición (0,0), dado que no hubo movimiento, y el puntero se puede diferenciar por estar entre corchetes.

Precondición

En una matriz cuadrada.

Repositorio

https://github.com/Compilador-Gobstone

Gramática

G={N,T,program, P}

N={program, initial, listaDeclaraciones, declaración, instruccion, loop, cond, hayColor, tablero, mover, poner, sacar}
T={**PONER**, **MOVER**, **SACAR**, **START**, **END**, **LPAREN**, **RPAREN**, **COMMA**, **SCOLON**, **LBRACE**, **RBRACE**, **TABLERO**, **DIR**, **COLOR**, **NUMBER, REPEAT, NEG, HAYCOLOR, IF**}

P={

program -> initial **START** listaInstrucciones **END**
initial -> tablero
tablero -> **TABLERO LPAREN NUMBER COMMA NUMBER RPAREN SCOLON**
listaDeclaraciones -> listaDeclaraciones declaración
listaDeclaraciones -> declaración
declaración -> instruccion
declaración -> loop
declaración -> cond
instruccion -> poner
instruccion -> mover
instruccion -> sacar

loop -> **REPEAT LPAREN NUMBER RPAREN LBRACE** poner **RBRACE SCOLON**

loop -> **REPEAT LPAREN NUMBER RPAREN LBRACE** mover **RBRACE SCOLON**

loop -> **REPEAT LPAREN NUMBER RPAREN LBRACE** sacar **RBRACE SCOLON**

cond -> **IF LPAREN** hayColor **RPAREN LBRACE** poner **RBRACE SCOLON**

cond **-> IF LPAREN NEG** hayColor **RPAREN LBRACE** poner **RBRACE SCOLON**

hayColor -> **HAYCOLOR LPAREN COLOR RPAREN**

poner -> **PONER LPAREN COLOR RPAREN SCOLON**

mover -> **MOVER LPAREN DIR RPAREN SCOLON**

sacar -> **SACAR LPAREN COLOR RPAREN SCOLON**

}

Acciones disponibles

El lenguaje contará con las siguientes instrucciones de movimientos del punto:

Poner(color): Pone una bolita del color indicado.

Sacar(Color): Saca una bolita del color indicado.

Mover(dirección): Mueve el puntero en la dirección indicada.

Las direcciones pueden ser: Norte, Sur, Este o Oeste.

Estructura de control

START, END: Para delimitar las estructuras de control.

repetir(num): Repite el bloque la cantidad de veces del valor num

if(hayColor(color)): Ejecuta el bloque si hay color en donde está el puntero

Errores en ejecución

- Sacar color cuando no hay en ese puntero
- Moverse fuera del tablero
- Si no se registra inicio y fin
- Si hay instrucciones fuera de orden

Capturas de pantallas

Archivo .jflex

Gobstone.jflex

```
1   import java_cup.runtime.Symbol;
2   %%
3   %public
4   %class Scanner
5   %standalone
6   %cup
7   %%
8   START {return new Symbol(sym.START);}
9   END {return new Symbol(sym.END);}
10  \d { return new Symbol(sym.NUMBER, Integer.valueOf(yytext())); }
11  Tablero { return new Symbol(sym.TABLERO);}
12  hayColor { return new Symbol(sym.HAYCOLOR); }
13  repeat { return new Symbol(sym.REPEAT);}
14  if { return new Symbol(sym.IF); }
15  Poner {return new Symbol(sym.PONER);}
16  Mover {return new Symbol(sym.MOVER);}
17  Sacar {return new Symbol(sym.SACAR);}
18  Dir {return new Symbol(sym.DIR);}
19  Color {return new Symbol(sym.COLOR);}
20  Norte | Sur | Este | Oeste { return new Symbol(sym.DIR, yytext()); }
21  Rojo | Azul { return new Symbol(sym.COLOR, yytext()); }
22  "(" { return new Symbol(sym.LPAREN); }
23  ")" { return new Symbol(sym.RPAREN); }
24  , { return new Symbol(sym.COMMA); }
25  ; { return new Symbol(sym.SCOLON); }
26  "{" { return new Symbol(sym.LBRACE); }
27  "}" { return new Symbol(sym.RBRACE); }
28  "!" { return new Symbol(sym.NEG); }
29  . {System.out.print(yytext() + "[Caracter Inválido]");}
```

Snipped

Archivo .cup

Gobstone.java

```
4   action code
5   {:
6       static FunctionAdapter funcionAdapter = new FunctionAdapter();
7
8       public static void ejecutarFuncion (Integer num, String param){
9           for (int i = 0; i < num; i++) {
10              funcionAdapter.exec(param);
11              Tablero.imprimirTablero();
12          };
13      };
14
15      public static void ponerEnTablero(String color){
16          funcionAdapter.poner();
17          ejecutarFuncion(1, color);
18      };
19
20      public static void sacarDelTablero(String color){
21          funcionAdapter.sacar();
22          ejecutarFuncion(1, color);
23      };
24
25      public static void moverPuntero(String direccion){
26          funcionAdapter.mover();
27          ejecutarFuncion(1, direccion);
28      };
29  :}
30
```

Snipped

Gobstone.java

```
31  terminal PONER, MOVER, SACAR, START, END, LPAREN, RPAREN, COMMA, SCOLON, LBRACE, RBRACE, TABLERO, REPEAT, NEG, HAYCOLOR, IF;
32  terminal String DIR, COLOR;
33  terminal Integer NUMBER;
34  non terminal program, initial, listaDeclaraciones, declaracion, instruccion, loop, cond;
35  non terminal Tablero tablero;
36  non terminal Boolean hayColor;
37  non terminal String mover, poner, sacar;
```

Snipped

Gobstone.java

```
39  program ::= initial START listaDeclaraciones END
40      {:
41          Tablero.imprimirTablero();
42          System.out.println("[Sintaxis completada satisfactoriamente]");
43      :};
44
45  initial ::= tablero
46      {:
47          Tablero.imprimirTablero();
48      :};
49
50  tablero ::= TABLERO LPAREN NUMBER:n1 COMMA NUMBER:n2 RPAREN SCOLON
51      {:
52          RESULT = new Tablero(n1, n2);
53      :};
54
55  listaDeclaraciones ::= listaDeclaraciones declaracion
56                       | declaracion;
57
58  declaracion ::= instruccion | loop | cond;
```

Snipped

Gobstone.java

```
60  instruccion ::= poner:c
61      {:
62          ponerEnTablero(c);
63      :};
64
65  instruccion ::= mover:d
66      {:
67          moverPuntero(d);
68      :};
69
70  instruccion ::= sacar:c
71      {:
72          sacarDelTablero(c);
73      :};
74
75  loop ::= REPEAT LPAREN NUMBER:n RPAREN LBRACE poner:c RBRACE SCOLON
76      {:
77          funcionAdapter.poner();
78          ejecutarFuncion(n, c);
79      :};
80
81  loop ::= REPEAT LPAREN NUMBER:n RPAREN LBRACE mover:d RBRACE SCOLON
82      {:
83          funcionAdapter.mover();
84          ejecutarFuncion(n, d);
85      :};
86
87  loop ::= REPEAT LPAREN NUMBER:n RPAREN LBRACE sacar:c RBRACE SCOLON
88      {:
89          funcionAdapter.sacar();
90          ejecutarFuncion(n, c);
91      :};
```

Snipped

Gobstone.java

```
93   cond ::= IF LPAREN hayColor:b RPAREN LBRACE poner:c RBRACE SCOLON
94       {:
95           if(b){
96               ponerEnTablero(c);
97           }
98       :};
99
100  cond ::= IF LPAREN NEG hayColor:b RPAREN LBRACE poner:c RBRACE SCOLON
101      {:
102          if(!b){
103              ponerEnTablero(c);
104          }
105      :};
106
107  hayColor ::= HAYCOLOR LPAREN COLOR:c RPAREN
108      {:
109          RESULT = Tablero.hayColor(String.valueOf(c.charAt(0)));
110      :};
```

Snipped

Gobstone.java

```
112  mover ::= MOVER LPAREN DIR:d RPAREN SCOLON
113      {:
114          RESULT = String.valueOf(d.charAt(0));
115      :};
116
117  poner ::= PONER LPAREN COLOR:c RPAREN SCOLON
118      {:
119          RESULT = String.valueOf(c.charAt(0));
120      :};
121
122  sacar ::= SACAR LPAREN COLOR:c RPAREN SCOLON
123      {:
124          RESULT = String.valueOf(c.charAt(0));
125      :};
```

Snipped

Programas de ejemplo

Ejemplo #1

Programa:

```
Tablero(5,5);
START
Poner(Rojo);
Mover(Norte);
Poner(Azul);
END
```

Salida:

```
Rojos: 0; Azules: 0 | Rojos: 0; Azules: 0 | Rojos: 0; Azules: 0 | Rojos: 0; Azules: 0 |
Rojos: 0; Azules: 0 | Rojos: 0; Azules: 0 | Rojos: 0; Azules: 0 | Rojos: 0; Azules: 0 |
Rojos: 0; Azules: 0 | Rojos: 0; Azules: 0 | Rojos: 0; Azules: 0 | Rojos: 0; Azules: 0 |
Rojos: 0; Azules: 0 | Rojos: 0; Azules: 0 | Rojos: 0; Azules: 0 | Rojos: 0; Azules: 0 |
[ Rojos: 1; Azules: 0 ] | Rojos: 0; Azules: 0 | Rojos: 0; Azules: 0 | Rojos: 0; Azules: 0 |


Rojos: 0; Azules: 0 | Rojos: 0; Azules: 0 | Rojos: 0; Azules: 0 | Rojos: 0; Azules: 0 |
Rojos: 0; Azules: 0 | Rojos: 0; Azules: 0 | Rojos: 0; Azules: 0 | Rojos: 0; Azules: 0 |
Rojos: 0; Azules: 0 | Rojos: 0; Azules: 0 | Rojos: 0; Azules: 0 | Rojos: 0; Azules: 0 |
[ Rojos: 0; Azules: 0 ] | Rojos: 0; Azules: 0 | Rojos: 0; Azules: 0 | Rojos: 0; Azules: 0 |
Rojos: 1; Azules: 0 | Rojos: 0; Azules: 0 | Rojos: 0; Azules: 0 | Rojos: 0; Azules: 0 |


Rojos: 0; Azules: 0 | Rojos: 0; Azules: 0 | Rojos: 0; Azules: 0 | Rojos: 0; Azules: 0 |
Rojos: 0; Azules: 0 | Rojos: 0; Azules: 0 | Rojos: 0; Azules: 0 | Rojos: 0; Azules: 0 |
Rojos: 0; Azules: 0 | Rojos: 0; Azules: 0 | Rojos: 0; Azules: 0 | Rojos: 0; Azules: 0 |
[ Rojos: 0; Azules: 1 ] | Rojos: 0; Azules: 0 | Rojos: 0; Azules: 0 | Rojos: 0; Azules: 0 |
Rojos: 1; Azules: 0 | Rojos: 0; Azules: 0 | Rojos: 0; Azules: 0 | Rojos: 0; Azules: 0 |
```

Ejemplo #2

Programa:

```
Tablero(5,5);
START
repeat(3){Poner(Rojo);};
if(hayColor(Rojo)){Poner(Rojo);};
END
```

Salida:

```
Rojos: 0; Azules: 0 | Rojos: 0; Azules: 0 | Rojos: 0; Azules: 0 | Rojos: 0; Azules: 0 |
Rojos: 0; Azules: 0 | Rojos: 0; Azules: 0 | Rojos: 0; Azules: 0 | Rojos: 0; Azules: 0 |
Rojos: 0; Azules: 0 | Rojos: 0; Azules: 0 | Rojos: 0; Azules: 0 | Rojos: 0; Azules: 0 |
Rojos: 0; Azules: 0 | Rojos: 0; Azules: 0 | Rojos: 0; Azules: 0 | Rojos: 0; Azules: 0 |
[ Rojos: 4; Azules: 0 ] | Rojos: 0; Azules: 0 | Rojos: 0; Azules: 0 | Rojos: 0; Azules: 0 |
```