

Compiladores

Martínez Coronel Brayan Yosafat

For. 3CM7

1/1/2021

Práctica 6:
Robot



Robot con for

Introducción

En esta práctica se desea agregar el FOR, otra estructura muy potente para lo que hemos desarrollado hasta ahora. Ahora debemos tener en cuenta varias cosas, si bien la idea es que el ciclo tenga forma como el lenguaje C, sin embargo, eso no tendría mucho sentido, ¿por qué?, porque nuestras variables no son numéricas, sólo contienen cadenas de caracteres, por lo que hacer un FOR con la forma del lenguaje C no es tan buena idea, pues la variable no sería numérica, o bien tendríamos que hacer toda la estructura para soportarlo.

Si bien sí se utilizan enteros, ahora tendremos producciones para generar enteros, pero es solamente en la multiplicación, de hecho, para hacer notar la diferencia entre la comparación de caracteres se usó el token EQUALS, pero para enteros se va a usar `==`, esto viene de gran utilidad porque los enteros ahora sí se pueden producir.

Sin embargo, la forma en la que se planeó fue hacer un FOR al estilo de Python, porque es mucho más potente para lo que queremos hacer, como ya detectamos los enteros es mucho más sencillo hacer un for que prácticamente va a hacer lo mismo que el de lenguaje C, porque la única cosa que se puede hacer con el iterador sería una multiplicación. Lo cual costaría muchas más producciones que no funcionarían para el dibujo de la secuencia del robot, y que para hacer que funcionaran más tendría que hacerse muchas más funciones para las secuencias y muchas más estructuras para el control del flujo, por ejemplo, una función que con un entero regrese el enésimo carácter de la secuencia, o una función que regrese una subsecuencia con dos enteros. Sin embargo, entonces tendríamos que crear muchos más IF de forma interna, y hasta bloque try catch.

Los objetivos de esta práctica son los siguientes:

- Producción del FOR y de INT
- Código de FORFUN
- Token del FOR
- Función de lengthpush
- Comparación de enteros

Desarrollo

La forma en la que procedimos para hacer el for es mediante la forma de rango de Python, y con ayuda de las producciones que ya teníamos sobre el bloque de código, el mapa del for en la memoria es muy directo, porque lo que tenemos que guardar es dónde empieza el rango y guardar dos enteros en la pila. Luego dónde empieza el bloque y dónde termina el for en general.

```
| for '(' range endlog ')' '{' block '}' end
{
  $$ = $1;
  int forIndex = ((Simbolo) $1.obj).getValor();
  maq.replace(forIndex + 1, $3.obj);
  maq.replace(forIndex + 2, $7.obj);
  maq.replace(forIndex + 3, $9.obj);
}

int      : INT
| '[' expr ']'      { $$ = new ParserVal(new Simbolo(maq.code("constpush"))); maq.code($1.obj); }
{ $$ = $2; maq.code("lengthpush"); }
```

Ahora veamos la producción range, que utiliza el carácter de : que requiere dos enteros, en este caso int es otra producción y también la presentamos aquí abajo.

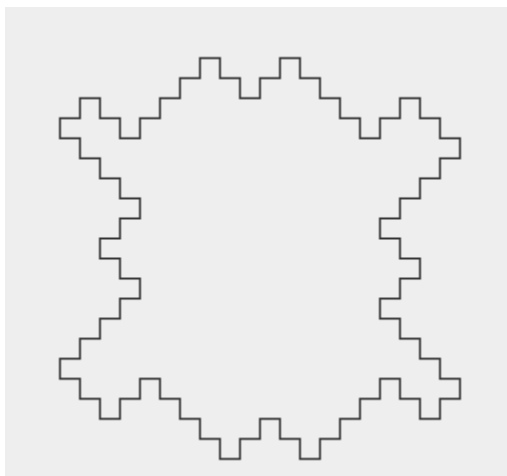
```
range      : int ':' int      { $$ = $1; }
```

Ahora, lo que hace la producción int es, como su nombre lo indica, tener un entero en la pila. Así que, para eso agregamos otro operador de utilidad. Como se unificó que las expresiones siempre generen una secuencia, entonces cualquier expresión tiene un tamaño, por ejemplo, [EEEE], es de tamaño 4, por lo que la sintaxis de un for sería de la siguiente forma:

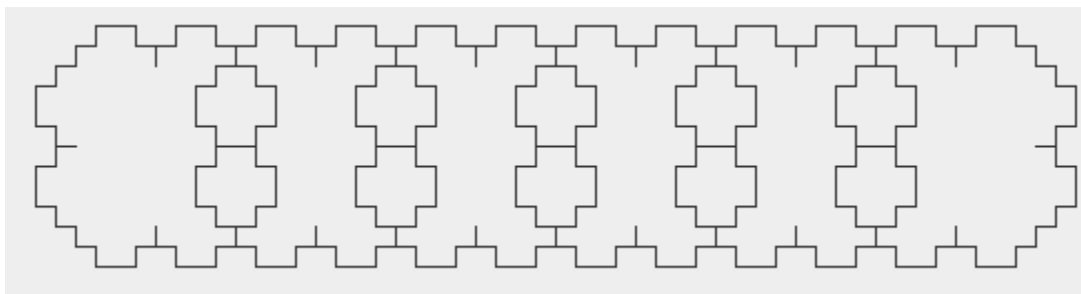
for (1 : [variable]) { variable = variable + - (variable) }

Esto hace un patrón complejo para dibujar, y la suma de la negación, de hecho, se agregaron muchas funciones útiles como la rotación, la negación en un solo eje, y la rotación negativa. Para ver el potencial de estas funciones presentamos el lienzo que se creó con este código, cambiando los signos y el eje durante 4 ejecuciones:

```
var = E N E N ;
for ( 1 : [ var ] ) {
  var = var + -y ( var ) ;
}
- ( var ) ;
```



Al igual, se hizo una captura de un código similar, pero con el uso de la rotación de una secuencia:



Conclusiones

Las rutas dibujadas en el lienzo gracias al for, son bastante interesantes, es como si estuviéramos dibujando patrones de mosaicos en el lienzo. Es bastante peculiar cómo se dibujaron los patrones, me asombra el hecho de que no tomara más de 6 líneas y de repetir varias veces la ejecución del ciclo. Incluso con todos estos incrementos, queda un incremento que hace que tenga mucha más funcionalidad. Faltan las funciones y procedimientos, algo que es complicado de implementar, pero que aumenta demasiado el nivel de funcionalidad. Ahora entiendo la importancia de añadir funciones de utilidad a un lenguaje, como length con Java en los Strings, o que sean fáciles de escribir como el foreach y que sean entendibles como los argumentos variables en una función.