Universidad Simón Bolívar

Departamento de Computación y Tecnología de la Información

CI3725 Traductores e Interpretadores

Enero – Marzo 2020

Profesor: Blai Bonet

Preparador: German Robayo

Informe Proyecto Willy

Integrantes:

Marval, Luis 12-10620

Suarez, Fabio 12-10578

Proyecto Traductores e Interpretadores: Willy*

El proyecto consiste en implementar el ambiente de programación Willy*.
Willy* está inspirado en el ambiente/lenguaje Willy definido por el Prof. Ernesto Hernandez-Novich.

Acerca del Proyecto

Proyecto del Curso CI-3725 para el trimestre Enero-Marzo 2020

Función: Interprete de Willy*

Descripción:

El ambiente consiste de un robot llamado Willy que se desenvuelve dentro de una cuadrícula finita en la cual existen paredes. Willy tiene sensores, brazos actuadores, una cesta para guardar objetos, y medios para desplazarse. El robot puede navegar libremente en la cuadrícula, puede recoger y dejar objetos en las celdas de la cuadrícula.
Willy es controlado por un programa que contiene una o más definiciones del mundo, la definición de los procedimientos y el programa principal. Willy es controlado por un programa que contiene una o más definiciones del mundo, y la definición de los procedimientos y el programa principal.

Implementado Con

- Python3
- Ply

Implementación

Fase 1

Esta Primera etapa consiste en un lexer, que con la ayuda de la librería PLY Se encarga de leer el string y devolver los tokens asociados o error en caso de conseguir uno.

Para la implementación primero se definieron los tokens a ser reconocidos mediante el uso de expresiones regulares, funciones y una lista de palabras reservadas.

Fase 2

Esta Segunda etapa consiste en un parser, con el AST y su tabla de símbolos, con la ayuda de la librería PLY.

Partiendo de donde termina la fase 1 (lexer) se procede en dicha segunda entrega crear las producciones que permiten satisfacer la sintaxis del lenguaje, luego de eso se procede a generar la tabla de símbolos que será importante para el análisis semántico y posteriormente la interpretación del lenguaje.

La estructura implementada para la parte de la tabla de símbolos fue un diccionario para los mundos, otro para las tareas y por último una pila que donde cada nivel es un marco nuevo de inmersión dentro del programa y a su vez cada marco tiene su diccionario para la declaración de variables en ese nivel.

Fase 3

Por último esta última fase consistió en corregir detalles de entregas anteriores e implementar el intérprete del lenguaje de modo que al suministrar un programa correcto de willy* se devuelve la corrida del mismo.

En nuestro intérprete no aceptamos defines como instrucciones primitivas como dice en el enunciado del proyecto, adicionalmente para que la ejecución de un programa se lleve a cabo, incluso las tareas que no se suministran para la corrida del intérprete deben ser correctas hasta su nivel de ejecución, de lo contrario se presentará el error.

Primero se hace la verificación léxica del programa, una vez verificada esta se hace una verificación gramática del programa, luego se revisa que este programa corra correctamente y por último se imprime por consola la corrida correspondiente a la instrucción que se indica, dependiendo del flag que se pase se tiene distintas formas de salida, consideramos necesario que se pase alguno de estos flags.

En caso de ser el flag automático se puede pasar un parámetro numérico que corresponde al intervalo de segundos que transcurre entre cada paso de la ejecución, si se pasa segundos en tipo de ejecución manual se considera un error, en caso de no pasar segundos en tipo automático, se imprime el resultado final de la ejecución.

Comenzando

Sigue los pasos

Prerequisitos

Python 3 debe estar instalado para el funcionamiento del interprete.
Ply debe estar provisto en la carpeta de ejecución para el funcionamiento de la herramienta.
Hay que otorgar permisos de ejecución con el comando chmod 777 a los archivos willy e interfaz donde se encuentra el script que permite analizar el programa

Instalación

Nada más es necesario para su ejecución

Como se Usa

Escriba su programa siguiendo las reglas del lenguaje willy*.
Ejecute el siguiente comando dentro de la carpeta donde tenga la librería de ply y se procederá a ejecutar su programa.

```
./willy nombreArchivo nombreTarea flag tiempo
```

En caso de no proveer los parámetros, se pedirá por consola.
Si prefiere ejecutar los comandos por separado sería de la siguiente forma.

```
python3 main.py nombreArchivo nombreTarea flag tiempo
```

Interfaz

Como adición extra al proyecto, se pudo implementar una interfaz gráfica con PyQt5. Para correr la misma no es necesario tener instalado el paquete
Se agregó otra forma de correr el programa en esta se muestra la corrida por consola y luego se abre la interfaz gráfica. Para correr la versión con interfaz es de la siguiente forma.

```
./interfaz nombreArchivo nombreTarea flag tiempo
```

Al igual que el anterior para ejecutar los comandos por separado seria.

```
python3 willy.py nombreArchivo nombreTarea flag tiempo
```

Dentro cada casilla de la interfaz si aparece algo de la forma corneta(5) significa que hay 5 objetos de tipo corneta en dicha posición.

Para poder visualizar mejor el contenido de las casillas, puede ampliar las filas y columnas acercando el mouse a los bordes de la tabla. (cualquier casilla que se vea con ... es necesario ampliarla para ver su contenido)

Cabe destacar que el último flag, el del tiempo es opcional si se va a ejecutar con el flag -m o --manual y por otro lado es opcional para la corrida del flag -a o --auto, dicho parámetro debe ser un entero y va sin corchetes.

Para el caso de la interfaz, si se ejecuta con el flag manual se avanza mediante el botón Next y para el caso de ser auto el flag con n segundos se avanza automáticamente cada n segundos

Nota: En el Archivo que se genera parselog.txt se puede ver el parseo del programa producción a producción

Impresión

```
El formato de impresión utilizado para el mundo es el de una matriz con el
siguiente formato para cada posicion:
        - Si la posición contiene "wall" es que se encuentra situada una pared
en esa casilla.
        - Si la posición contiene "0" es que se encuentra vacia.
        - Si la posición contiene "willy" es que se encuentra situado solamente
willy en esa casilla.
        - Si la posición contiene una X es que se encuentra situado al menos un
objeto en dicha posición.
        - Si la posición contiene willy-X es que se encuentra el robot willy y
al menos un objeto.
```

En general para cada vez que se imprime una instancia de la matriz del mundo, en su parte superior izquierda se imprime la instrucción que se ejecutó. Luego de eso se procede a imprimir en la parte inferior de la matriz la cantidad de objetos que se encuentran en dicha posición y su indexación.

Los objetos que se encuentran bajo la misma indexación están impresos de manera continua y luego se imprime un salto de línea, para colocar la posición del siguiente grupo de objetos.

```
##Ejemplo ilustrativo:

        move:
                wall    wall    wall    wall    wall    wall
                wall    0       0       0       0       0
                wall    0       0       0       0       0
                wall    0       0       0       0       0
                wall    X       X       0       willy-X 0
                wall    0       0       0       0       X

        posicion  row: 2, col: 2
        X =  {'id': 'corneta', 'color': 'blue', 'type': 'object', 'quantity': 1}

        posicion  row: 2, col: 3
        X =  {'id': 'pelota', 'color': 'red', 'type': 'object', 'quantity': 1}

        posicion  row: 2, col: 5
        X =  {'id': 'corneta', 'color': 'red', 'type': 'object', 'quantity': 1}
        X =  {'id': 'celular', 'color': 'red', 'type': 'object', 'quantity': 2}

        posicion  row: 1, col: 6
        X =  {'id': 'pelota', 'color': 'red', 'type': 'object', 'quantity': 5}



cabe destacar que este formato es tanto para el modo manual como el auto
```