

TRABAJO PRÁCTICO 1

Técnicas de Compilación

Fecha: 11/05/2022

Integrantes: Szulman Pablo, Tello Eric.

Temas: Reglas de compilación, expresiones regulares, reglas semánticas y sintácticas.

Profesor: Maximiliano Andrés Eschoyez.

Herramientas: ANTLR – Visual Studio Code – Java.

Consigna: Dado un archivo de entrada en lenguaje C, se debe generar como salida el Árbol Sintáctico (ANTLR) correcto. Para lograr esto se debe construir un parser que tenga como mínimo la implementación de los siguientes puntos:

-Reconocimiento de un bloque de código, que puede estar en cualquier parte del código fuente, controlando balance de llaves.

-Verificación de:

- Declaraciones y asignaciones,
- Operaciones aritmeticológicas,
- Declaración/llamada a función.

-Verificación de las estructuras de control if, for y while. Ante el primer error léxico o sintáctico el programa deberá terminar.

DESARROLLO Y CONCLUSIONES

Para abordar el trabajo hemos usado los recursos de expresiones regulares de las primeras clases (detectar parámetros como fechas y horas, o distinguir letras a números) conjuntamente con lo visto en “balances” (llaves, corchetes y paréntesis) para crear nuestras primeras estructuras de expresiones comunes en el lenguaje como por ejemplo:

```
LA : '{' ;  
LC : '}' ;  
PYC : ';' ;  
INT : 'int' ;
```

También ha sido de gran ayuda la ejecución de expresiones regulares visualizando el árbol sintáctico (en clases visto con balance de llaves y distintas entradas) para poder ir ejecutando el árbol y así armando nuestras expresiones regulares.

Una vez definidos los primeros elementos aislados del programa como mostramos anteriormente (signos, comparadores, llaves, tipos, entre otros) comenzamos con la lógica del árbol.

Por los conceptos adquiridos sabemos que la siguiente conformación es correcta:

```
s : programa;  
  
//programa tiene instrucciones  
programa: instrucciones ;  
  
//instrucciones tiene una instruccion, seguida de más instrucciones  
instrucciones : instruccion instrucciones  
               |  
               ;  
  
//bloque es {instrucciones}  
bloque : LA instrucciones LC  
        ;
```

A partir de la estructura básica de un programa podemos desglosar cada sentencia y captarla según armemos nuestras expresiones.

Así fuimos armando las funcionalidades básicas (declaraciones, asignaciones, operaciones, etc.) para posteriormente abarcar los ciclos y funciones (estructuras más complejas).

La primera dificultad que tuvimos se nos presentó a la hora de captar las operaciones lógicas (disyunción, conjunción y la comparación), desarrollamos primero el ciclo if y while y últimamente el for, para terminar con el desarrollo de llamada a funciones y estructuras de las mismas.

El trabajo nos ha servido mucho para comprender el trabajo que está atrás de las ejecuciones a nuestros códigos que nunca hasta el momento habíamos visto, ambos trabajamos hace bastante en desarrollo y se

automatizan tanto ciertos procesos que en la rutina simplemente se programa, se dá click en compilar y “levanta o no levanta”. Consideramos que como ingenieros es fundamental entender el cómo y poder desglosarlo para captar un lenguaje. Entender las estructuras, y reglas sintácticas y semánticas son una gran herramienta, tanto para captar funcionamientos o mecánicas de lenguajes como para el uso cotidiano, como por ejemplo validar un mail en un formulario web (con expresiones regulares).