# Diseño de Compiladores

## Extensión de Trabajo Práctico de Cursada

## Operador “++”


Marcelo Rodríguez

mrodriguez@alumnos.exa.unicen.edu.ar

**Profesor :** Mg. Marcela Ridao

**Resumen**

En este documento se describe someramente la incorporación de una operación de suma, el operador ++, como sintaxis para el lenguaje desarrollado en calidad de cursada.

## Introducción

El Lenguaje desarrollado, utiliza tipos de datos uint, long, impresiones de cadenas, identificadores con name mangling, conversiones explícitas tolong, y operaciones aritméticas comunes sin paréntesis y chequeos en tiempo de ejecución: división por cero y resultados negativos en restas son signo.
Como extensión a este lenguaje, se ha solicitado añadir el operador ++ ( sumar uno y asignar ) sobre identificadores de tipo “long”.

**Nota:**

- Repositorio del proyecto **actual** ,desarrollado en lenguaje Java, aquí.
    - En https://github.com/buhtigexa/Lexer/tree/master/src/parser, el script **run.sh** se utiliza para generar el parser y los archivos autómatas de salida.
- El informe, y trabajo original se encuentra aquí

## Descripción de los cambios.

Añadir estados en el autómata finito correspondiente al analizador léxico ; como también el token correspondiente a los dos signos ‘+’ ( T_PLUS_PLUS en el parser) . Este último es entregado sin otros atributos adicionales.

En la etapa de análisis sintáctico, se añade a la gramática una nueva regla para reconocer tal construcción: **T_IDENTIFIER T_PLUS_PLUS** ; en la regla factor. Así, el operador tiene efecto sobre identificadores ; para determinar que los identificadores de tipo **long** sean exclusivos para tal operación, se añaden las acciones semánticas en dicha regla.

En la etapa de generación de código fué necesario generar dos tercetos: suma y asignación sobre el identificador.

Como última modificación, en la tabla de símbolos se agrega la constante “1”, tipo long. Esta última, puede ser reutilizada a fin de economizar el espacio de dicha tabla.

```
<exp>       :<exp> ‘+’ <term>
            | <term>
            ;

<term>      : <term> ‘*’ <factor>
            |  <factor>
            ;

<factor>    :   <variable>
            |   <constante>
```

```
            ;

<variable> :    T_VARIABLE
            |   T_VARIABLE T_PLUS_PLUS    ; Nueva regla !.
            ;
```

## Ejecución de la aplicación:

**$> java -jar compilador.jar** *[path_programa_a_compilar] [directorio_archivos_asm] [directorio_de_salidas]*

Sí los directorios no están creados, se crean . A excepción de los archivos assembler, el resto se sobreescriben como resultado de cada nueva compilación.

**Directorio de salidas:** incluye archivo de errores, warnings y mensajes del compilador ( código intermedio generado, tabla de símbolos y mensajes ).

**Directorio archivos asm:** contiene los archivos en lenguaje ensamblador generados.

**Nota:** los archivos de prueba en https://github.com/buhtigexa/Lexer/tree/master/test_files , son los originales correspondientes al proyecto de cursada, con las modificaciones para incorporar el operador en cuestión; el archivo *errors.txt* , contiene errores intencionales para apreciar la salida del compilador eventualmente.