

Reporte

Matías Greco, Javier Reyes

2 de Octubre, 2018

Introducción

El presente reporte explica el trabajo realizado para el desarrollo de un analizador léxico y sintáctico para el lenguaje de programación C++.

El lenguaje C++ corresponde a un subconjunto del lenguaje C, con la adición de algunas características de C++, como la posibilidad de definir una función con paso por valor o paso por referencia.

El analizador léxico y sintáctico fue desarrollado en la herramienta ANTLR4 (ANother Tool for Language Recognition) con lenguaje de salida C++.

Características del lenguaje

El lenguaje C++ incluye las siguientes características, propias del standard C89

- Funciones.
- Declaraciones.
- Asignaciones.
- Expresiones lógicas y de operaciones.
- If.
- Switch (case y default).
- While.
- For.
- Do While.
- Struct

Algunas características interesantes:

- Comma expression

Características no incluidas

- **Macros:**

No se incluyó debido a que complica todo el proceso. Requeriría una precompilación y la capacidad de incluir otros archivos.

- **Punteros:**

No se incluyó debido a que complica la gramática, por la aparición de una infinidad de tipos distintos (del estilo `int**`). También requiere una administración de memoria que se escapa un poco de nuestro objetivo.

- **Typedef:**

No se incluyó debido a que complica la gramática. Eso permitiría utilizar una expresión de tipo `VAR` como una definición de tipo. Al no incluirlo, el lenguaje no pierde capacidades, ya que una estructura personalizada `S` tiene tipo `struct S`.

Definiciones léxicas

La definición de los componentes léxicos del lenguaje C+-es similar al lenguaje C, y se define de la siguiente forma:

- **Identificadores:** Puede componerse de letras, números y guiones bajos, pero no pueden empezar con un número.
- **Valores constantes:** Pueden ser números enteros con o sin signo (expresables en base 8, 10 y 16), números de punto flotante, caracteres y strings.
- **Operadores aritméticos:** `+` para suma, `-` para resta, `*` para multiplicación `/` para división y `%` para el resto de la división.
- **Operadores de comparación:** `==`, `!=`, `<=`, `>=`, `<`, `>`.
- **Operadores unarios:** `++`, `--`, `+`, `-`, `!`, `~`.
- **Operadores de shift:** `<<`, `>>`.
- **Operadores bitwise:** `&`, `^`, `|`.
- **Operadores lógicos:** `&&`, `||`.
- **Operador ternario:** `?` `:`
- **Operador coma:** `exp1,exp2` ejecuta `a`, luego `b` y retorna `b`.
- **Operadores varios:** `sizeof` retorna el tamaño en bytes de una expresión o tipo; llamadas a métodos (`f(exp1,exp2)`); acceso a miembros (`estructura.miembro`); y acceso a elementos de un array (`arr[i]`).

Conclusiones