

Gramática

Gramática léxica

La gramática define patrones para los tokens o elementos léxicos que se pueden encontrar en el código fuente, como operadores, palabras clave, identificadores, literales de cadena, números, caracteres, símbolos y espacios en blanco.

Por ejemplo, el patrón para identificar la palabra clave "if" es simplemente la cadena "if", mientras que el patrón para identificar un número entero es una o más ocurrencias de cualquier dígito del 0 al 9.

Cada patrón se asigna a un token, que se utiliza para identificar el tipo de elemento léxico encontrado y proporcionar información al compilador o intérprete sobre cómo procesar ese elemento. Por ejemplo, la palabra clave "if" se puede utilizar para indicar el comienzo de una declaración condicional en el programa.

La gramática también incluye patrones para ignorar comentarios y espacios en blanco, así como para detectar errores léxicos, como caracteres no válidos o tokens desconocidos.

Gramática sintáctica

Esta es una gramática en formato BNF (Backus-Naur Form) que describe la sintaxis de un lenguaje de programación. Cada regla de la gramática se define utilizando símbolos no terminales y terminales. Los símbolos no terminales se definen en términos de otros símbolos no terminales o terminales, mientras que los símbolos terminales son símbolos que aparecen literalmente en el código fuente del lenguaje de programación.

En esta gramática, la regla "program" define la estructura básica de un programa, que puede constar de una lista de declaraciones y expresiones o estar vacío. La regla "statements" define una lista de declaraciones y expresiones, mientras que la regla "statement" define un tipo particular de declaración o expresión, como una asignación, una construcción integrada, una declaración condicional, una declaración iterativa, etc.

Las reglas "assignment", "built_in", "conditional", "declaration", "iterative" y "unary" definen cada uno un tipo particular de declaración o expresión, con reglas más específicas para cada tipo de declaración o expresión.

Las reglas "expression", "arithmetic", "logical", "relational" y "value" definen diferentes tipos de expresiones que pueden aparecer en el código fuente, como expresiones aritméticas, lógicas, relaciones o valores simples como enteros, cadenas o booleanos.

Las reglas "if", "else_if", "code_block", "switch", "cases" y "default" definen construcciones condicionales y de control de flujo, como un if-else, un switch-case, etc.

La regla "declaration" define una declaración de variable con un tipo de dato explícito, mientras que las reglas "data_type" definen los diferentes tipos de datos que se pueden utilizar en el código fuente, como booleanos, caracteres, números enteros, números de coma flotante o cadenas de caracteres.

Finalmente, las reglas "do_while", "for" y "while" definen diferentes tipos de construcciones de control de flujo iterativas.