



INSITUTO POLITECNICO NACIONAL

ANALIZADOR SINTACTICO

ESCUELA SUPERIOR DE CÓMPUTO

INGENIERÍA EN SISTEMAS COMPUTACIONALES

MATERIA: COMPILADORES

URIBE HERNÁNDEZ CARLOS DANIEL

GROUP: 5CV1

2022

Introducción

Un analizador es un componente crucial en el proceso de compilación de un lenguaje de programación. Se encarga de analizar la estructura sintáctica del código fuente, asegurándose de que sigue las reglas gramaticales y sintácticas correctas del lenguaje. El analizador convierte el código fuente a un formato más comprensible, como un árbol de sintaxis abstracta (AST) o un árbol de análisis. Esto permite que el compilador genere código de máquina eficiente o interprete el programa. Hay varios tipos de analizadores, como los analizadores descendentes recursivos, los analizadores LL y los analizadores LR, cada uno con sus propias ventajas y desventajas. Comprender el papel y la función de los analizadores en el diseño del compilador es esencial para crear compiladores eficientes y efectivos.

3.1.1 LR PARSER

Un analizador LR, también conocido como analizador de derivación de izquierda a derecha, más a la derecha, es un tipo de analizador utilizado en informática para analizar gramáticas libres de contexto. Los analizadores LR están diseñados para trabajar con gramáticas formales, que proporcionan un conjunto de reglas para generar cadenas de símbolos. Un analizador LR se llama "LR" porque lee la entrada de izquierda a derecha y construye una derivación más a la derecha de la cadena de entrada. El "LR" en el nombre se refiere al orden en que el analizador lee la entrada y la dirección de la derivación más a la derecha.

Los analizadores LR se dividen en dos tipos: LR(0) y LR(1). La diferencia entre los dos es que los analizadores LR(0) utilizan un conjunto de elementos LR(0) para analizar la entrada, mientras que los analizadores LR(1) utilizan un conjunto de elementos LR(1). Los analizadores LR(1) se consideran más potentes que los analizadores LR(0), ya que pueden manejar gramáticas más complejas. Los analizadores LR utilizan una tabla de análisis, también conocida como tabla de transición, para analizar la entrada. La tabla de análisis contiene un conjunto de estados y, para cada estado, especifica el siguiente estado al que se realizará la transición en función del símbolo de entrada actual. La tabla de análisis se genera a partir de la gramática mediante un proceso llamado construcción de tablas.

Una de las ventajas de los analizadores sintácticos LR es que pueden manejar una amplia gama de gramáticas, incluidas aquellas que no son LR(k) para cualquier valor finito de k. Además, los analizadores LR pueden detectar errores en la cadena de entrada y pueden proporcionar mensajes de error detallados al usuario.

Sin embargo, los analizadores LR también tienen algunas desventajas. Una de las principales desventajas es que pueden ser bastante complejos de implementar, ya que requieren una gran cantidad de memoria y potencia de procesamiento. Además, los analizadores LR pueden ser lentos para analizar cadenas de entrada grandes, ya que necesitan examinar todas las posibles derivaciones del extremo derecho de la entrada. En general, los analizadores LR son una herramienta poderosa para analizar gramáticas libres de contexto y se pueden usar en una amplia gama de aplicaciones, incluidos compiladores, intérpretes y otros tipos de herramientas de procesamiento de lenguaje.

3.1.1 LL PARSER

Un analizador LL, también conocido como analizador de derivación de izquierda a derecha, es un tipo de analizador utilizado en informática para analizar gramáticas libres de contexto. Los analizadores LL están diseñados para trabajar con gramáticas formales, que proporcionan un conjunto de reglas para generar cadenas de símbolos.

Un analizador LL se llama "LL" porque lee la entrada de izquierda a derecha y construye una derivación más a la izquierda de la cadena de entrada. El "LL" en el nombre se refiere al orden en que el analizador lee la entrada y la dirección de la derivación más a la izquierda. Los analizadores LL se dividen en dos tipos: LL(1) y LL(k) donde k es un número natural. La diferencia entre los dos es que los analizadores LL(1) utilizan un conjunto de elementos LL(1) para analizar la entrada, mientras que los analizadores LL(k) utilizan un conjunto de elementos LL(k). Los analizadores LL(k) se consideran más potentes que los analizadores LL(1), ya que pueden manejar gramáticas más complejas.

Los analizadores LL utilizan una tabla de análisis, también conocida como tabla de análisis predictivo, para analizar la entrada. La tabla de análisis contiene un conjunto de estados y, para cada estado, especifica el siguiente estado al que se realizará la transición según el símbolo de entrada actual y los siguientes k símbolos. La tabla de análisis se genera a partir de la gramática mediante un proceso llamado construcción de tablas. Una de las ventajas de los analizadores LL es que son fáciles de entender, construir e implementar. Además, pueden detectar errores en la cadena de entrada y pueden proporcionar mensajes de error detallados al usuario. Los analizadores LL también tienen una huella de memoria baja y son eficientes en términos de complejidad de tiempo y espacio.

Sin embargo, los analizadores LL también tienen algunas desventajas. Una de las principales desventajas es que solo pueden manejar un subconjunto de gramáticas libres de contexto, específicamente aquellas que son LL(k) para algún valor finito de k. Además, los analizadores LL pueden ser lentos para analizar cadenas de entrada grandes, ya que necesitan examinar todas las posibles derivaciones del extremo izquierdo de la entrada. En general, los analizadores LL son una herramienta poderosa para analizar un subconjunto específico de gramáticas libres de contexto y se pueden usar en una amplia gama de aplicaciones, incluidos compiladores, intérpretes y otros tipos de herramientas de procesamiento de lenguaje donde la gramática es lo suficientemente simple como para ser LL. (k) analizable.

Conclusión

En conclusión, los analizadores LL y LR son tipos de analizadores utilizados en informática para analizar gramáticas libres de contexto. Ambos tipos de analizadores leen la entrada de izquierda a derecha y construyen una derivación de la cadena de entrada. Sin embargo, la principal diferencia entre los dos es la dirección de la derivación. Los analizadores LL construyen una derivación más a la izquierda, mientras que los analizadores LR construyen una derivación más a la derecha.

Los analizadores LL son fáciles de entender, construir e implementar y son eficientes en términos de complejidad de tiempo y espacio. Son adecuados para un subconjunto específico de gramáticas libres de contexto, específicamente aquellas que son $LL(k)$ para algún valor finito de k . Por otro lado, los analizadores LR pueden manejar una amplia gama de gramáticas, incluidas aquellas que no son $LR(k)$ para cualquier valor finito de k , pero que son más complejas de implementar y requieren más memoria y potencia de procesamiento.

Ambos tipos de analizadores tienen ventajas y desventajas, y la elección de cuál usar depende de los requisitos específicos de la aplicación y la naturaleza de la gramática que se analiza. Los analizadores LL y LR se utilizan ampliamente en una variedad de aplicaciones, incluidos compiladores, intérpretes y otros tipos de herramientas de procesamiento de lenguaje.