Instituto Tecnológico de Costa Rica.

Unidad de Computación

Josué Esquivel V.

Katherine Salazar P.

Análisis Contextual Compilador Mini\_C#

Sede San Carlos.

13-noviembre-2017.

Introducción

Un compilador es una herramienta fundamental en el área de programación que, a pesar de su importancia y uso frecuente, los estudiantes no conocen a profundidad los detalles del proceso que realiza. El compilador es un programa que se encarga de traducir un código escrito en un lenguaje de programación en alto nivel a un lenguaje de bajo nivel que la máquina pueda ejecutar las instrucciones.

Anteriormente se concluyeron las etapas de análisis sintáctico y análisis contextual, donde se creó la lógica de las instrucciones para el código Mini C#, además se comprobó la correcta escritura del código por parte del usuario. Además de mostrar el árbol de sintaxis abstracta donde se muestra el recorrido del código token a token.

Para culminar el proyecto se debe de comprobar el correcto funcionamiento de las etapas anteriores cuando se ejecute las instrucciones del código dado por el usuario.

Soluciones e implementaciones

Para el proceso de ejecutar las instrucciones de código ingresadas, se debe de descomponer el código en tokens, validando que estén en el orden correcto según las reglas ya escritas.

Después se procede a crear un árbol de sintaxis abstracta, donde están almacenadas las instrucciones en el preciso orden ingresadas, se realiza un recorrido en profundidad del árbol para generar un resultado.

Antes de mostrar el resultado, el recorrido del árbol de tokens se realiza mediante un recorrido en profundidad, *hijo izquierdo* – *padre* – *hijo derecho*. Para almacenar cada parámetro y resultado es necesario utilizar una pila, donde se almacene del árbol el *hijo izquierdo*, luego el *hijo derecho*, después se debe de operar ambos resultados según el dato que tenga el *padre*, y el resultado de la operación se guarda en la pila, en el árbol se sería el *hijo izquierdo* pero esta vez de un nivel más arriba. Para almacenar los tokens se hace diferencia entre las variables, métodos y clases, esto para mantener en orden la secuencia de declaraciones, por ejemplo, cuando se declara una variable dentro o fuera de un método, en el proyecto se tiene los archivos *almacenClases, almacenMetodos y almacenVariables* para este fin.

El archivo *ejecucionVisitor* es donde se realizan las visitas a cada regla, para comprobar la escritura del código, en este punto es donde se almacena en la pila los datos para generar resultados, push, y también donde se generan otros resultados de operaciones y se actualiza la pila, pop.

Análisis de resultados

|  |  |  |
| --- | --- | --- |
| Tarea | Estado | Observaciones |
| Implementación de visitors. | Completo |  |
| Ejecución de métodos predefinidos (ord,chr,len) | Completo |  |
| Llamada de funciones | Completo |  |
| Reconocimiento de parámetros. | Completo |  |

**Conclusiones y recomendaciones**

El proyecto le brinda al estudiante la oportunidad de analizar a fondo, en grandes rasgos, el proceso de un compilador, en específico la fase de análisis de la composición del código. Se logra analizar la validez de los datos ingresados por el usuario al editor de texto web. Esta última fase del proyecto logra mostrar el funcionamiento completo del editor, ejecutando las instrucciones del código, mostrando el resultado de la ejecución, aunque lo más importante del proyecto es comprender el proceso que se requiere para la ejecución del código.

La funcionalidad del editor se ve limitada, esto porque se utilizan solo algunas de las sentencias condicionales y funcionalidades del lenguaje de programación C#, aun así, el editor de texto es útil pero básico. El paso a paso de la creación del software queda abierto para el uso e investigación del estudiante, puede ver la esencia del mismo, además de agregar y modificar las funciones para investigación y creación de nuevos lenguajes de programación

**Manual de prueba**

Manual de prueba adjunto.

Referencias

Para esta fase no fue necesario consultar fuentes externas.