**Unidad 1. Análisis semántico.**

1.1 Árboles de expresiones.

1.2 Acciones semánticas de un analizador sintáctico.

1.3 Comprobaciones de tipos en expresiones.

1.4 Pila semántica en un analizador sintáctico.

1.5 Esquema de traducción.

1.6 Generación de la tabla de símbolo y tabla de direcciones.

1.7 Manejo de errores semánticos.

**1.2 Acciones semánticas de un analizador sintáctico**

***Tareas y Actividades.***

Tarea 1.

1. Agregar acciones semánticas a la estructura de la gramática.
2. Manipular la tabla de conversión de símbolos y de errores y direcciones.

***Recursos.***



Definición de un analizador sintáctico: es la fase del analizador que se encarga de chequear el texto de entrada en base a una gramática dada. Y en caso de que el programa de entrada sea válido, suministra el árbol sintáctico que lo reconoce.

En teoría, se supone que la salida del analizador sintáctico es alguna representación del árbol sintáctico que reconoce la secuencia de token suministrada por el analizador léxico.}

En la práctica, el analizador sintáctico también hace:

* Acceder a la tabla de símbolos (para hacer parte del trabajo del analizador semántico).
* Chequeo de tipos (del analizador semántico).
* Generar código intermedio.
* Generar errores cuando se producen.
* Controlar el flujo tokens reconocidos por parte del analizador léxico.
* Estrategia para recuperarse de errores sintácticos.
* Recuperación del modo pánico.
* Recuperación a nivel de frase.
* Producción de errores.
* Corrección global.
* Manejo de errores sintácticos.
* En definitiva, realiza casi todas las operaciones de la compilación. Este método de trabajo da lugar a los métodos de compilación dirigidos por sintaxis.

Manejo de errores sintácticos

Los errores sintácticos son dados por una expresión aritmética o paréntesis no equilibrados.

El manejo de errores de sintaxis es el más complicado desde el punto de vista de la creación de compiladores. Nos interesa que cuando el compilador encuentre un error, se recupere y siga buscando errores. Por lo tanto el manejador de errores de un analizador sintáctico tiene como objetivos:

• Indicar los errores de forma clara y precisa. Aclarar el tipo de error y su localización.

• Recuperarse del error, para poder seguir examinando la entrada.

• No ralentizar significativamente la compilación

Tipo de gramática que acepta un analizador sintáctico

Nosotros nos centraremos en el análisis sintáctico para lenguajes basados en gramáticas formales, ya que de otra forma se hace muy difícil la comprensión del compilador, y se pueden corregir, quizás más fácilmente, errores de muy difícil localización, como es la ambigüedad en el reconocimiento de ciertas sentencias.

La gramática que acepta el analizador sintáctico es una gramática de contexto libre:

Gramática: G (N, T, P, S)

N = No terminales.

T = Terminales.

P = Reglas de Producción.

S = Axioma Inicial.

ACCIONES SEMANTICAS

El análisis semántico se realiza después del sintáctico y es más difícil de formalizar que éste. Se trata de determinar el tipo de los resultados intermedios, comprobar que los argumentos que tiene un operador pertenecen al conjunto de los operadores posibles, y si son compatibles entre sí, es decir, comprobará que el significado de lo que se va leyendo es válido. El análisis semántico utiliza como entrada el árbol sintáctico detectado para comprobar restricciones de tipo y otras limitaciones semánticas y preparar la generación de código.

La salida “teórica” de la fase de análisis semántico sería un árbol semántico. Consiste

en un árbol sintáctico en el que cada una de sus ramas ha adquirido el significado que debe tener. En el caso de los operadores polimórficos (un único símbolo con varios significados), el análisis semántico determina cuál es el aplicable. Por ejemplo, consideremos la siguiente sentencia de asignación: A := B + C

En Pascal, el signo “+” sirve para sumar enteros y reales, concatena

r cadenas de caracteres y unir conjuntos. El análisis semántico debe comprobar que B y C sean de un tipo común o compatible y que se les pueda aplicar dicho operador. Si B y C son enteros o reales los sumará, si son cadenas las concatenará y si son conjuntos calculará su unión.

Dependiendo del tipo de sentencias, las acciones semánticas pueden agruparse en:

* Sentencias de Declaración: Completar la sección de tipos de la Tabla de Símbolos.
* Sentencias “ejecutables”: Realizar comprobaciones de tipos entre los operandos implicados.
* Funciones y procedimientos: Comprobar el número, orden y tipo de los parámetros actuales en cada      llamada a una función o procedimiento.
* Identificación de variables: Comprobar si un identificador ha sido declarado antes de utilizarlo.
* Etiquetas: Comprobar si hay etiquetas repetidas y validación.
* Constantes: Comprobar que no se utilicen en la parte izquierda de una asignación.
* Conversiones y equivalencias de tipo: Verificación.
* Sobrecarga de operadores y funciones: Detectar y solventar.