

Universidad Iberoamericana

Santo Domingo, RD.

# Repositorio de Informaciones de la Semana 10

# 

# 

Isaac Bonilla 19-0754

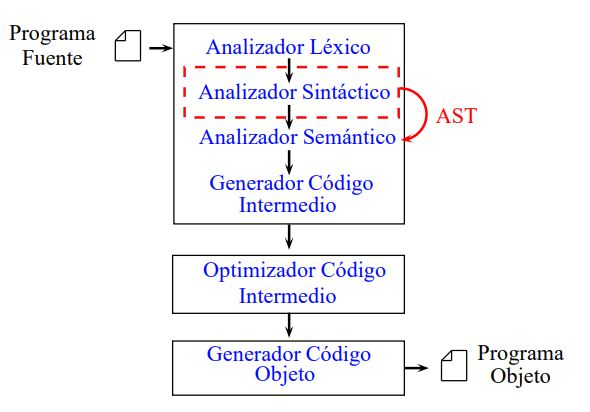
Prof. Rina Maria Familia

**Teoría de Compiladores**

Noviembre 15, 2021

**El analizador semántico**

El analizador semántico es la fase que sigue al análisis sintáctico. En esta fase se explora el AST (árbol de sintaxis abstracta) con el fin de detectar los errores semánticos.



**Funciones del Analizador Semántico:**

•Identificar cada tipo de instrucción y sus componentes

•Completar la Tabla de Símbolos

•Realizar distintas comprobaciones y validaciones:

•Comprobaciones de tipos.

•Comprobaciones del flujo de control.

•Comprobaciones de unicidad.

•Comprobaciones de emparejamiento.

•El Analizador Semántico finaliza la fase de Análisis del compilador y comienza la fase de Síntesis, en la cual se comienza a generar el código objeto.

•La especificación de la semántica puede realizarse de dos formas:

̧

* Lenguaje natural
* Especificación formal: Semántica Operacional, semántica denotacional, semántica Axiomática, Gramáticas con Atributos

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.

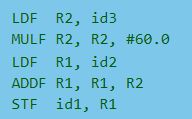
•**Sobrecarga de operadores y funciones**: Detectar y solventar

El analizador semántico utiliza el árbol sintáctico y la información en la tabla de símbolos para comprobar la consistencia semántica del programa fuente con la definición del lenguaje. También recopila información sobre el tipo y la guarda, ya sea en el árbol sintáctico o en la tabla de símbolos, para usarla más tarde durante la generación de código intermedio.

Una parte importante del análisis semántico es la comprobación (verificación) de tipos, en donde el compilador verifica que cada operador tenga operando que coincidan. Por ejemplo, muchas definiciones de lenguajes de programación requieren que el índice de un arreglo sea entero; el compilador debe reportar un error si se utiliza un número de punto flotante para indexar el arreglo.

La especificación del lenguaje puede permitir ciertas conversiones de tipo conocidas como coerciones. Por ejemplo, puede aplicarse un operador binario aritmético a un par de enteros o a un par de números de punto flotante. Si el operador se aplica a un número de punto flotante y a un entero, el compilador puede convertir u obligar a que se convierta en un número de punto flotante.

Dicha conversión aparece en la figura que se muestra debajo. Suponga que posición, inicial y velocidad se han declarado como números de punto flotante, y que el lexema 60 por sí solo forma un entero. El comprobador de tipo en el analizador semántico de la figura de abajo descubre que se aplica el operador \* al número de punto flotante velocidad y al entero 60. En este caso, el entero puede convertirse en un número de punto flotante. Observe en la figura de abajo que la salida del analizador semántico tiene un nodo adicional para el operador inttofloat, que convierte de manera explícita su argumento tipo entero en un número de puinto flotante.



**ANÁLISIS SEMÁNTICO EN PROCESADORES DE LENGUAJE**

La fase de análisis semántico de un procesador de lenguaje es aquélla que computa la información adicional necesaria para el procesamiento de un lenguaje, una vez que la estructura sintáctica de un programa haya sido obtenida. Es por tanto la fase posterior a la de análisis sintáctico y la última dentro del proceso de síntesis de un lenguaje de programación [Aho90].

Sintaxis de un lenguaje de programación es el conjunto de reglas formales que especifican la estructura de los programas pertenecientes a dicho lenguaje. Semántica de un lenguaje de programación es el conjunto de reglas que especifican el significado de cualquier sentencia sintácticamente válida. Finalmente, el análisis semántico 1 de un procesador de lenguaje es la fase encargada de detectar la validez semántica de las sentencias aceptadas por el analizador sintáctico.

Ejemplo: este es un ejemplo de una línea de código simple en lenguaje C.

superficie = base \* altura / 2;

La sintaxis del lenguaje C indica que las expresiones se pueden formar con un conjunto de operadores y un conjunto de elementos básicos. Entre los operadores, con sintaxis binaria infija, se encuentran la asignación, el producto y la división. Entre los elementos básicos de una expresión existen los identificadores y las constantes enteras sin signo (entre otros). Su semántica identifica que en el registro asociado al identificador superficie se le va a asociar el valor resultante del producto de los valores asociados a base y altura, divididos por dos (la superficie de un triángulo). Finalmente, el análisis semántico del procesador de lenguaje, tras haber analizado correctamente que la sintaxis es válida, deberá comprobar que se satisfacen las siguientes condiciones:

− Que todos los identificadores que aparecen en la expresión hayan sido declarados en el ámbito actual, o en alguno de sus ámbitos (bloques^2 ) previos.

− Que la subexpresión de la izquierda sea semánticamente válida, es decir, que sea un lvalue^3.

− Que a los tipos de los identificadores base y altura se les pueda aplicar el operador de multiplicación. Un registro en C, por ejemplo, no sería válido.

bloque^2: El lenguaje C es un lenguaje orientado a bloques. Los bloques se especifican mediante la pareja de caracteres { y }. Dentro de un bloque, es posible declarar variables que ocultan a las variables declaradas en bloques de un nivel menor de anidamiento.

^3lvalue: Una expresión es un lvalue (left value, valor a la izquierda) si puede estar a la izquierda en una expresión de asignación, es decir, si se puede obtener su dirección de memoria y modifica el contenido de ésta. Otros ejemplos de expresiones lvalue en C, son: \*puntero, array[n], registro.campo...

Referencias:

[[PDF] ANÁLISIS SEMÁNTICO. Especificación formal: Semántica Operacional, semántica denotacional, semántica Axiomática, Gramáticas con Atributos. - Free Download PDF (silo.tips)](https://silo.tips/download/analisis-semantico-especificacion-formal-semantica-operacional-semantica-denotac)

[2.7. Generación de código - Teoría de Lenguajes y Compiladores (google.com)](https://sites.google.com/site/teoriadelenguajesycompiladores/procesadores-de-lenguaje/generacion)

[(Microsoft Word - Sem\341ntico.0.32.doc) (uniovi.es)](http://di002.edv.uniovi.es/~cueva/publicaciones/monografias/38_semantico.pdf)

[Teoría de Autómatas y Compiladores [ICI-445] Capítulo 5: Análisis Semántico (ucv.cl)](http://zeus.inf.ucv.cl/~rsoto/cursos/ICI445/Cap5_ICI445.pdf)