

# UNIVERSIDAD AUTÓNOMA DE CHIAPAS



FACULTAD DE CONTADURÍA Y ADMINISTRACIÓN  
CAMPUS 1

ING. EN DESARROLLO Y TECNOLOGÍAS DE SOFTWARE

6 "M"

COMPILADORES

SUBCOMPETENCIA III. - ANÁLISIS SEMÁNTICO

ACT. 3.1. REALIZAR UN ARTEX REFERENTE A ACCIONES  
SEMÁNTICAS DE UN ANALIZADOR SINTÁCTICO

ALUMNO: MARCO ANTONIO ZÚÑIGA MORALES – A211121

DOCENTE: DR. LUIS GUTIÉRREZ ALFARO

TUXTLA GUTIÉRREZ, CHIAPAS  
VIERNES, 27 DE OCTUBRE DE 2023

### 3.1 Árboles de expresiones - ARTEX

**Nombre del estudiante:** Marco Antonio Zúñiga Morales

**Síntesis:** Los árboles de expresiones son una representación visual y estructurada de las expresiones matemáticas o lógicas que se utilizan en programación y en análisis sintáctico. Estos árboles permiten descomponer una expresión compleja en sus componentes más simples y jerárquicos, lo que facilita su evaluación y procesamiento por parte de un analizador sintáctico.

#### Conceptos importantes

- **Expresiones:** una expresión es una combinación de operandos y operadores que se evalúa para producir un valor. Ejemplos de expresiones incluyen operaciones matemáticas (como suma, resta, multiplicación, división), expresiones lógicas (como operadores booleanos), y mucho más.
- **Árbol de expresiones:** Un árbol de expresiones es una estructura jerárquica que se utiliza para representar una expresión de manera clara y organizada. En este tipo de árbol, cada nodo representa un operador u operando, y las ramas conectan los nodos siguiendo la jerarquía de la expresión.
- **Analizador sintáctico:** Un analizador sintáctico es una parte esencial de un compilador o intérprete que se encarga de analizar la estructura gramatical de un programa informático.

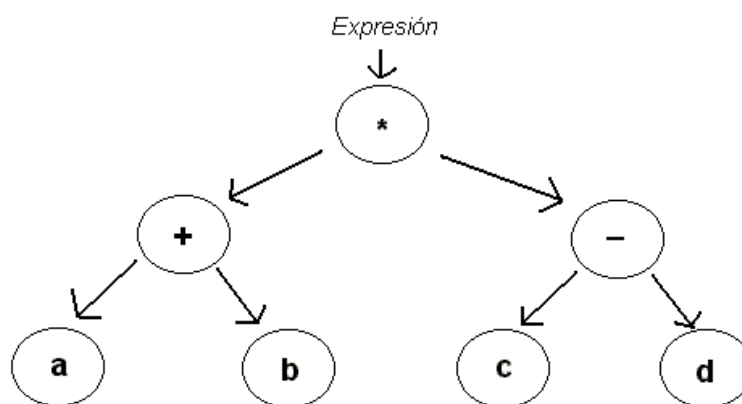
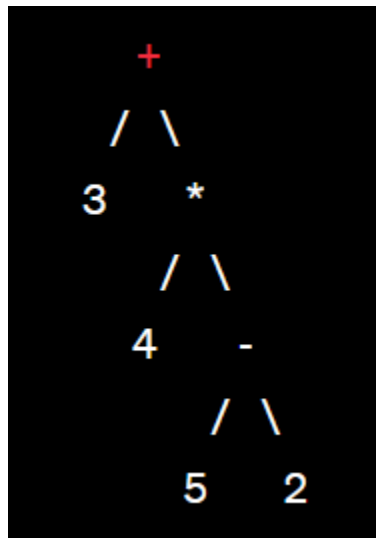
#### Ideas principales

- **Representación Jerárquica:** Los árboles de expresiones representan las expresiones de manera jerárquica, donde los operadores se sitúan en los nodos internos y los operandos en las hojas.
- **Facilitan la Evaluación:** Al descomponer una expresión en un árbol, es más fácil evaluarla paso a paso, siguiendo la jerarquía de los operadores.
- **Orden de Evaluación:** Los árboles de expresiones también indican el orden de evaluación de los operadores, lo que es esencial para obtener el resultado correcto de una expresión.
- **Compatibilidad con Analizadores Sintácticos:** Los analizadores sintácticos utilizan los árboles de expresiones para comprender y procesar las expresiones en el código fuente, lo que es crucial en la compilación o interpretación de programas.

## Esquema



## Imagen o dibujo



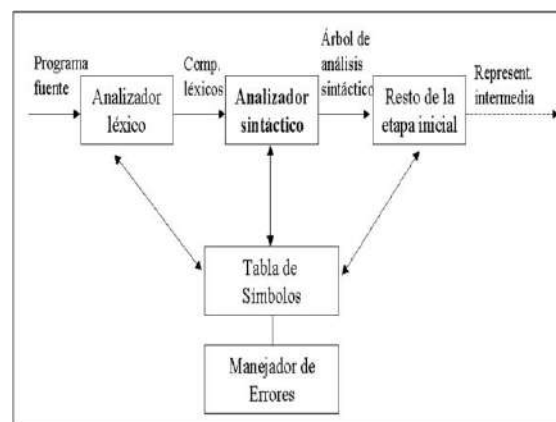
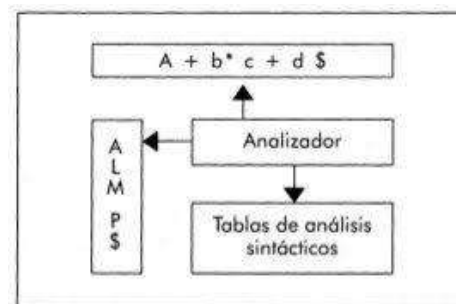
### 3.2 Acciones semánticas de analizador sintáctico - ARTEX

<b>Nombre del estudiante:</b> Marco Antonio Zúñiga Morales	
<b>Síntesis:</b> En el proceso de análisis sintáctico de un programa informático, las acciones semánticas desempeñan un papel fundamental. Estas acciones permiten la asignación de significado a la estructura gramatical de un programa, lo que es esencial para su posterior ejecución.	
<b>Conceptos importantes</b> <ul style="list-style-type: none"><li>• <b>Acciones semánticas:</b> son procedimientos o funciones asociados a reglas gramaticales en un analizador sintáctico. Estas acciones se ejecutan cuando se reconoce una producción gramatical y se utilizan para asignar significado y llevar a cabo acciones específicas en el programa.</li><li>• <b>Análisis sintáctico:</b> es la etapa del proceso de compilación o interpretación que verifica la estructura gramatical de un programa. Las acciones semánticas complementan esta etapa al permitir la construcción de árboles de sintaxis abstracta y la asignación de significado a las estructuras reconocidas.</li><li>• <b>Árbol de Sintaxis Abstracta (ATS):</b> es una estructura de datos que representa la estructura jerárquica de un programa después del análisis sintáctico. Las acciones semánticas contribuyen a la construcción de un AST, lo que facilita la posterior interpretación o compilación.</li></ul>	<b>Ideas principales</b> <ul style="list-style-type: none"><li>• <b>Asignación de Significado:</b> Las acciones semánticas permiten asignar significado a la estructura gramatical del programa. Esto implica dar sentido a las variables, operadores y otras construcciones sintácticas.</li><li>• <b>Construcción del Árbol de Sintaxis Abstracta:</b> Durante el proceso de análisis sintáctico, se construye un árbol de sintaxis abstracta. Las acciones semánticas contribuyen a la creación de este árbol, que representa la estructura lógica del programa.</li><li>• <b>Validación Semántica:</b> Las acciones semánticas también se utilizan para realizar verificaciones semánticas, como la comprobación de tipos, la resolución de nombres y la detección de errores semánticos.</li><li>• <b>Interpretación y Compilación:</b> Una vez que las acciones semánticas han asignado significado y construido el AST, el programa está listo para la interpretación o compilación, lo que permite su ejecución.</li></ul>

## Esquema



## Imagen o dibujo



### 3.3 Comprobaciones de tipos en expresiones - ARTEX

**Nombre del estudiante:** Marco Antonio Zúñiga Morales

**Síntesis:** Las comprobaciones de tipos en expresiones son un componente esencial del análisis semántico en la construcción de compiladores e intérpretes. Estas comprobaciones se realizan para garantizar que las operaciones y asignaciones en un programa se realicen de acuerdo con las reglas de tipos definidas.

#### Conceptos importantes

- **Comprobaciones de Tipos:** Las comprobaciones de tipos se refieren al proceso de verificar que los tipos de datos involucrados en una expresión sean compatibles con las operaciones que se están realizando. Esto incluye la verificación de la coherencia entre los tipos de los operandos y el tipo del resultado de una operación.
- **Tipos de Datos:** Los tipos de datos representan la naturaleza de la información contenida en una variable, constante o expresión. Ejemplos comunes de tipos de datos incluyen enteros, flotantes, caracteres, booleanos y estructuras personalizadas.
- **Análisis Semántico:** El análisis semántico es una fase crucial en la construcción de compiladores e intérpretes que se encarga de verificar la coherencia semántica de un programa, lo que incluye las comprobaciones de tipos.

#### Ideas principales

- **Detección de Errores:** Las comprobaciones de tipos ayudan a detectar errores de tipo, como sumar un entero con una cadena de caracteres, lo que podría generar un comportamiento inesperado o errores en tiempo de ejecución.
- **Garantía de Coherencia:** Aseguran que las operaciones se realicen de acuerdo con las reglas definidas para los tipos de datos, lo que evita resultados incorrectos o incoherentes.
- **Conversión de Tipos:** En algunos casos, las comprobaciones de tipos pueden requerir la conversión de tipos de datos para que una operación sea válida. Por ejemplo, convertir un entero en un flotante antes de realizar una operación aritmética.
- **Mejora de Rendimiento:** Las comprobaciones de tipos también pueden optimizar el rendimiento al evitar conversiones innecesarias o realizar operaciones especializadas según los tipos involucrados.

## Esquema



## Imagen o dibujo

**Tipo Simple : X: Integer**

