

# Clase #24 de 27

## Expresiones: Sintaxis & Semántica

*Octubre 14, Martes*  
*Octubre 15, Martes*

# Agenda para esta clase

- Expresiones
- Sintaxis de C: Expresiones

# Expresiones

# Conceptos de Expresiones

- **Expresión**
  - **Secuencia** de **operandos** y **operadores** que al evaluarla genera un **valor** y, opcionalmente, produce un **efecto**
- **Operando**
  - Nodos hojas del árbol
- **Operador**
  - Nodos internos del árbol
- **Aridad** de los operadores
  - Unario (e.g., \*p)
  - Binario (e.g., a[i])
  - Ternario ( a ? b : c )
  - N-ario ( f(a, b, c) )
- **Valor** de la expresión
- **Tipo** de la expresión
- **Efecto** de lado (ó colateral, secundario)
- **Orden de evaluación** de los operandos y "corto circuito"
- **Precedencia** (orden) de ejecución de las operaciones
- **Asociatividad** (agrupación) de los operadores:
  - ID, DI, sin asociatividad
- **Categoría de expresión:**
  - rvalue, lvalue, lvalue modificable
- **Expresión Primaria**
  - *identificador*
  - *constante*
  - *constante-cadena*
  - ( *expresión* )

# Operadores propios de C, poco comunes en otros LP

*// Operador Coma*

exp1, exp2, exp3

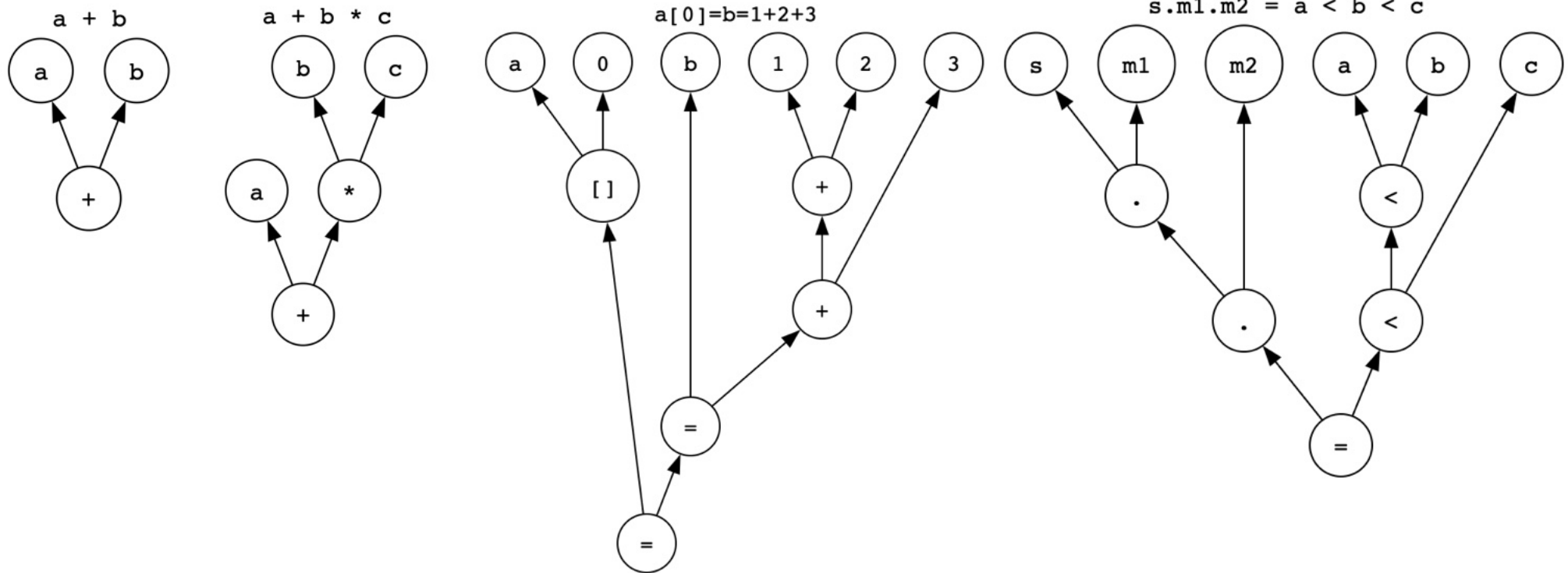
*// Operador Condicional o Ternario*

exp1 ? exp2 : exp3

*// Operador sizeof*

sizeof exp  
sizeof(*type*)

# Árboles de Expresión (I): Operadores, Operandos, Precedencia, y Asociatividad



# Sintaxis de C: Expresiones

# Expresión (1/10)

## Coma, Asignación y Condicional

*expresión*

*expresión-de-asignación*

*expresión , expresión-de-asignación*

*expresión-de-asignación*

*expresión-condicional*

*expresión-unaria operador-de-asignación expresión-de-asignación*

*operador-de-asignación uno de*

**= \*= /= %= += -= <<= >>= &= ^= |=**

*expresión-condicional*

*expresión-O-lógico*

*expresión-O-lógico ? expresión : expresión-condicional*



# Expresión (2/10)

## Conectores lógicos

*expresión-O-lógico*

*expresión-Y-lógico*

*expresión-O-lógico* || *expresión-Y-lógico*

*expresión-Y-lógico*

*expresión-O-inclusivo*

*expresión-Y-lógico* && *expresión-O-inclusivo*

# Expresión (3/10)

## Operadores binarios a nivel bits

*expresión-O-inclusivo*

*expresión-O-excluyente*

*expresión-O-inclusivo* | *expresión-O-excluyente*

*expresión-O-excluyente*

*expresión-Y*

*expresión-O-excluyente* ^ *expresión-Y*

*expresión-Y*

*expresión-de-igualdad*

*expresión-Y* & *expresión-de-igualdad*

# Expresión (4/10)

## Igualdad y relación

*expresión-de-igualdad*

*expresión-relacional*

*expresión-de-igualdad == expresión-relacional*

*expresión-de-igualdad != expresión-relacional*

*expresión-relacional*

*expresión-de-corrimento*

*expresión-relacional < expresión-de-corrimento*

*expresión-relacional > expresión-de-corrimento*

*expresión-relacional <= expresión-de-corrimento*

*expresión-relacional >= expresión-de-corrimento*

# Expresión (5/10)

## Corrimiento de bits

*expresión-de-corrimiento*

*expresión-aditiva*

*expresión-de-corrimiento* << *expresión-aditiva*

*expresión-de-corrimiento* >> *expresión-aditiva*

# Expresión (6/10)

## Adiciones y Multiplicaciones

*expresión-aditiva*

*expresión-multiplicativa*

*expresión-aditiva + expresión-multiplicativa*

*expresión-aditiva - expresión-multiplicativa*

*expresión-multiplicativa*

*expresión-de-conversión*

*expresión-multiplicativa \* expresión-de-conversión*

*expresión-multiplicativa / expresión-de-conversión*

*expresión-multiplicativa % expresión-de-conversión*

# Expresión (7/10)

## Casteo

*expresión-de-conversión*

*expresión-unaria*

*( nombre-de-tipo ) expresión-de-conversión*

# Expresión (8/10) Unarios, pre-incremento y decremento, y sizeof

*expresión-unaria*

*expresión-sufijo*

**++** *expresión-unaria*

**--** *expresión-unaria*

*operador-unario* *expresión-de-conversión*

**sizeof** *expresión-unaria*

**sizeof** ( *nombre-de-tipo* )

*operador-unario* uno de

**& \* + - ~ !**

# Expresión (9/10)

## Operadores sufijos

*expresión-sufijo*

*expresión-primaria*

*expresión-sufijo* [ *expresión* ]

Subindicación

*expresión-sufijo* ( *lista-de-argumentos?* )

Invocación

*expresión-sufijo* . *identificador*

Estructura

*expresión-sufijo* -> *identificador*

Puntero a estructura

*expresión-sufijo* ++

Posincremento

*expresión-sufijo* --

Posdecremento

*lista-de-argumentos*

*expresión-de-asignación*

*lista-de-argumentos* , *expresión-de-asignación*



# Expresión (10/10)

## Expresiones primarias

*expresión-primaria*

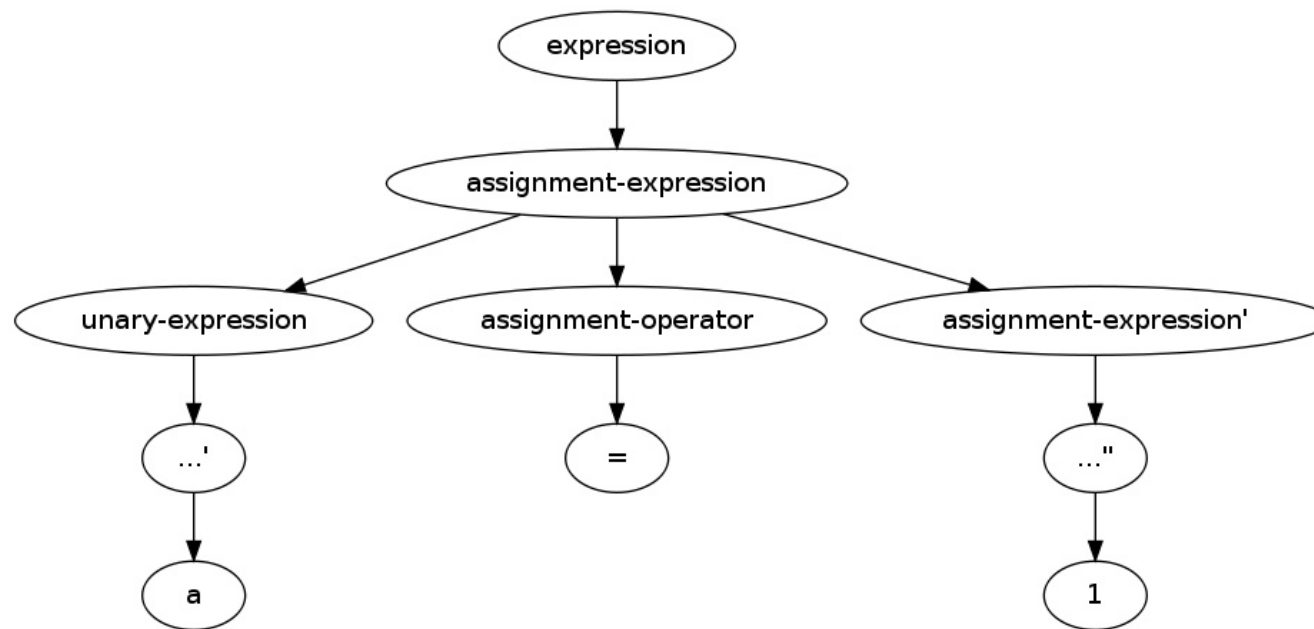
*identificador*

*constante*

*constante-cadena*

*( expresión )*

# Congruencia entre Árbol de Derivación y Árbol de Expresión



# ¿Consultas?

**Fin de la clase**