Árboles de Sintaxis/Expresión
Árboles de Codificación

Estructuras de Datos

Andrea Rueda

Pontificia Universidad Javeriana
Departamento de Ingeniería de Sistemas

Árbol de Sintaxis

Árbol de Sintaxis

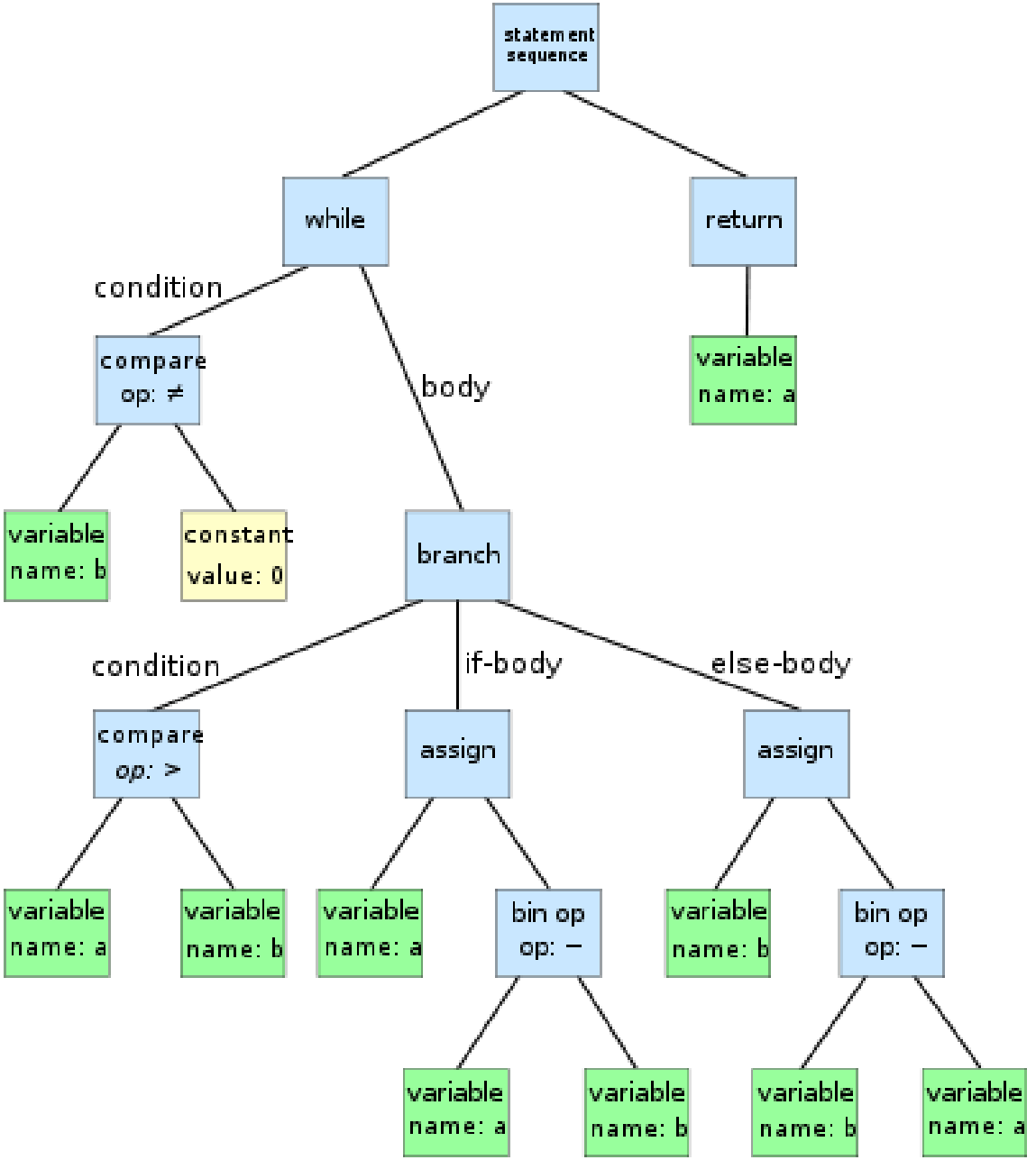
- Representación en árbol de la estructura sintáctica del código fuente.
- Abstracto en el sentido de que no incorpora de forma exacta cada detalle presente en el código (por ejemplo, los paréntesis están implícitos).
- Facilita la visualización de la estructura del código, la localización de elementos dentro del mismo, y la anotación de información adicional.

Árbol de Sintaxis

- Ejemplo:

```
while (b != 0)
        if (a > b)
            a = a – b;
        else
            b = b – a;
    return a;
```

Árbol de Sintaxis



http://en.wikipedia.org/wiki/Abstract_syntax_tree

Árbol de Sintaxis

Interpretación del lenguaje:

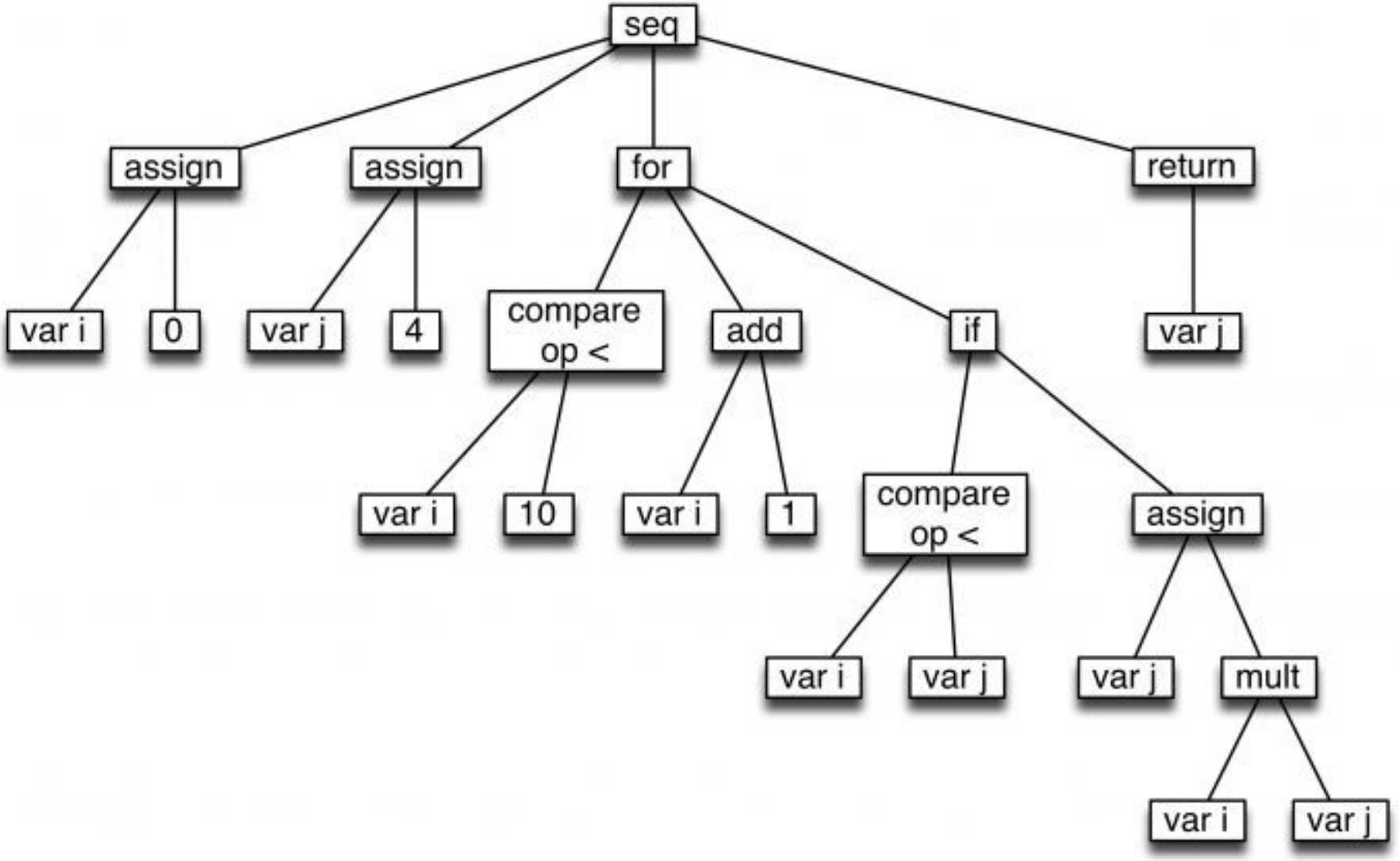
- ¿Qué representan las hojas?
- ¿Qué representan los demás nodos?
- ¿Qué significa recorrer el árbol?
  - Preorden.
  - Inorden.
  - Posorden.
  - Niveles.

Árbol de Sintaxis

- Ejercicio:

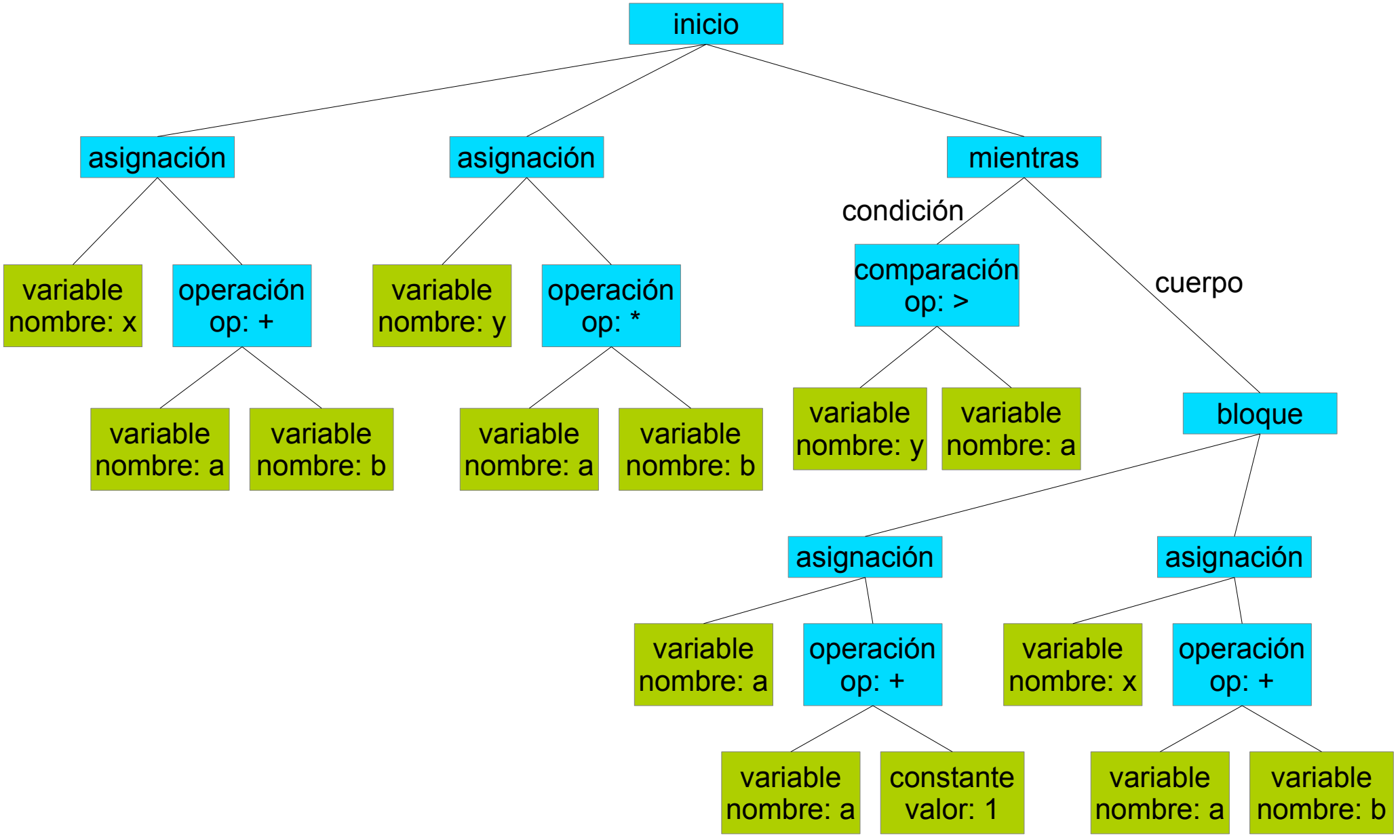
```
int i = 0;
int j = 4;
for ( ; i<10; i++) {
   if (i<j) {
      j = i*j;
   }
}
return j;
```

Árbol de Sintaxis



http://blog.zenika.com/index.php?post/2012/08/23/Static-analysis-%28part-1/4%29

Árbol de Sintaxis



Árbol de Sintaxis

```
x = a + b;
y = a * b;
while (y > a) {
    a = a + 1;
    x = a + b;
}
```

Árbol de Expresión

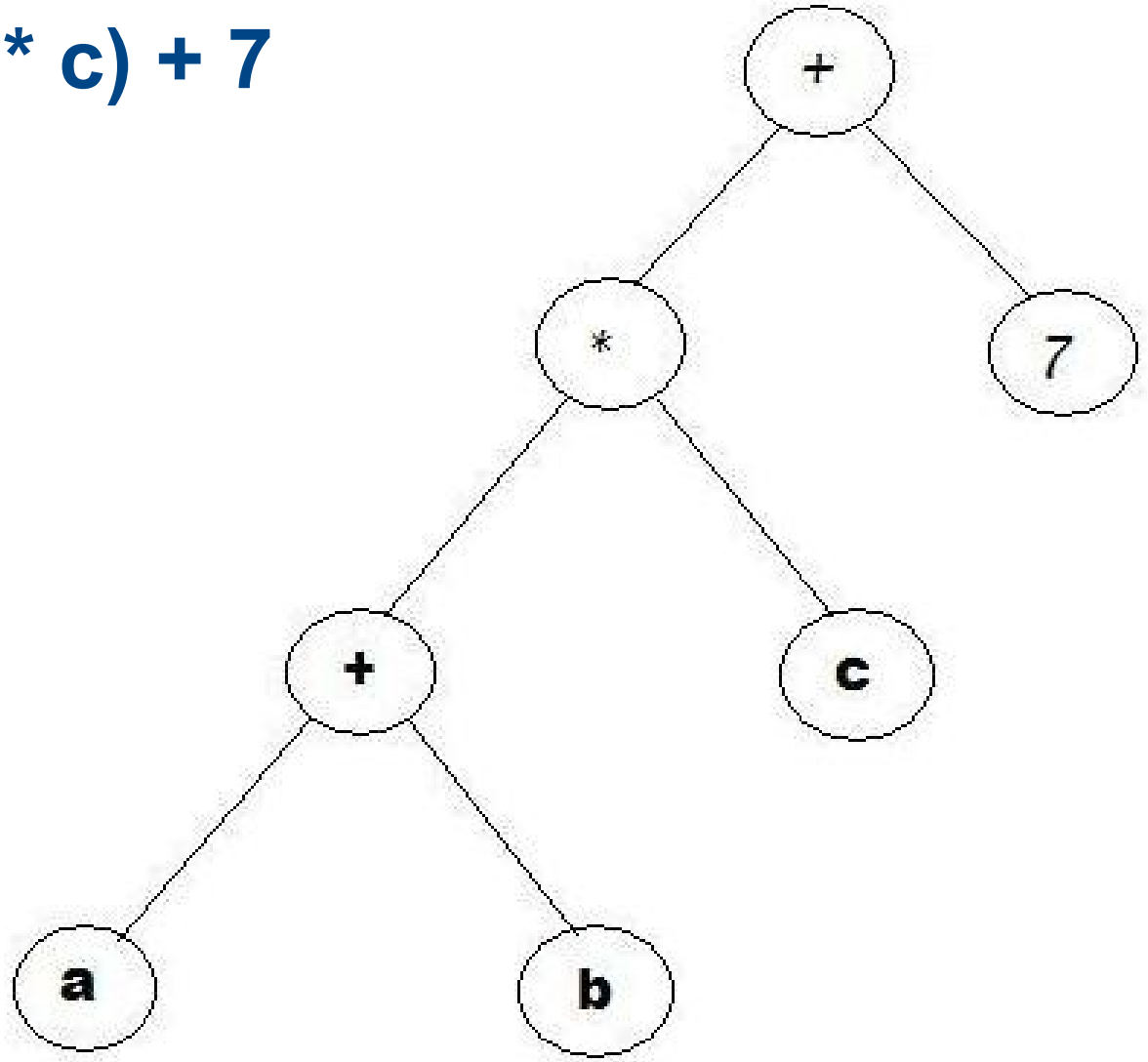
Árbol de Expresión

Árbol binario de expresión:

- Aplicación específica de un árbol binario para la evaluación de ciertas expresiones:
  - Algebraicas o Booleanas.
  - Operadores unarios o binarios.

Árbol de Expresión

((a + b) * c) + 7



http://en.wikipedia.org/wiki/Binary_expression_tree

Árbol de Expresión

- Cada nodo hoja contiene un único operando.
- Cada nodo interno contiene un único operador (unario o binario).
- Los subárboles izquierdo y derecho de un nodo operador representan subexpresiones que deben ser evaluadas antes de aplicar el operador en la raíz del subárbol.

Árbol de Expresión

- Los niveles de los nodos indican la precedencia relativa de su evaluación (no se requieren paréntesis).
- Operaciones en los primeros niveles (niveles altos) se evalúan después que operaciones en los últimos niveles (niveles bajos).
- La operación en la raíz es siempre la última en ejecutarse.

Árbol de Expresión

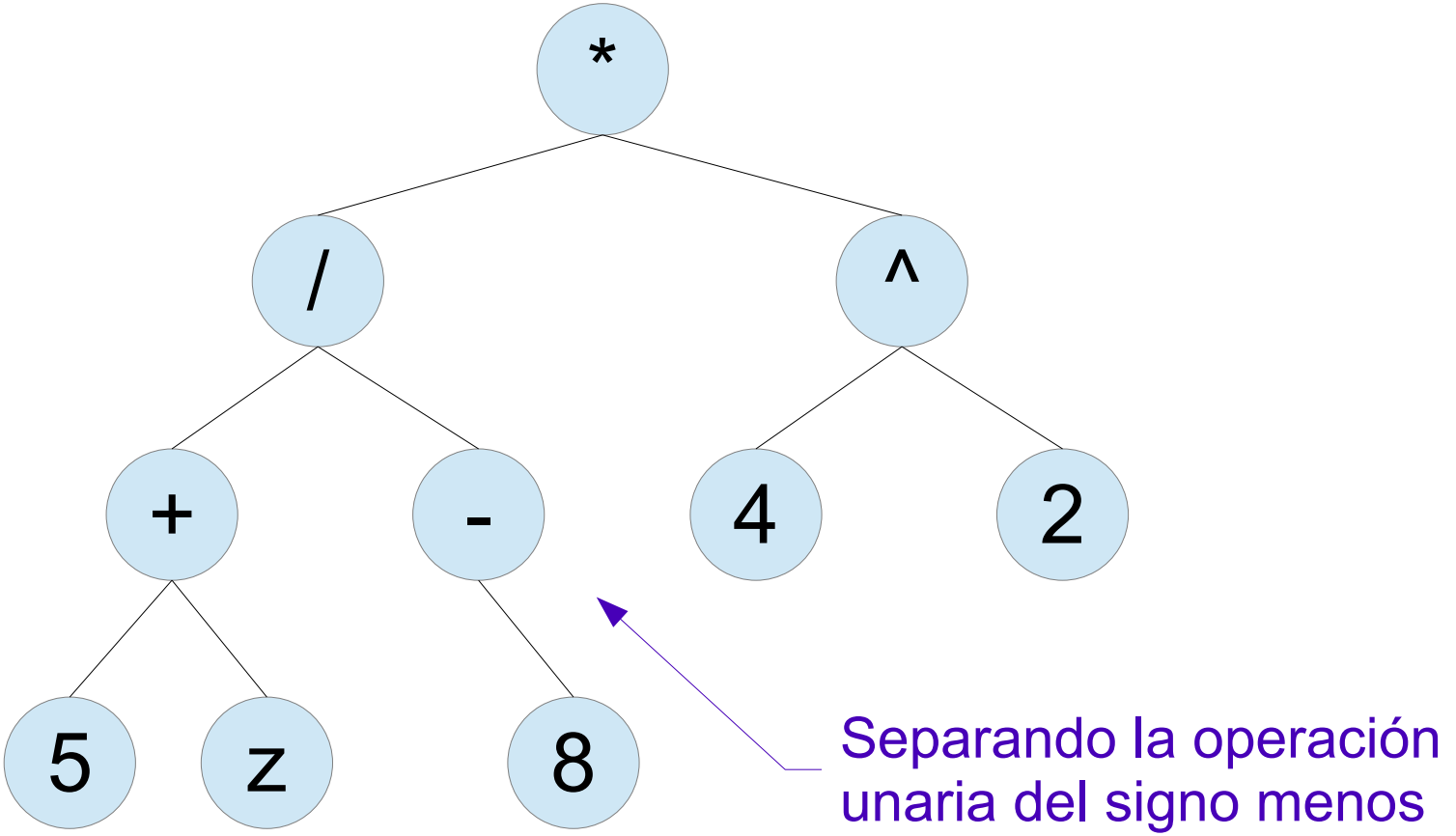
- Ejercicio

  Construir el árbol asociado a la expresión:

  ((5 + z) / (- 8)) * (4 ^ 2)

Árbol de Expresión

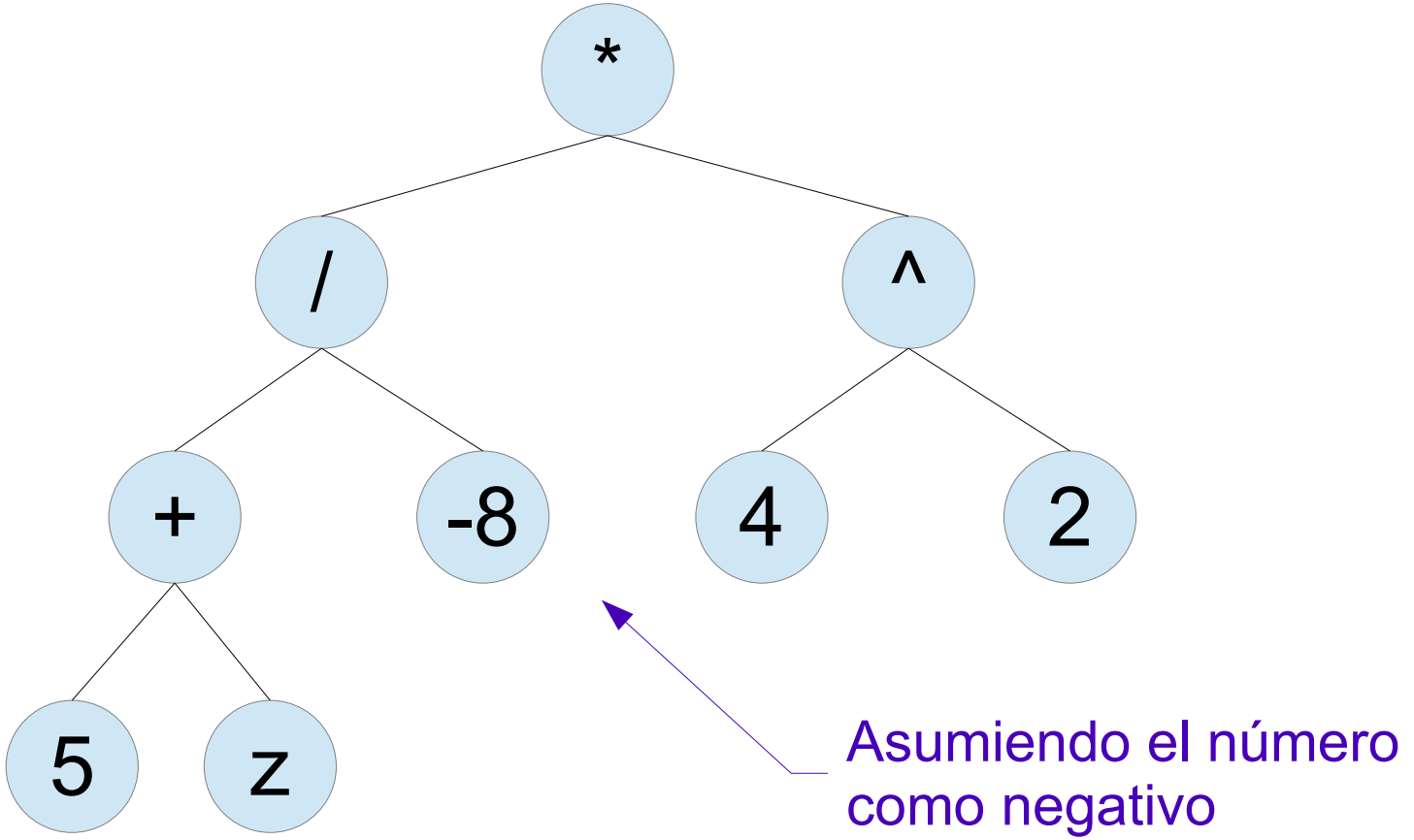
((5 + z) / (- 8)) * (4 ^ 2)



commons.wikimedia.org/wiki/File:Binary_Algebraic_Expression_Tree.JPG

Árbol de Expresión

((5 + z) / (- 8)) * (4 ^ 2)



commons.wikimedia.org/wiki/File:Binary_Algebraic_Expression_Tree.JPG

Árbol de Expresión

- ¿Recorridos?

  Preorden:

  Posorden:

  Inorden:

Árbol de Expresión

- ¿Recorridos?

  Preorden:

  * / + 5 z - 8 ^ 4 2 — * / + 5 z -8 ^ 4 2

  Posorden:

  5 z + 8 - / 4 2 ^ * — 5 z + -8 / 4 2 ^ *

  Inorden:

  5 + z / - 8 * 4 ^ 2 — 5 + z / -8 * 4 ^ 2

Separando la operación unaria del signo menos

Asumiendo el número como negativo

Árbol de Expresión

- Recorridos:

  Asociados con la generación de expresiones en notación:

  - Polaca (notación prefija): **+ 3 5**

    recorrido en preorden.
  - Polaca inversa (notación posfija): **3 5 +**

    recorrido en posorden.
  - Infija (la más común): **3 + 5**

    recorrido en inorden.

Árbol de Expresión

- Recorridos:

  Asociados con la generación de expresiones en notación:

  - Polaca (notación prefija):

    recorrido en preorden.

    + 3 5
    no ambigua

  - Polaca inversa (notación posfija):

    recorrido en posorden.

    3 5 +
    no ambigua

  - Infija (la más común):

    recorrido en inorden.

    3 + 5
    ambigua (requiere paréntesis)

Árbol de Expresión

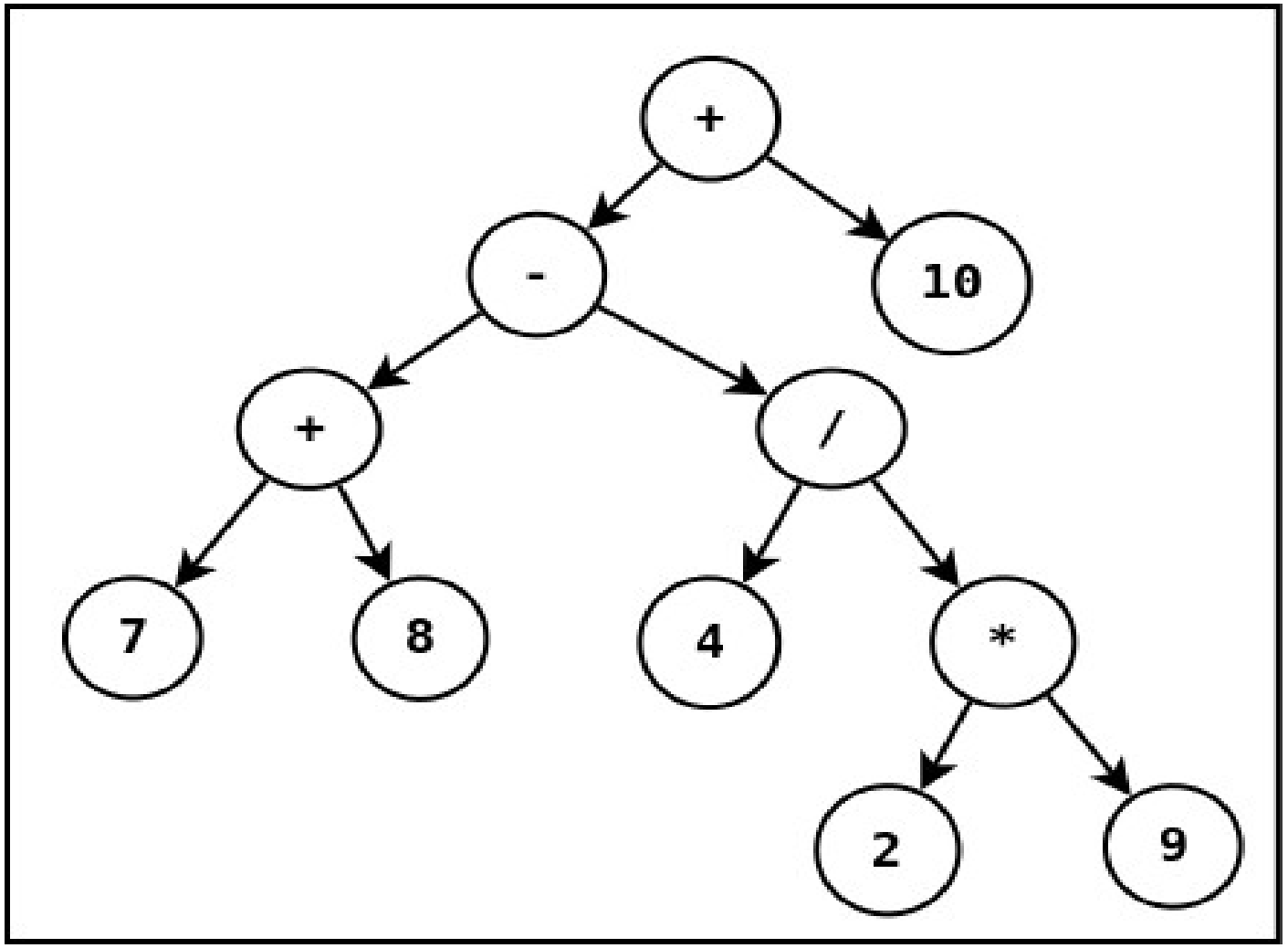
- ¿Generación del árbol a partir de la expresión?

Árbol de Expresión

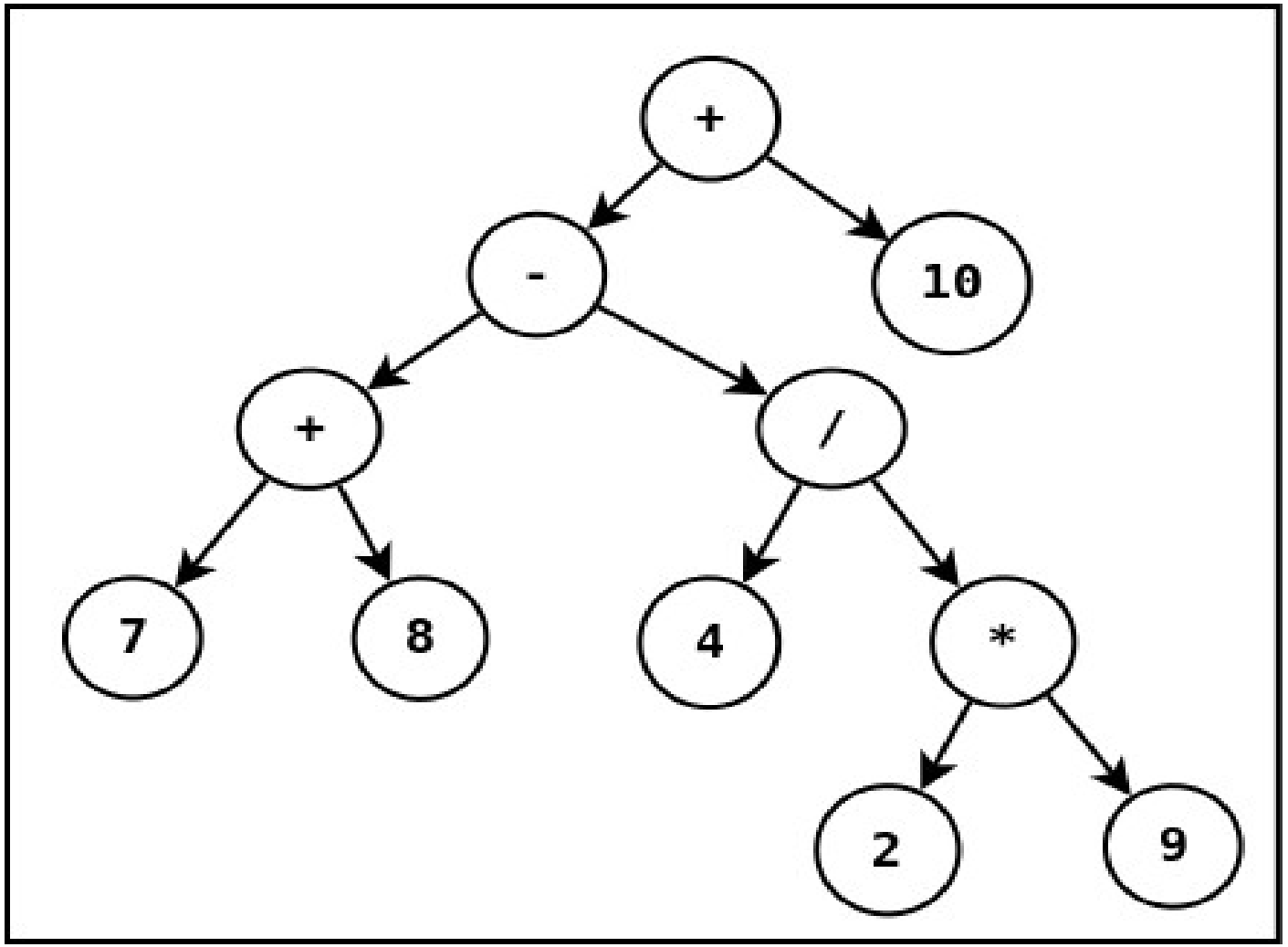
- Ejercicio: (7 + 8 – (4 / (2 * 9))) + 10

Árbol de Expresión

- Ejercicio: (7 + 8 – (4 / (2 * 9))) + 10

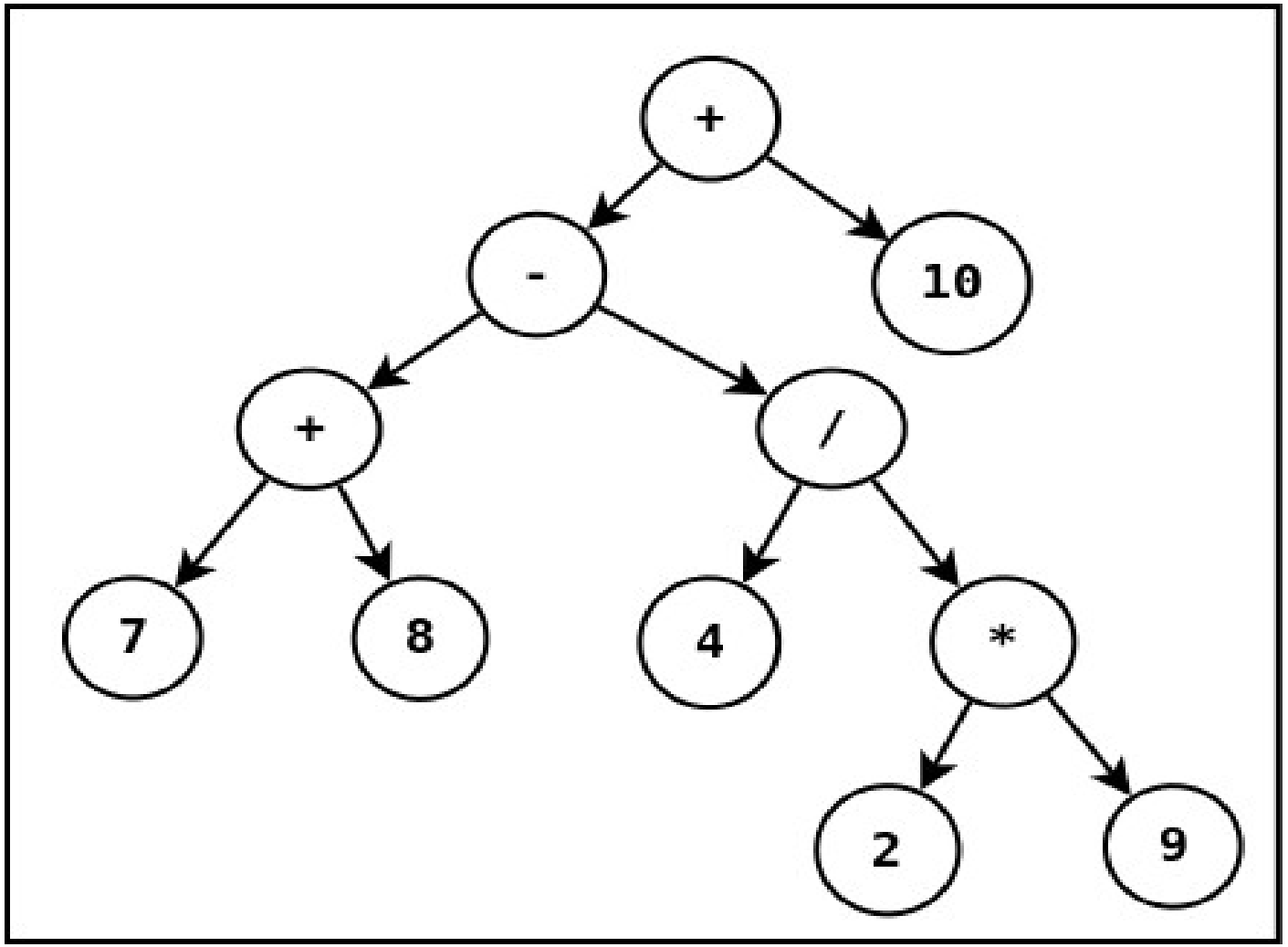


Árbol de Expresión



- Preorden (prefija):
- Posorden (posfija):

Árbol de Expresión



- Preorden (prefija): + - + 7 8 / 4 * 2 9 10
- Posorden (posfija): 7 8 + 4 2 9 * / - 10 +

Árbol de Expresión

- Generación del árbol a partir de la expresión

  ¿Expresión posfija?

  - Empezar por el extremo derecho de la expresión.
  - Ese elemento debe ser un operador, el cual se ubica como la raíz del árbol.
  - Para cada elemento siguiente (hacia la izquierda):
    - Si el nodo actual es operador, poner el elemento como hijo derecho, si ya hay hijo derecho, entonces como hijo izquierdo.
    - Si el nodo actual es operando, regresar tantos ancestros como sea necesario para insertar el elemento ahí.

Árbol de Expresión

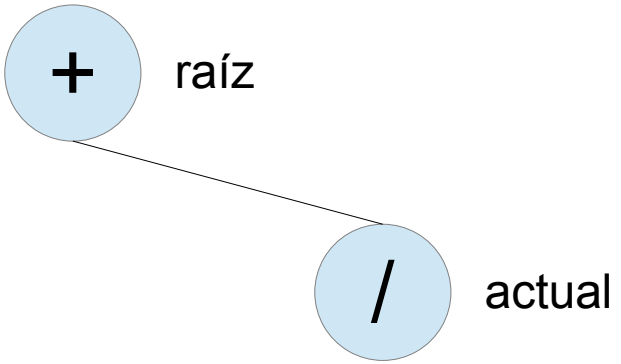
- Ejemplo expresión posfija: 2 6 * 3 8 / +

Árbol de Expresión

- Ejemplo expresión posfija: 2 6 * 3 8 / +

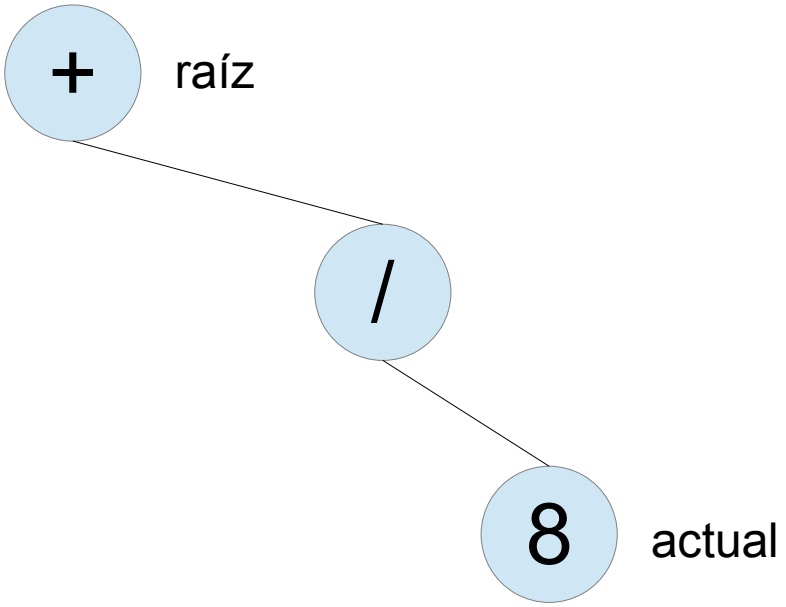
Árbol de Expresión

- Ejemplo expresión posfija: 2 6 * 3 8 / +



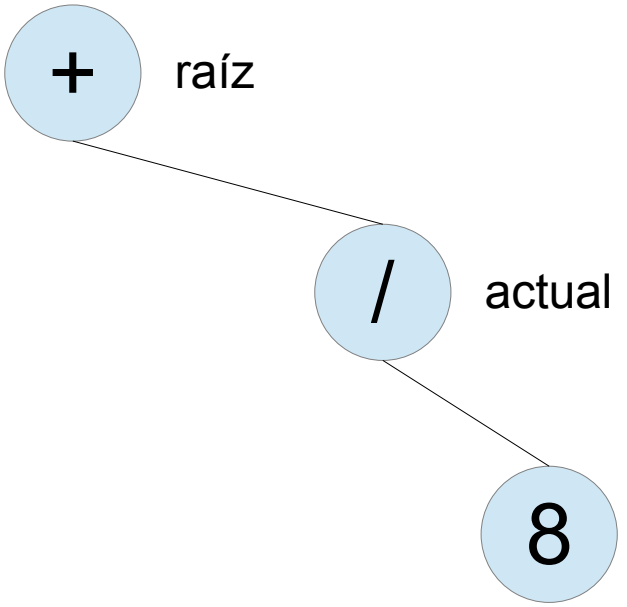
Árbol de Expresión

- Ejemplo expresión posfija: 2 6 * 3 8 / +



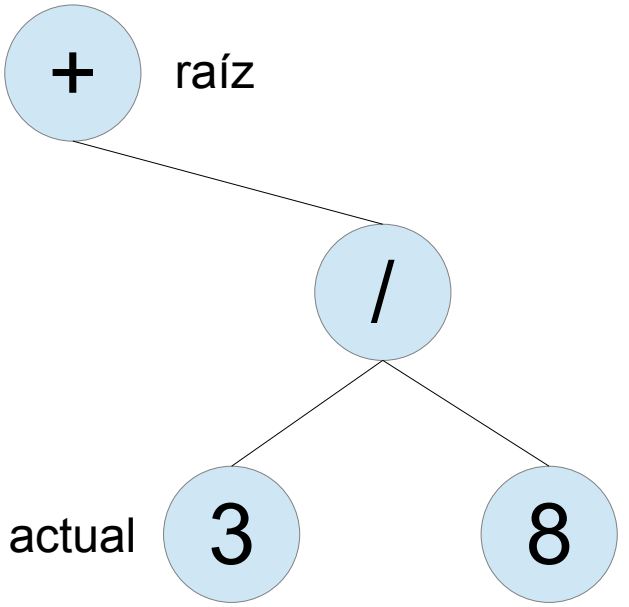
Árbol de Expresión

- Ejemplo expresión posfija: 2 6 * 3 8 / +



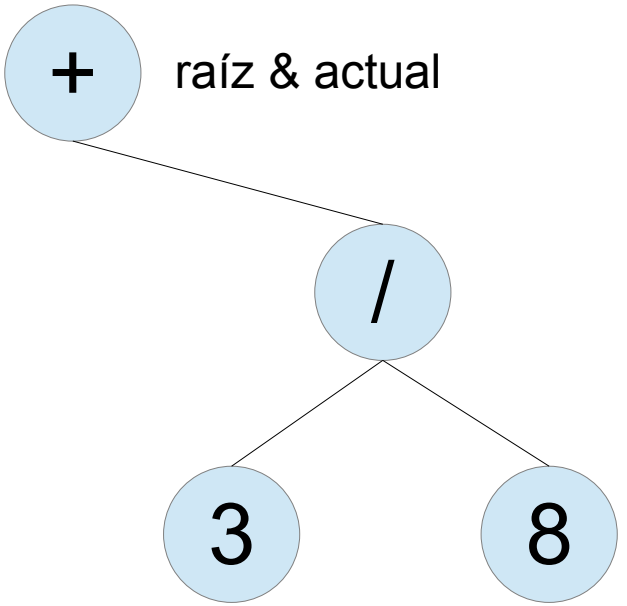
Árbol de Expresión

- Ejemplo expresión posfija: 2 6 * 3 8 / +



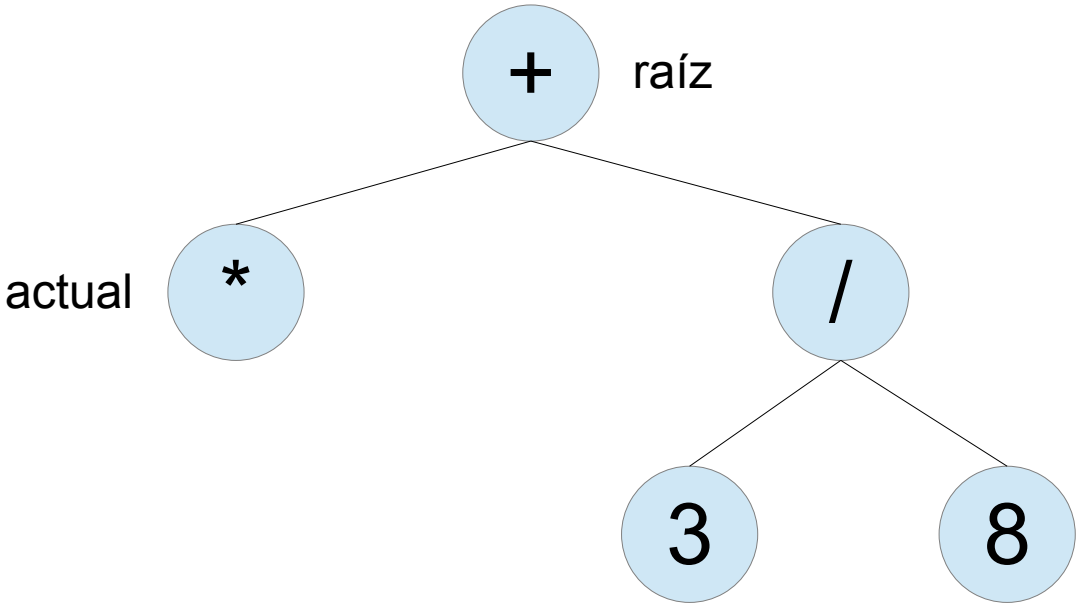
Árbol de Expresión

- Ejemplo expresión posfija: 2 6 * 3 8 / +



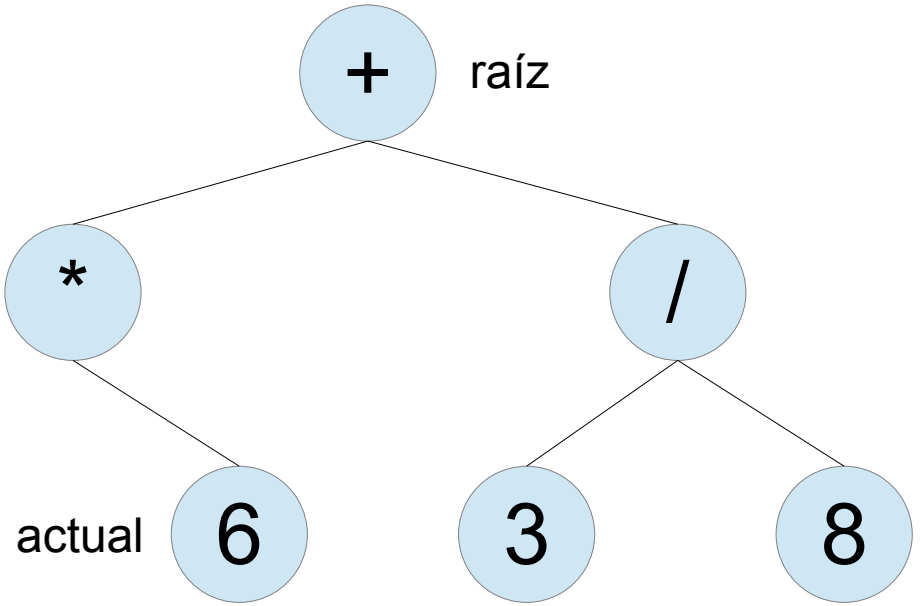
Árbol de Expresión

- Ejemplo expresión posfija: 2 6 * 3 8 / +



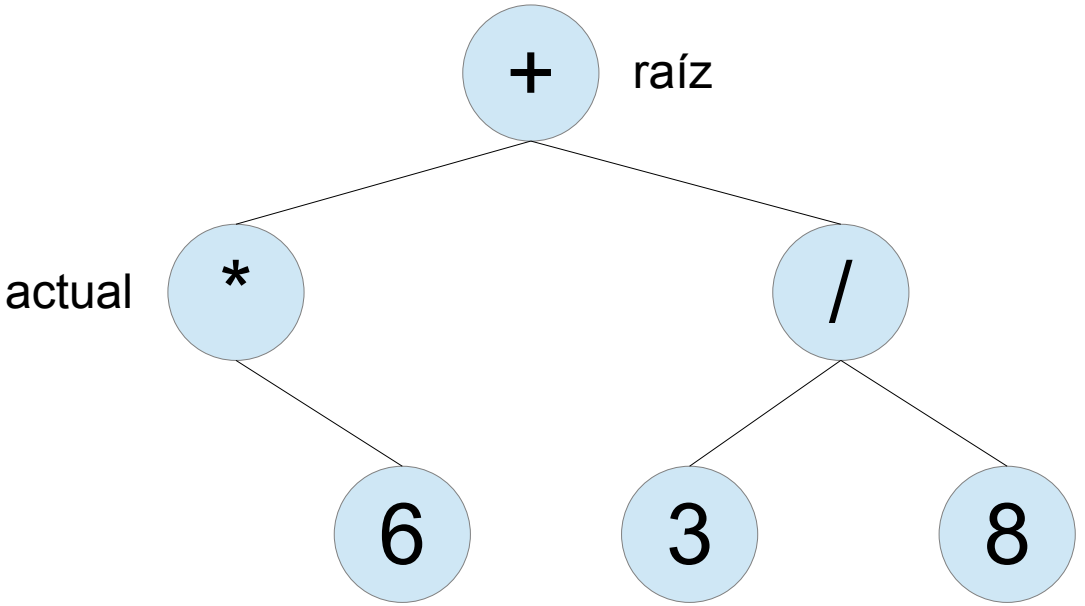
Árbol de Expresión

- Ejemplo expresión posfija: 2 6 * 3 8 / +



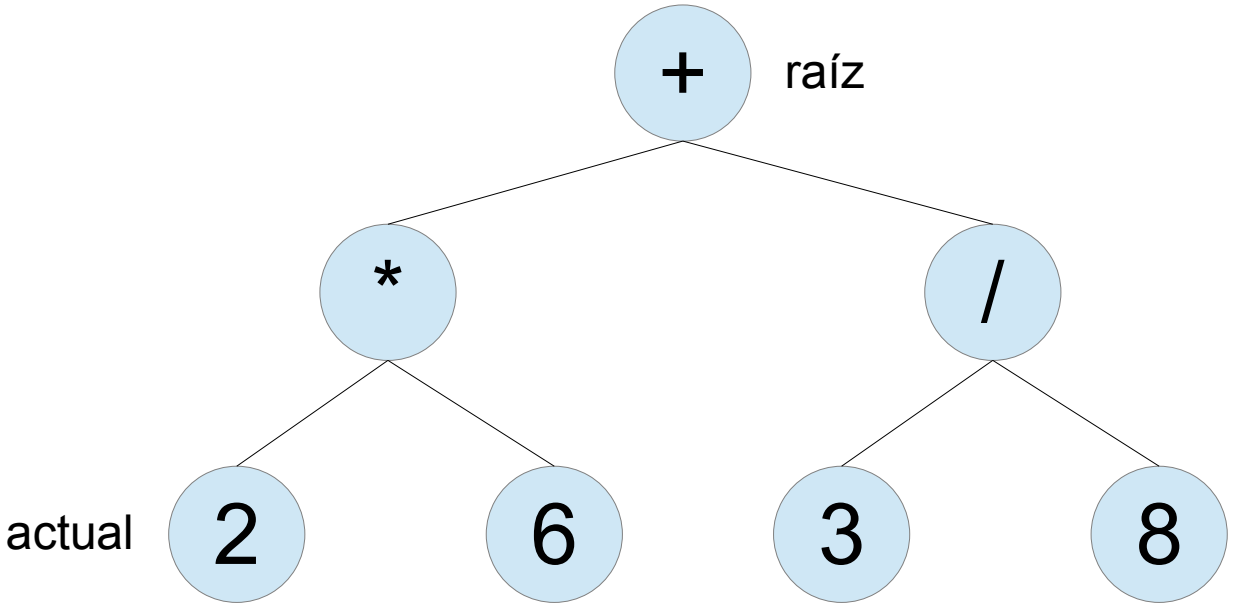
Árbol de Expresión

- Ejemplo expresión posfija: 2 6 * 3 8 / +



Árbol de Expresión

- Ejemplo expresión posfija: 2 6 * 3 8 / +



Árbol de Expresión

- Generación del árbol a partir de la expresión

  ¿Expresión prefija?

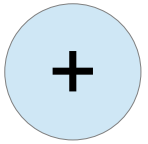
  - Empezar por el extremo izquierdo de la expresión.
  - Ese elemento debe ser un operador, el cual se ubica como la raíz del árbol.
  - Para cada elemento siguiente (hacia la derecha):
    - Si el nodo actual es operador, poner el elemento como hijo izquierdo, si ya hay hijo izquierdo, entonces como hijo derecho.
    - Si el nodo actual es operando, regresar tantos ancestros como sea necesario para insertar el elemento ahí.

Árbol de Expresión

- Ejemplo expresión prefija: + * 2 6 / 3 8

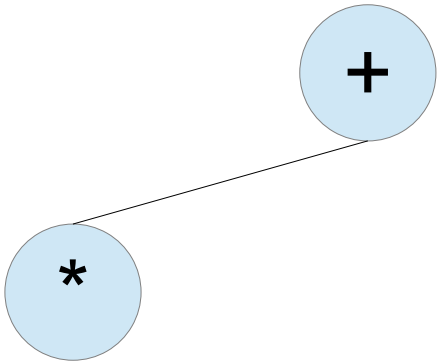
Árbol de Expresión

- Ejemplo expresión prefija: + * 2 6 / 3 8



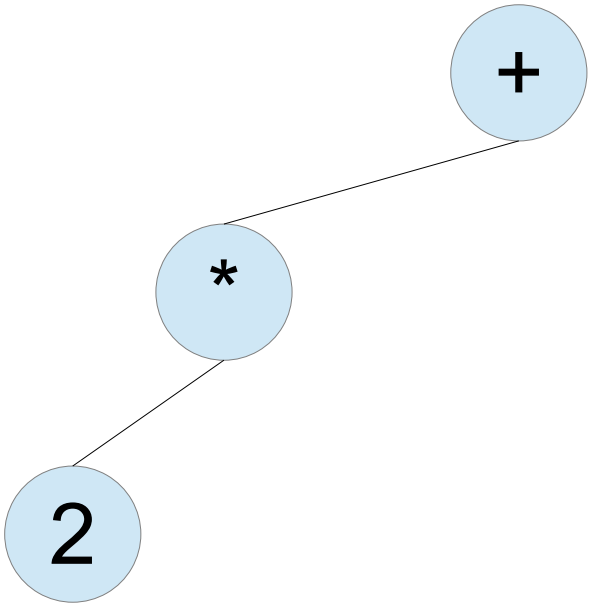
Árbol de Expresión

- Ejemplo expresión prefija: + * 2 6 / 3 8



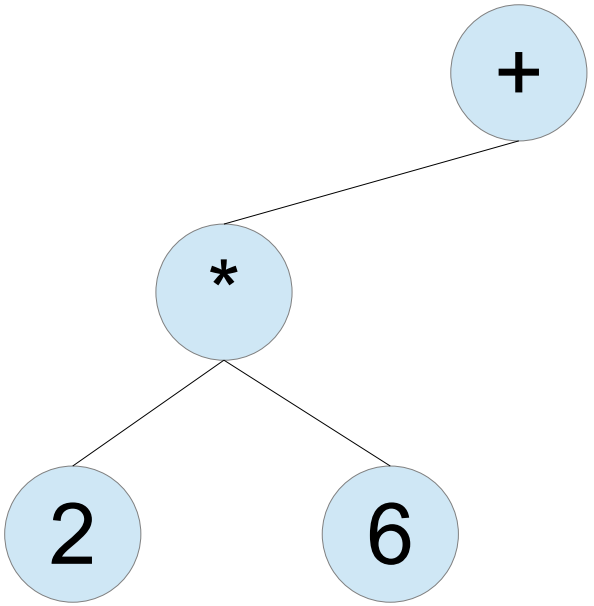
Árbol de Expresión

- Ejemplo expresión prefija: + * 2 6 / 3 8



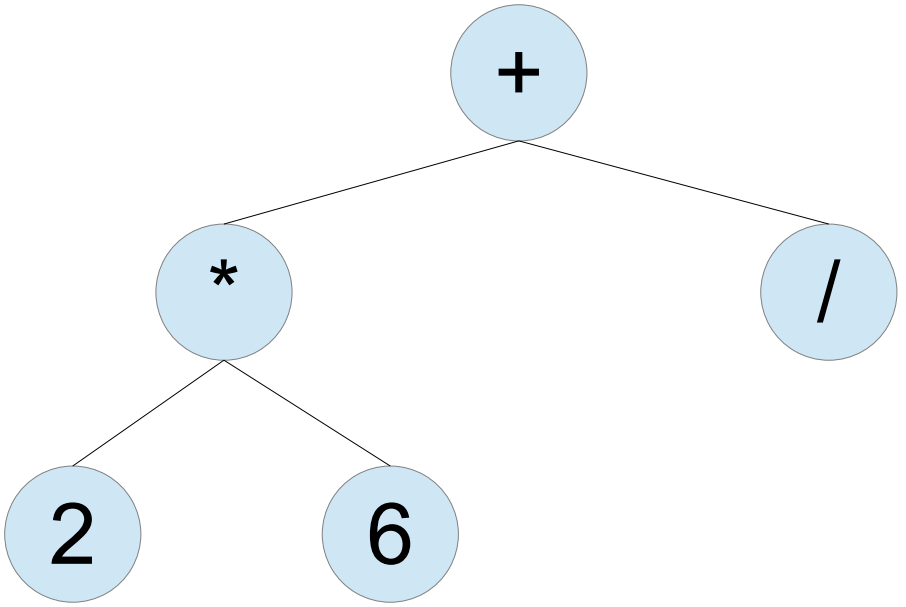
Árbol de Expresión

- Ejemplo expresión prefija: + * 2 6 / 3 8



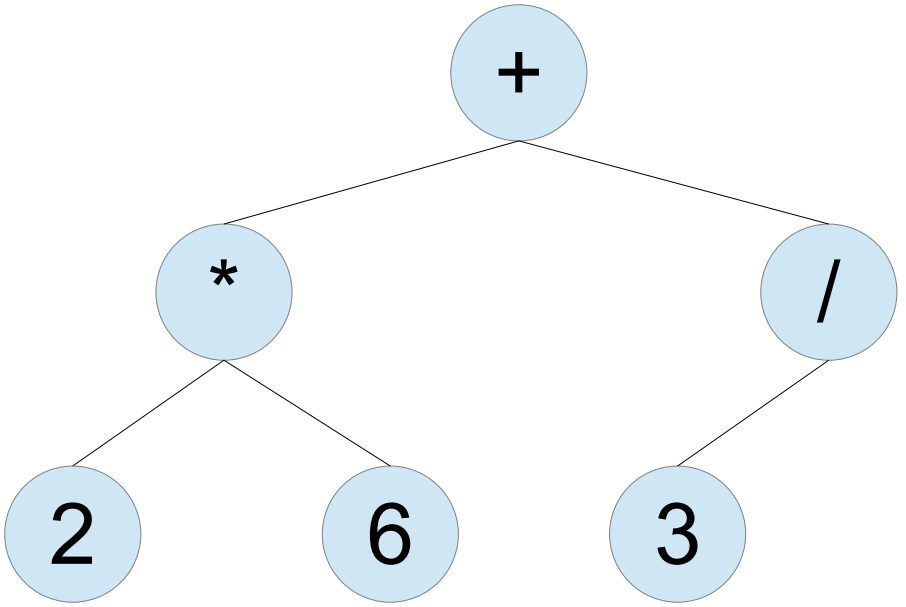
Árbol de Expresión

- Ejemplo expresión prefija: + * 2 6 / 3 8



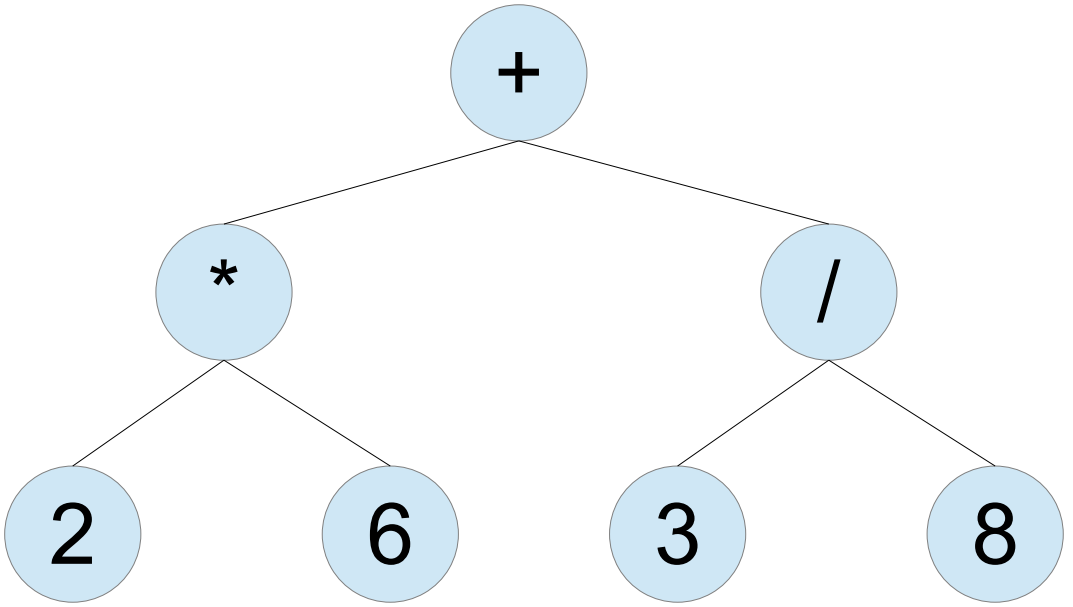
Árbol de Expresión

- Ejemplo expresión prefija: + * 2 6 / 3 8



Árbol de Expresión

- Ejemplo expresión prefija: + * 2 6 / 3 8



Árbol de Expresión

- Evaluación de expresiones:

  Encontrar el valor numérico resultado de la expresión.

  Implica:

  - Sustituir variables (si las hay) por valores numéricos.

  - Aplicar las operaciones en el orden dado por la expresión (precedencia de paréntesis u orden dado por prefija o posfija).

Árbol de Expresión

- Evaluación de expresiones:

  ¿Notación infija?

  ¿Notación posfija?

  ¿Notación prefija?

Árbol de Expresión

- Evaluación de expresiones.

  Notación posfija:

  - Evaluar la expresión de izquierda a derecha.
  - Si el elemento es un operando, insertar en la pila.
  - Si el elemento es un operador, extraer los dos operandos de la pila, aplicar el operador, e insertar el resultado en la pila.
  - Al final, retornar el elemento en el tope de la pila como resultado de la expresión.

Árbol de Expresión

- Evaluación de expresiones.

  Notación prefija:

  - Evaluar la expresión de derecha a izquierda.
  - Si el elemento es un operando, insertar en la pila.
  - Si el elemento es un operador, extraer los dos operandos de la pila, aplicar el operador, e insertar el resultado en la pila.
  - Al final, retornar el elemento en el tope de la pila como resultado de la expresión.

Árbol de Codificación

Árbol de Codificación

- En los textos (en español) hay algunas palabras que se repiten más que otras. ¿Cómo representar estos textos en forma óptima?

- Árbol binario (ordenado) óptimo: estructura recurrente donde los elementos que se buscan "más seguido" se ubican más cerca de la raíz.

Árbol de Codificación

- ¿Cómo modelar elementos repetidos? Histograma, representa la probabilidad (o frecuencia) de la búsqueda (o presencia) de un elemento.

- Ejemplo: histograma de “dabale arroz a la zorra el abad”

a	b	d	e	l	o	r	z
8	2	2	2	3	2	4	2

Árbol de Codificación

- Codificación:
  Representar cada elemento (carácter) con una cadena binaria única.

- Ejemplo: archivo con 100.000 caracteres

	a	b	c	d	e	f
Frecuencia (miles)	45	13	12	16	9	5
Código de longitud fija	000	001	010	011	100	101
Código de longitud variable	0	101	100	111	1101	1100

Árbol de Codificación

- Codificación:
  Concatenación de códigos binarios.
  abc → 0 101 100 → 0101100

- Códigos prefijo:
  Ningún código es prefijo de otro.
  Simplifican la decodificación, no es ambigua.

  001011101 → 0 0 101 1101 → aabe

Árbol de Codificación

- Árbol de codificación:

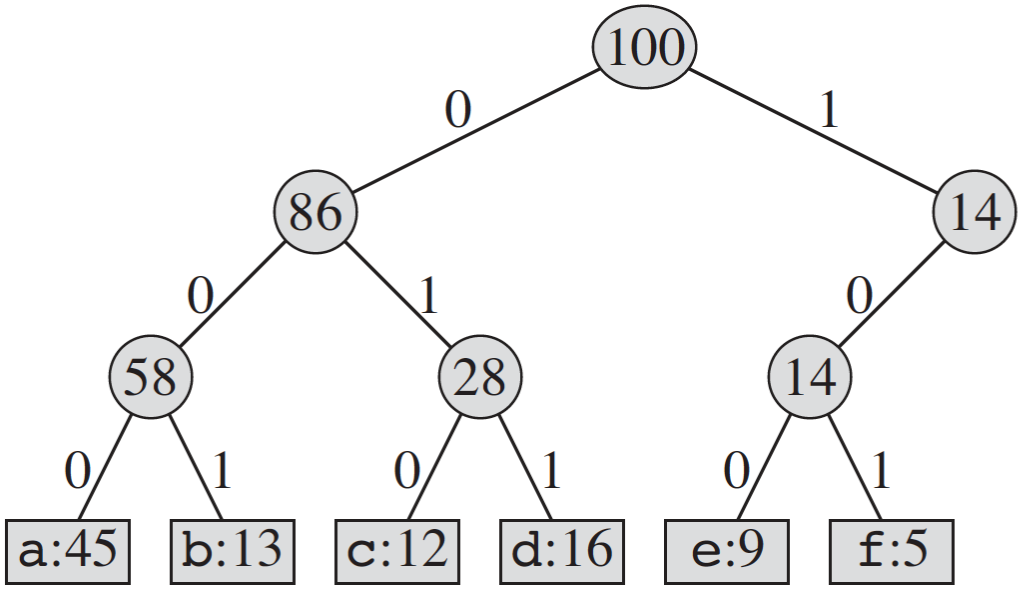
  - árbol binario.

  - caracteres a codificar en las hojas.

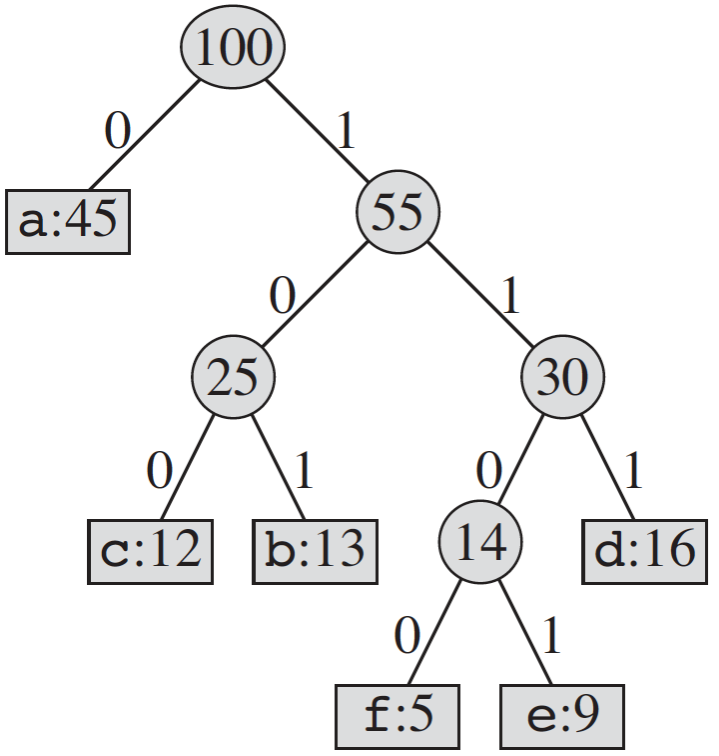
  - raíz y nodos internos almacenan la suma de frecuencias de sus hijos.

  - código: ruta desde la raíz hasta cada hoja, 0 representa hijo izquierdo y 1 representa hijo derecho.

Árbol de Codificación



Árbol de codificación para el código de longitud fija



Árbol de codificación para el código de longitud variable

Árbol de Codificación

- Código de Huffman: código prefijo óptimo.
- Árbol de Huffman: almacena un código de Huffman.
- ¿Cómo construírlo?

  - C un conjunto de caracteres

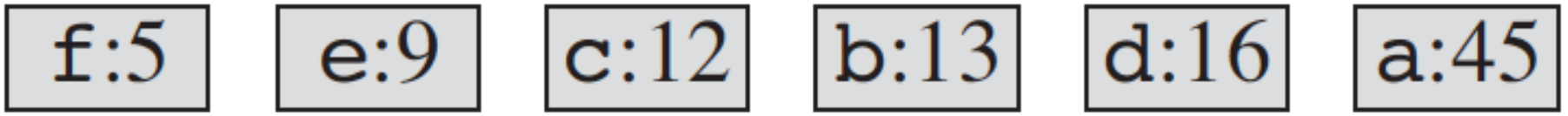
  - Q una cola de prioridad mínima (se extrae siempre el mínimo)

Árbol de Codificación

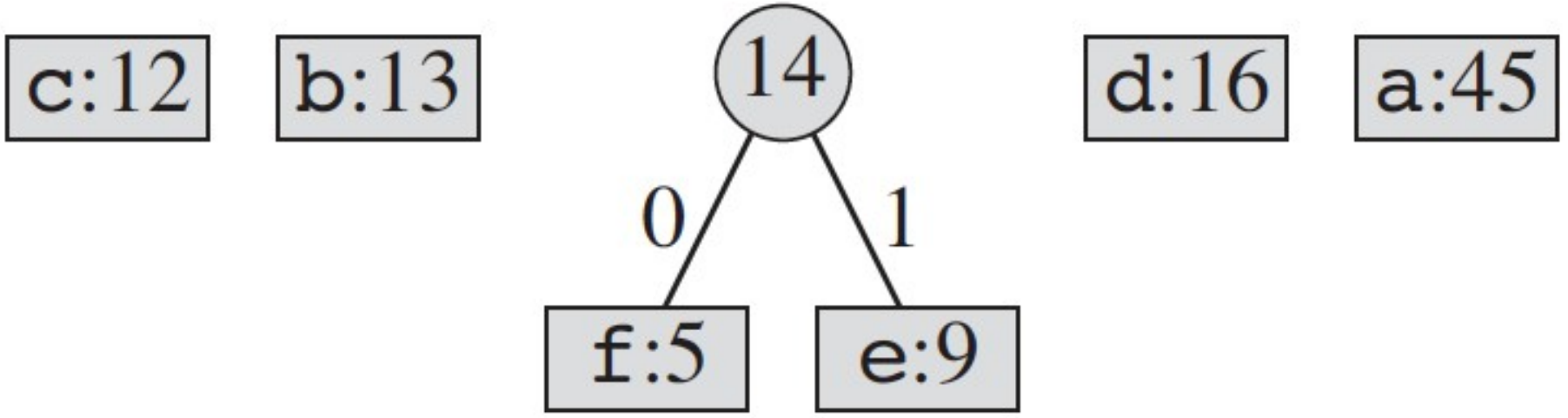
- Pseudocódigo

```
Huffman(C):
  n = |C|
  Q = C
  para i=1 to n-1
    crear nuevo nodo z
    z.hijoIzq = extraer_min(Q)
    z.hijoDer = extraer_min(Q)
    z.frecuencia =
                    frecuencia(z.hijoIzq) +
                    frecuencia(z.hijoDer)
    insertar(Q,z)
  retornar extraer_min(Q) // raíz
```

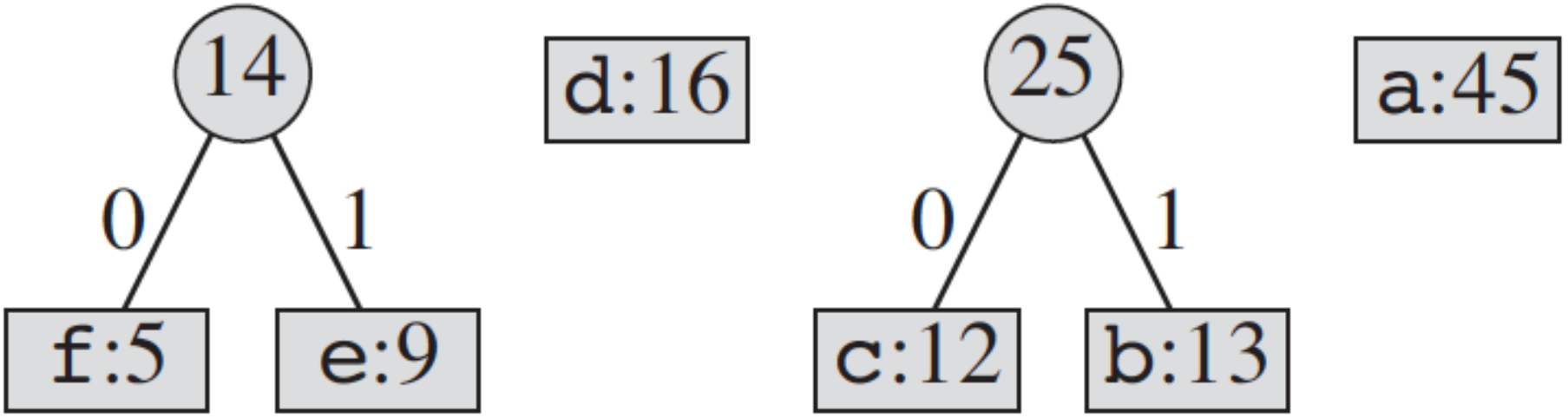
Árbol de Codificación



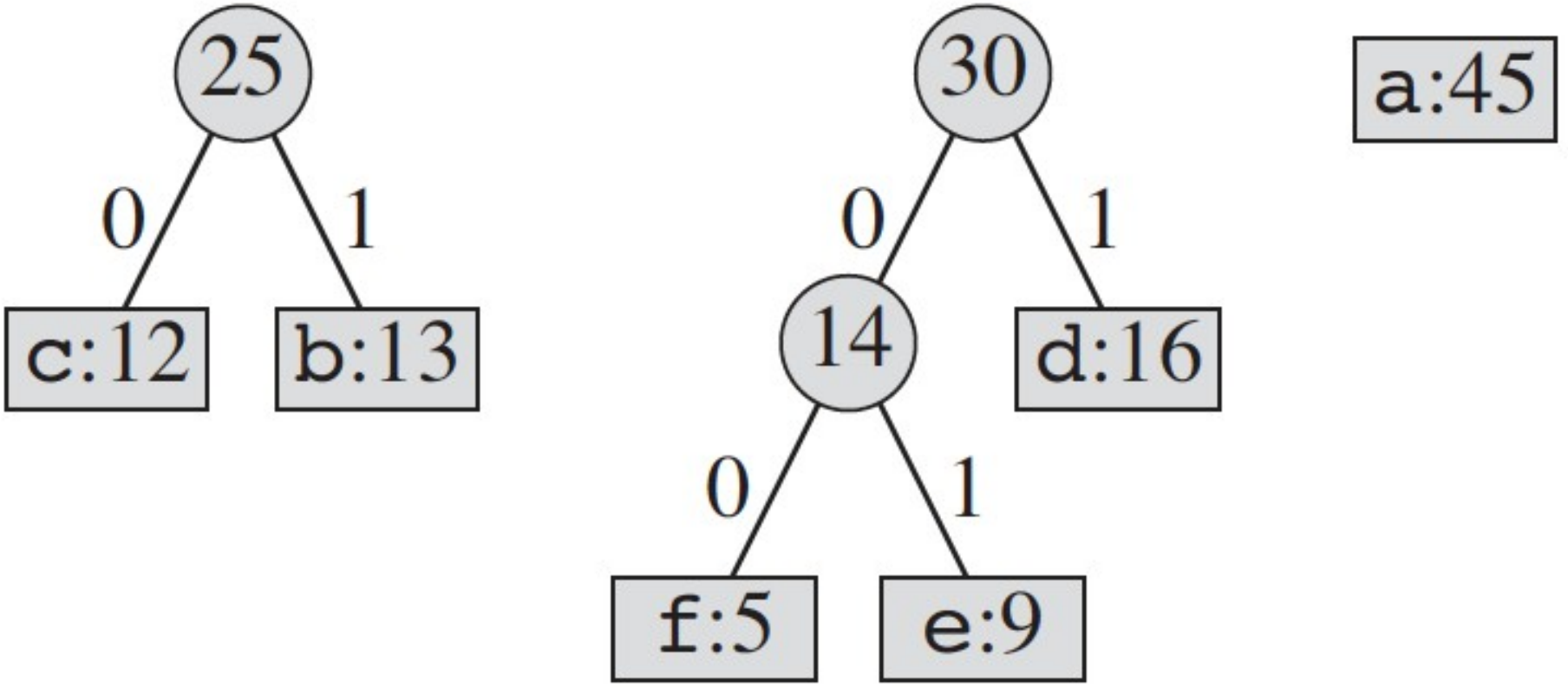
Árbol de Codificación



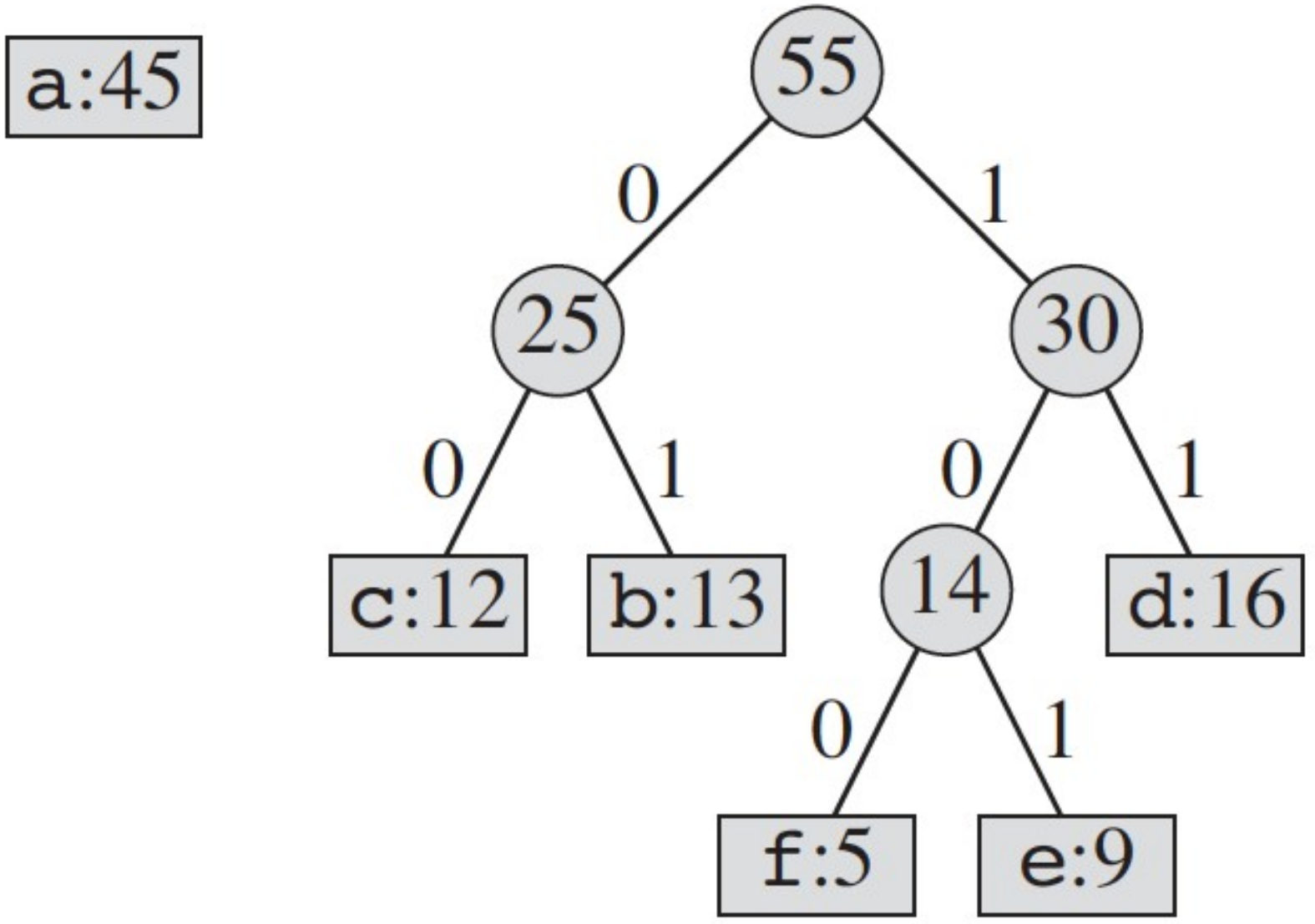
Árbol de Codificación



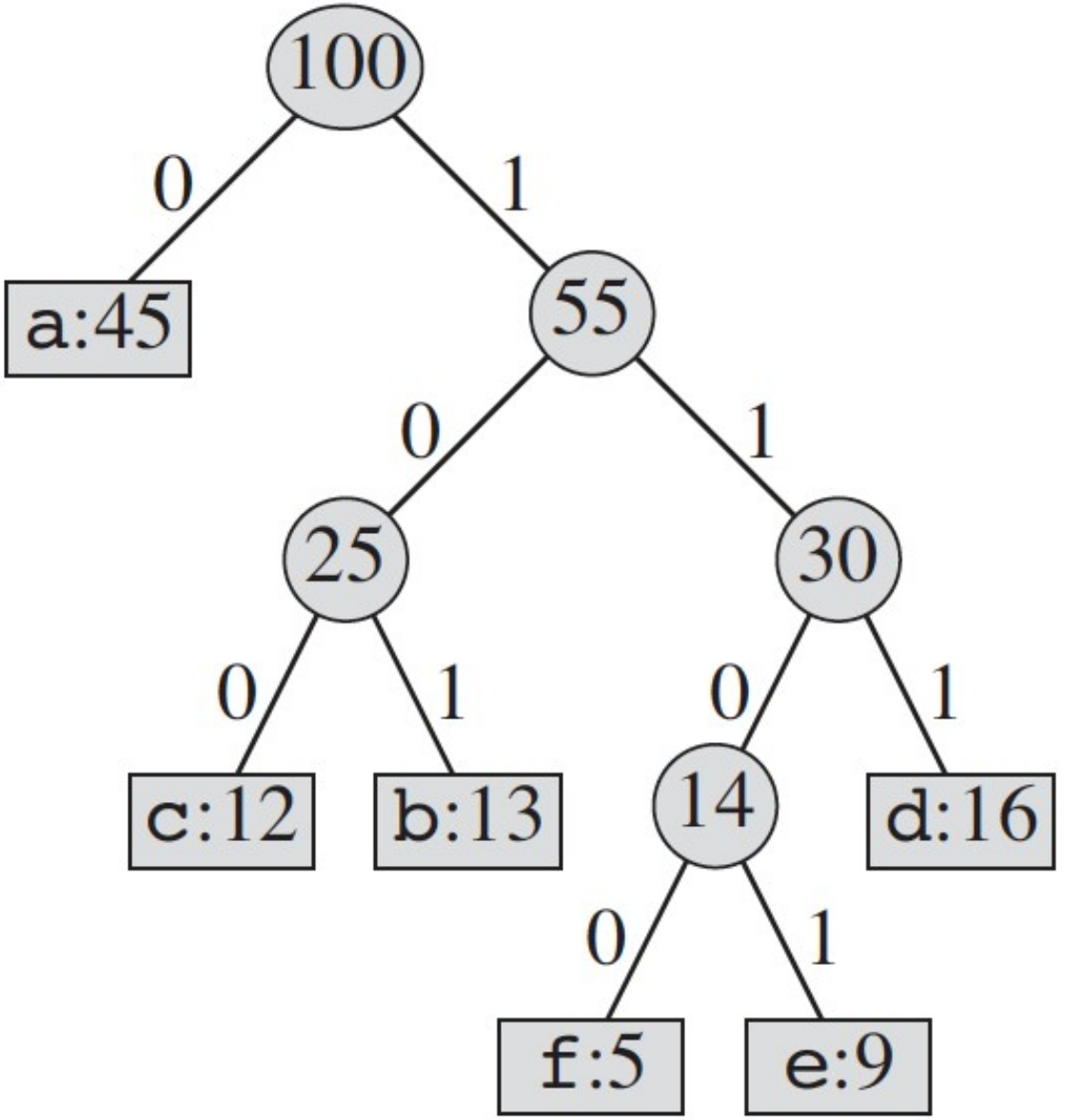
Árbol de Codificación



Árbol de Codificación



Árbol de Codificación



Referencias

- en.wikipedia.org/wiki/Abstract_syntax_tree
- en.wikipedia.org/wiki/Binary_expression_tree
- en.wikipedia.org/wiki/Polish_notation
- en.wikipedia.org/wiki/Reverse_Polish_notation
- Cormen, Leiserson, Rivest y Stein. Introduction to Algorithms, third edition. The MIT Press, 2009.