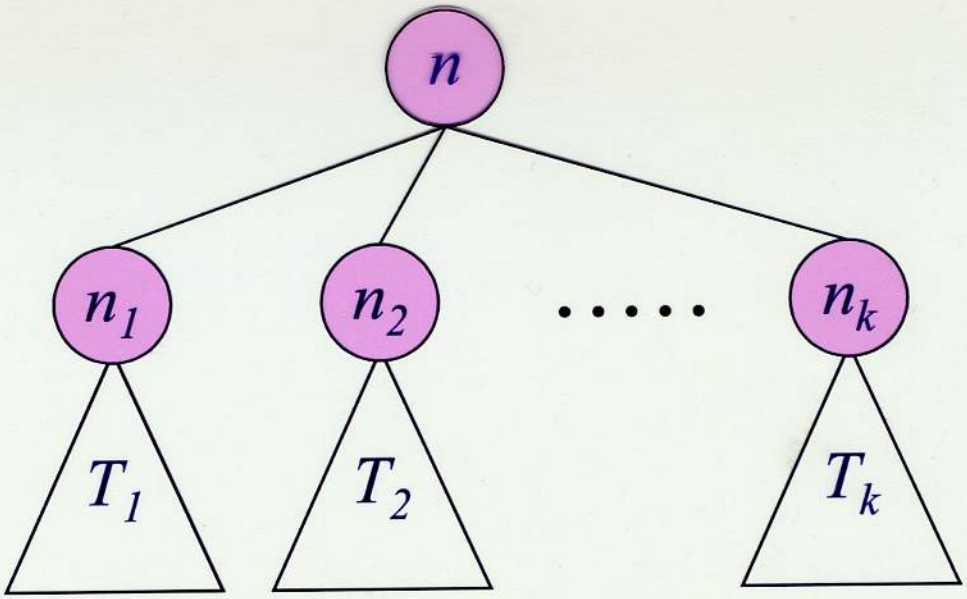
Conceptos sobre árboles

Árbol n-ario

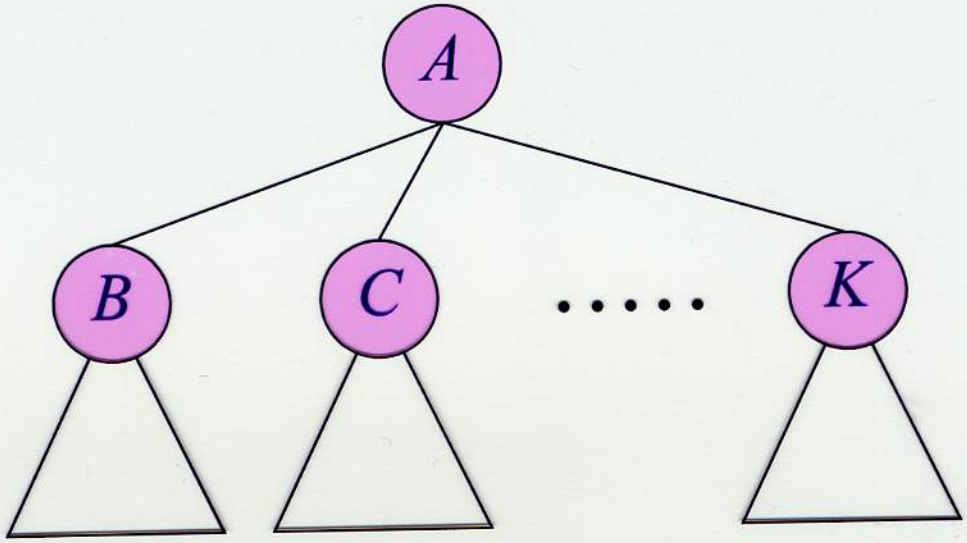
Base: Un nodo es un árbol n-ario (si un árbol tiene un solo nodo, éste es el nodo raíz).

Recurrencia: Si n es un nodo y T_1, ..., T_k son árboles n-arios con raíces n_1, ..., n_k respectivamente, entonces podemos construir un árbol que tenga como raíz el nodo n y como subárboles T_1, ..., T_k.



Árbol etiquetado

Se dice que un árbol está **etiquetado** si todos los nodos contienen una etiqueta.



Nodos hermanos

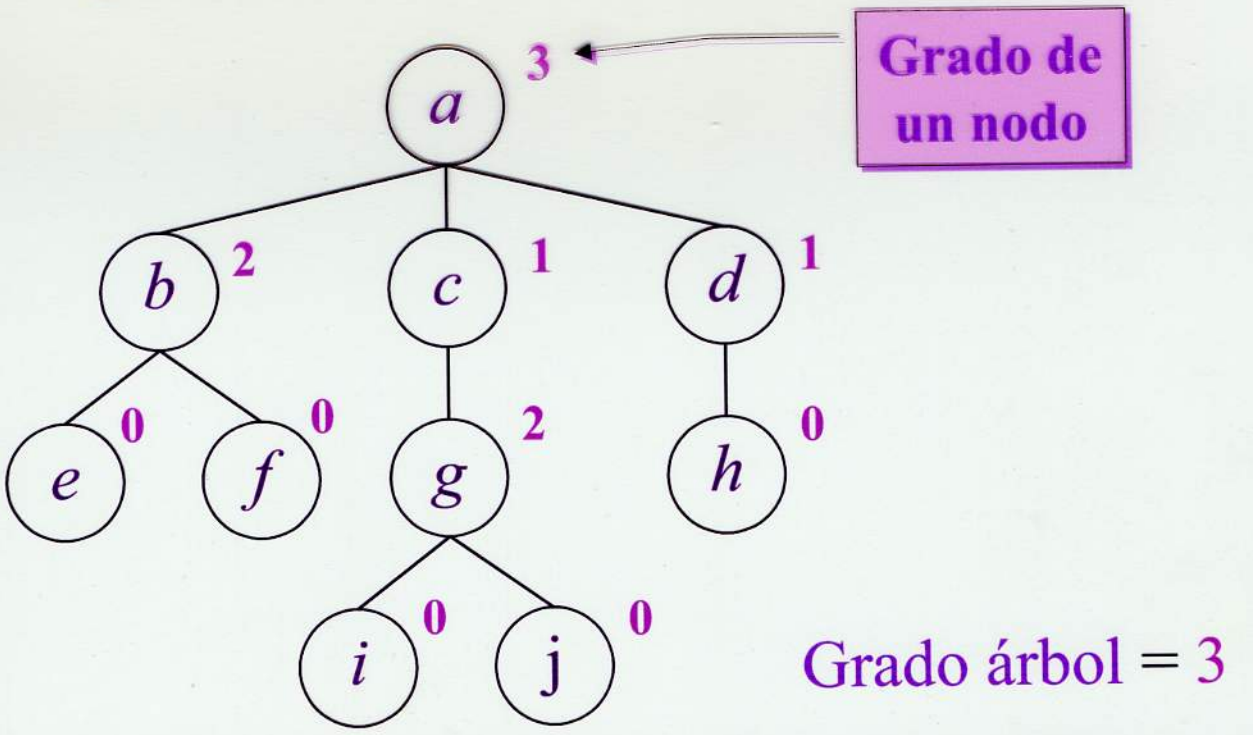
A los nodos que son hijos de un mismo padre se les denomina **hermanos**.

Grado de un nodo

Se llama **grado de un nodo** al número de subárboles que tiene dicho nodo. Los nodos de grado 0 se denominan **hojas o nodos terminales**. El resto se llaman **nodos no terminales o interiores**.

Grado de un árbol

El **grado de un árbol** es el máximo de los grados de los nodos del árbol.

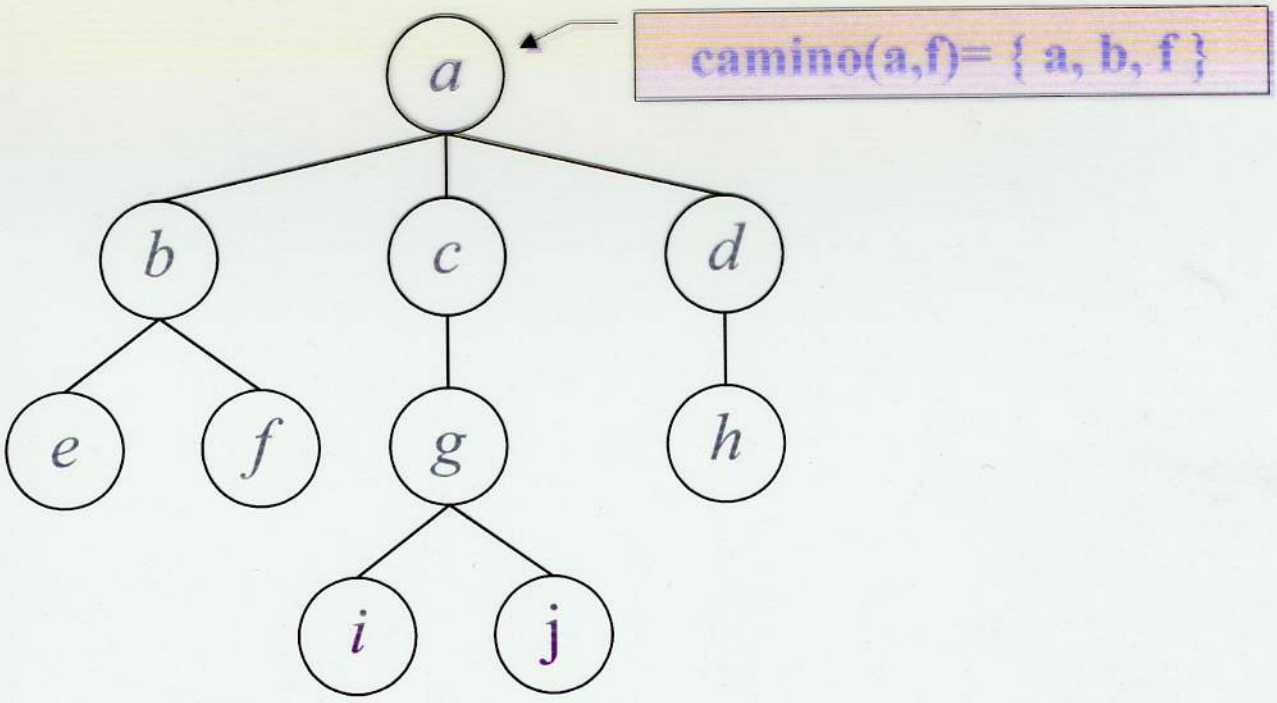


Camino entre dos nodos

El **camino entre dos nodos** n_i, n_j se define como la secuencia de nodos del árbol necesaria para alcanzar el nodo n_j desde el n_i.

Longitud de un camino

La **longitud del camino entre dos nodos** es igual al nº de nodos que forman este camino menos 1 (nº de ejes que hay en el camino).



Nivel de un nodo

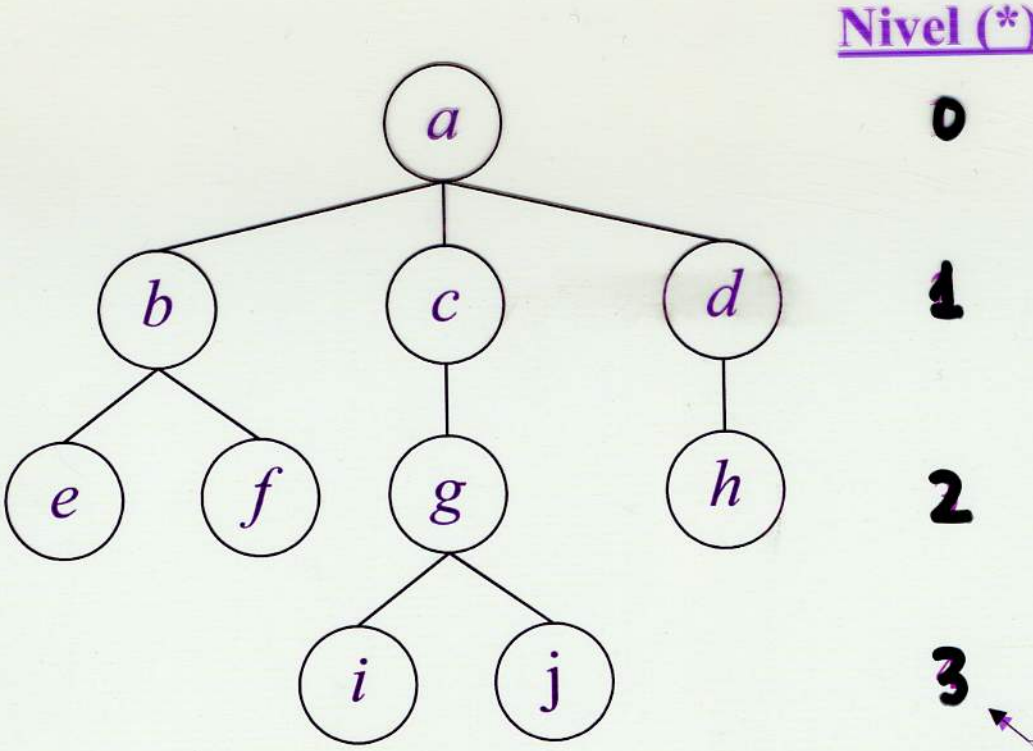
Simplicidad computacional (*)

Base: El nivel del nodo raíz es 0.

Recurrencia: Si el nodo n está en el nivel i, entonces todos sus hijos están en el nivel $i+1$.

Altura de un árbol

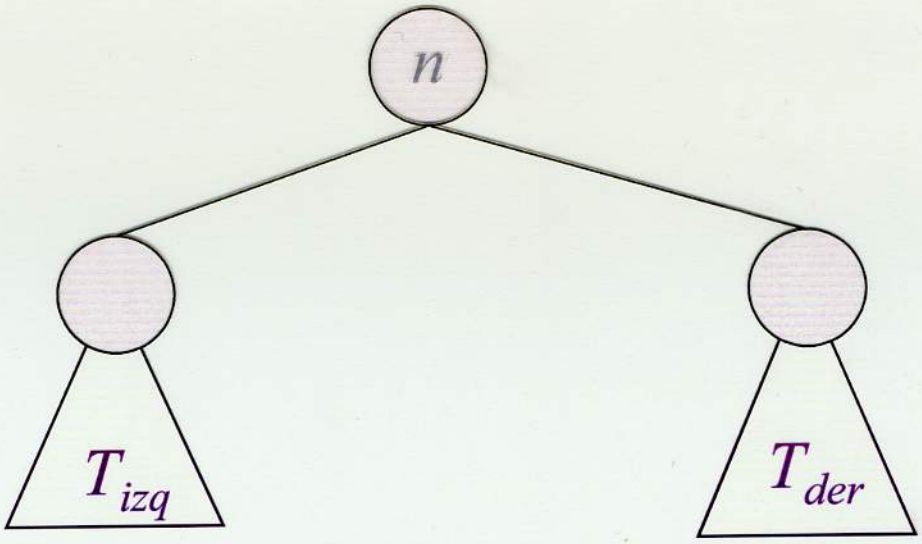
La [redacted] de un árbol es el máximo de los niveles de los nodos del árbol.



Árbol binario

Base: Un árbol vacío es un árbol binario

Recurrencia: Si n es un nodo y T_{izq}, T_{der} son árboles binarios, entonces podemos construir un nuevo árbol binario que tenga como raíz el nodo n y como subárboles T_{izq} y T_{der} (subárbol izquierdo y subárbol derecho de n, respectivamente).



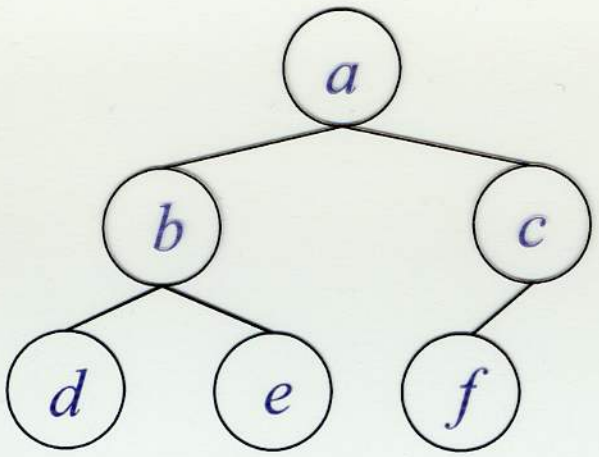
¡ Un árbol binario NO es un árbol n-ario de grado 2 !

Árbol binario homogéneo

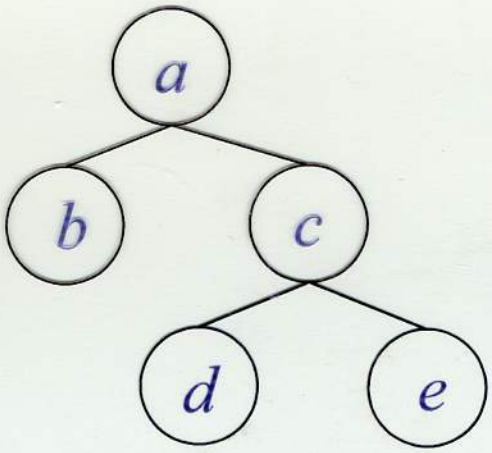
Es aquel cuyos nodos tienen grado 0 ó 2 (no hay ninguno de grado 1).

Árbol binario completo

Es aquel que tiene todos los niveles llenos excepto, quizás, el último en cuyo caso los huecos deben quedar a la derecha.



Completo

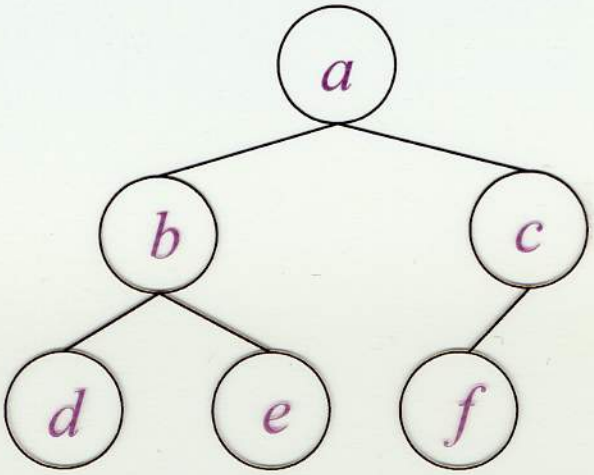


Homogéneo

En un árbol **binario completo** con ***n*** nodos, el camino más largo desde la raíz a las hojas no atraviesa más de ***log n*** nodos.

Número máximo de nodos por nivel

En un árbol binario el número máximo de nodos que puede haber en el nivel i es 2^i.



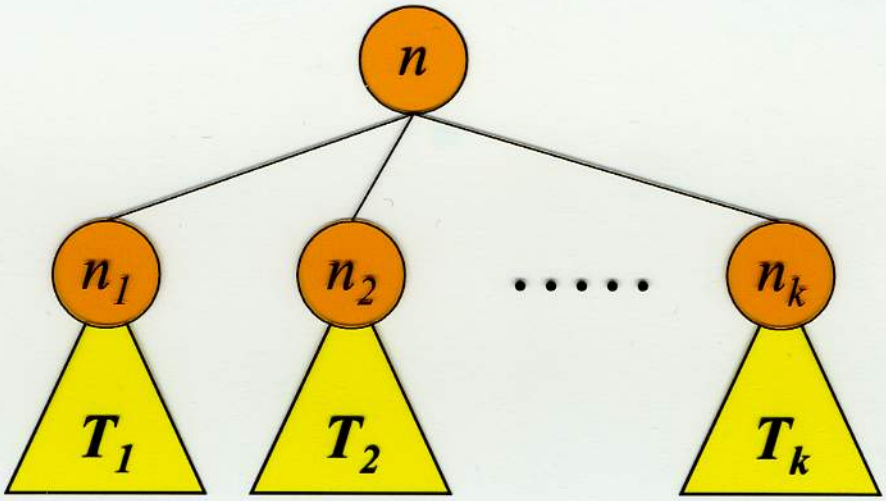
Nivel	Nodos
0	$2^0 = 1$
1	$2^1 = 2$
2	$2^2 = 4$

En un árbol **binario completo** con altura ***k***, el número máximo de nodos es 2^k-1 nodos.

Esquemas de recorrido en árboles n-arios

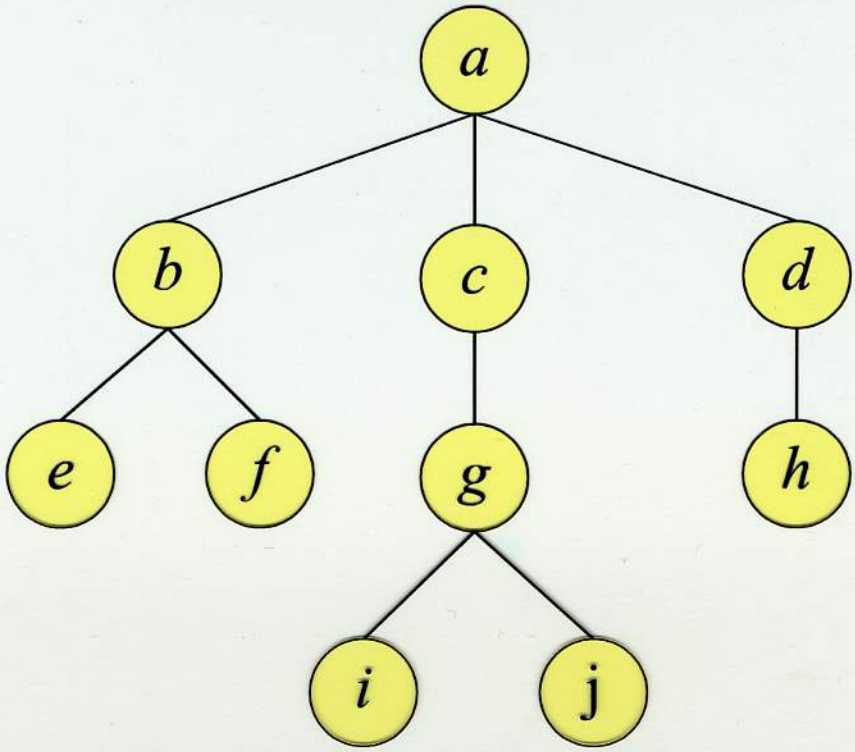
Recorridos en profundidad:

- **Preorden**: raíz, Pre(T1), Pre(T2), …, Pre(Tk)
- **Inorden**: In(T1), raíz, In(T2), In(T3),…, In(Tk)
- **Postorden**: Post(T1), Post(T2), …, Post(Tk), raíz



Recorrido en anchura:

- **Por niveles**: De arriba a abajo y de izquierda a derecha, empezando en la raíz.



Recorrido en preorden: ***a b e f c g i j d h***

Recorrido en inorden: ***e b f a i g j c h d***

Recorrido en postorden: ***e f b i j g c h d a***

Recorrido por niveles: ***a b c d e f g h i j***

Recorrido de árboles binarios

Existen cuatro formas posibles de recorrer todos los nodos de un árbol:

- Recorrido en **preorden**
- Recorrido en **inorden**
- Recorrido en **postorden**

En profundidad

- Recorrido **por niveles** de izquierda a derecha

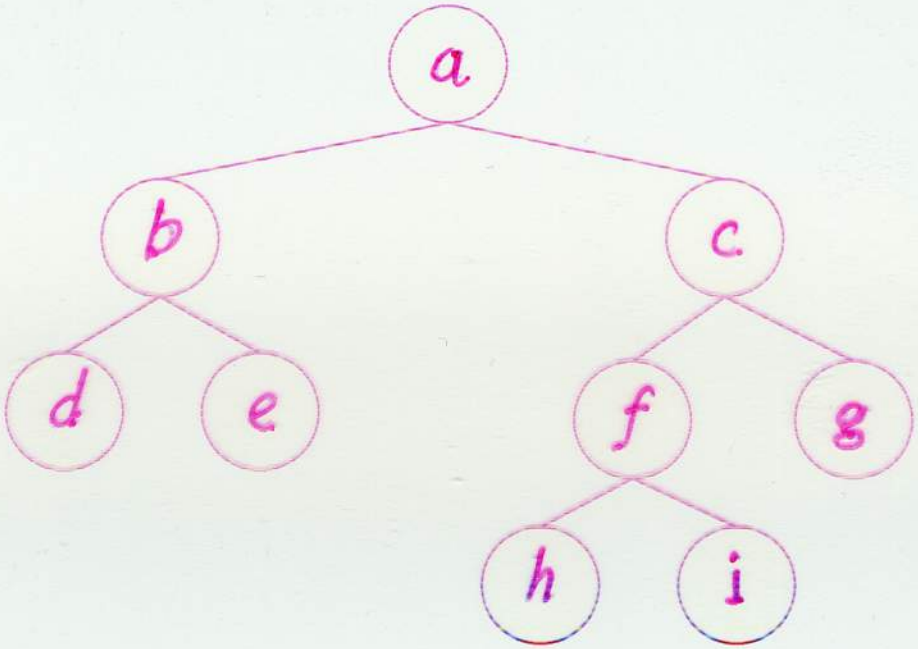
En anchura

Los tres primeros se pueden realizar de forma recursiva, aplicando el esquema de construcción de funciones recursivas para árboles binarios.

Por el contrario, el recorrido por niveles se realiza de forma iterativa.

Descripción de los recorridos en profundidad:

- **Preorden:** raíz Preorden(izq) Preorden(der)
- **Inorden:** Inorden(izq) raíz Inorden(der)
- **Postorden:** Postorden(izq) Postorden(der) raíz

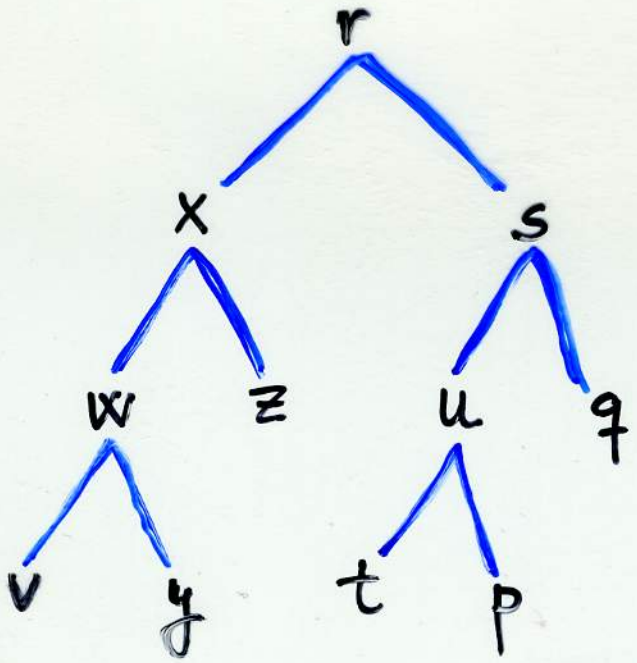


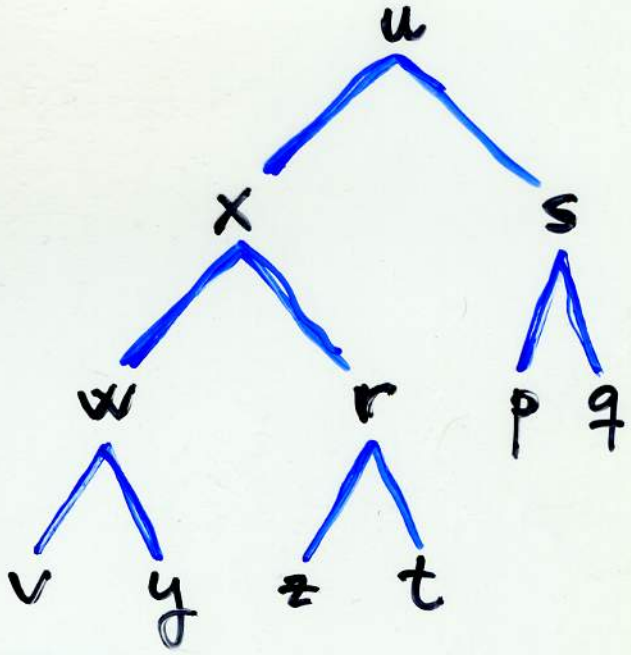
- **Preorden:** a b d e c f h i g
- **Inorden:** d b e a h f i c g
- **Postorden:** d e b h i f g c a
- **Por niveles:** a b c d e f g h i

	Pre(n) < Pre(m)	In(n) < In(m)	Post(n) < Post(m)
n regla de m			
n derda de m			
n desc de m			
n ancestro de m			

Pre: G, E, A, I, B, M, C, L, D, F, K, J, H

In: I, A, B, E, G, L, D, C, F, M, K, H, J





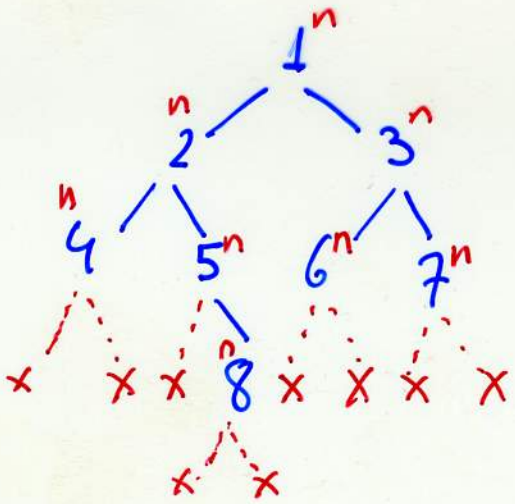
INORDEN: v, w, y, x, z, r, t, u, p, s, q

INORDEN: v, w, y, x, z, r, t, u, p, s, q

[En general, un árbol no puede recuperarse con uno solo de sus recorridos.]

Lectura/escritura de un árbol

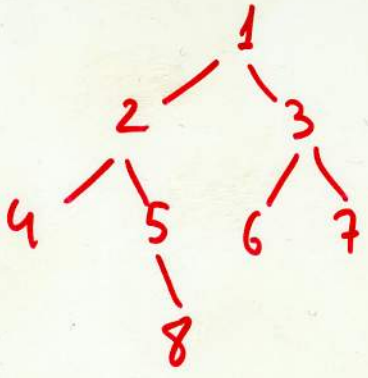
Preorden



n 1 n 2 n 4 x x n 5 x n 8 x x n 3 n 6 x x n 7 x x

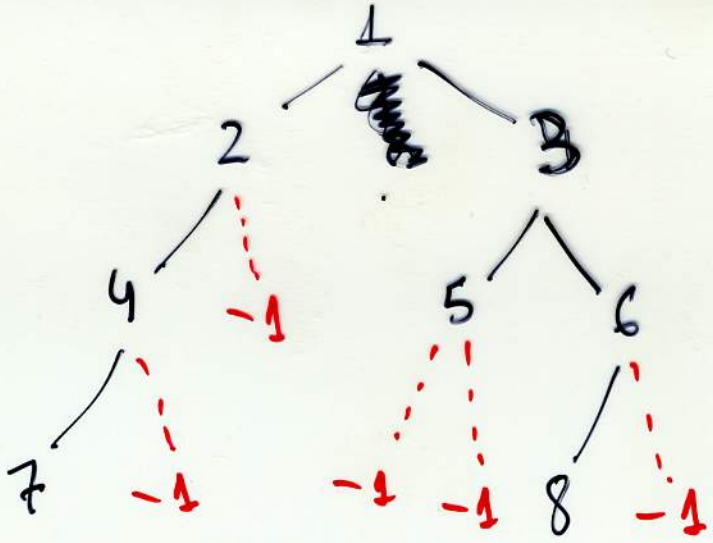
Pre: 1, 2, 4, 5, 8, 3, 6, 7

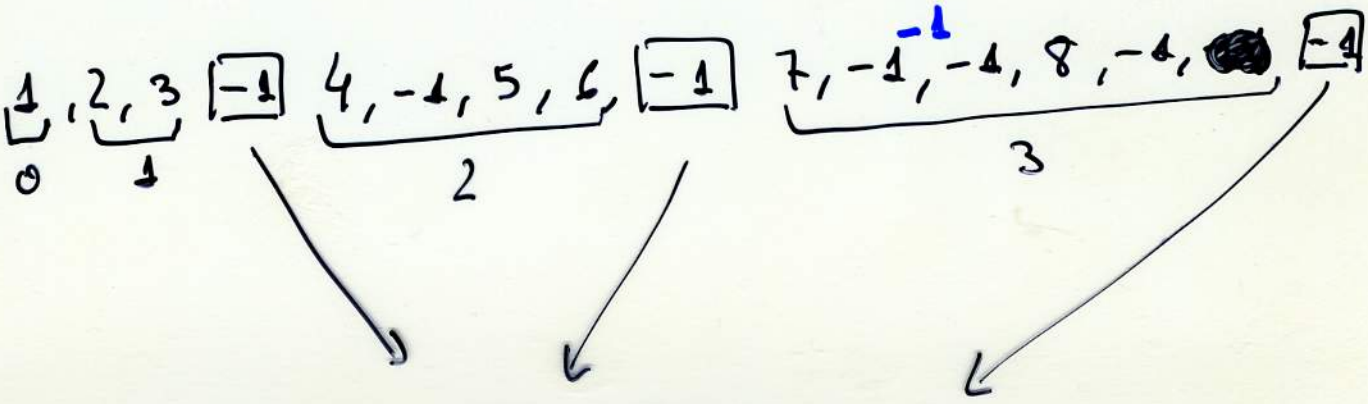
Niveles



1, 2, 3 -1 4, 5, 6, 7 -1 -1 -1 8 -1 -1 -1 -1 -1

fin de distinto hijos



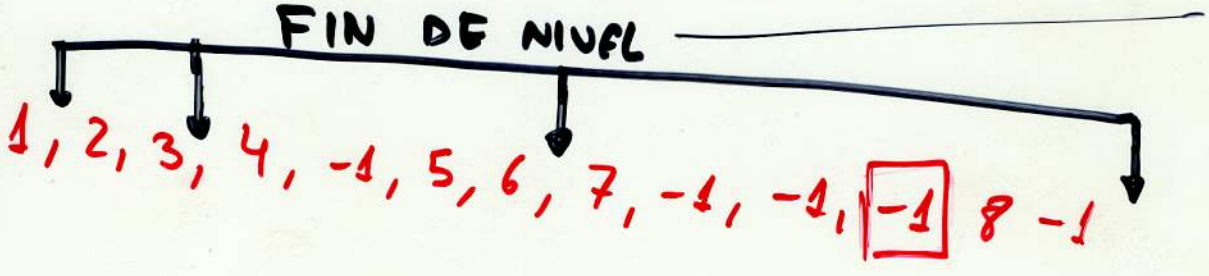


Fin de nivel

Guardar en disco respetando la jerarquía

↓

-1 ≡ CENTINELA

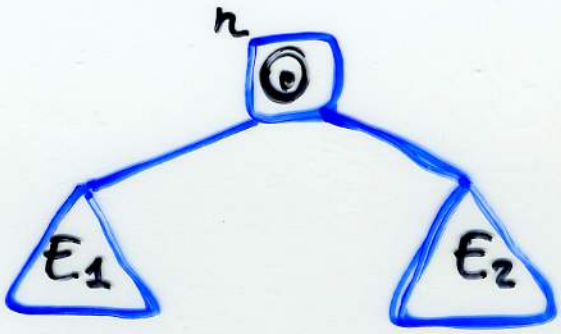


UNA APLICACIÓN: ÁRBOLES DE EXPRESIÓN.

Árboles sintácticos: árboles que contienen las derivaciones de una gramática necesarias para obtener una determinada frase del lenguaje.

Árboles de expresión: etiquetan

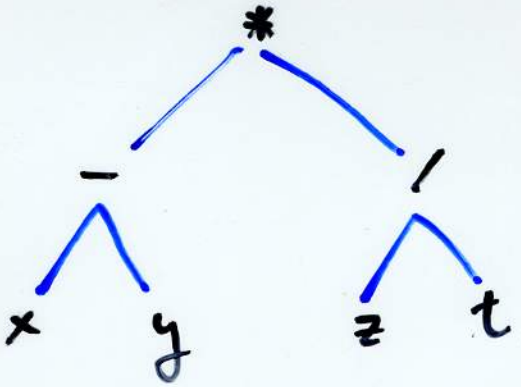
- hojas con un solo operando.
- nodos interiores con un operador.



$(E_1) \odot (E_2)$

$\Downarrow\Downarrow$

$E_1 \odot E_2$



((x)-(y)) * ((z)/(t))

$\Downarrow\Downarrow$

(x-y) * (z/t)

INORDEN

PREORDEN: * - xy / zt → Representación prefija

POSTORDEN: xy - zt / * → Representación Postfija

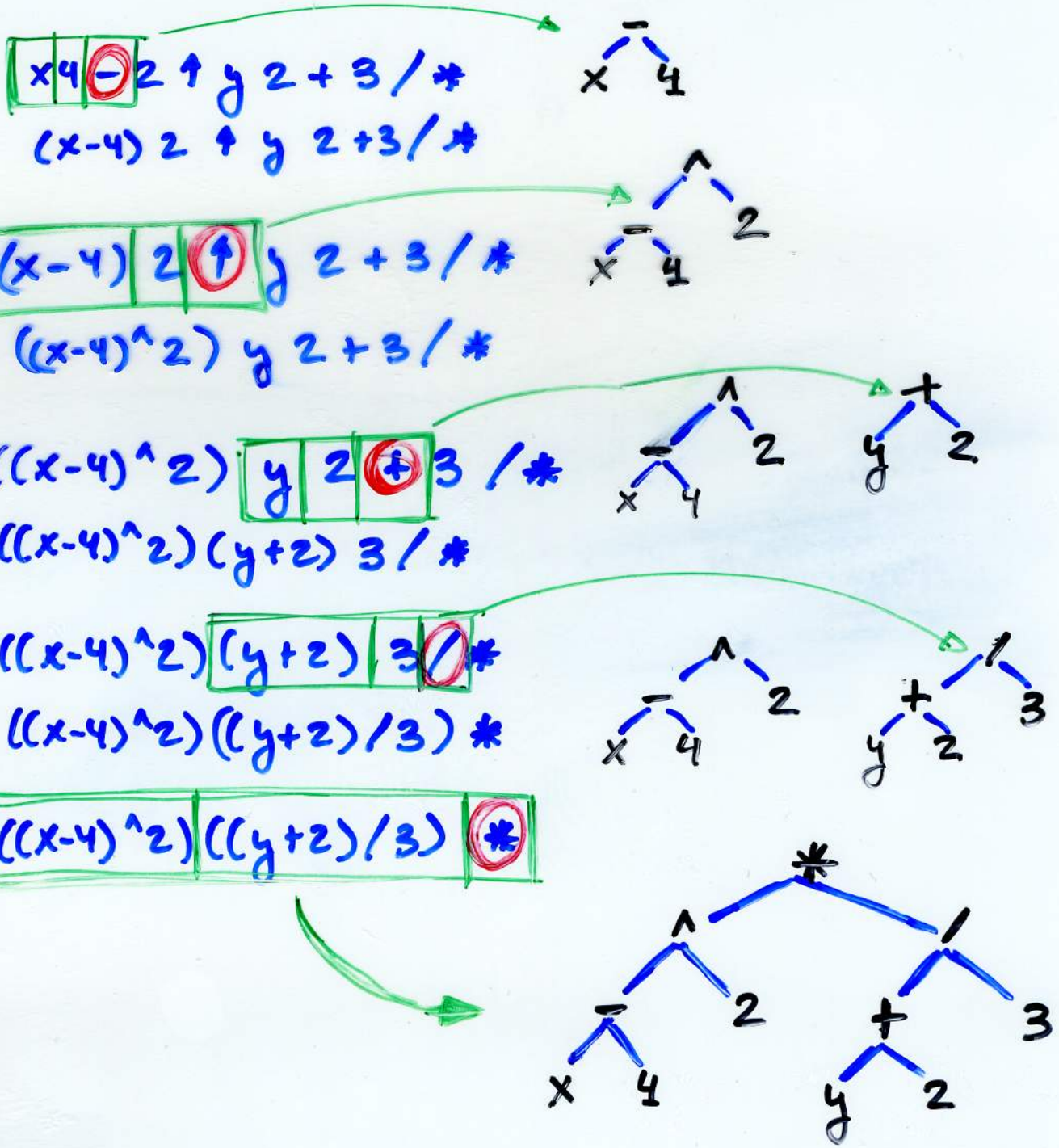
¡NO SE REQUIEREN LOS PARÉNTESIS!

Resolución de ambigüedades:

- * Recorridos preorden o postorden más
- * 1) Nivel de cada nodo, o'
  2) Número de hijos de cada nodo.

P.e.: X4-2↑y2+3/* (POSTFIJO)

Los operadores -, ↑, +, / y * son binarios.



$\underbrace{[(a+b)+(c*(d+e)+f)]}_{E_1} * \underbrace{(g+h)}_{E_2}$

$\boxed{* E_1 E_2}$

$[\underbrace{(a+b)}_{E_{11}} \boxed{+} \underbrace{((*(d+e)+f)}_{E_{12}]}$ $\Big\}$ $* + E_{11} E_{12} E_2$

$E_{11} \equiv +ab \Longrightarrow *++ab\ E_{12}\ E_2$

E_{12} $\underbrace{c*(d+e)}_{E_{121}} + \underbrace{f}_{E_{122}}$ $\Big\}$ $+ E_{121}\ E_{122}$

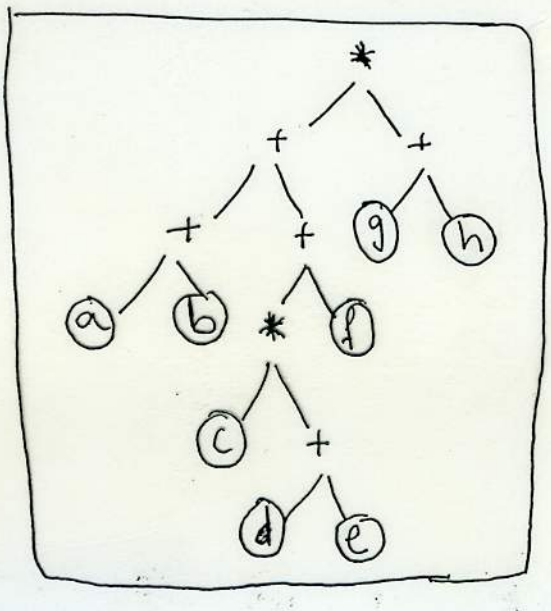
$\Big\}$ $*++ab+E_{121}\ E_{122}\ E_2$

E_{121} $\underbrace{c}_{E_{1211}} * \underbrace{(d+e)}_{E_{1212}} \Rightarrow *\ c\ E_{1212} \equiv *\ c + de$

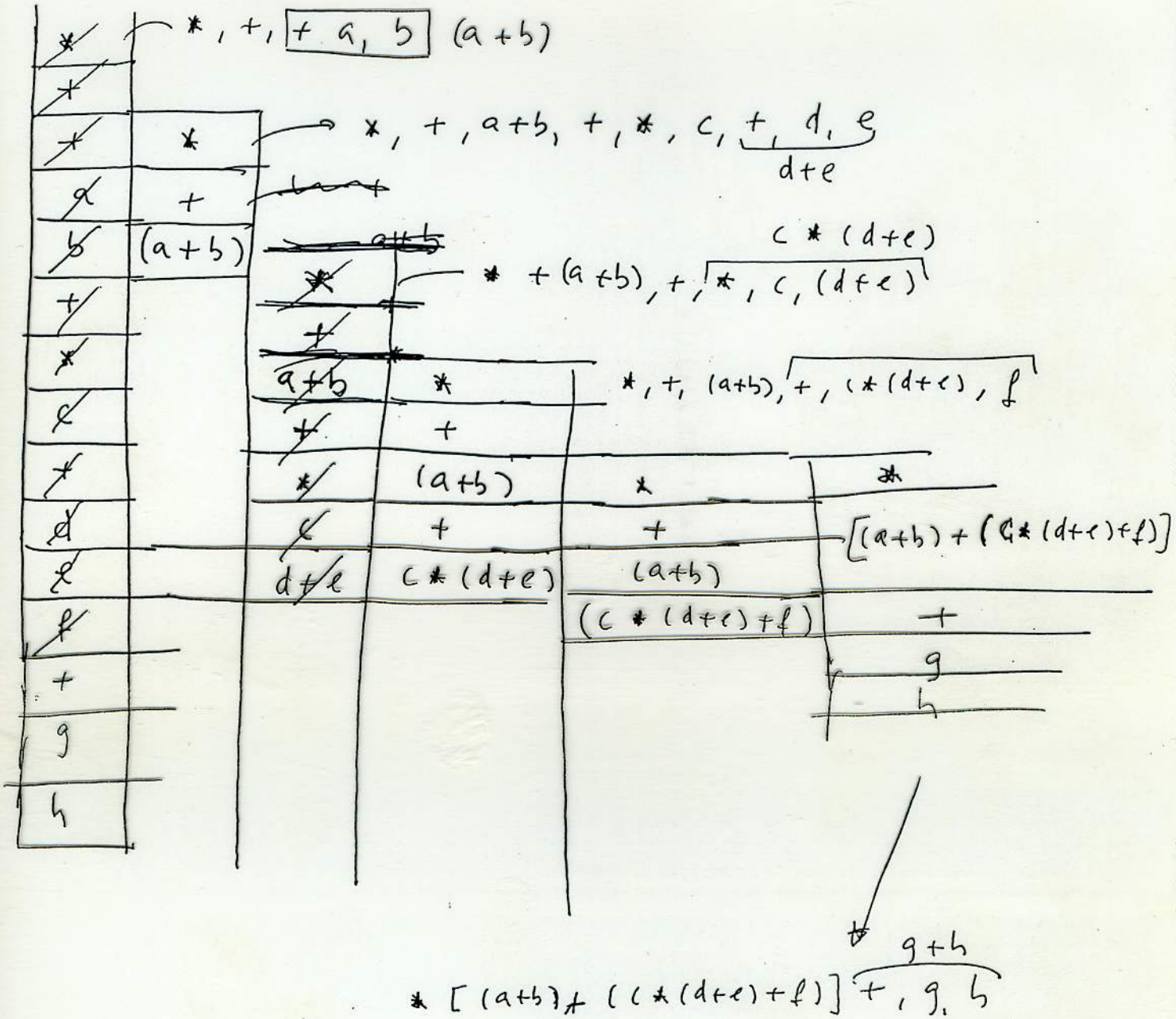
$*++ab+*c+def\ E_2$

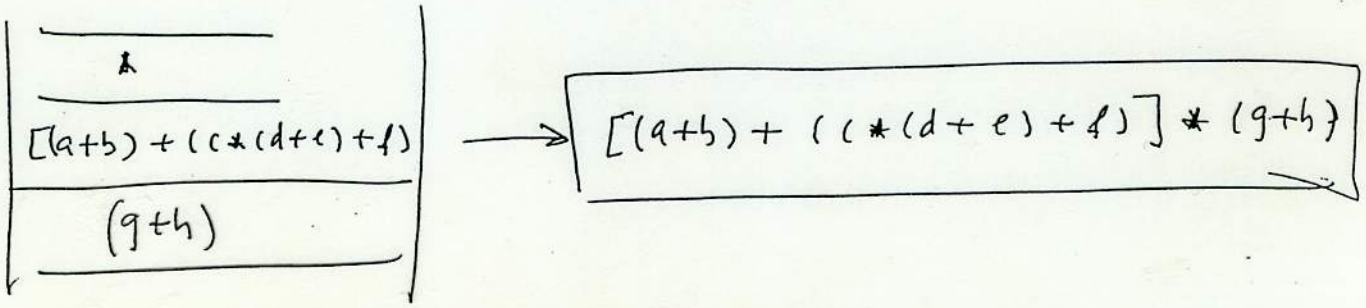
E_2 $(g+h) \equiv +gh$

$\boxed{*++ab+*c+def+gh}$



* + + a b + * c + d e f + g h


*, +, + a, b (a+b)
*, +, a+b, +, *, c, + d, e
d+e
c * (d+e)
* + (a+b), +, *, c, (d+e)
, +, (a+b), +, ((d+e), f
[(a+b) + (c*(d+e)+f)]
g+h
* [(a+b) + ((*(d+e)+f)] + g, h


*
[(a+b) + ((*(d+e)+f)
(g+h)
[(a+b) + ((*(d+e)+f)] * (g+h)