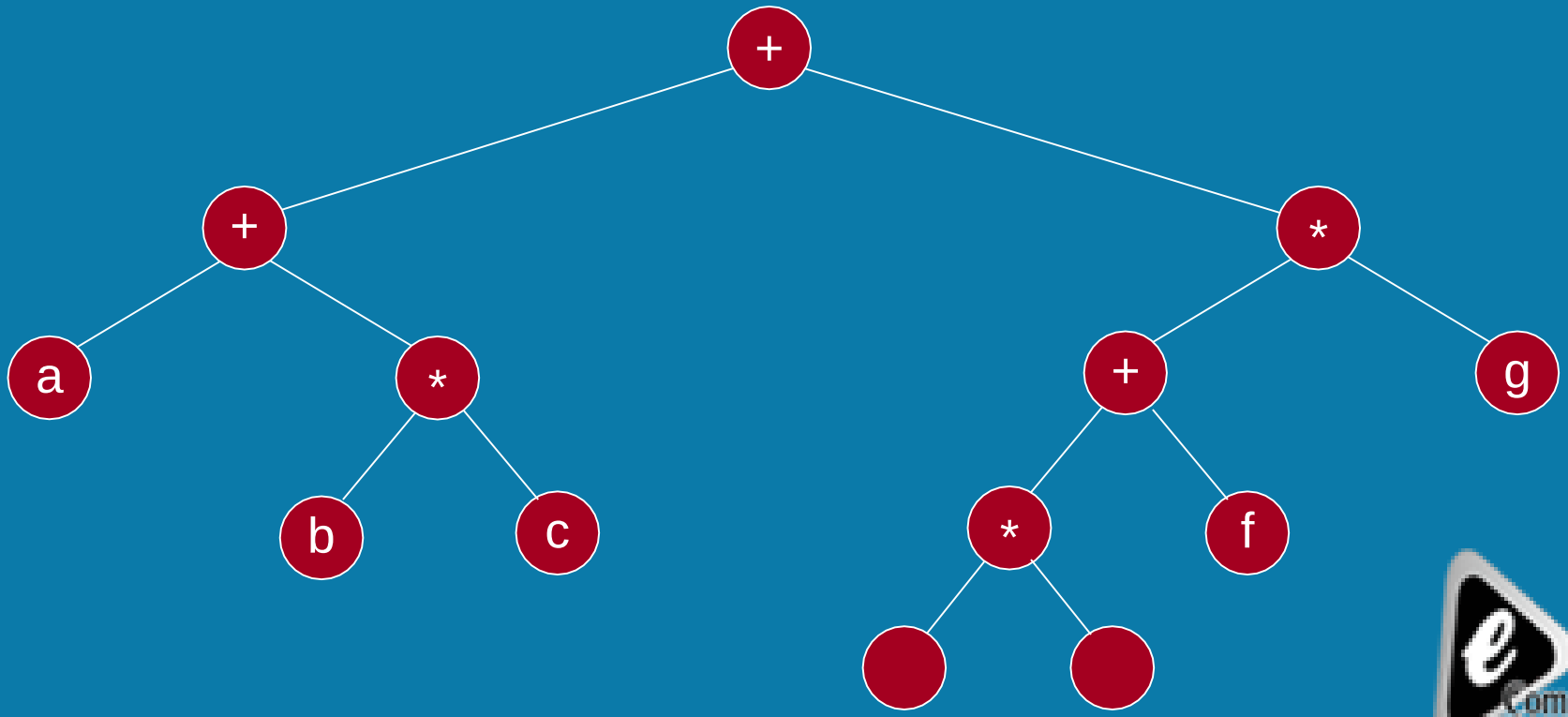
Expressions trees

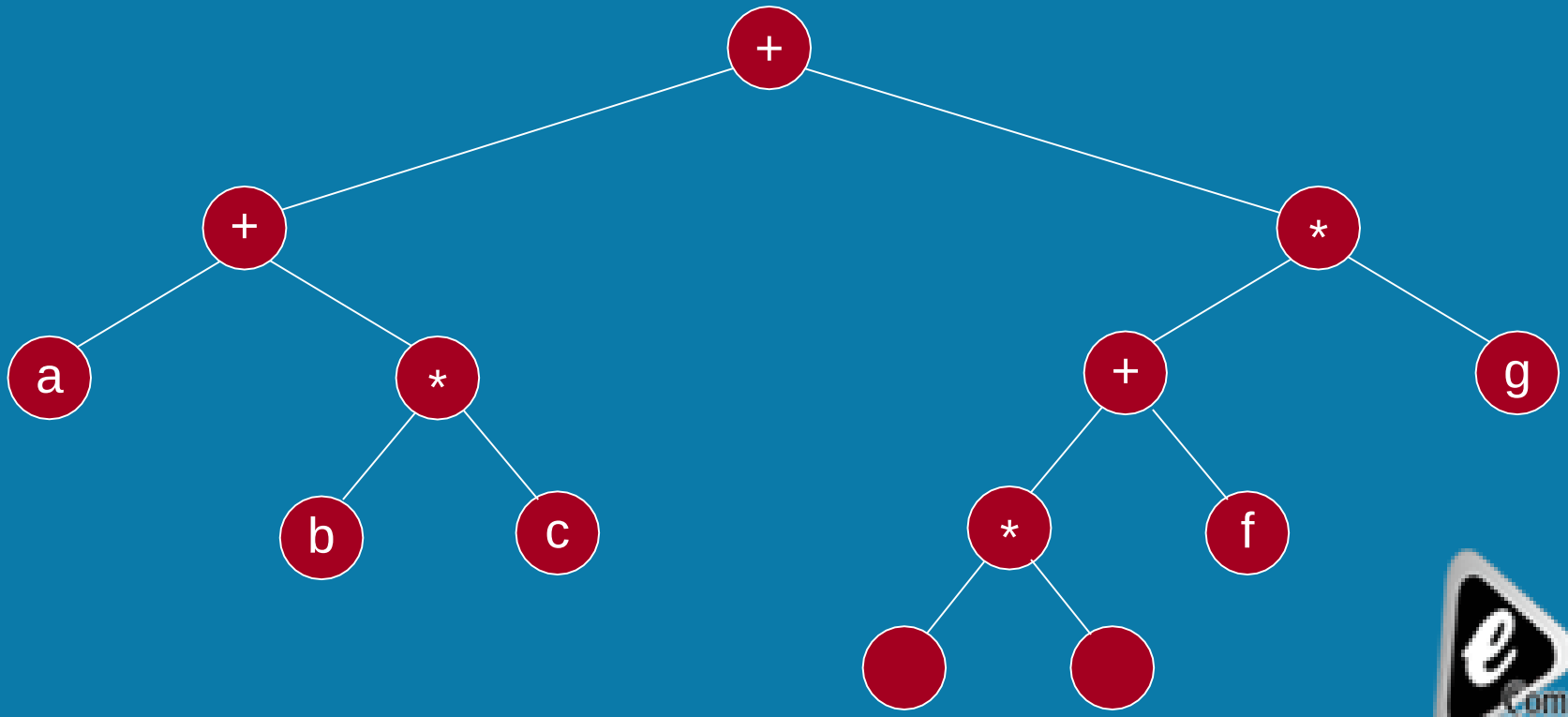
Expression Tree

- The inner nodes contain operators while leaf nodes contain operands.



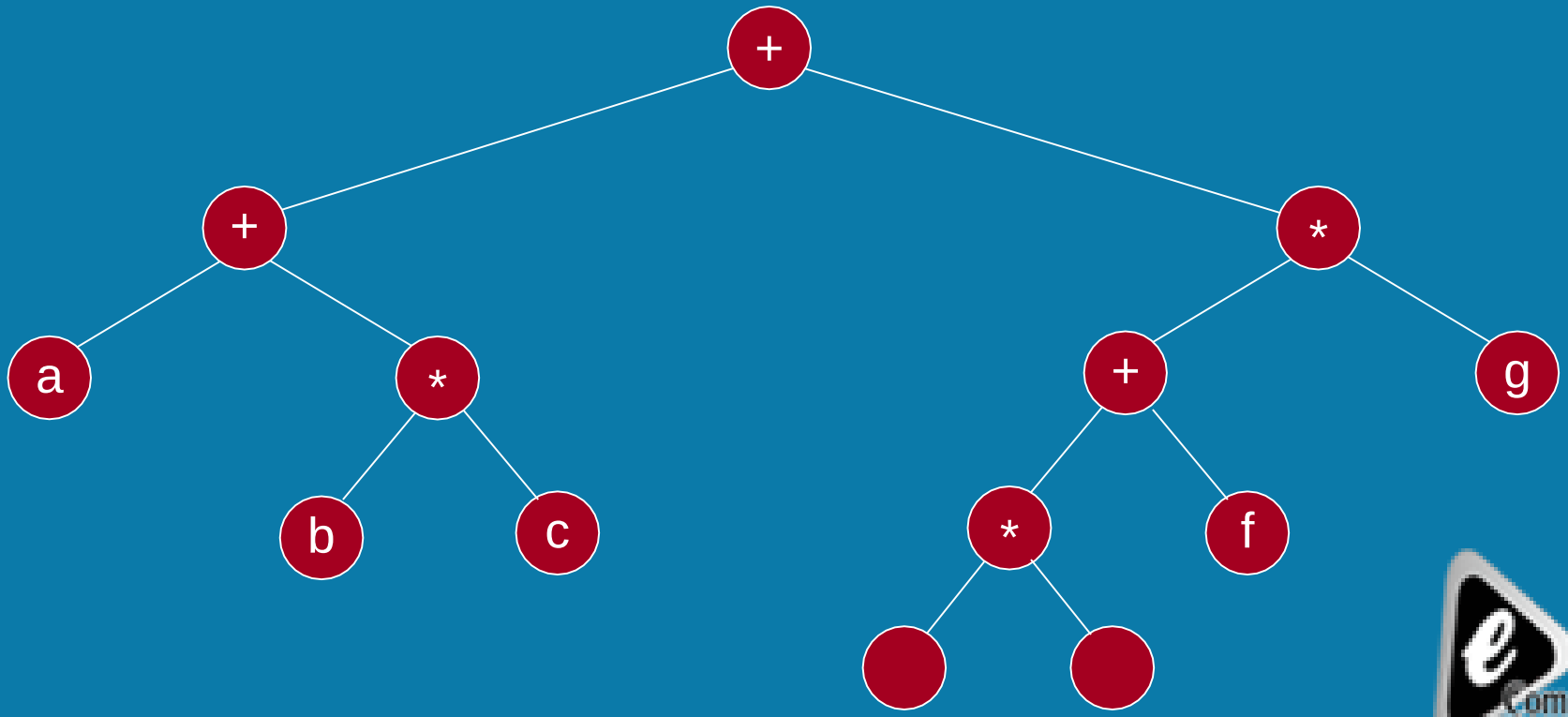
Expression Tree

- The tree is binary because the operators are binary.



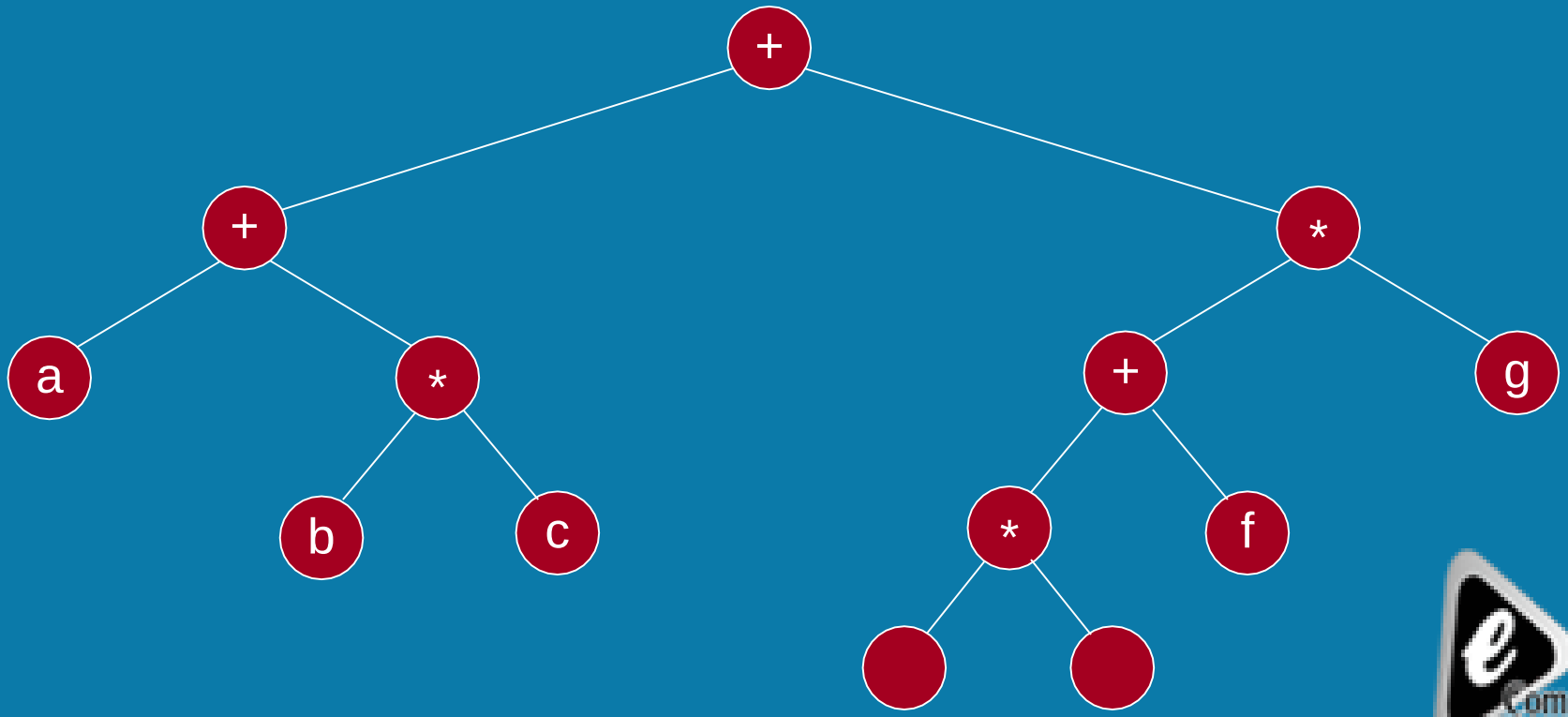
Expression Tree

- This is not necessary. A unary operator (!, e.g.) will have only one subtree.



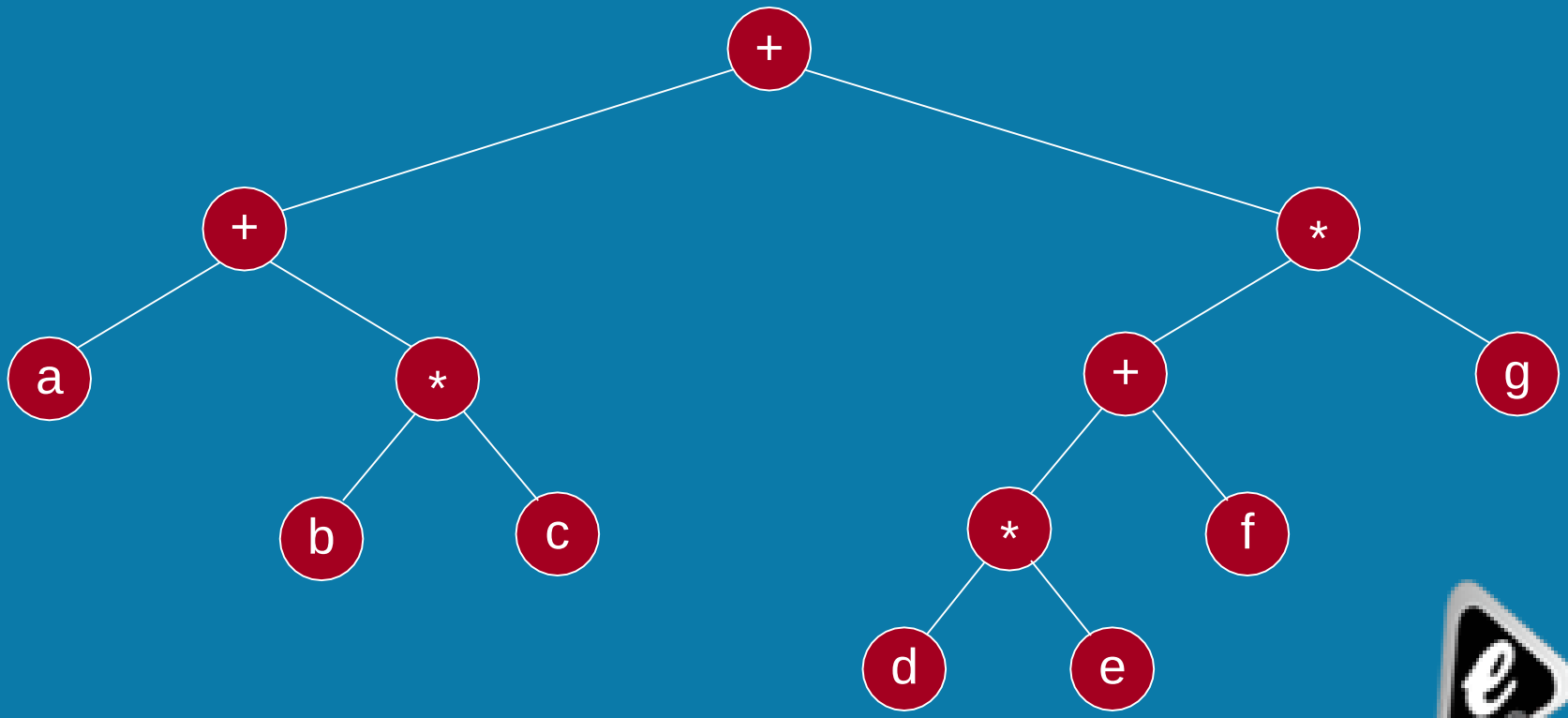
Expression Tree

- Inorder :
  (a+(b*c))+(((d*e)+f)*g)



Expression Tree

- Postorder traversal: a b c * + d e * f + g * + which is the postfix form.



Constructing Expression Tree

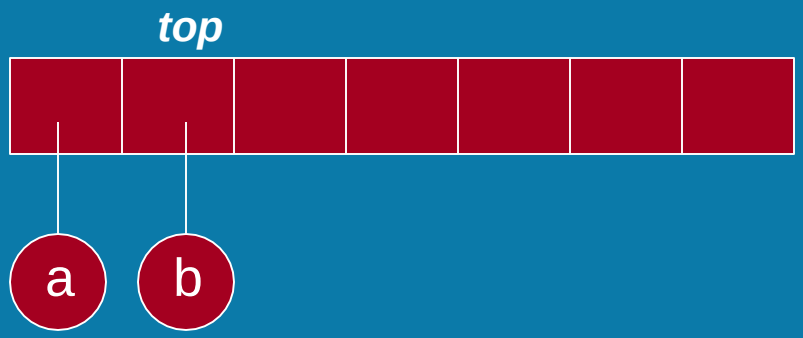
- Algorithm to convert postfix expression into an expression tree.
- We already have an expression to convert an infix expression to postfix.
- Read a symbol from the postfix expression.
- If symbol is an operand, put it in a one node tree and push it on a stack.
- If symbol is an operator, pop two trees from the stack, form a new tree with operator as the root and T_1 and T_2 as left and right subtrees and push this tree on the stack.

Constructing Expression Tree

- a b + c d e + * *

Constructing Expression Tree

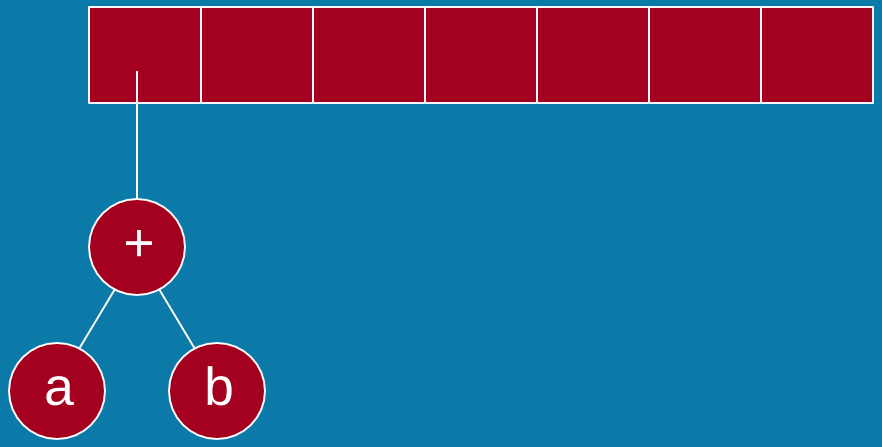
- a b + c d e + * *



If symbol is an operand, put it in a one node tree and push it on a stack.

Constructing Expression Tree

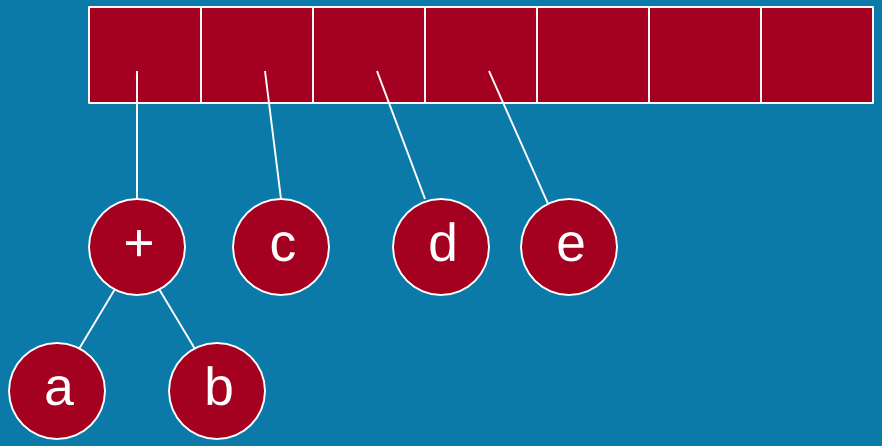
- a b + c d e + * *



If symbol is an operator, pop two trees from the stack, form a new tree with operator as the root and T_1 and T_2 as left and right subtrees and push this tree on the stack.

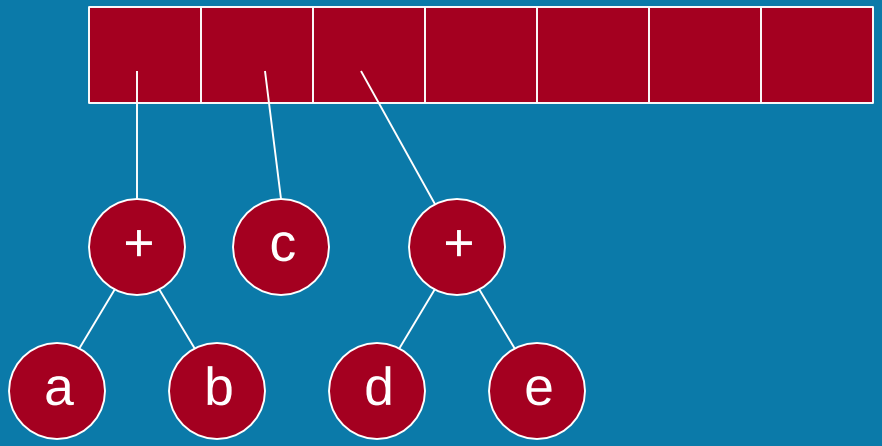
Constructing Expression Tree

- a b + c d e + * *



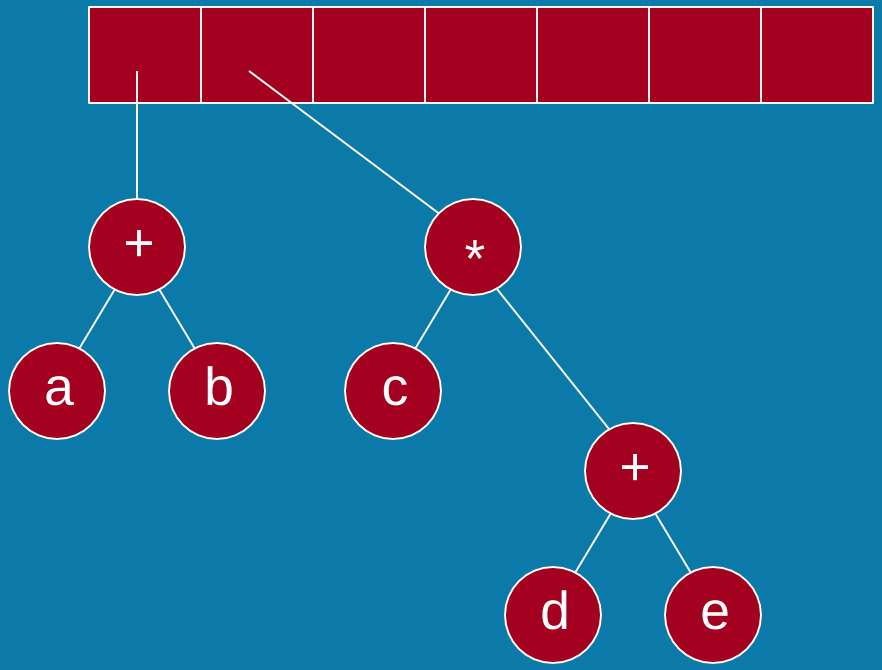
Constructing Expression Tree

- a b + c d e + * *



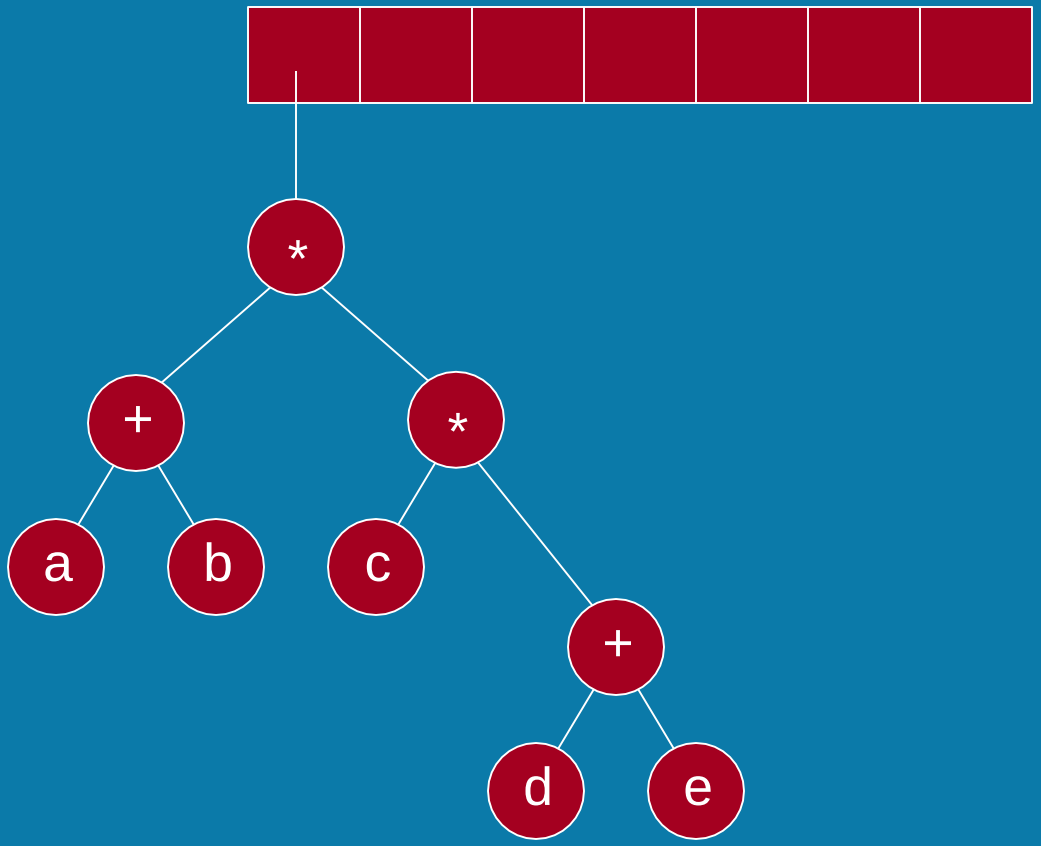
Constructing Expression Tree

- a b + c d e + * *

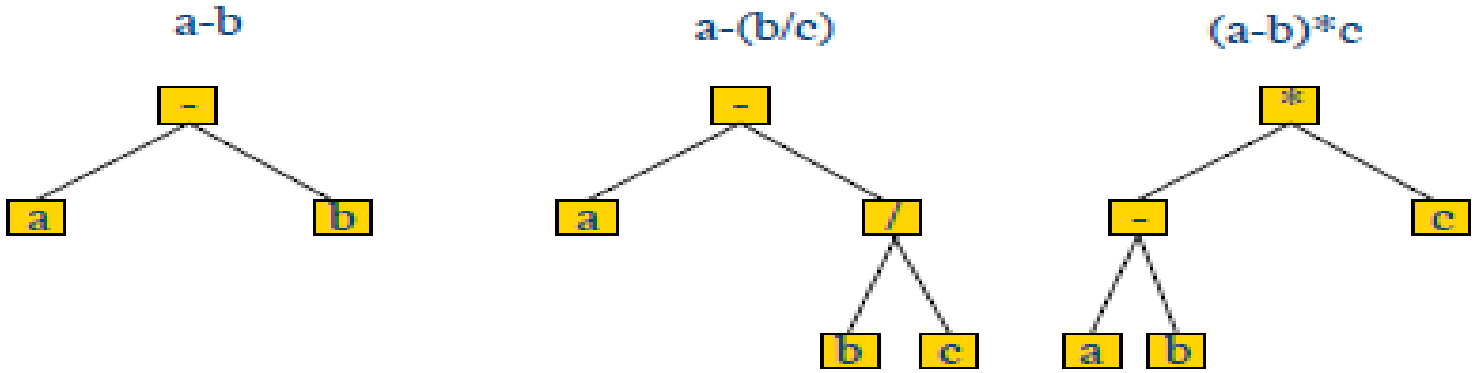


Constructing Expression Tree

- a b + c d e + * *



Arithmetic Expressions as Trees



hmetic expressions are often represented as binary trees.

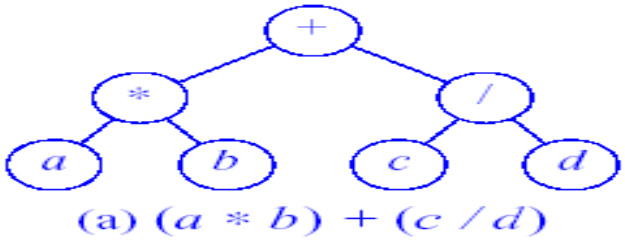
rnal nodes are **operations** - Leaves are **numbers/variables**.

rator **precedence** is enforced by the tree shape.

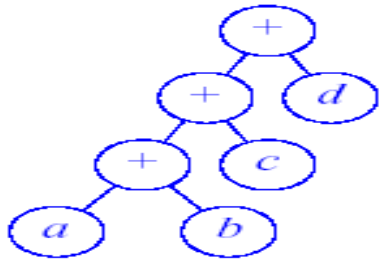
Image source : Google

Arithmetic Expressions as trees(cont.)

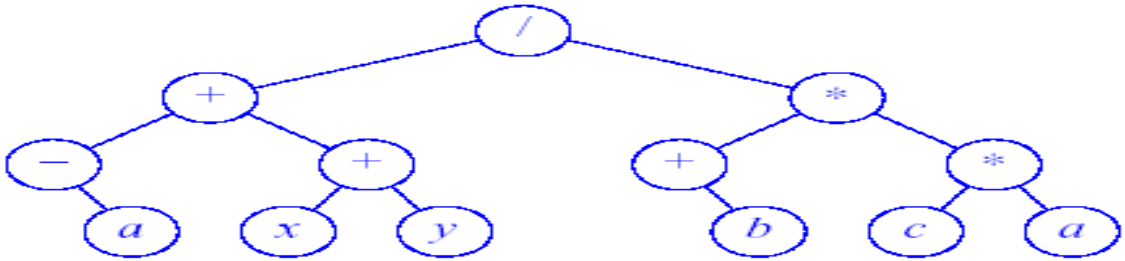
- A queue in which we are able to insert items or remove items from any position based on someproperty is (based on the priority assigned to the tasks) is knows as Priority Queue.



(a) $(a * b) + (c / d)$



(b) $((a + b) + c) + d)$



(c) $((-a) + (x + y)) / ((+b) * (c * a))$

Arithmetic Expressions as trees(cont.)

1. 3 + ((5+9)*2)
2. a + (b * c) + d * (e + f)
1. Infix expression:

(a+(b*c))+(d*(e + f))

2. Postfix Expression:

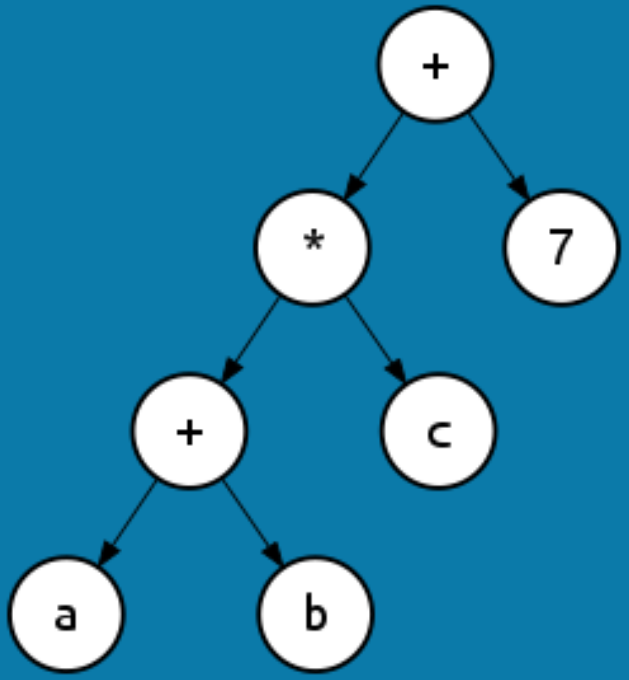
a b c * + d e f + * +

3. Prefix Expression:

+ + a * b c * d + e f

Arithmetic Expressions as trees(cont.)

(a+b)*c+7



Arithmetic Expressions as trees(cont.)

(5-x)*y+6/(x + z)

Post fix : 5x-y*6xz+/+

Pre fix : +*-5xy/6+xz