We will cover Sections

11.2 *Applications of trees* (game trees)

11.3 *Tree Traversal*

11.2 *Applications of Trees*

Game Trees

Trees can be used to analyze certain types of games such as *tic-tac-toe*, *nim*, *checkers*, *chess*, etc.

11.2 *Applications of Trees*

Game Trees

Trees can be used to analyze certain types of games such as *tic-tac-toe*, *nim*, *checkers*, *chess*, etc.

In each of these games:

- two players take turns
- each player knows (sees) the moves of the other player
- no element of chance

We model such games using game trees.

11.2 *Applications of Trees*

Game Trees

vertices: the positions that a game can be in as it is progresses
edges: legal moves between the positions
root: the starting position

11.2 *Applications of Trees*

Game Trees

vertices: the positions that a game can be in as it is progresses
edges: legal moves between the positions
root: the starting position

Game trees are usually large, hence *simplifications* are necessary.
Let's represent *symmetric positions* in the game by the *same vertex*.

11.2 *Applications of Trees*

Game Trees

vertices: the positions that a game can be in as it is progresses
edges: legal moves between the positions
root: the starting position

Game trees are usually large, hence *simplifications* are necessary.
Let's represent *symmetric positions* in the game by the *same vertex*.
However, if *different sequences of moves* lead to the same position of the game, it may be represented by *different vertices*.

11.2 *Applications of Trees*

Game Trees

vertices: the positions that a game can be in as it is progresses
edges: legal moves between the positions
root: the starting position

Another set of conventions:
Vertices at *even levels* are represented by *boxes* (*first player's move*), and those at *odd* by *circles* (*second player's move*).

11.2 *Applications of Trees*

Game Trees

vertices: the positions that a game can be in as it is progresses
edges: legal moves between the positions
root: the starting position

Another set of conventions:
Vertices at *even levels* are represented by *boxes* (*first player's move*), and those at *odd* by *circles* (*second player's move*).

Game trees can be infinite when the games they represent never end (games can enter an infinite loop), but for most games there are rules that lead to finite game trees.

11.2 *Applications of Trees*

Game Trees

vertices: the positions that a game can be in as it is progresses
edges: legal moves between the positions
root: the starting position
leaves: the final positions of a game
leaves' values: payoff to the first player
for win-lose games: +1 and a circle, if first player wins
-1 and a square, otherwise
If draws are allowed: 0

11.2 *Applications of Trees*

Game Trees

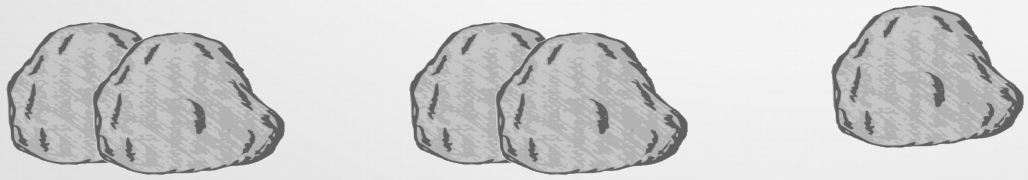
Example: Game of Nim
We are given a number of piles of stones.
Two players take turns by making moves.
A legal move: removing one or more stones from one of the piles, without removing all the stones left.
A player without a legal move loses.

11.2 *Applications of Trees*

Game Trees

Example: Game of Nim
We are given a number of piles of stones.
Two players take turns by making moves.
A legal move: removing one or more stones from one of the piles, without removing all the stones left.
A player without a legal move loses.

Three piles of stones (2-2-1):

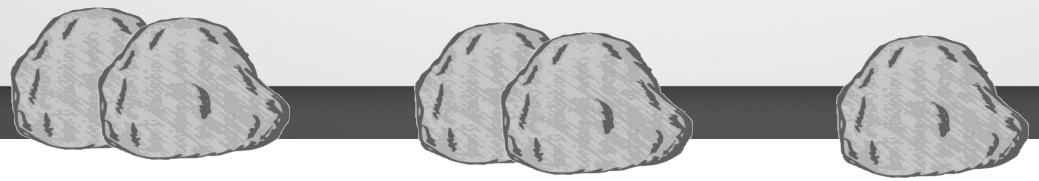


11.2 *Applications of Trees*

Game Trees

Example: Game of Nim : three piles of stones (2-2-1):

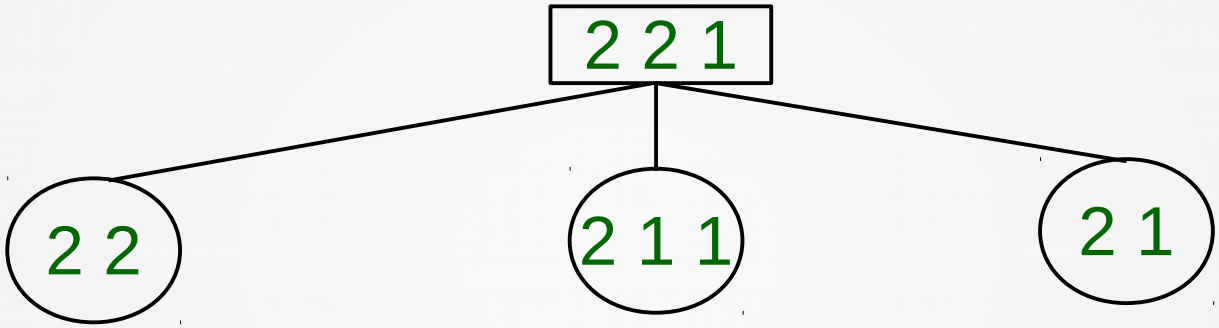
2 2 1

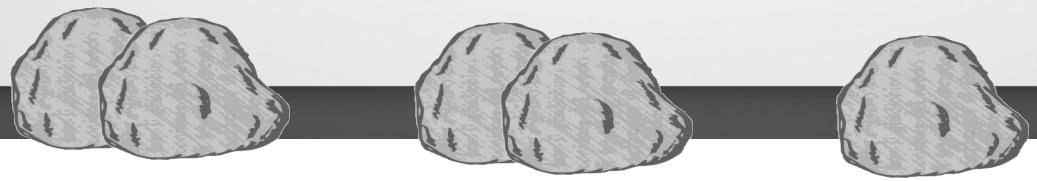


11.2 *Applications of Trees*

Game Trees

Example: Game of Nim : three piles of stones (2-2-1):

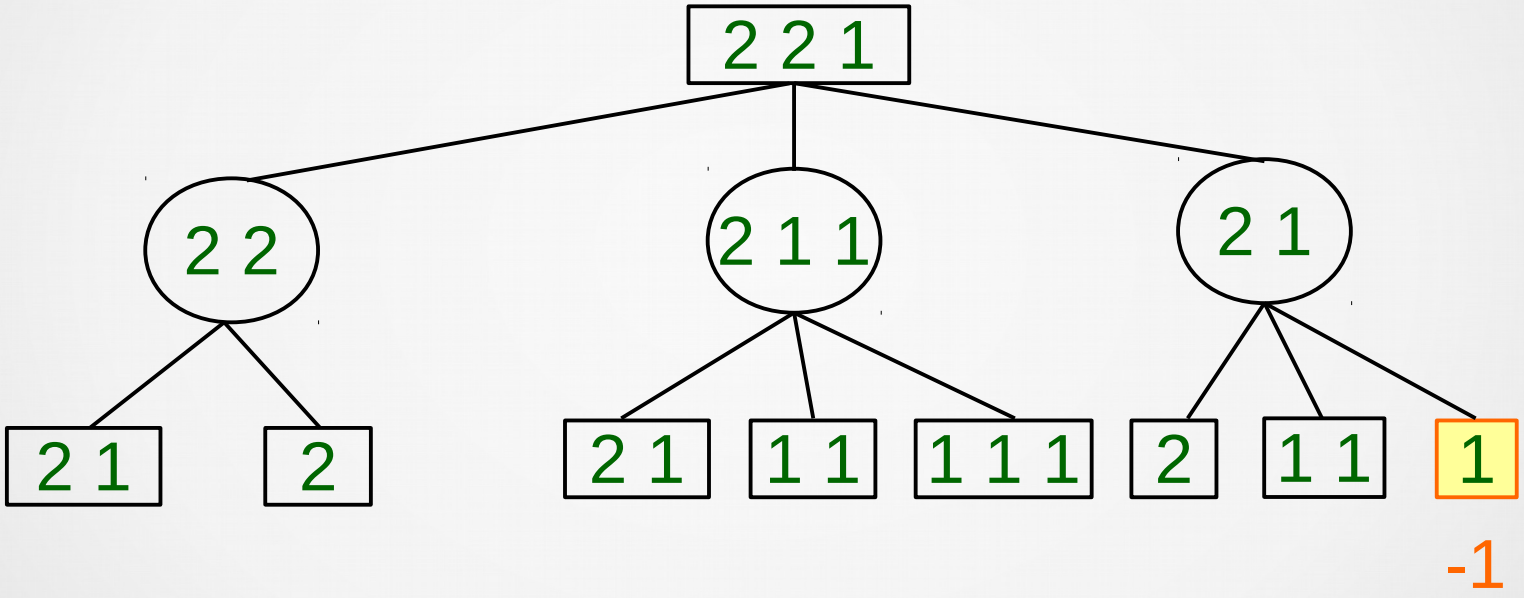




11.2 *Applications of Trees*

Game Trees

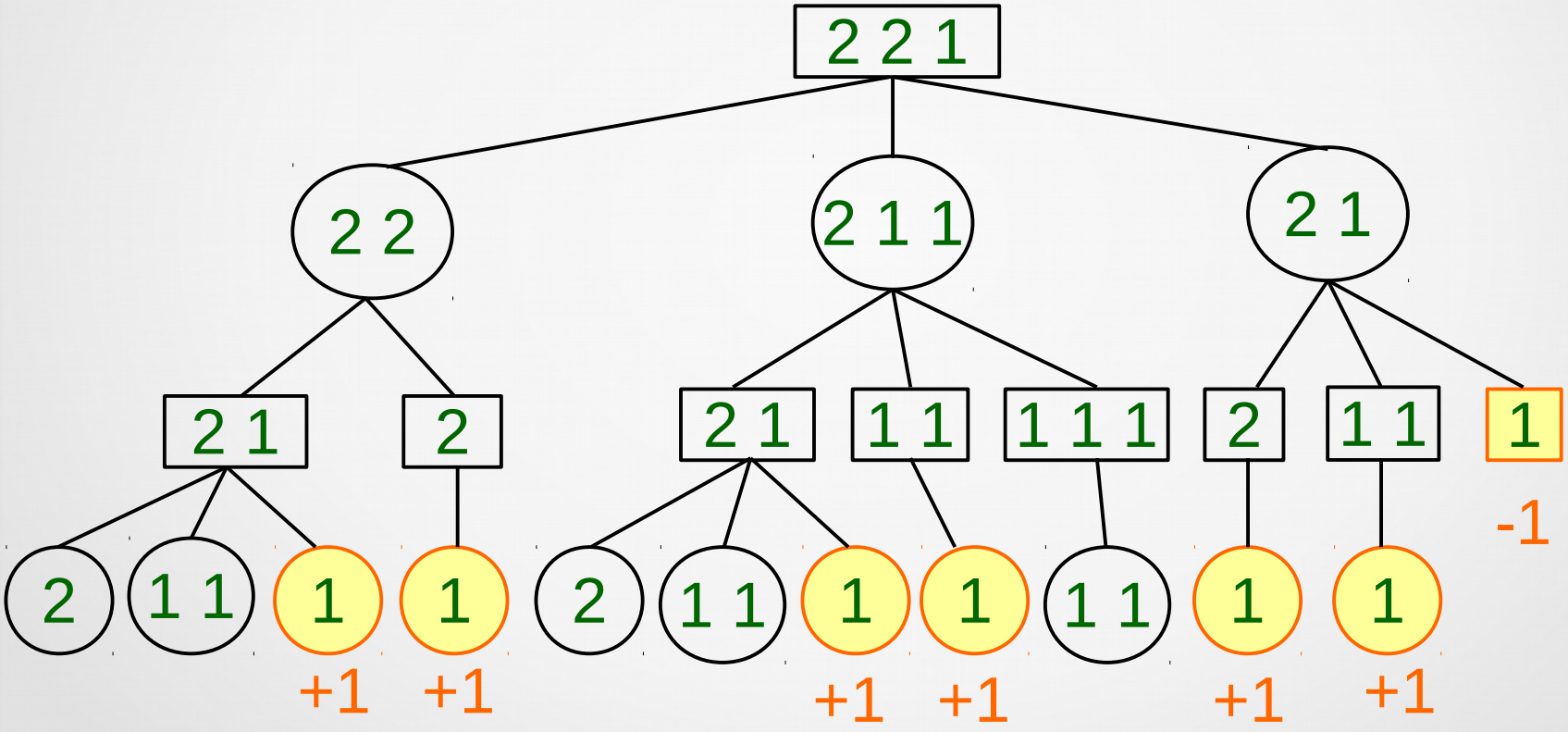
Example: Game of Nim : three piles of stones (2-2-1):



11.2 *Applications of Trees*

Game Trees

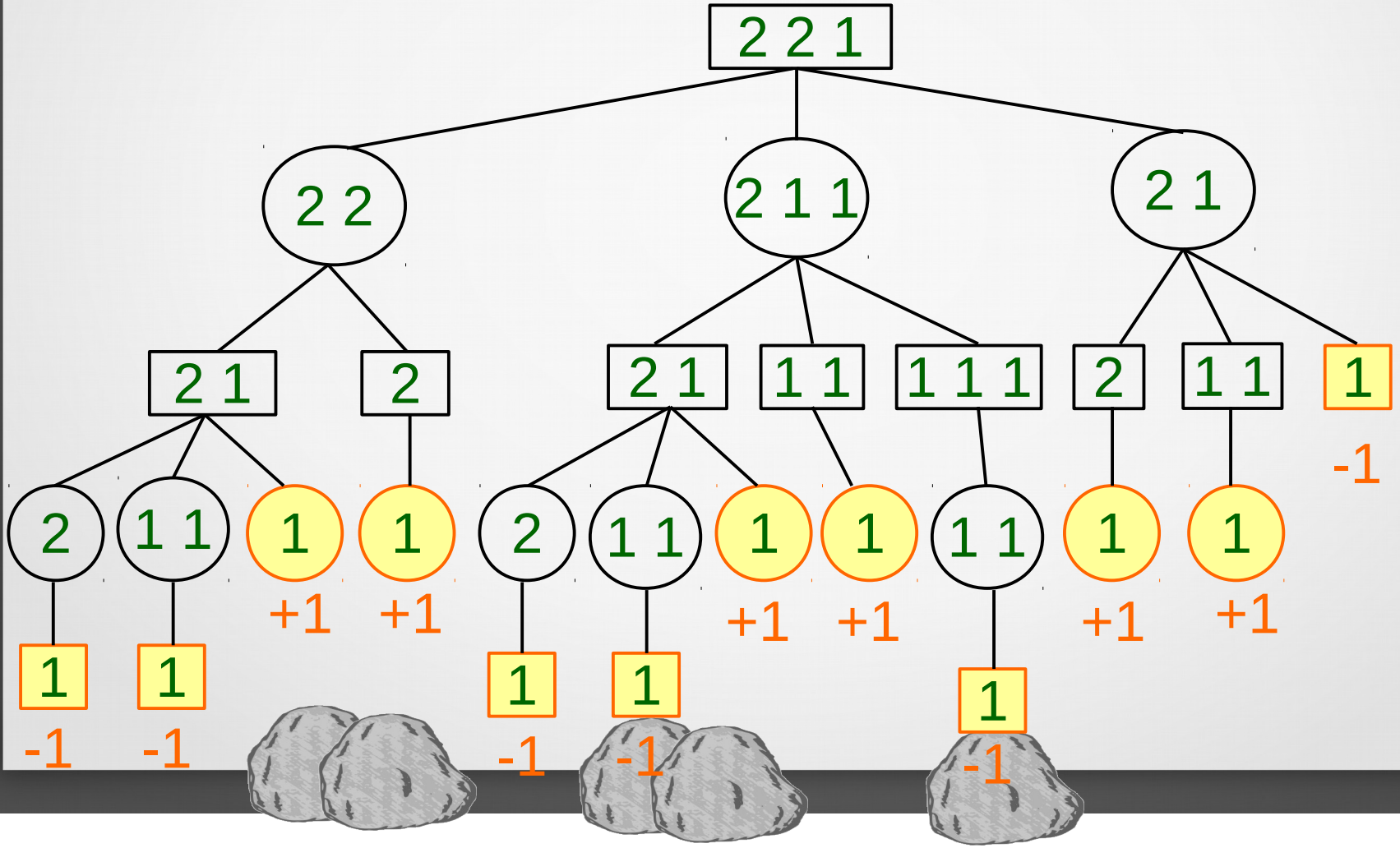
Example: Game of Nim : three piles of stones (2-2-1):



11.2 *Applications of Trees*

Game Trees

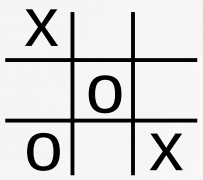
Example: Game of Nim : three piles of stones (2-2-1):



11.2 *Applications of Trees*

Game Trees

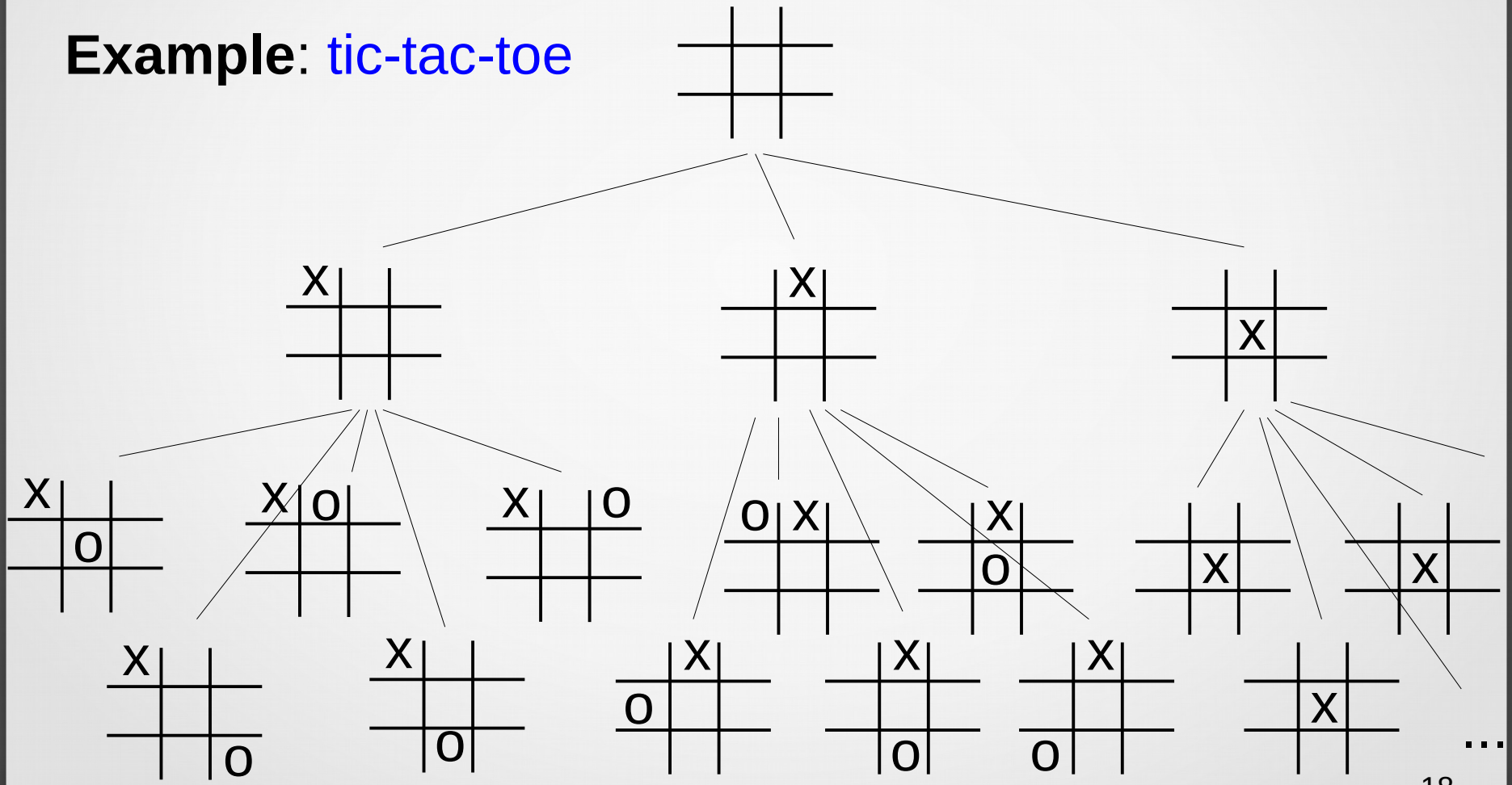
Example: tic-tac-toe



11.2 *Applications of Trees*

Game Trees

Example: tic-tac-toe



11.2 *Applications of Trees*

Game Trees

A game tree can be systematically analyzed to determine optimal playing strategies for each player.

However, in practice for most games, the corresponding game tree is much too large to build and analyze in its entirety.

Instead, partial game trees starting from the current configuration in the game and estimate the best next move based on the results of the partial tree are built.

11.2 *Applications of Trees*

Game Trees

Tic-tac-toe and chess are examples of deterministic games whose outcome is completely determined by the choices made by each player.

A game that involves rolling a pair of dice or shuffling a deck of cards introduces randomization into the game. Probability theory as well as game trees are required to analyze games with an element of chance.

11.3 *Tree traversal*

Ordered rooted trees are often used to store information, as well as to represent various types of expressions (e.g. arithmetic expressions).

We need procedures for visiting each vertex/node of such a tree to access data.

11.3 *Tree traversal*

Ordered rooted trees are often used to store information, as well as to represent various types of expressions (e.g. arithmetic expressions).

We need procedures for visiting each vertex/node of such a tree to access data.

Note that *trees have recursive structure*.

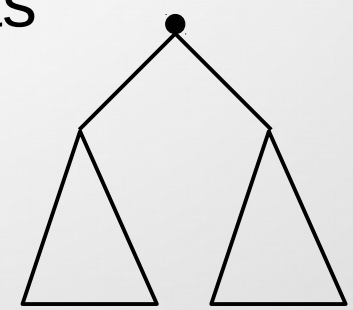
11.3 *Tree traversal*

Ordered rooted trees are often used to store information, as well as to represent various types of expressions (e.g. arithmetic expressions).

We need procedures for visiting each vertex/node of such a tree to access data.

Note that *trees have recursive structure*.
For example, a *binary tree* can be defined as either

- an empty binary tree (base case), or as
- a vertex with two children:
  - left child is a binary tree, and
  - right child is a binary tree



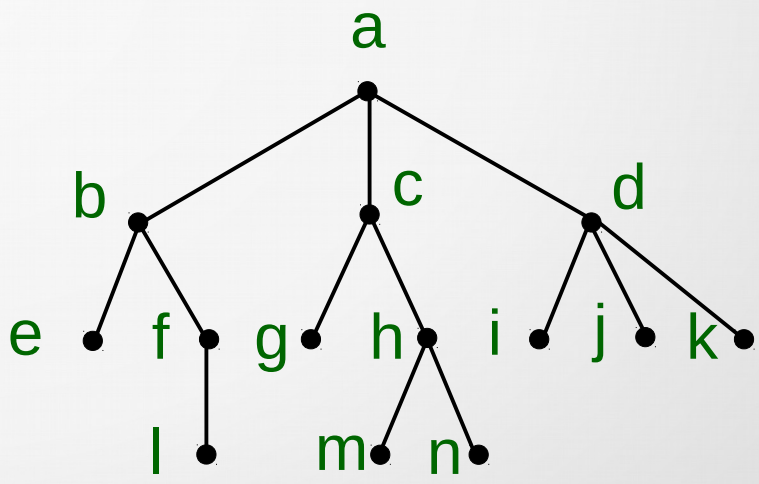
11.3 *Tree traversal*

Preorder Traversal

Let T be an ordered rooted tree with root *r*, that has *n* children enumerated from left to right: T_1, ...,T_n . Then

preorder traversal procedure:

- begins by visiting *r*, then
- traverse T_1 in preorder, then
- traverse T_2 in preorder, then
- …
- traverse T_n in preorder.



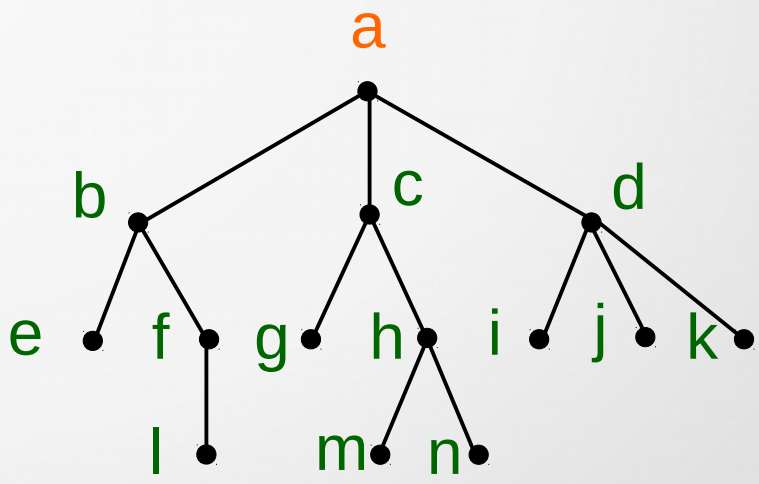
11.3 *Tree traversal*

Preorder Traversal

Let T be an ordered rooted tree with root *r*, that has *n* children enumerated from left to right: T_1, ...,T_n . Then

preorder traversal procedure:

- begins by visiting *r*, then
- traverse T_1 in preorder, then
- traverse T_2 in preorder, then
- …
- traverse T_n in preorder.



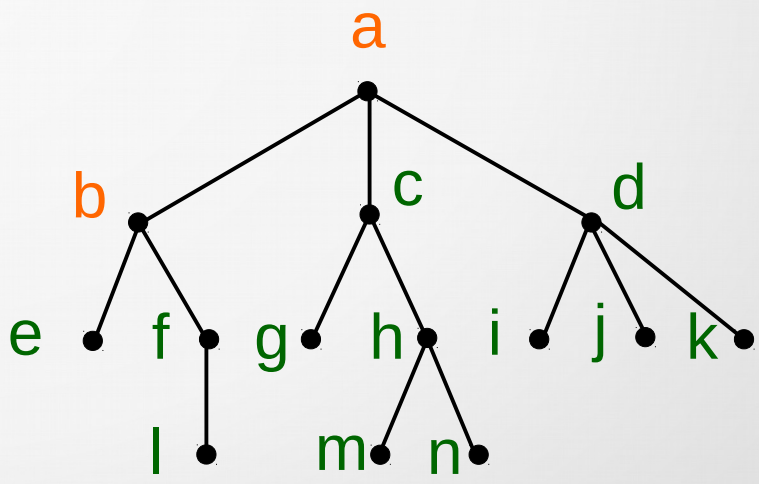
11.3 *Tree traversal*

Preorder Traversal

Let T be an ordered rooted tree with root *r*, that has *n* children enumerated from left to right: T_1, ...,T_n . Then

preorder traversal procedure:

- begins by visiting *r*, then
- traverse T_1 in preorder, then
- traverse T_2 in preorder, then
- …
- traverse T_n in preorder.



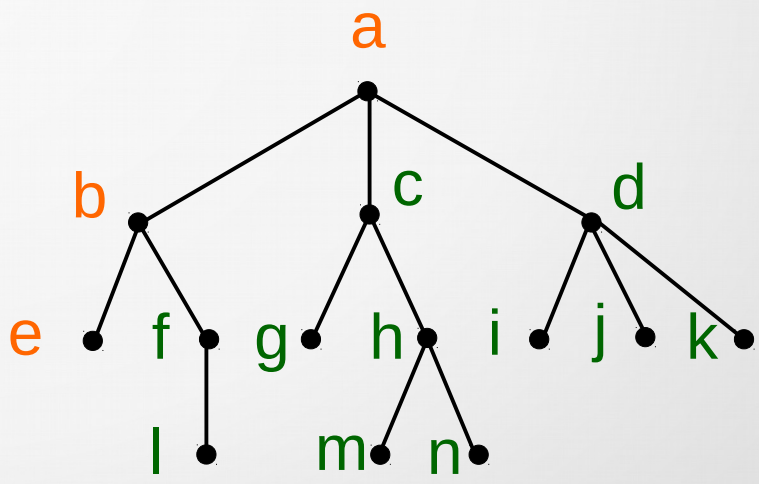
11.3 *Tree traversal*

Preorder Traversal

Let T be an ordered rooted tree with root *r*, that has *n* children enumerated from left to right: T_1, ...,T_n . Then

preorder traversal procedure:

- begins by visiting *r*, then
- traverse T_1 in preorder, then
- traverse T_2 in preorder, then
- …
- traverse T_n in preorder.



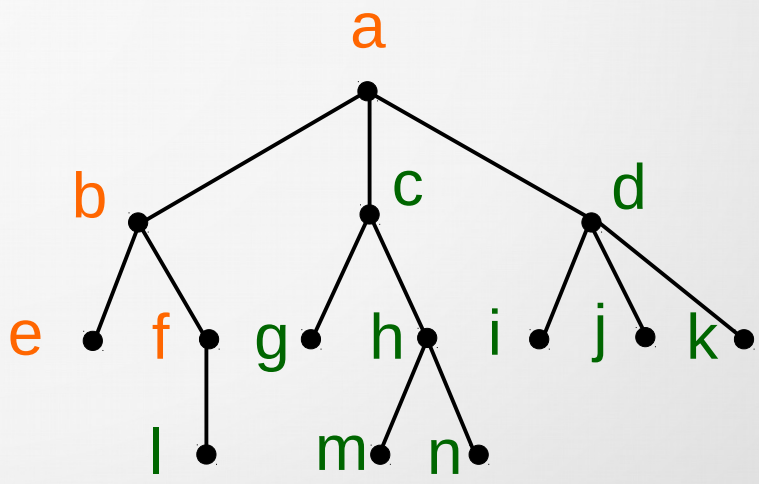
11.3 *Tree traversal*

Preorder Traversal

Let T be an ordered rooted tree with root *r*, that has *n* children enumerated from left to right: T_1, ...,T_n . Then

preorder traversal procedure:

- begins by visiting *r*, then
- traverse T_1 in preorder, then
- traverse T_2 in preorder, then
- …
- traverse T_n in preorder.



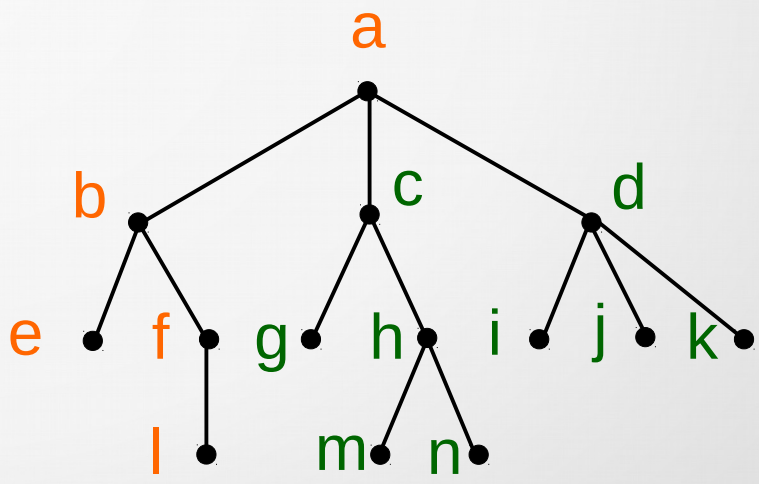
11.3 *Tree traversal*

Preorder Traversal

Let T be an ordered rooted tree with root *r*, that has *n* children enumerated from left to right: T_1, ...,T_n . Then

preorder traversal procedure:

- begins by visiting *r*, then
- traverse T_1 in preorder, then
- traverse T_2 in preorder, then
- …
- traverse T_n in preorder.



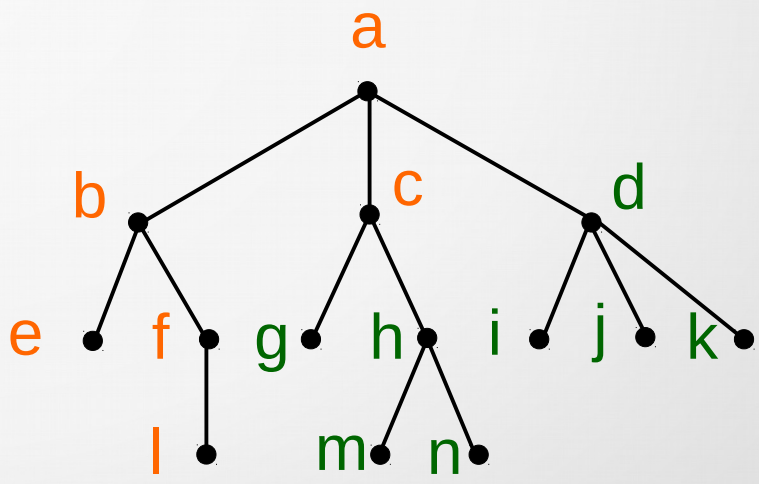
11.3 *Tree traversal*

Preorder Traversal

Let T be an ordered rooted tree with root *r*, that has *n* children enumerated from left to right: T_1, ...,T_n . Then

preorder traversal procedure:

- begins by visiting *r*, then
- traverse T_1 in preorder, then
- traverse T_2 in preorder, then
- …
- traverse T_n in preorder.



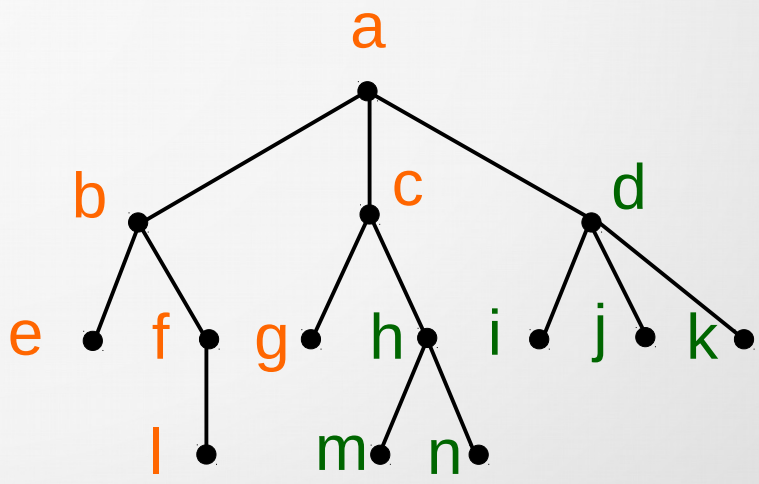
11.3 *Tree traversal*

Preorder Traversal

Let T be an ordered rooted tree with root *r*, that has *n* children enumerated from left to right: T_1, ...,T_n . Then

preorder traversal procedure:

- begins by visiting *r*, then
- traverse T_1 in preorder, then
- traverse T_2 in preorder, then
- …
- traverse T_n in preorder.



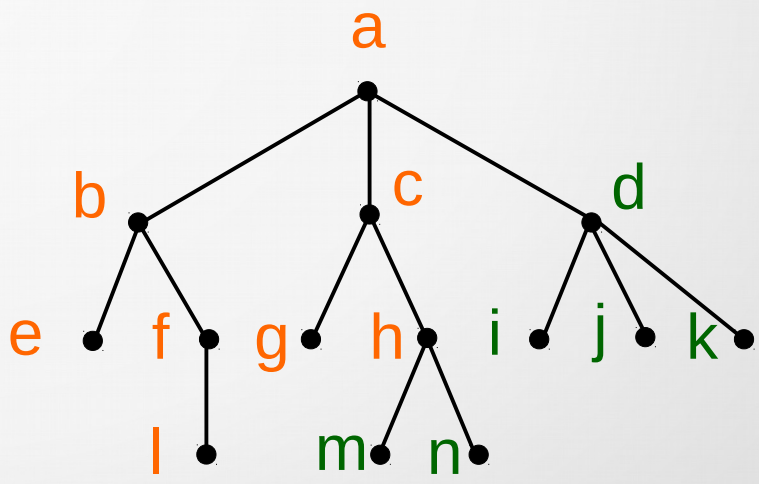
11.3 *Tree traversal*

Preorder Traversal

Let T be an ordered rooted tree with root *r*, that has *n* children enumerated from left to right: T_1, ...,T_n . Then

preorder traversal procedure:

- begins by visiting *r*, then
- traverse T_1 in preorder, then
- traverse T_2 in preorder, then
- …
- traverse T_n in preorder.



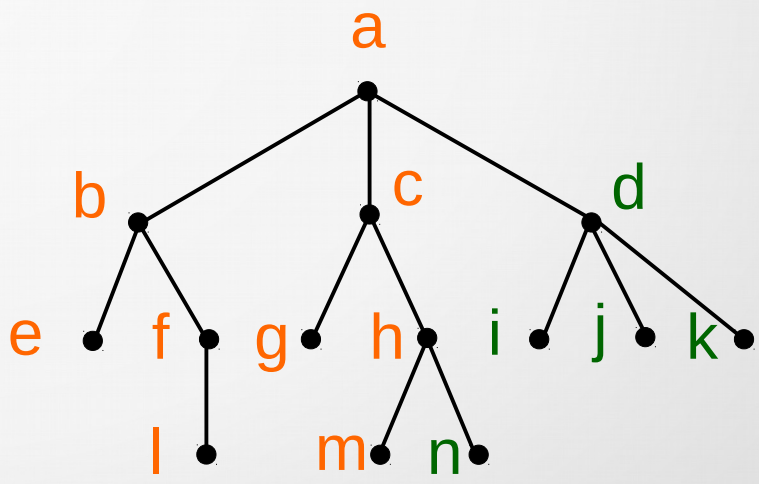
11.3 *Tree traversal*

Preorder Traversal

Let T be an ordered rooted tree with root *r*, that has *n* children enumerated from left to right: T_1, ...,T_n . Then

preorder traversal procedure:

- begins by visiting *r*, then
- traverse T_1 in preorder, then
- traverse T_2 in preorder, then
- …
- traverse T_n in preorder.



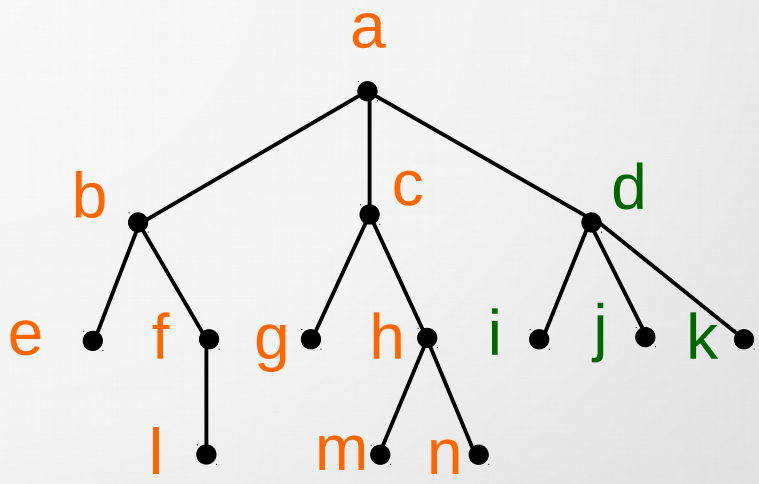
11.3 *Tree traversal*

Preorder Traversal

Let T be an ordered rooted tree with root *r*, that has *n* children enumerated from left to right: T_1, ...,T_n . Then

preorder traversal procedure:

- begins by visiting *r*, then
- traverse T_1 in preorder, then
- traverse T_2 in preorder, then
- …
- traverse T_n in preorder.



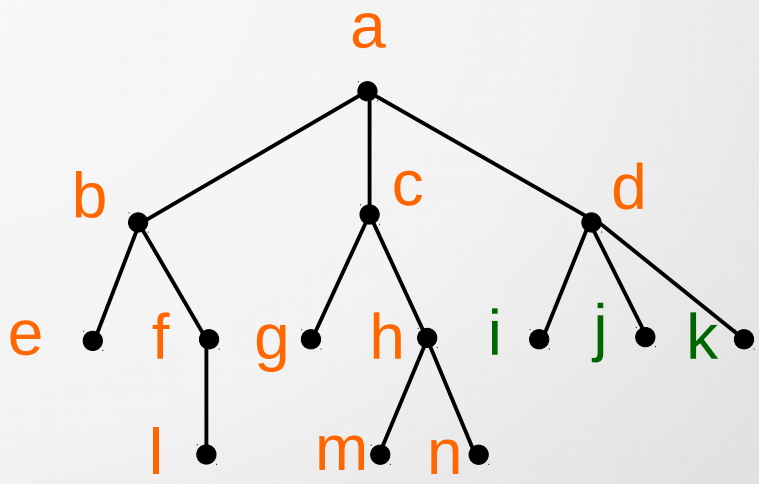
11.3 *Tree traversal*

Preorder Traversal

Let T be an ordered rooted tree with root *r*, that has *n* children enumerated from left to right: T_1, ...,T_n . Then

preorder traversal procedure:

- begins by visiting *r*, then
- traverse T_1 in preorder, then
- traverse T_2 in preorder, then
- …
- traverse T_n in preorder.



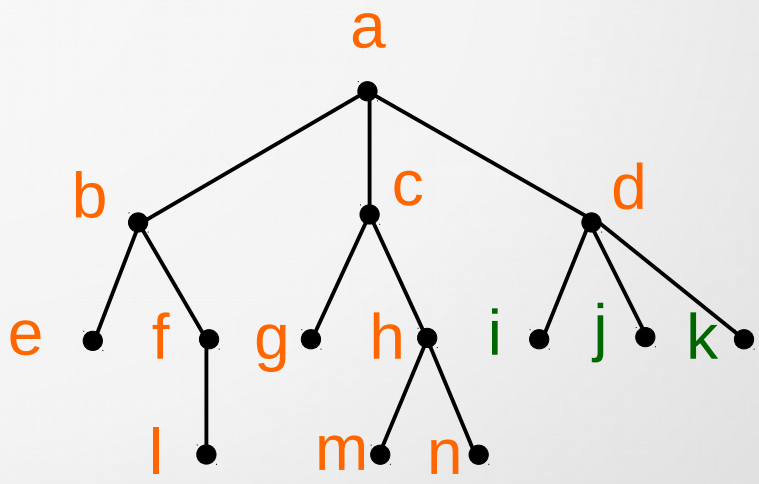
11.3 *Tree traversal*

Preorder Traversal

Let T be an ordered rooted tree with root *r*, that has *n* children enumerated from left to right: T_1, ...,T_n . Then

preorder traversal procedure:

- begins by visiting *r*, then
- traverse T_1 in preorder, then
- traverse T_2 in preorder, then
- …
- traverse T_n in preorder.



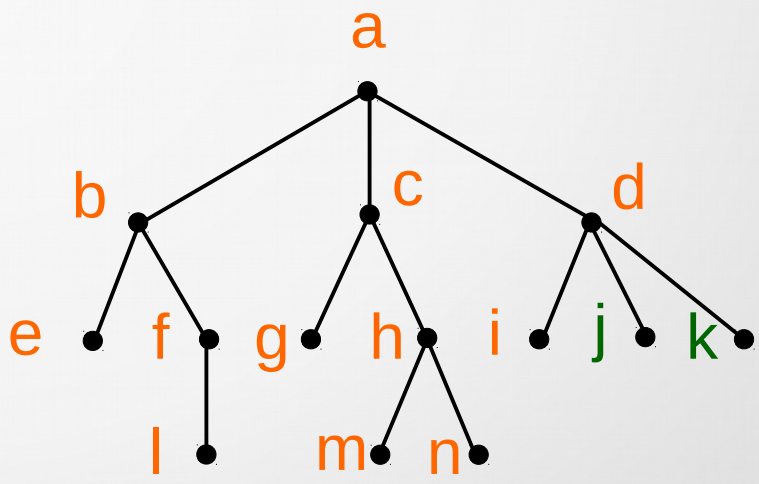
11.3 *Tree traversal*

Preorder Traversal

Let T be an ordered rooted tree with root *r*, that has *n* children enumerated from left to right: T_1, ...,T_n . Then

preorder traversal procedure:

- begins by visiting *r*, then
- traverse T_1 in preorder, then
- traverse T_2 in preorder, then
- …
- traverse T_n in preorder.



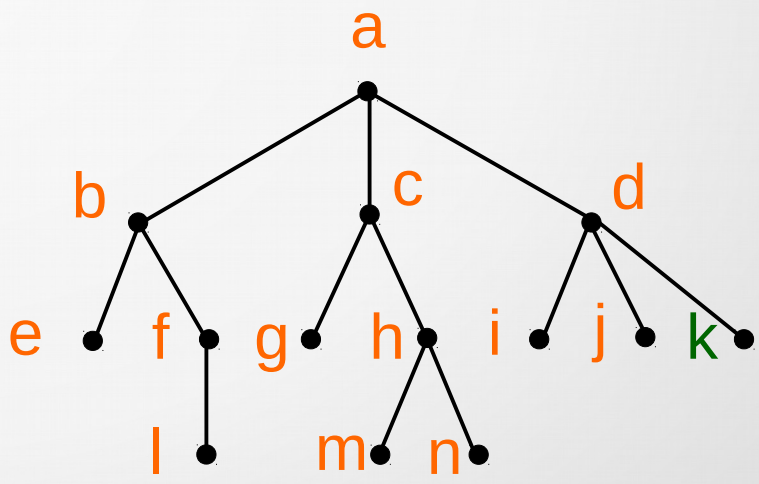
11.3 *Tree traversal*

Preorder Traversal

Let T be an ordered rooted tree with root *r*, that has *n* children enumerated from left to right: T_1, ...,T_n . Then

preorder traversal procedure:

- begins by visiting *r*, then
- traverse T_1 in preorder, then
- traverse T_2 in preorder, then
- …
- traverse T_n in preorder.



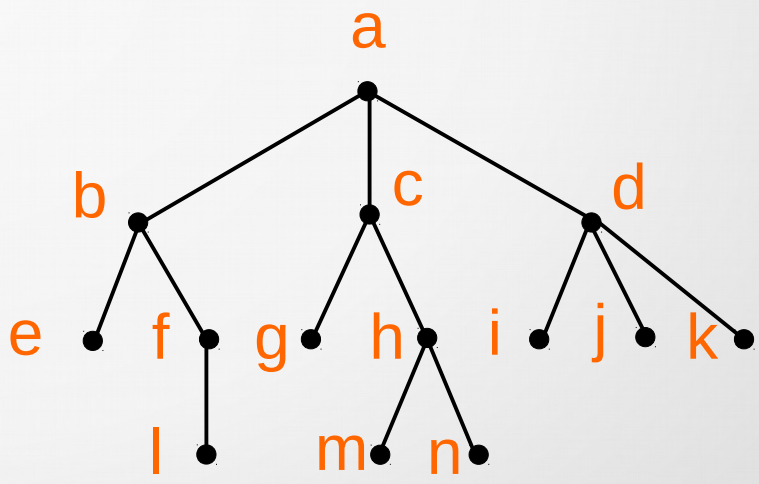
11.3 *Tree traversal*

Preorder Traversal

Let T be an ordered rooted tree with root *r*, that has *n* children enumerated from left to right: T_1, ...,T_n . Then

preorder traversal procedure:

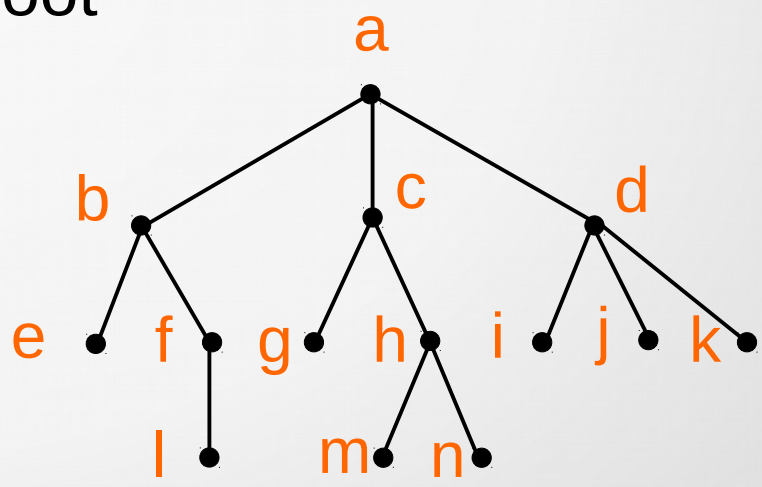
- begins by visiting *r*, then
- traverse T_1 in preorder, then
- traverse T_2 in preorder, then
- …
- traverse T_n in preorder.



11.3 *Tree traversal*

Preorder Traversal

procedure *preorder*(*T*: ordered rooted tree)
r := root of *T*
list *r*
for each child *c* of *r* from left to right
    T(c) := subtree with *c* as its root
    preorder(*T(c)*)



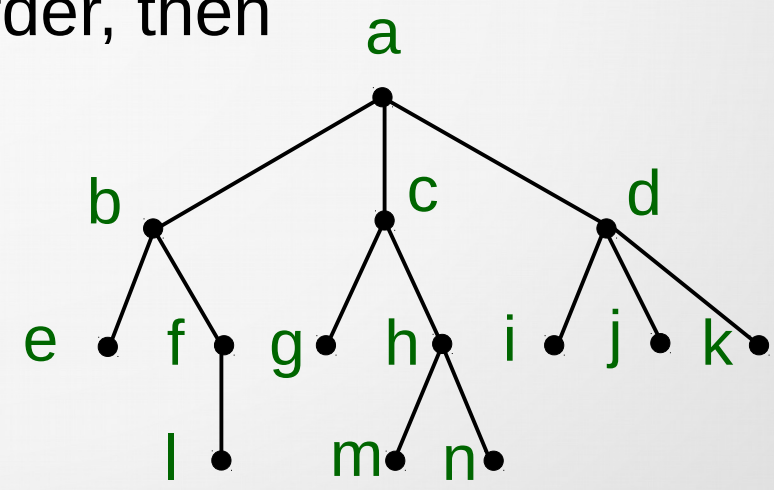
11.3 *Tree traversal*

Inorder Traversal

Let T be an ordered rooted tree with root *r*, that has *n* children enumerated from left to right: T_1, ...,T_n . Then

inorder traversal procedure:

- begins by traversing T_1 in inorder, then
- visiting *r*, then
- traverse T_2 in inorder, then
- …
- finishes by traversing T_n in inorder.



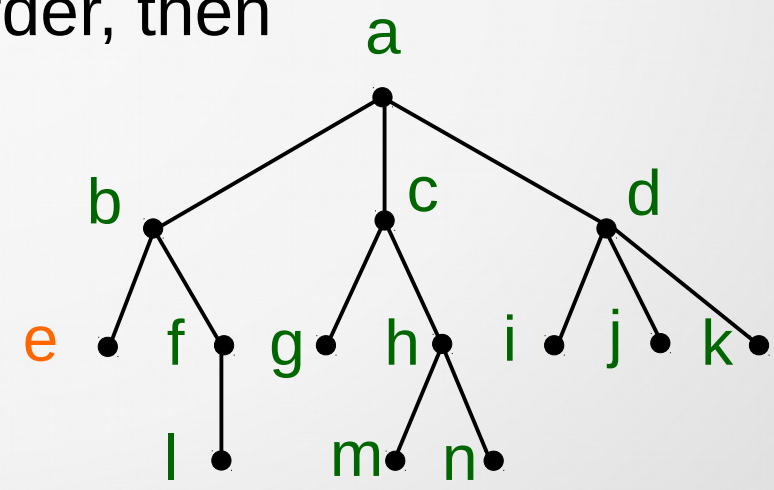
11.3 *Tree traversal*

Inorder Traversal

Let T be an ordered rooted tree with root *r*, that has *n* children enumerated from left to right: T_1, ...,T_n . Then

inorder traversal procedure:

- begins by traversing T_1 in inorder, then
- visiting *r*, then
- traverse T_2 in inorder, then
- …
- finishes by traversing T_n in inorder.



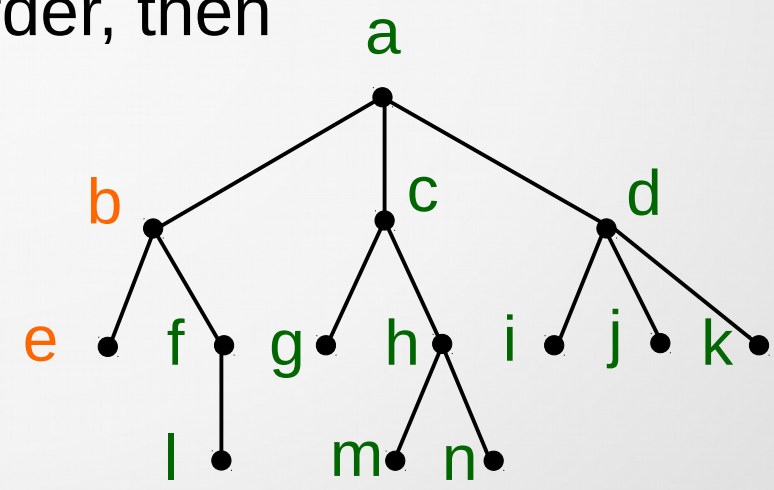
11.3 *Tree traversal*

Inorder Traversal

Let T be an ordered rooted tree with root *r*, that has *n* children enumerated from left to right: T_1, ...,T_n . Then

inorder traversal procedure:

- begins by traversing T_1 in inorder, then
- visiting *r*, then
- traverse T_2 in inorder, then
- …
- finishes by traversing T_n in inorder.



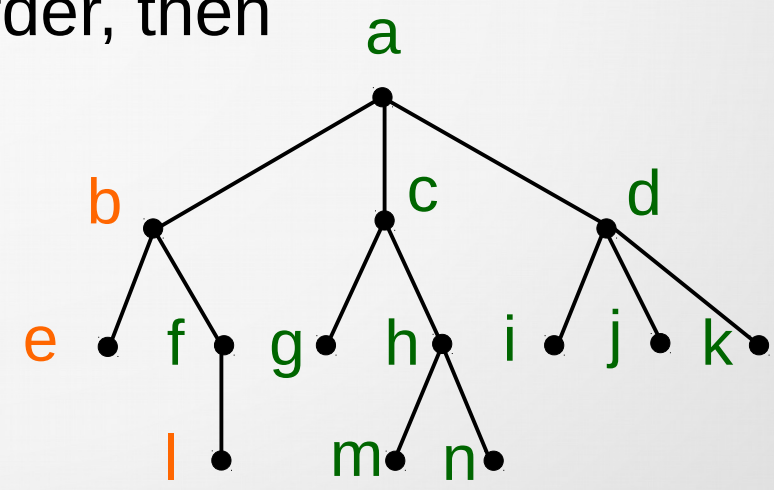
11.3 *Tree traversal*

Inorder Traversal

Let T be an ordered rooted tree with root *r*, that has *n* children enumerated from left to right: T_1, ...,T_n . Then

inorder traversal procedure:

- begins by traversing T_1 in inorder, then
- visiting *r*, then
- traverse T_2 in inorder, then
- …
- finishes by traversing T_n in inorder.



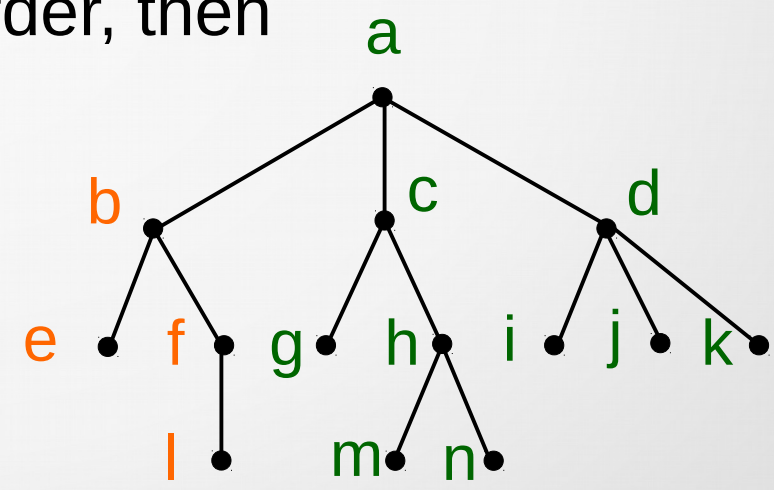
11.3 *Tree traversal*

Inorder Traversal

Let T be an ordered rooted tree with root *r*, that has *n* children enumerated from left to right: T_1, ...,T_n . Then

inorder traversal procedure:

- begins by traversing T_1 in inorder, then
- visiting *r*, then
- traverse T_2 in inorder, then
- …
- finishes by traversing T_n in inorder.



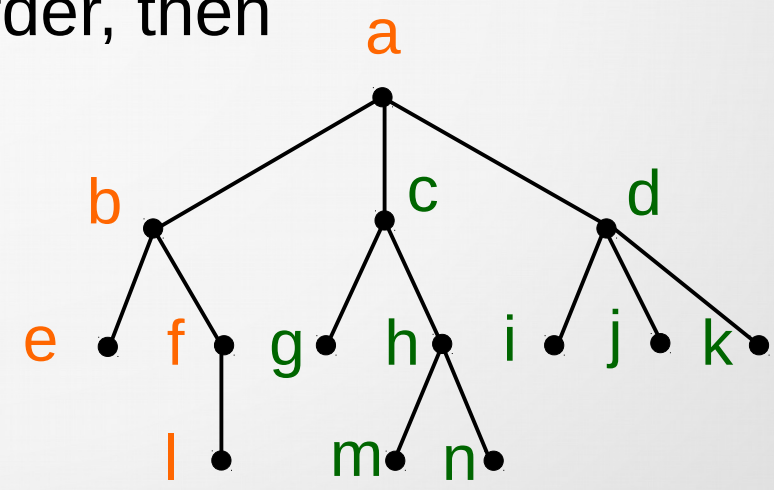
11.3 *Tree traversal*

Inorder Traversal

Let T be an ordered rooted tree with root *r*, that has *n* children enumerated from left to right: T_1, ...,T_n . Then

inorder traversal procedure:

- begins by traversing T_1 in inorder, then
- visiting *r*, then
- traverse T_2 in inorder, then
- …
- finishes by traversing T_n in inorder.



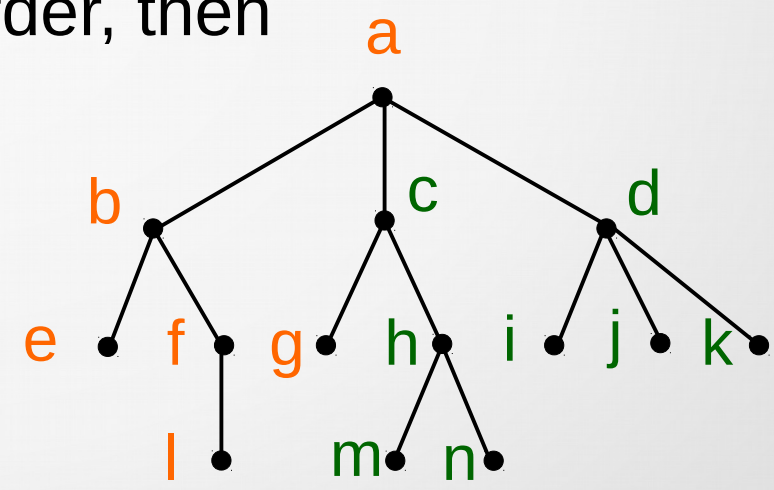
11.3 *Tree traversal*

Inorder Traversal

Let T be an ordered rooted tree with root *r*, that has *n* children enumerated from left to right: T_1, ...,T_n . Then

inorder traversal procedure:

- begins by traversing T_1 in inorder, then
- visiting *r*, then
- traverse T_2 in inorder, then
- …
- finishes by traversing T_n in inorder.



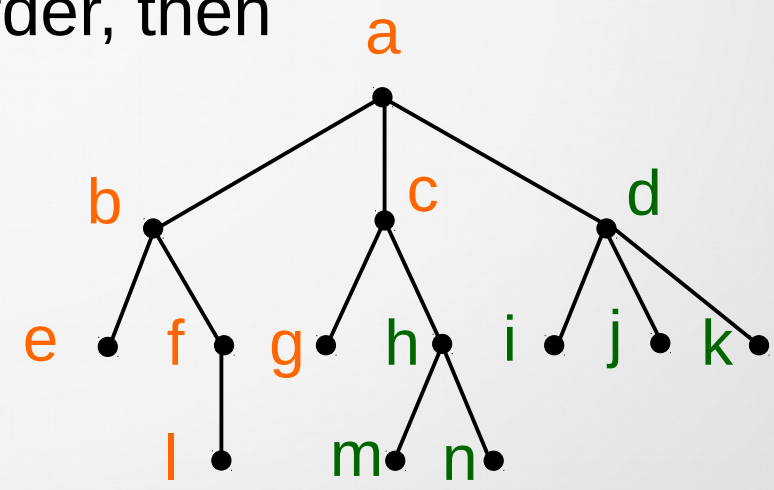
11.3 *Tree traversal*

Inorder Traversal

Let T be an ordered rooted tree with root *r*, that has *n* children enumerated from left to right: T_1, ...,T_n . Then

inorder traversal procedure:

- begins by traversing T_1 in inorder, then
- visiting *r*, then
- traverse T_2 in inorder, then
- …
- finishes by traversing T_n in inorder.



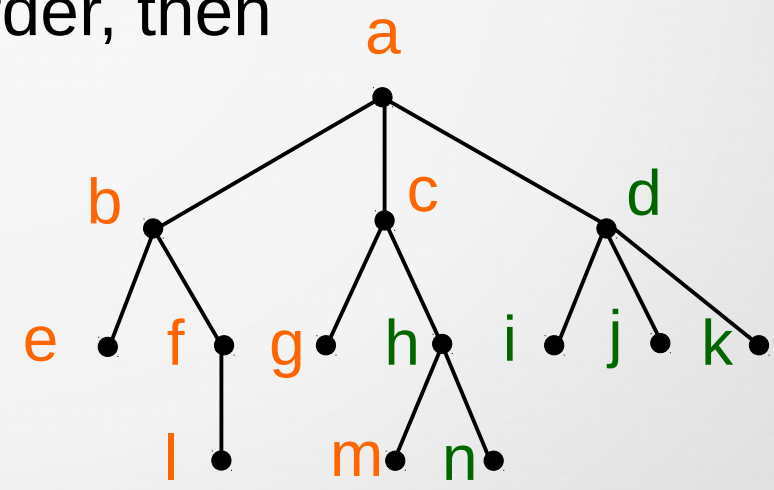
11.3 *Tree traversal*

Inorder Traversal

Let T be an ordered rooted tree with root *r*, that has *n* children enumerated from left to right: T_1, ...,T_n . Then

inorder traversal procedure:

- begins by traversing T_1 in inorder, then
- visiting *r*, then
- traverse T_2 in inorder, then
- …
- finishes by traversing T_n in inorder.



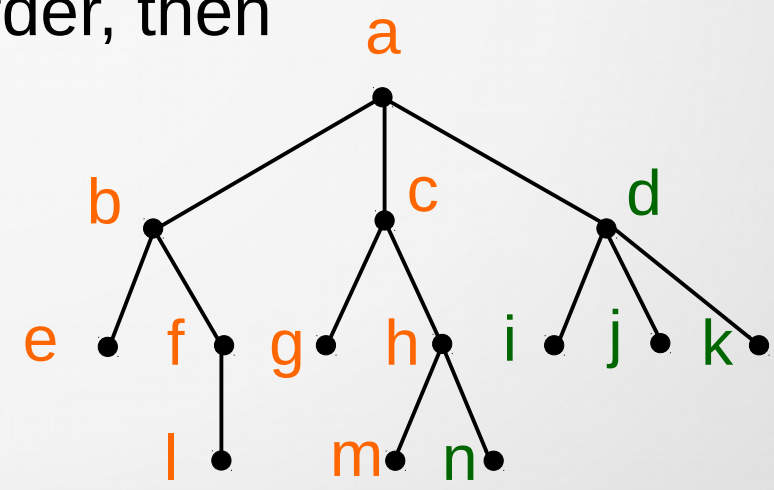
11.3 *Tree traversal*

Inorder Traversal

Let T be an ordered rooted tree with root *r*, that has *n* children enumerated from left to right: T_1, ...,T_n . Then

inorder traversal procedure:

- begins by traversing T_1 in inorder, then
- visiting *r*, then
- traverse T_2 in inorder, then
- …
- finishes by traversing T_n in inorder.



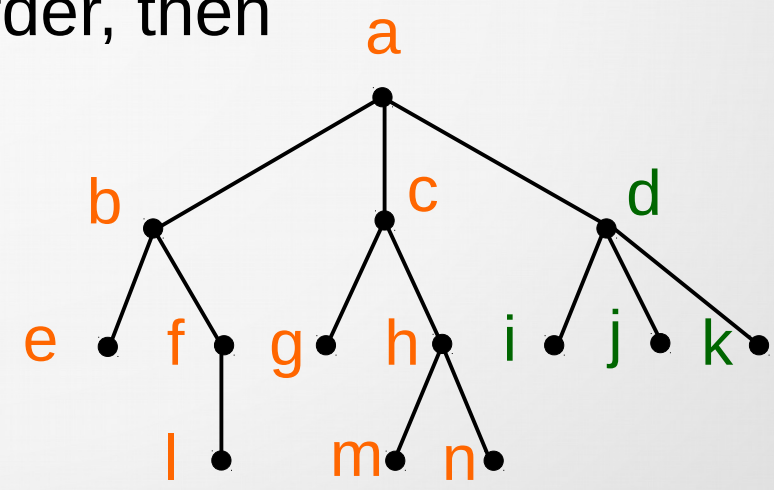
11.3 *Tree traversal*

Inorder Traversal

Let T be an ordered rooted tree with root *r*, that has *n* children enumerated from left to right: T_1, ...,T_n . Then

inorder traversal procedure:

- begins by traversing T_1 in inorder, then
- visiting *r*, then
- traverse T_2 in inorder, then
- …
- finishes by traversing T_n in inorder.



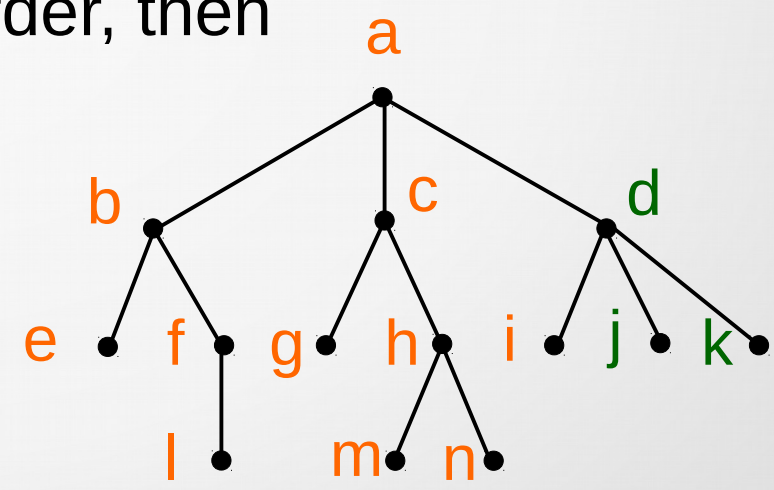
11.3 *Tree traversal*

Inorder Traversal

Let T be an ordered rooted tree with root *r*, that has *n* children enumerated from left to right: T_1, ...,T_n . Then

inorder traversal procedure:

- begins by traversing T_1 in inorder, then
- visiting *r*, then
- traverse T_2 in inorder, then
- …
- finishes by traversing T_n in inorder.



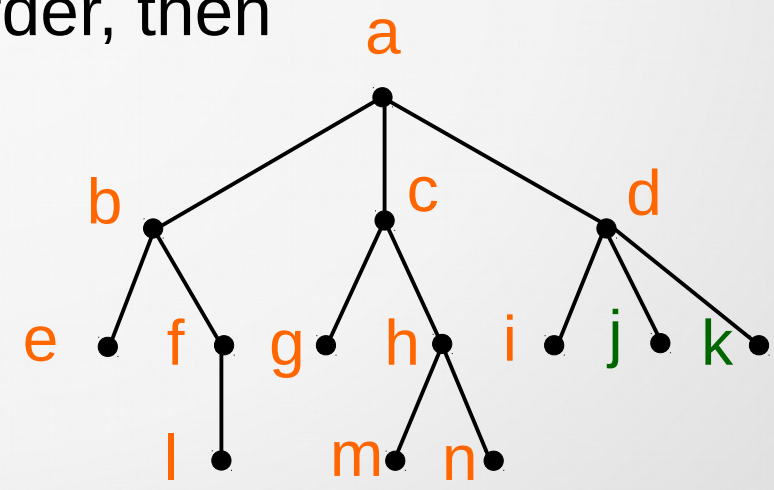
11.3 *Tree traversal*

Inorder Traversal

Let T be an ordered rooted tree with root *r*, that has *n* children enumerated from left to right: T_1, ...,T_n . Then

inorder traversal procedure:

- begins by traversing T_1 in inorder, then
- visiting *r*, then
- traverse T_2 in inorder, then
- …
- finishes by traversing T_n in inorder.



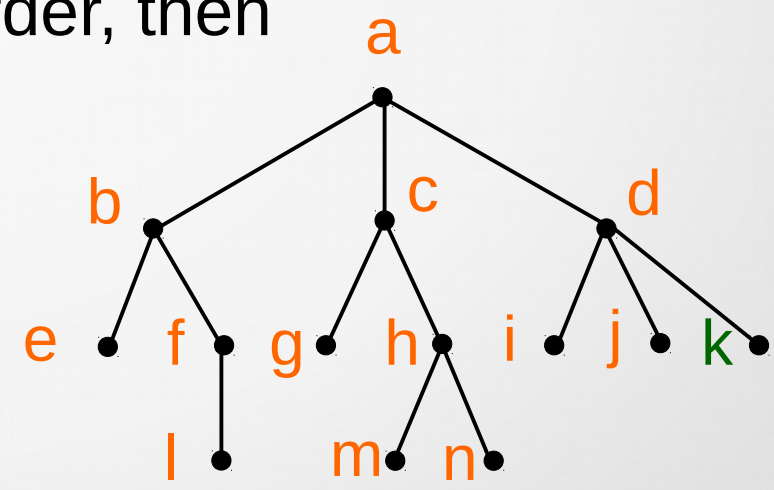
11.3 *Tree traversal*

Inorder Traversal

Let T be an ordered rooted tree with root *r*, that has *n* children enumerated from left to right: T_1, ...,T_n . Then

inorder traversal procedure:

- begins by traversing T_1 in inorder, then
- visiting *r*, then
- traverse T_2 in inorder, then
- …
- finishes by traversing T_n in inorder.



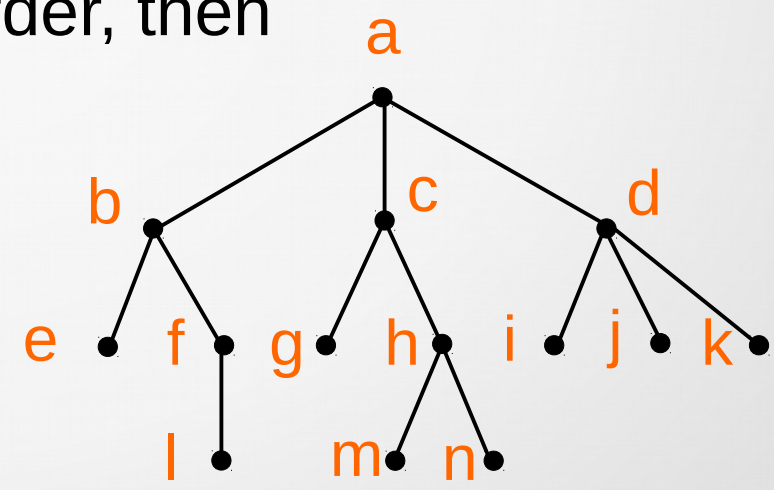
11.3 *Tree traversal*

Inorder Traversal

Let T be an ordered rooted tree with root *r*, that has *n* children enumerated from left to right: T_1, ...,T_n . Then

inorder traversal procedure:

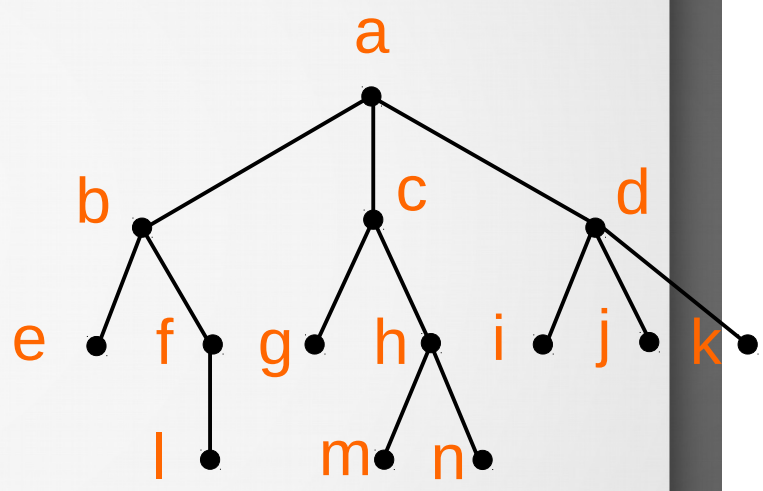
- begins by traversing T_1 in inorder, then
- visiting *r*, then
- traverse T_2 in inorder, then
- …
- finishes by traversing T_n in inorder.



11.3 *Tree traversal*

Inorder Traversal

procedure *inorder*(*T*: ordered rooted tree)
r := root of *T*
if r is a leaf, then list *r*
else
    l := first child of *r* from left to right
    T(l) := subtree with l as its root
    inorder(*T(l)*)
    list *r*
    for each child *c* of *r* from left to right, excluding *l*
        T(c) := subtree with *c* as its root
        inorder(*T(c)*)



11.3 *Tree traversal*

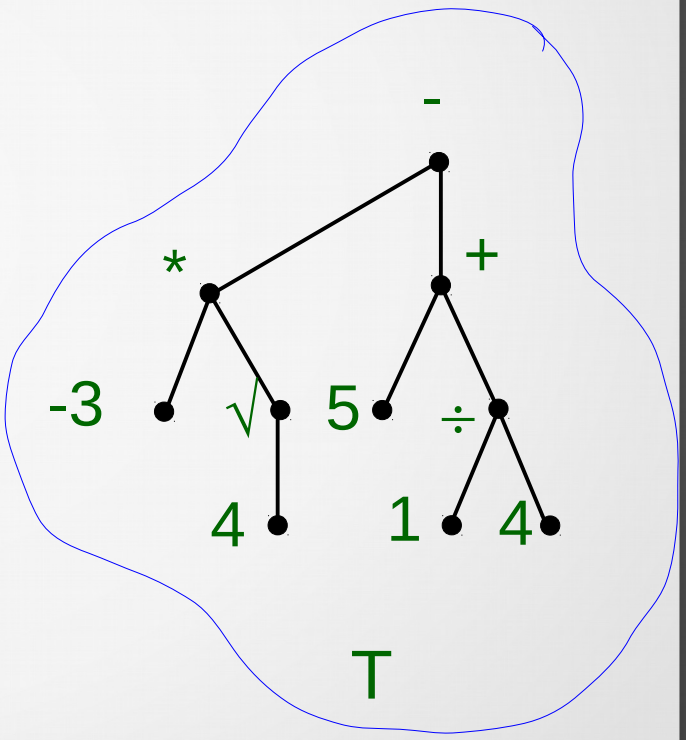
Inorder Traversal

Example 1: recall algebraic expressions

Expression $(-3)\sqrt{4}-(5+\frac{1}{4})$

is represented by the tree T.

The *inorder traversal* of the tree T will give us the expression above.



Note that:
leaves are *integers*, and
inner vertices are *operation symbols*.

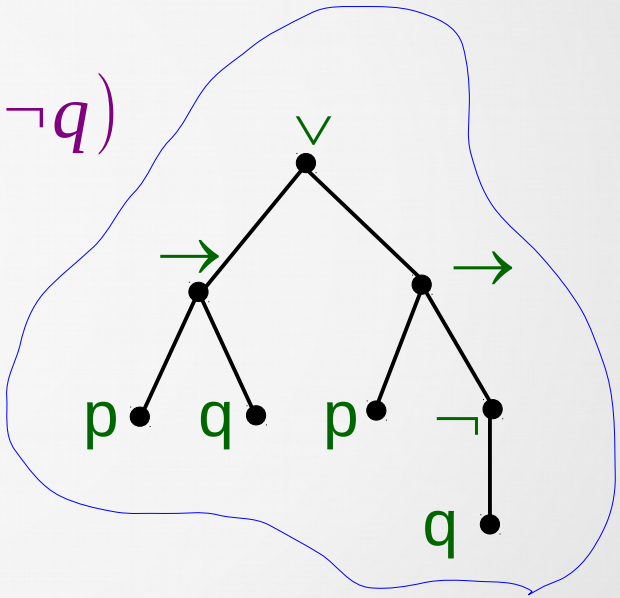
11.3 *Tree traversal*

Inorder Traversal

Example 2: recall Propositional Logic

Compound proposition $(p \rightarrow q) \vee (p \rightarrow \neg q)$

is represented by the tree S.



S

The *inorder traversal* of the tree T will give us the expression above.

Note that:
leaves are *propositional variables*, and
inner vertices are *logical connectives*.

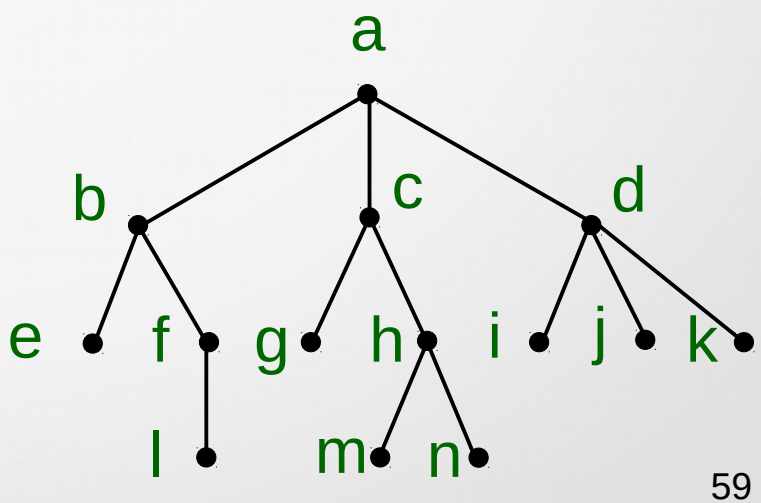
11.3 *Tree traversal*

Postorder Traversal

Let T be an ordered rooted tree with root *r*, that has *n* children enumerated from left to right: T_1, ...,T_n . Then

postorder traversal procedure:

- begins by traversing T_1 in postorder, then
- traverses T_2 in postorder, then
- …
- traverses T_n in postorder, then
- ends by visiting *r*.



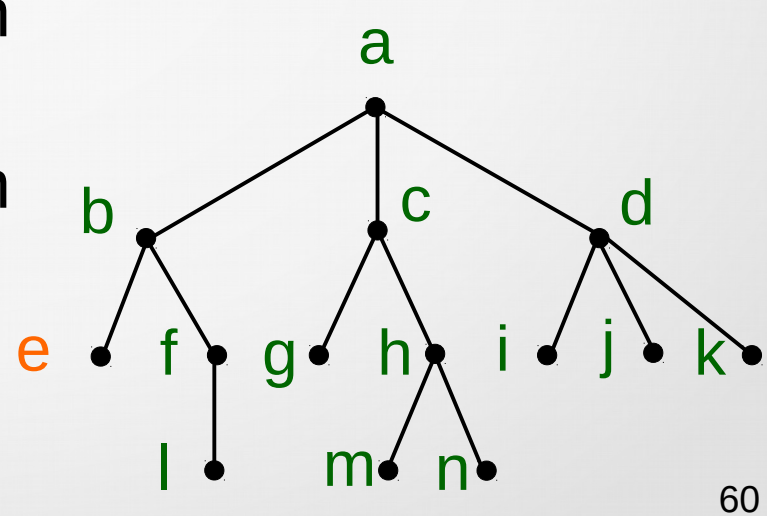
11.3 *Tree traversal*

Postorder Traversal

Let T be an ordered rooted tree with root *r*, that has *n* children enumerated from left to right: T_1, ...,T_n . Then

postorder traversal procedure:

- begins by traversing T_1 in postorder, then
- traverses T_2 in postorder, then
- …
- traverses T_n in postorder, then
- ends by visiting *r*.



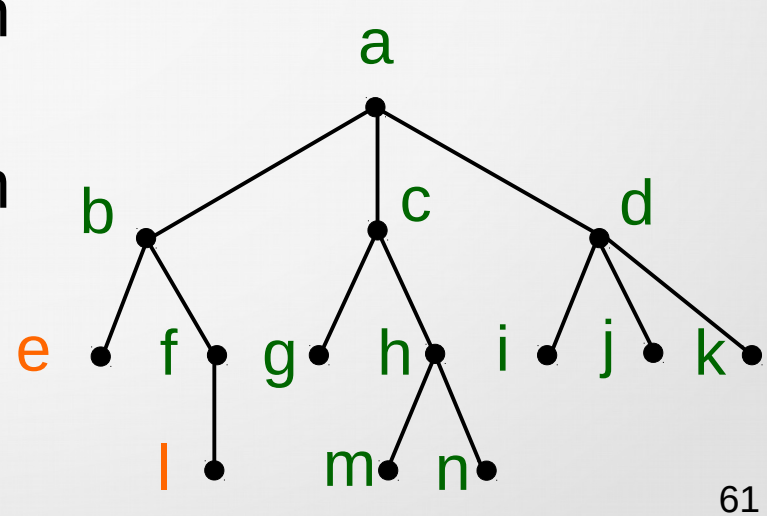
11.3 *Tree traversal*

Postorder Traversal

Let T be an ordered rooted tree with root *r*, that has *n* children enumerated from left to right: T_1, ...,T_n . Then

postorder traversal procedure:

- begins by traversing T_1 in postorder, then
- traverses T_2 in postorder, then
- …
- traverses T_n in postorder, then
- ends by visiting *r*.



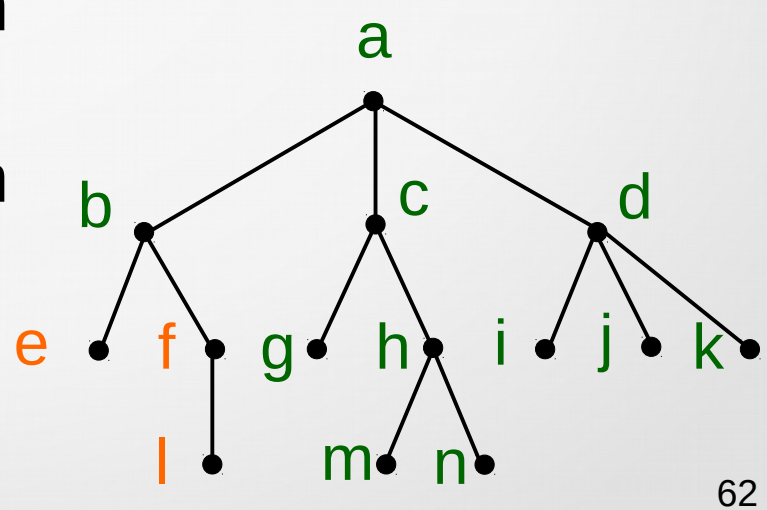
11.3 *Tree traversal*

Postorder Traversal

Let T be an ordered rooted tree with root *r*, that has *n* children enumerated from left to right: T_1, ...,T_n . Then

postorder traversal procedure:

- begins by traversing T_1 in postorder, then
- traverses T_2 in postorder, then
- …
- traverses T_n in postorder, then
- ends by visiting *r*.



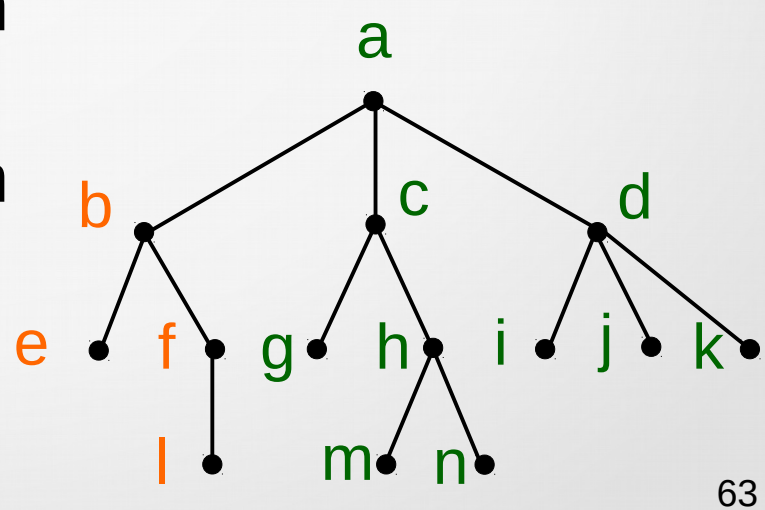
11.3 *Tree traversal*

Postorder Traversal

Let T be an ordered rooted tree with root *r*, that has *n* children enumerated from left to right: T_1, ...,T_n . Then

postorder traversal procedure:

- begins by traversing T_1 in postorder, then
- traverses T_2 in postorder, then
- …
- traverses T_n in postorder, then
- ends by visiting *r*.



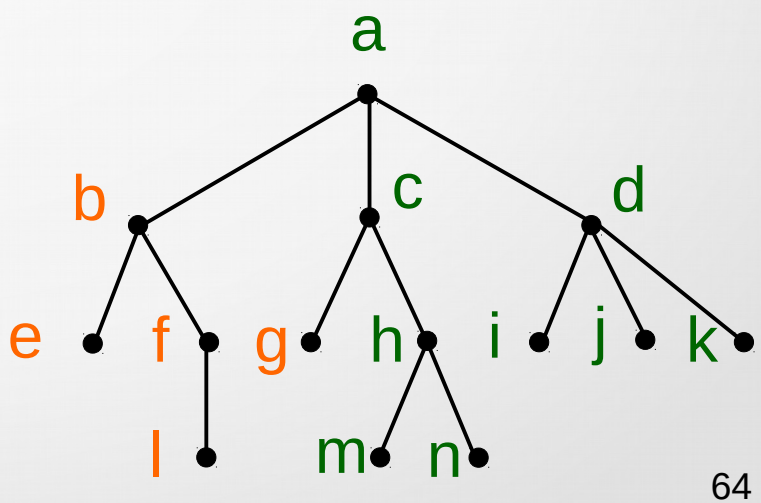
11.3 *Tree traversal*

Postorder Traversal

Let T be an ordered rooted tree with root *r*, that has *n* children enumerated from left to right: T_1, ...,T_n . Then

postorder traversal procedure:

- begins by traversing T_1 in postorder, then
- traverses T_2 in postorder, then
- …
- traverses T_n in postorder, then
- ends by visiting *r*.



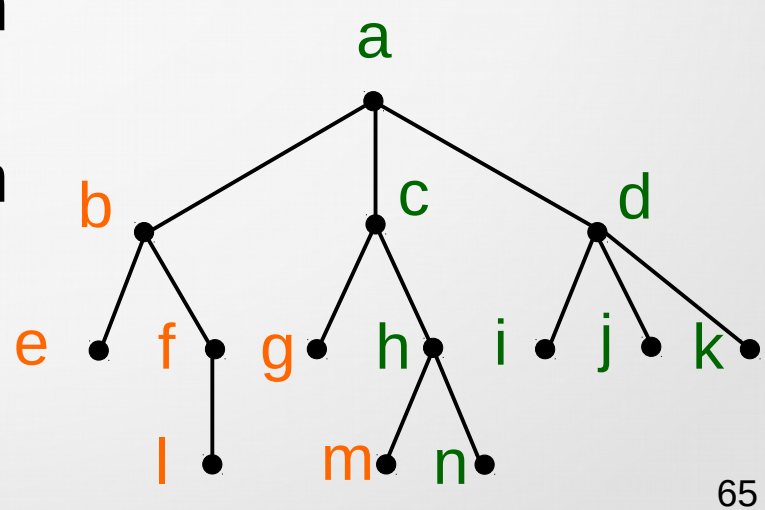
11.3 *Tree traversal*

Postorder Traversal

Let T be an ordered rooted tree with root *r*, that has *n* children enumerated from left to right: T_1, ...,T_n . Then

postorder traversal procedure:

- begins by traversing T_1 in postorder, then
- traverses T_2 in postorder, then
- …
- traverses T_n in postorder, then
- ends by visiting *r*.



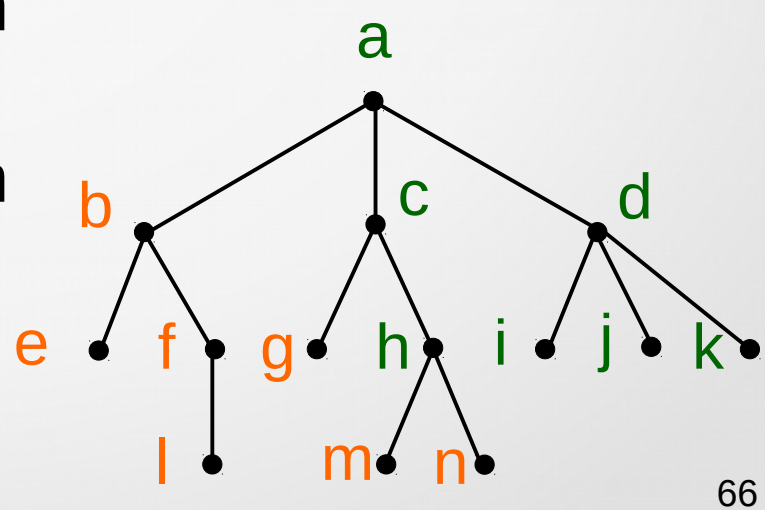
11.3 *Tree traversal*

Postorder Traversal

Let T be an ordered rooted tree with root *r*, that has *n* children enumerated from left to right: T_1, ...,T_n . Then

postorder traversal procedure:

- begins by traversing T_1 in postorder, then
- traverses T_2 in postorder, then
- …
- traverses T_n in postorder, then
- ends by visiting *r*.



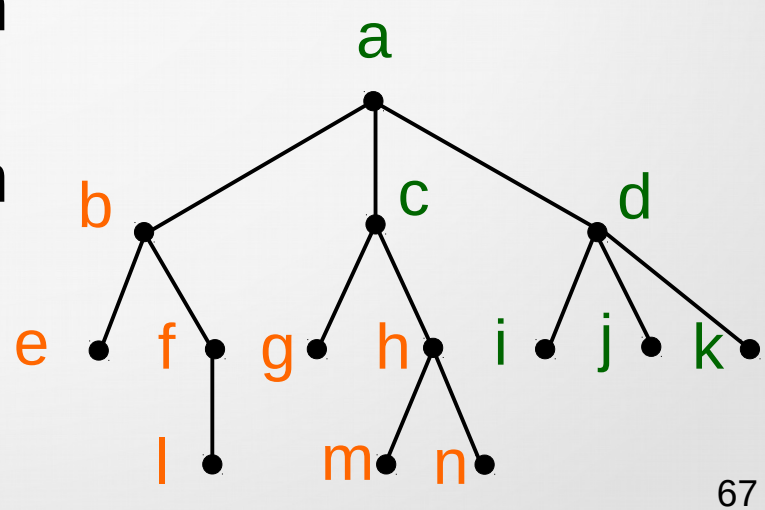
11.3 *Tree traversal*

Postorder Traversal

Let T be an ordered rooted tree with root *r*, that has *n* children enumerated from left to right: T_1, ...,T_n . Then

postorder traversal procedure:

- begins by traversing T_1 in postorder, then
- traverses T_2 in postorder, then
- …
- traverses T_n in postorder, then
- ends by visiting *r*.



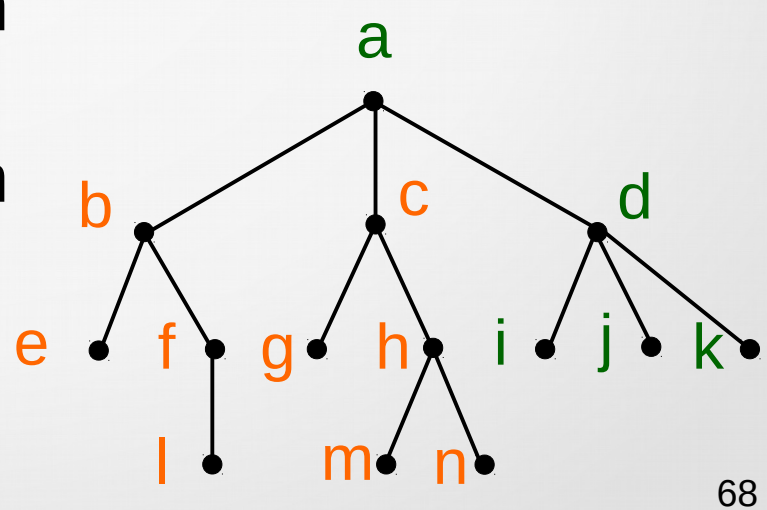
11.3 *Tree traversal*

Postorder Traversal

Let T be an ordered rooted tree with root *r*, that has *n* children enumerated from left to right: T_1, ...,T_n . Then

postorder traversal procedure:

- begins by traversing T_1 in postorder, then
- traverses T_2 in postorder, then
- …
- traverses T_n in postorder, then
- ends by visiting *r*.



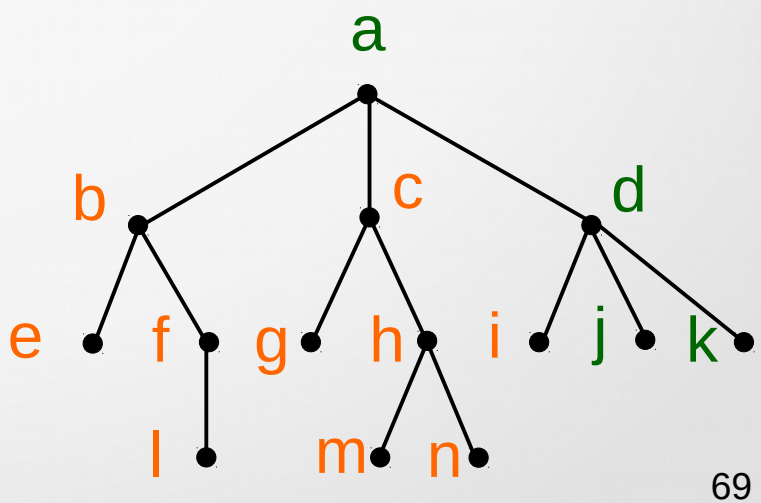
11.3 *Tree traversal*

Postorder Traversal

Let T be an ordered rooted tree with root *r*, that has *n* children enumerated from left to right: T_1, ...,T_n . Then

postorder traversal procedure:

- begins by traversing T_1 in postorder, then
- traverses T_2 in postorder, then
- …
- traverses T_n in postorder, then
- ends by visiting *r*.



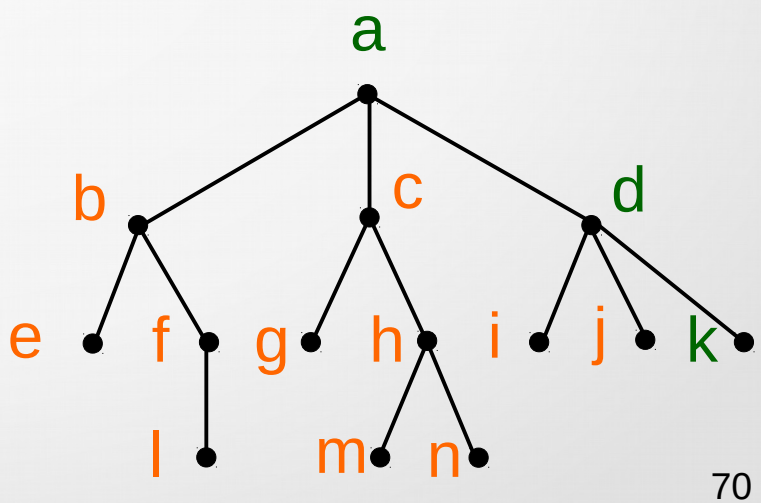
11.3 *Tree traversal*

Postorder Traversal

Let T be an ordered rooted tree with root *r*, that has *n* children enumerated from left to right: T_1, ...,T_n . Then

postorder traversal procedure:

- begins by traversing T_1 in postorder, then
- traverses T_2 in postorder, then
- …
- traverses T_n in postorder, then
- ends by visiting *r*.



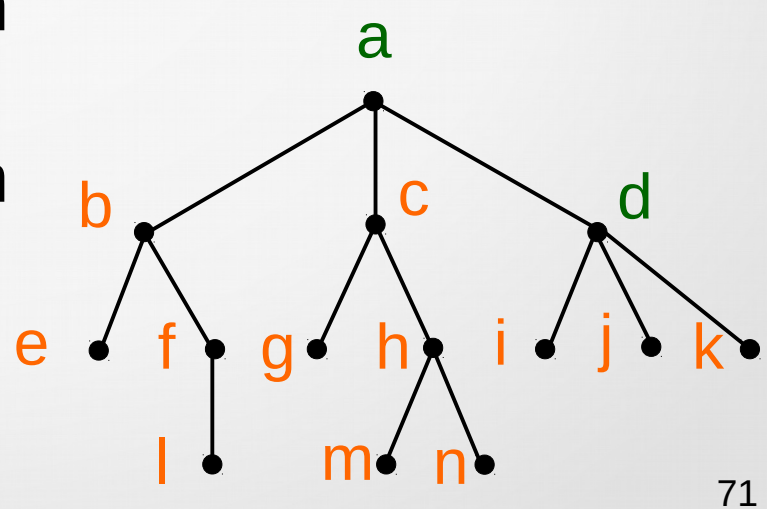
11.3 *Tree traversal*

Postorder Traversal

Let T be an ordered rooted tree with root *r*, that has *n* children enumerated from left to right: T_1, ...,T_n . Then

postorder traversal procedure:

- begins by traversing T_1 in postorder, then
- traverses T_2 in postorder, then
- …
- traverses T_n in postorder, then
- ends by visiting *r*.



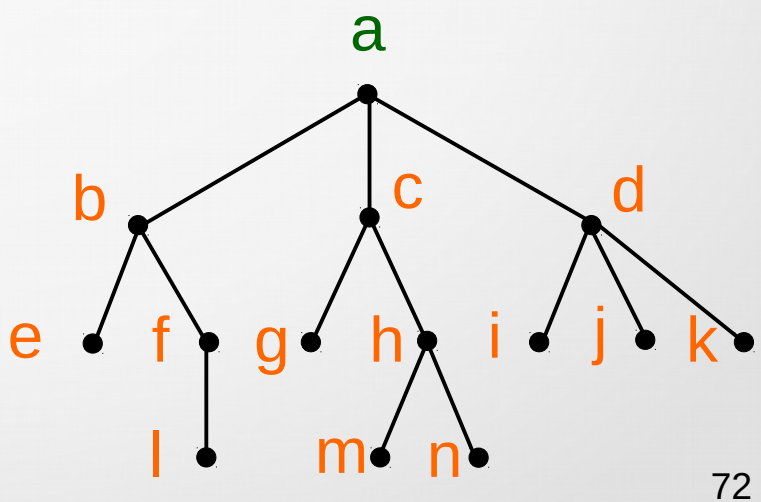
11.3 *Tree traversal*

Postorder Traversal

Let T be an ordered rooted tree with root *r*, that has *n* children enumerated from left to right: T_1, ...,T_n . Then

postorder traversal procedure:

- begins by traversing T_1 in postorder, then
- traverses T_2 in postorder, then
- …
- traverses T_n in postorder, then
- ends by visiting *r*.



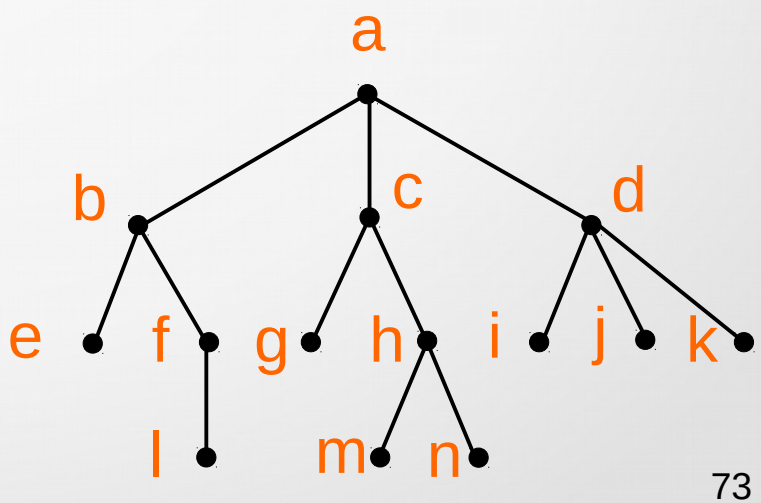
11.3 *Tree traversal*

Postorder Traversal

Let T be an ordered rooted tree with root *r*, that has *n* children enumerated from left to right: T_1, ...,T_n . Then

postorder traversal procedure:

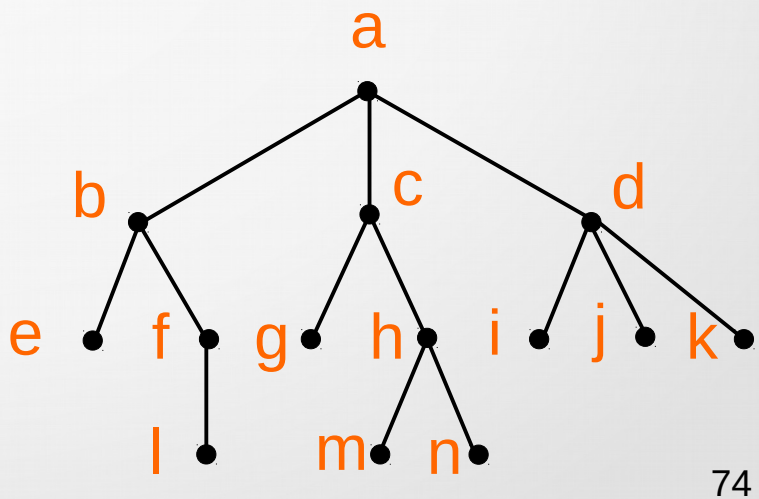
- begins by traversing T_1 in postorder, then
- traverses T_2 in postorder, then
- …
- traverses T_n in postorder, then
- ends by visiting *r*.



11.3 *Tree traversal*

Postorder Traversal

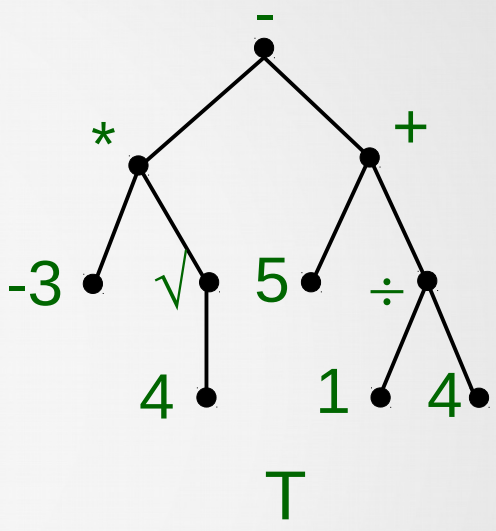
procedure *postorder*(*T*: ordered rooted tree)
r := root of *T*
for each child *c* of *r* from left to right
    T(c) := subtree with *c* as its root
    postorder(*T(c)*)
list *r*



11.3 *Tree traversal*

Infix, Prefix, and Postfix Notation

recall **Example 1**:



T

An *inorder traversal* of T (fully parenthesised) produces the expression in *infix notation*:

$$(-3)\sqrt{4}-(5+\frac{1}{4})$$

The *preorder traversal* of T produces the expression in *prefix notation*: $-*(-3)\sqrt{}\,4+5\div 1\,4$

The *postorder traversal* of T produces the expression in *postfix notation (Polish notation)*: $(-3)\,4\sqrt{}\,*5\;1\;4\div+-$

was named after Polish logician Jan Łukasiewicz

11.3 *Tree traversal*

Infix, Prefix, and Postfix Notation

the expression in *prefix notation*:

$$-*(-3)\sqrt{\ }\,4+5\div 1\,4$$

the expression in *postfix notation (Polish notation)*:

$$(-3)\,4\sqrt{\ }\,*5\;1\;4\;\div+-$$

11.3 *Tree traversal*

Infix, Prefix, and Postfix Notation

the expression in *prefix notation*:

$$-*(-3)\sqrt{\ }\ 4+5\div 1\ 4$$

the expression in *postfix notation (Polish notation)*:

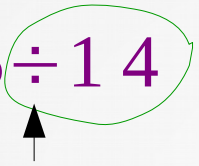
$$(-3)\ 4\sqrt{\ }\ *5\ 1\ 4\ \div+-$$

11.3 *Tree traversal*

Infix, Prefix, and Postfix Notation

the expression in *prefix notation*:

$$-*(-3)\sqrt{\ }\ 4+5\div 1\ 4$$



the expression in *postfix notation (Polish notation)*:

$$(-3)\ 4\sqrt{\ }\ *5\ 1\ 4\ \div+-$$

11.3 *Tree traversal*

Infix, Prefix, and Postfix Notation

the expression in *prefix notation*:

$$-*(-3)\sqrt{\ }4+5\div 1\ 4$$

$$-*(-3)\sqrt{\ }4+5\left(\frac{1}{4}\right)$$

the expression in *postfix notation (Polish notation)*:

$$(-3)\ 4\sqrt{\ }*5\ 1\ 4\ \div+-$$

11.3 *Tree traversal*

Infix, Prefix, and Postfix Notation

the expression in *prefix notation*:

$$-*(-3)\sqrt{\ }\,4+5\div 1\,4$$

$$-*(-3)\sqrt{\ }\,4+5\left(\frac{1}{4}\right)$$

the expression in *postfix notation (Polish notation)*:

$$(-3)\,4\sqrt{\ }\,*5\;1\;4\;\div+-$$

11.3 *Tree traversal*

Infix, Prefix, and Postfix Notation

the expression in *prefix notation*:

$$-*(-3)\sqrt{\ }\,4+5\div 1\,4$$

$$-*(-3)\sqrt{\ }\,4+5\left(\frac{1}{4}\right)$$

the expression in *postfix notation (Polish notation)*:

$$(-3)\,4\sqrt{\ }\,*5\;1\;4\;\div+-$$

11.3 *Tree traversal*

Infix, Prefix, and Postfix Notation

the expression in *prefix notation*:

$$-*(-3)\sqrt{\cdot}\,4+5\div 1\,4$$

$$-*(-3)\sqrt{\cdot}\,4+5\left(\frac{1}{4}\right)$$

$$-*(-3)\sqrt{\cdot}\,4\left(5+\frac{1}{4}\right)$$

the expression in *postfix notation (Polish notation)*:

$$(-3)\,4\sqrt{\cdot}\,*5\;1\;4\;\div+-$$

11.3 *Tree traversal*

Infix, Prefix, and Postfix Notation

the expression in *prefix notation*:

$$-*(-3)\sqrt{\cdot}\,4+5\div 1\,4$$

$$-*(-3)\sqrt{\cdot}\,4+5\left(\frac{1}{4}\right)$$

$$-*(-3)\sqrt{\cdot}\,4\,\left(5+\frac{1}{4}\right)$$

the expression in *postfix notation (Polish notation)*:

$$(-3)\,4\sqrt{\cdot}\,*5\;1\;4\;\div+-$$

11.3 *Tree traversal*

Infix, Prefix, and Postfix Notation

the expression in *prefix notation*:

$$-*(-3)\sqrt{\cdot}\,4+5\div 1\,4$$

$$-*(-3)\sqrt{\cdot}\,4+5\left(\frac{1}{4}\right)$$

$$-*(-3)\sqrt{\cdot}\,4\left(5+\frac{1}{4}\right)$$

the expression in *postfix notation (Polish notation)*:

$$(-3)\,4\sqrt{\cdot}\,*5\;1\;4\;\div+-$$

11.3 *Tree traversal*

Infix, Prefix, and Postfix Notation

the expression in *prefix notation*:

$$-*(-3)\sqrt{\cdot}\,4+5\div 1\,4$$

$$-*(-3)\sqrt{\cdot}\,4+5\left(\frac{1}{4}\right)$$

$$-*(-3)\sqrt{\cdot}\,4\left(5+\frac{1}{4}\right)$$

$$-*(-3)\sqrt{4}\left(5+\frac{1}{4}\right)$$

the expression in *postfix notation (Polish notation)*:

$$(-3)\,4\sqrt{\cdot}\,*5\;1\;4\;\div+-$$

11.3 *Tree traversal*

Infix, Prefix, and Postfix Notation

the expression in *prefix notation*:

$$-*(-3)\sqrt{\cdot}\,4+5\div 1\,4$$

$$-*(-3)\sqrt{\cdot}\,4+5\left(\frac{1}{4}\right)$$

$$-*(-3)\sqrt{\cdot}\,4\left(5+\frac{1}{4}\right)$$

$$-*(-3)\sqrt{4}\left(5+\frac{1}{4}\right)$$

the expression in *postfix notation (Polish notation)*:

$$(-3)\,4\sqrt{\cdot}\,*5\;1\;4\;\div+-$$

11.3 *Tree traversal*

Infix, Prefix, and Postfix Notation

the expression in *prefix notation*:

$$-*(-3)\sqrt{\cdot}\,4+5\div 1\,4$$

$$-*(-3)\sqrt{\cdot}\,4+5\left(\frac{1}{4}\right)$$

$$-*(-3)\sqrt{\cdot}\,4\left(5+\frac{1}{4}\right)$$

$$-*(-3)\sqrt{4}\left(5+\frac{1}{4}\right)$$

the expression in *postfix notation (Polish notation)*:

$$(-3)\,4\sqrt{\cdot}\,*5\;1\;4\;\div+-$$

11.3 *Tree traversal*

Infix, Prefix, and Postfix Notation

the expression in *prefix notation*:

$$-*(-3)\sqrt{\cdot}\,4+5\div 1\,4$$

$$-*(-3)\sqrt{\cdot}\,4+5\left(\frac{1}{4}\right)$$

$$-*(-3)\sqrt{\cdot}\,4\left(5+\frac{1}{4}\right)$$

$$-*(-3)\sqrt{4}\left(5+\frac{1}{4}\right)$$

$$-((-3)*\sqrt{4})\left(5+\frac{1}{4}\right)$$

the expression in *postfix notation (Polish notation)*:

$$(-3)\,4\sqrt{\cdot}\,*5\;1\;4\;\div+-$$

11.3 *Tree traversal*

Infix, Prefix, and Postfix Notation

the expression in *prefix notation*:

$$-*(-3)\sqrt{\ }4+5\div 1\ 4$$

$$-*(-3)\sqrt{\ }4+5(\frac{1}{4})$$

$$-*(-3)\sqrt{\ }4(5+\frac{1}{4})$$

$$-*(-3)\sqrt{4}(5+\frac{1}{4})$$

$$-((-3)*\sqrt{4})(5+\frac{1}{4})$$

the expression in *postfix notation (Polish notation)*:

$$(-3)\ 4\sqrt{\ }*5\ 1\ 4\ \div+-$$

11.3 *Tree traversal*

Infix, Prefix, and Postfix Notation

the expression in *prefix notation*:

$$-*(-3)\sqrt{\ }4+5\div 1\ 4$$

$$-*(-3)\sqrt{\ }4+5(\frac{1}{4})$$

$$-*(-3)\sqrt{\ }4(5+\frac{1}{4})$$

$$-*(-3)\sqrt{4}(5+\frac{1}{4})$$

$$-((-3)*\sqrt{4})(5+\frac{1}{4})$$

$$(-3)*\sqrt{4}-(5+\frac{1}{4})$$

the expression in *postfix notation (Polish notation)*:

$$(-3)\ 4\sqrt{\ }*5\ 1\ 4\div+-$$

11.3 *Tree traversal*

Infix, Prefix, and Postfix Notation

the expression in *prefix notation*:

$$-*(-3)\sqrt{\ }\,4+5\div 1\,4$$

$$-*(-3)\sqrt{\ }\,4+5\left(\frac{1}{4}\right)$$

$$-*(-3)\sqrt{\ }\,4\left(5+\frac{1}{4}\right)$$

$$-*(-3)\sqrt{4}\left(5+\frac{1}{4}\right)$$

$$-\left((-3)*\sqrt{4}\right)\left(5+\frac{1}{4}\right)$$

$$(-3)*\sqrt{4}-\left(5+\frac{1}{4}\right)$$

the expression in *postfix notation (Polish notation)*:

$$(-3)\,4\sqrt{\ }\,*5\;1\;4\div+-$$

11.3 *Tree traversal*

Infix, Prefix, and Postfix Notation

the expression in *prefix notation*:

$$-*(-3)\sqrt{\ }4+5\div 1\ 4$$

$$-*(-3)\sqrt{\ }4+5(\frac{1}{4})$$

$$-*(-3)\sqrt{\ }4(5+\frac{1}{4})$$

$$-*(-3)\sqrt{4}(5+\frac{1}{4})$$

$$-((-3)*\sqrt{4})(5+\frac{1}{4})$$

$$(-3)*\sqrt{4}-(5+\frac{1}{4})$$

the expression in *postfix notation (Polish notation)*:

$$(-3)\ 4\sqrt{\ }*5\ 1\ 4\ \div+-$$

11.3 *Tree traversal*

Infix, Prefix, and Postfix Notation

the expression in *prefix notation*:

$$-*(-3)\sqrt{\cdot}\,4+5\div 1\,4$$

$$-*(-3)\sqrt{\cdot}\,4+5\left(\frac{1}{4}\right)$$

$$-*(-3)\sqrt{\cdot}\,4\left(5+\frac{1}{4}\right)$$

$$-*(-3)\sqrt{4}\left(5+\frac{1}{4}\right)$$

$$-((-3)*\sqrt{4})\left(5+\frac{1}{4}\right)$$

$$(-3)*\sqrt{4}-\left(5+\frac{1}{4}\right)$$

the expression in *postfix notation (Polish notation)*:

$$(-3)\,4\sqrt{\cdot}\,*5\;1\;4\div+-$$

11.3 *Tree traversal*

Infix, Prefix, and Postfix Notation

the expression in *prefix notation*:

$$-*(-3)\sqrt{\cdot}\,4+5\div 1\,4$$

$$-*(-3)\sqrt{\cdot}\,4+5\left(\frac{1}{4}\right)$$

$$-*(-3)\sqrt{\cdot}\,4\left(5+\frac{1}{4}\right)$$

$$-*(-3)\sqrt{4}\left(5+\frac{1}{4}\right)$$

$$-((-3)*\sqrt{4})\left(5+\frac{1}{4}\right)$$

$$(-3)*\sqrt{4}-\left(5+\frac{1}{4}\right)$$

the expression in *postfix notation (Polish notation)*:

$$(-3)\,4\sqrt{\cdot}\,*5\;1\;4\div+-$$

11.3 *Tree traversal*

Infix, Prefix, and Postfix Notation

the expression in *prefix notation*:

$$-*(-3)\sqrt{\ }4+5\div 1\ 4$$

$$-*(-3)\sqrt{\ }4+5(\frac{1}{4})$$

$$-*(-3)\sqrt{\ }4(5+\frac{1}{4})$$

$$-*(-3)\sqrt{4}(5+\frac{1}{4})$$

$$-((-3)*\sqrt{4})(5+\frac{1}{4})$$

$$(-3)*\sqrt{4}-(5+\frac{1}{4})$$

the expression in *postfix notation (Polish notation)*:

$$(-3)\ 4\sqrt{\ }*5\ 1\ 4\ \div+-$$

11.3 *Tree traversal*

Infix, Prefix, and Postfix Notation

the expression in *prefix notation*:

$$-*(-3)\sqrt{\ }\,4+5\div 1\,4$$

$$-*(-3)\sqrt{\ }\,4+5\left(\frac{1}{4}\right)$$

$$-*(-3)\sqrt{\ }\,4\left(5+\frac{1}{4}\right)$$

$$-*(-3)\sqrt{4}\left(5+\frac{1}{4}\right)$$

$$-\left((-3)*\sqrt{4}\right)\left(5+\frac{1}{4}\right)$$

$$(-3)*\sqrt{4}-\left(5+\frac{1}{4}\right)$$

the expression in *postfix notation (Polish notation)*:

$$(-3)\,4\sqrt{\ }\,*5\ 1\ 4\ \div+-$$

$$(-3)\ \sqrt{4}\ *5\ 1\ 4\ \div+-$$

11.3 *Tree traversal*

Infix, Prefix, and Postfix Notation

the expression in *prefix notation*:

$$-*(-3)\sqrt{\ }4+5\div 1\ 4$$

$$-*(-3)\sqrt{\ }4+5(\frac{1}{4})$$

$$-*(-3)\sqrt{\ }4(5+\frac{1}{4})$$

$$-*(-3)\sqrt{4}(5+\frac{1}{4})$$

$$-((-3)*\sqrt{4})(5+\frac{1}{4})$$

$$(-3)*\sqrt{4}-(5+\frac{1}{4})$$

the expression in *postfix notation (Polish notation)*:

$$(-3)4\sqrt{\ }*5\ 1\ 4\div+-$$

$$(-3)\sqrt{4}*5\ 1\ 4\div+-$$

11.3 *Tree traversal*

Infix, Prefix, and Postfix Notation

the expression in *prefix notation*:

$$-*(-3)\sqrt{\ }4+5\div 1\ 4$$

$$-*(-3)\sqrt{\ }4+5(\frac{1}{4})$$

$$-*(-3)\sqrt{\ }4(5+\frac{1}{4})$$

$$-*(-3)\sqrt{4}(5+\frac{1}{4})$$

$$-((-3)*\sqrt{4})(5+\frac{1}{4})$$

$$(-3)*\sqrt{4}-(5+\frac{1}{4})$$

the expression in *postfix notation (Polish notation)*:

$$(-3)\ 4\sqrt{\ }*5\ 1\ 4\div+-$$

$$(-3)\ \sqrt{4}\ *5\ 1\ 4\div+-$$

11.3 *Tree traversal*

Infix, Prefix, and Postfix Notation

the expression in *prefix notation*:

$$-*(-3)\sqrt{\ }\,4+5\div 1\,4$$

$$-*(-3)\sqrt{\ }\,4+5\left(\frac{1}{4}\right)$$

$$-*(-3)\sqrt{\ }\,4\left(5+\frac{1}{4}\right)$$

$$-*(-3)\sqrt{4}\left(5+\frac{1}{4}\right)$$

$$-\left((-3)*\sqrt{4}\right)\left(5+\frac{1}{4}\right)$$

$$(-3)*\sqrt{4}-\left(5+\frac{1}{4}\right)$$

the expression in *postfix notation (Polish notation)*:

$$(-3)\,4\sqrt{\ }*5\ 1\ 4\div+-$$

$$(-3)\,\sqrt{4}\,*5\ 1\ 4\div+-$$

$$(-3)*\sqrt{4}\ 5\ 1\ 4\div+-$$

11.3 *Tree traversal*

Infix, Prefix, and Postfix Notation

the expression in *prefix notation*:

$$-*(-3)\sqrt{\ }4+5\div 1\ 4$$

$$-*(-3)\sqrt{\ }4+5(\frac{1}{4})$$

$$-*(-3)\sqrt{\ }4(5+\frac{1}{4})$$

$$-*(-3)\sqrt{4}(5+\frac{1}{4})$$

$$-((-3)*\sqrt{4})(5+\frac{1}{4})$$

$$(-3)*\sqrt{4}-(5+\frac{1}{4})$$

the expression in *postfix notation (Polish notation)*:

$$(-3)4\sqrt{\ }*5\ 1\ 4\div+-$$

$$(-3)\sqrt{4}*5\ 1\ 4\div+-$$

$$(-3)*\sqrt{4}\ 5\ 1\ 4\div+-$$

11.3 *Tree traversal*

Infix, Prefix, and Postfix Notation

the expression in *prefix notation*:

$$-*(-3)\sqrt{\cdot}\,4+5\div 1\,4$$

$$-*(-3)\sqrt{\cdot}\,4+5\left(\frac{1}{4}\right)$$

$$-*(-3)\sqrt{\cdot}\,4\left(5+\frac{1}{4}\right)$$

$$-*(-3)\sqrt{4}\left(5+\frac{1}{4}\right)$$

$$-\left((-3)*\sqrt{4}\right)\left(5+\frac{1}{4}\right)$$

$$(-3)*\sqrt{4}-\left(5+\frac{1}{4}\right)$$

the expression in *postfix notation (Polish notation)*:

$$(-3)\,4\sqrt{\cdot}\,*5\;1\;4\div+-$$

$$(-3)\,\sqrt{4}\,*5\;1\;4\div+-$$

$$(-3)*\sqrt{4}\;5\;1\;4\div+-$$

11.3 *Tree traversal*

Infix, Prefix, and Postfix Notation

the expression in *prefix notation*:

$$-*(-3)\sqrt{\ }4+5\div 1\ 4$$

$$-*(-3)\sqrt{\ }4+5(\frac{1}{4})$$

$$-*(-3)\sqrt{\ }4(5+\frac{1}{4})$$

$$-*(-3)\sqrt{4}(5+\frac{1}{4})$$

$$-((-3)*\sqrt{4})(5+\frac{1}{4})$$

$$(-3)*\sqrt{4}-(5+\frac{1}{4})$$

the expression in *postfix notation (Polish notation)*:

$$(-3)4\sqrt{\ }*5\ 1\ 4\div+-$$

$$(-3)\sqrt{4}*5\ 1\ 4\div+-$$

$$(-3)*\sqrt{4}\ 5\ 1\ 4\div+-$$

11.3 *Tree traversal*

Infix, Prefix, and Postfix Notation

the expression in *prefix notation*:

$$-*(-3)\sqrt{\ }4+5\div 1\ 4$$

$$-*(-3)\sqrt{\ }4+5(\frac{1}{4})$$

$$-*(-3)\sqrt{\ }4(5+\frac{1}{4})$$

$$-*(-3)\sqrt{4}(5+\frac{1}{4})$$

$$-((-3)*\sqrt{4})(5+\frac{1}{4})$$

$$(-3)*\sqrt{4}-(5+\frac{1}{4})$$

the expression in *postfix notation (Polish notation)*:

$$(-3)4\sqrt{\ }*5\ 1\ 4\div+-$$

$$(-3)\sqrt{4}*5\ 1\ 4\div+-$$

$$(-3)*\sqrt{4}\ 5\ 1\ 4\div+-$$

11.3 *Tree traversal*

Infix, Prefix, and Postfix Notation

the expression in *prefix notation*:

$$-*(-3)\sqrt{\ }\,4+5\div 1\,4$$

$$-*(-3)\sqrt{\ }\,4+5\left(\frac{1}{4}\right)$$

$$-*(-3)\sqrt{\ }\,4\left(5+\frac{1}{4}\right)$$

$$-*(-3)\sqrt{4}\left(5+\frac{1}{4}\right)$$

$$-\left((-3)*\sqrt{4}\right)\left(5+\frac{1}{4}\right)$$

$$(-3)*\sqrt{4}-\left(5+\frac{1}{4}\right)$$

the expression in *postfix notation (Polish notation)*:

$$(-3)\,4\sqrt{\ }\,*5\ 1\ 4\ \div+-$$

$$(-3)\ \sqrt{4}\ *5\ 1\ 4\ \div+-$$

$$(-3)*\sqrt{4}\ 5\ 1\ 4\ \div+-$$

11.3 *Tree traversal*

Infix, Prefix, and Postfix Notation

the expression in *prefix notation*:

$$-*(-3)\sqrt{\ }\,4+5\div 1\,4$$

$$-*(-3)\sqrt{\ }\,4+5\left(\frac{1}{4}\right)$$

$$-*(-3)\sqrt{\ }\,4\left(5+\frac{1}{4}\right)$$

$$-*(-3)\sqrt{4}\left(5+\frac{1}{4}\right)$$

$$-\left((-3)*\sqrt{4}\right)\left(5+\frac{1}{4}\right)$$

$$(-3)*\sqrt{4}-\left(5+\frac{1}{4}\right)$$

the expression in *postfix notation (Polish notation)*:

$$(-3)\,4\sqrt{\ }*5\ 1\ 4\div+-$$

$$(-3)\,\sqrt{4}\,*5\ 1\ 4\div+-$$

$$(-3)*\sqrt{4}\ 5\ 1\ 4\div+-$$

$$(-3)*\sqrt{4}\ 5\ \frac{1}{4}\ +-$$

11.3 *Tree traversal*

Infix, Prefix, and Postfix Notation

the expression in *prefix notation*:

$$-*(-3)\sqrt{\ }\,4+5\div 1\,4$$

$$-*(-3)\sqrt{\ }\,4+5\left(\frac{1}{4}\right)$$

$$-*(-3)\sqrt{\ }\,4\left(5+\frac{1}{4}\right)$$

$$-*(-3)\sqrt{4}\left(5+\frac{1}{4}\right)$$

$$-\left((-3)*\sqrt{4}\right)\left(5+\frac{1}{4}\right)$$

$$(-3)*\sqrt{4}-\left(5+\frac{1}{4}\right)$$

the expression in *postfix notation (Polish notation)*:

$$(-3)\,4\sqrt{\ }\,*5\ 1\ 4\ \div+-$$

$$(-3)\ \sqrt{4}\ *5\ 1\ 4\ \div+-$$

$$(-3)*\sqrt{4}\ 5\ 1\ 4\ \div+-$$

$$(-3)*\sqrt{4}\ 5\ \frac{1}{4}\ +-$$

11.3 *Tree traversal*

Infix, Prefix, and Postfix Notation

the expression in *prefix notation*:

$$-*(-3)\sqrt{\ }\,4+5\div 1\,4$$

$$-*(-3)\sqrt{\ }\,4+5\left(\frac{1}{4}\right)$$

$$-*(-3)\sqrt{\ }\,4\left(5+\frac{1}{4}\right)$$

$$-*(-3)\sqrt{4}\left(5+\frac{1}{4}\right)$$

$$-\left((-3)*\sqrt{4}\right)\left(5+\frac{1}{4}\right)$$

$$(-3)*\sqrt{4}-\left(5+\frac{1}{4}\right)$$

the expression in *postfix notation (Polish notation)*:

$$(-3)\,4\sqrt{\ }\,*5\ 1\ 4\ \div+-$$

$$(-3)\ \sqrt{4}\ *5\ 1\ 4\ \div+-$$

$$(-3)*\sqrt{4}\ 5\ 1\ 4\ \div+-$$

$$(-3)*\sqrt{4}\ 5\ \frac{1}{4}\ +-$$

11.3 *Tree traversal*

Infix, Prefix, and Postfix Notation

the expression in *prefix notation*:

$$-*(-3)\sqrt{\ }4+5\div 1\ 4$$

$$-*(-3)\sqrt{\ }4+5(\frac{1}{4})$$

$$-*(-3)\sqrt{\ }4(5+\frac{1}{4})$$

$$-*(-3)\sqrt{4}(5+\frac{1}{4})$$

$$-((-3)*\sqrt{4})(5+\frac{1}{4})$$

$$(-3)*\sqrt{4}-(5+\frac{1}{4})$$

the expression in *postfix notation (Polish notation)*:

$$(-3)4\sqrt{\ }*5\ 1\ 4\div+-$$

$$(-3)\sqrt{4}*5\ 1\ 4\div+-$$

$$(-3)*\sqrt{4}\ 5\ 1\ 4\div+-$$

$$(-3)*\sqrt{4}\ 5\ \frac{1}{4}\ +-$$

$$(-3)*\sqrt{4}\ (5+\frac{1}{4})-$$

11.3 *Tree traversal*

Infix, Prefix, and Postfix Notation

the expression in *prefix notation*:

$$-*(-3)\sqrt{\cdot}\,4+5\div 1\,4$$

$$-*(-3)\sqrt{\cdot}\,4+5\left(\frac{1}{4}\right)$$

$$-*(-3)\sqrt{\cdot}\,4\left(5+\frac{1}{4}\right)$$

$$-*(-3)\sqrt{4}\left(5+\frac{1}{4}\right)$$

$$-\left((-3)*\sqrt{4}\right)\left(5+\frac{1}{4}\right)$$

$$(-3)*\sqrt{4}-\left(5+\frac{1}{4}\right)$$

the expression in *postfix notation (Polish notation)*:

$$(-3)\,4\sqrt{\cdot}\,*5\;1\;4\div+-$$

$$(-3)\,\sqrt{4}\,*5\;1\;4\div+-$$

$$(-3)*\sqrt{4}\;5\;1\;4\div+-$$

$$(-3)*\sqrt{4}\;5\;\frac{1}{4}\;+-$$

$$(-3)*\sqrt{4}\left(5+\frac{1}{4}\right)-$$

11.3 *Tree traversal*

Infix, Prefix, and Postfix Notation

the expression in *prefix notation*:

$$-*(-3)\sqrt{\ }4+5\div 1\ 4$$

$$-*(-3)\sqrt{\ }4+5(\frac{1}{4})$$

$$-*(-3)\sqrt{\ }4(5+\frac{1}{4})$$

$$-*(-3)\sqrt{4}(5+\frac{1}{4})$$

$$-((-3)*\sqrt{4})(5+\frac{1}{4})$$

$$(-3)*\sqrt{4}-(5+\frac{1}{4})$$

the expression in *postfix notation (Polish notation)*:

$$(-3)4\sqrt{\ }*5\ 1\ 4\div+-$$

$$(-3)\sqrt{4}*5\ 1\ 4\div+-$$

$$(-3)*\sqrt{4}\ 5\ 1\ 4\div+-$$

$$(-3)*\sqrt{4}\ 5\ \frac{1}{4}\ +-$$

$$(-3)*\sqrt{4}\ (5+\frac{1}{4})\ -$$

$$(-3)*\sqrt{4}-(5+\frac{1}{4})$$

11.3 *Tree traversal*

Infix, Prefix, and Postfix Notation

Prefix and *postfix* expressions are unambiguous and can be evaluated easily without scanning back and forth, hence they are used extensively in computer science.

In CSI 33: postfix notation expressions are evaluated using stacks.