Programming Languages and Techniques (CIS120)

Lecture 6

Jan 25, 2012

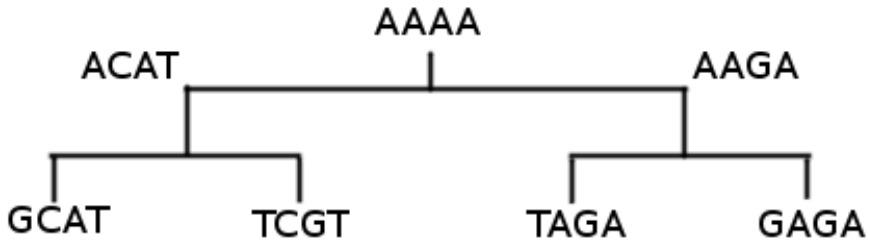
Binary Trees

Announcements

- Homework 2 is on the web pages.
  - On-time due date: Jan. 30 at 11:59:59pm
  - Get started early, and seek assistance if you get stuck!

DNA Computing Abstractions

- Nucleotide
  - Adenine (A), Guanine (G), Thymine (T), or Cytosine (C)
- Codon
  - three nucleotides : e.g. (A,A,T) or (T,G,C)
  - codons map to amino acids and other markers
- Helix
  - a sequence of nucleotides: e.g. AGTCCGATTACAGAGA…
- Phylogenetic tree
  - Binary (2-child) tree with helices (species) at the nodes and leaves



DNA Computing Abstractions

Simple datatypes

```
type nucleotide =
  | G  (* Guanine  *)
  | C  (* Cytosine *)
  | A  (* Adenine  *)
  | T  (* Thymine  *)
```

Tuples

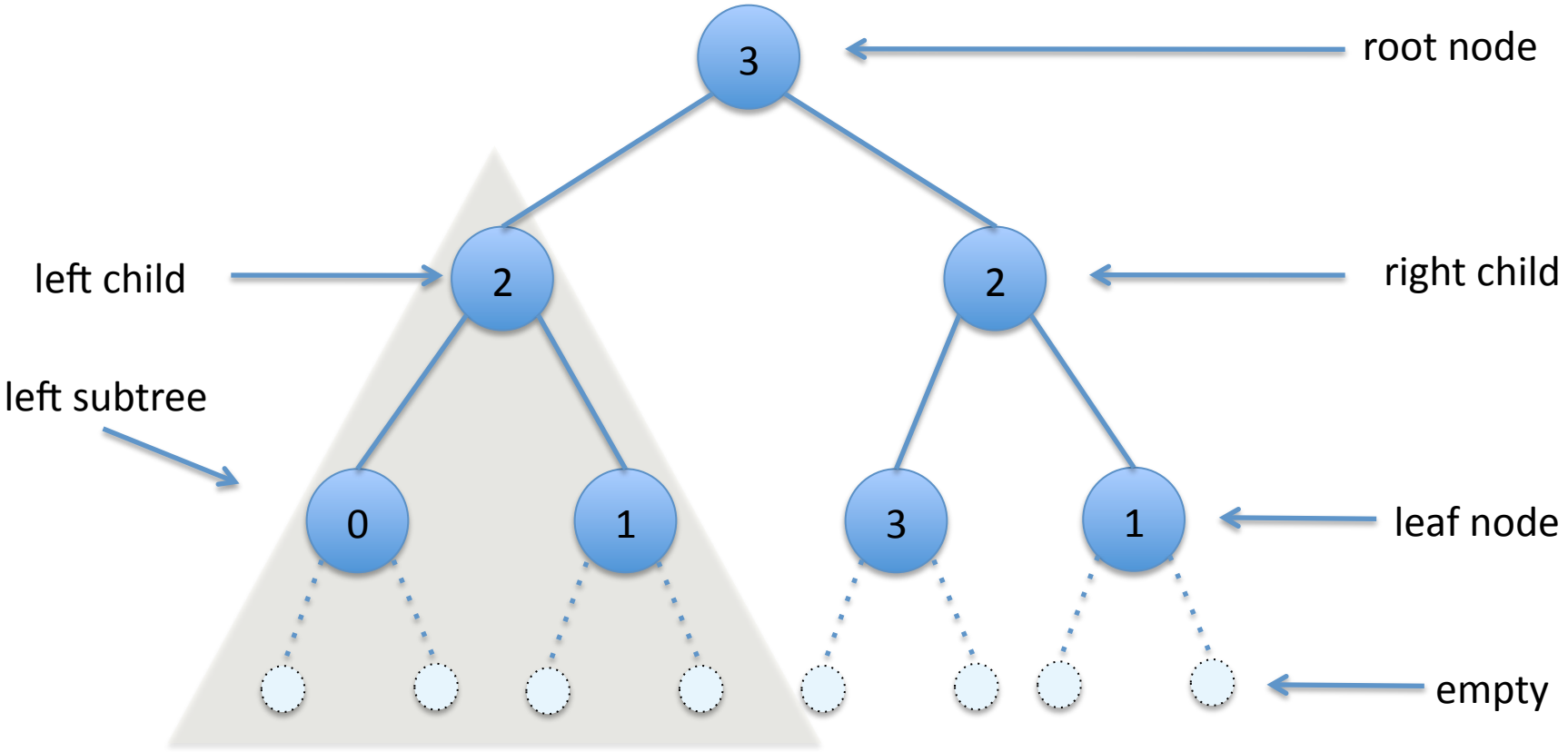
```
type codon = nucleotide * nucleotide * nucleotide

type helix = nucleotide list
```

Recursive datatypes

```
type tree =
  | Leaf of helix
  | Node of tree * helix * tree
```

Recap: Binary Trees



A binary tree is either *empty*, or a *node* with at most two children, both of which are also binary trees.

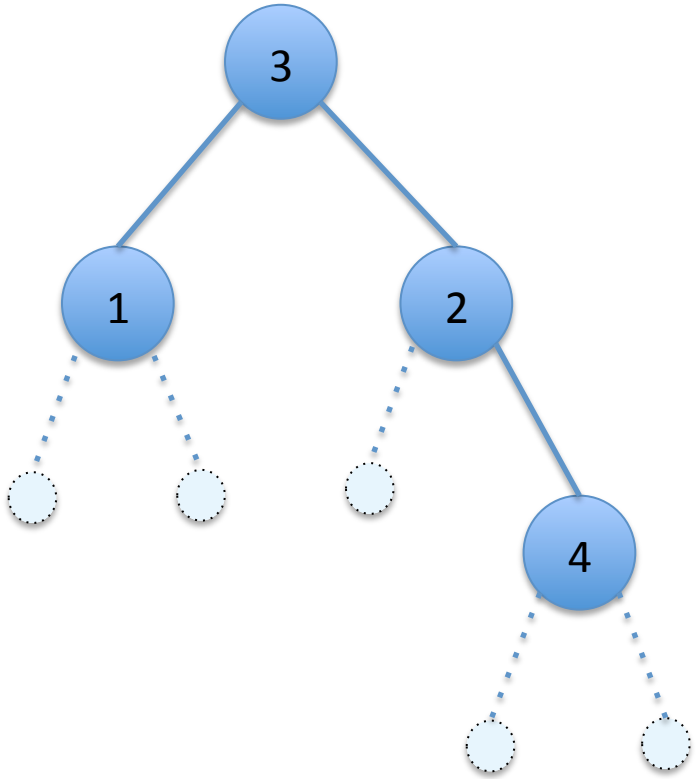
A *leaf* is a node whose children are both empty.

Integer Binary Trees in OCaml

```
type tree =
| Empty
| Node of tree * int * tree
```

```
let t : tree =
  Node (Node (Empty, 1, Empty),
    3,
    Node (Empty, 2,
      Node (Empty, 4, Empty)))
```

=

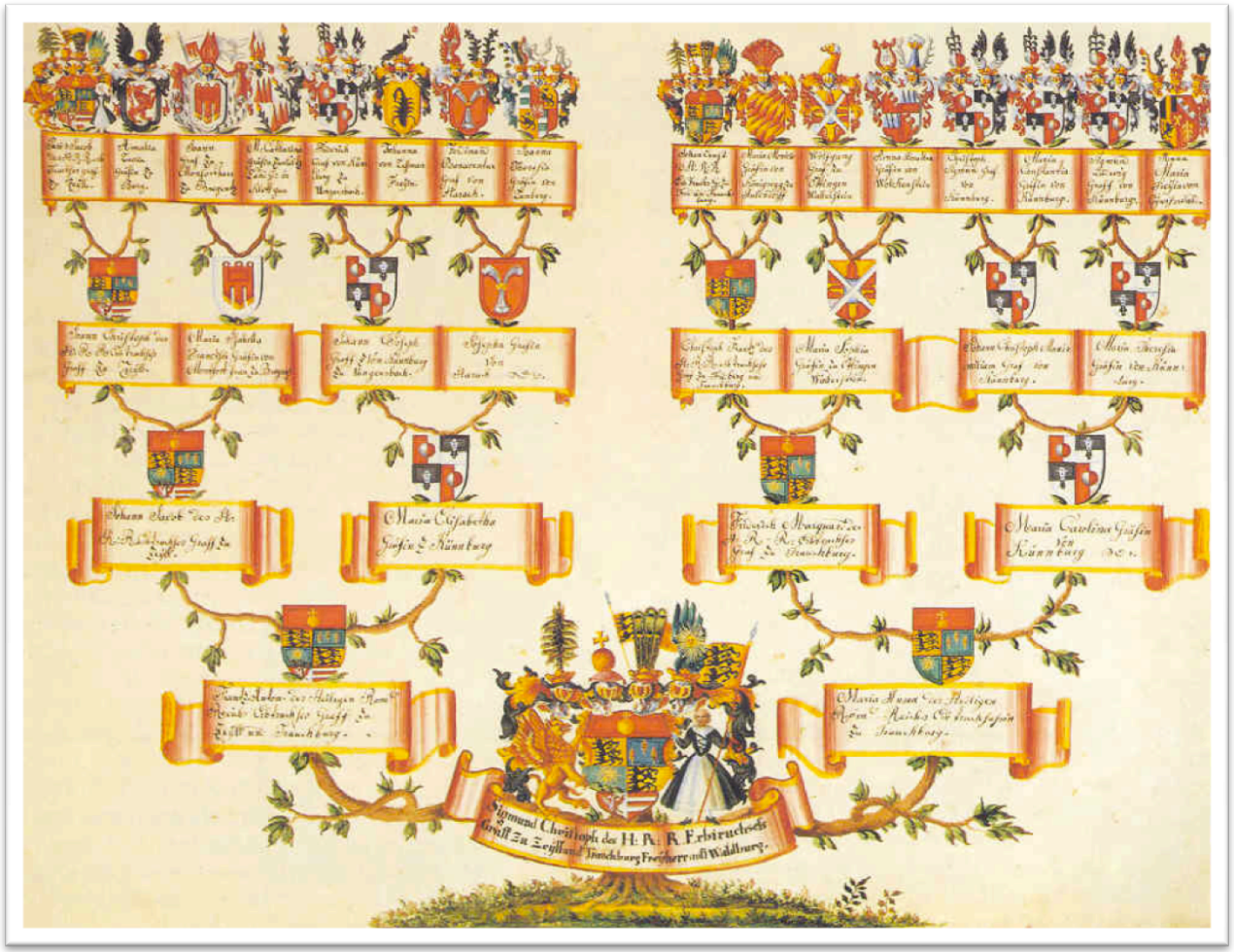


Demo

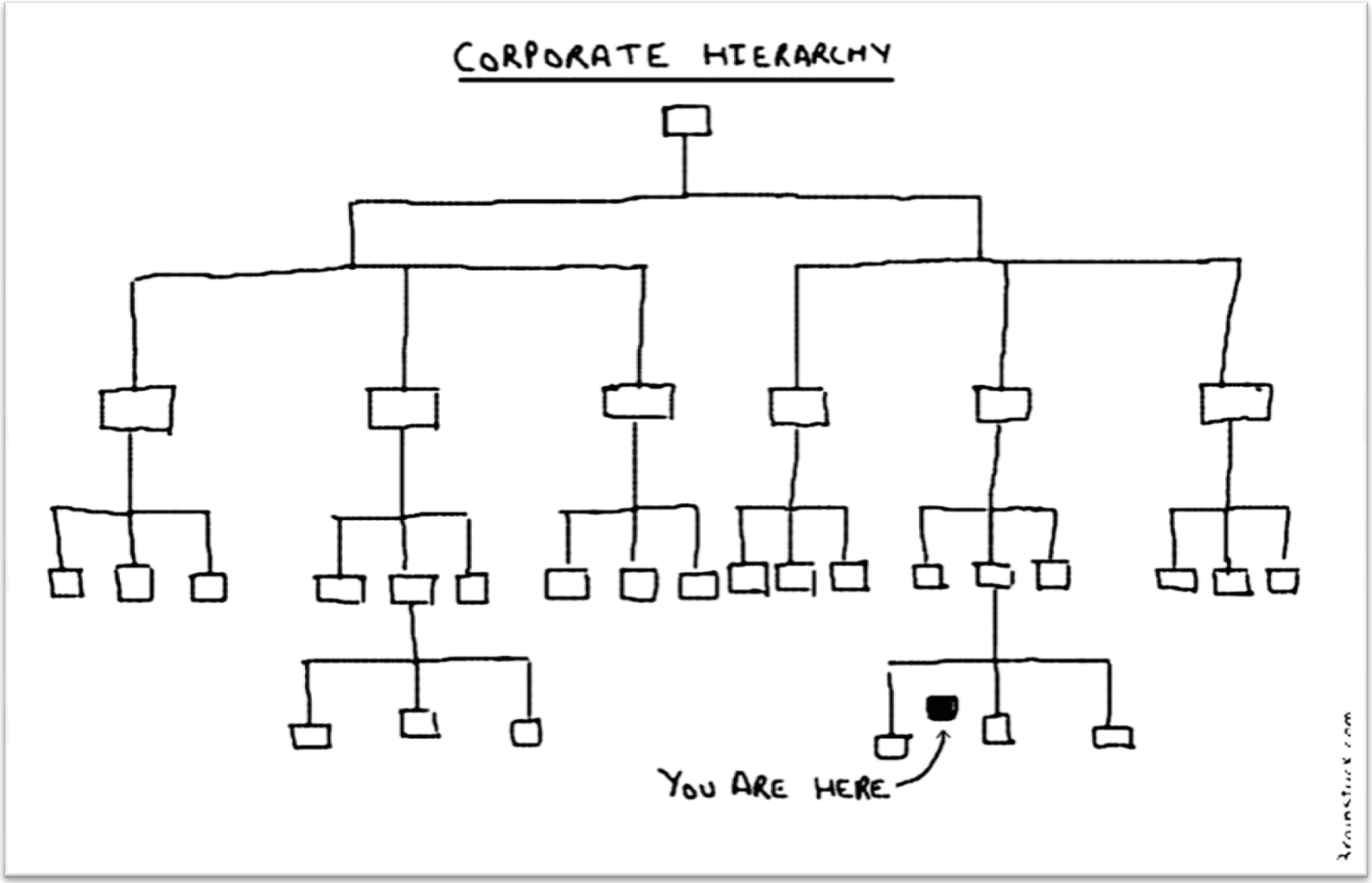
see demotree.ml

Other uses for trees?

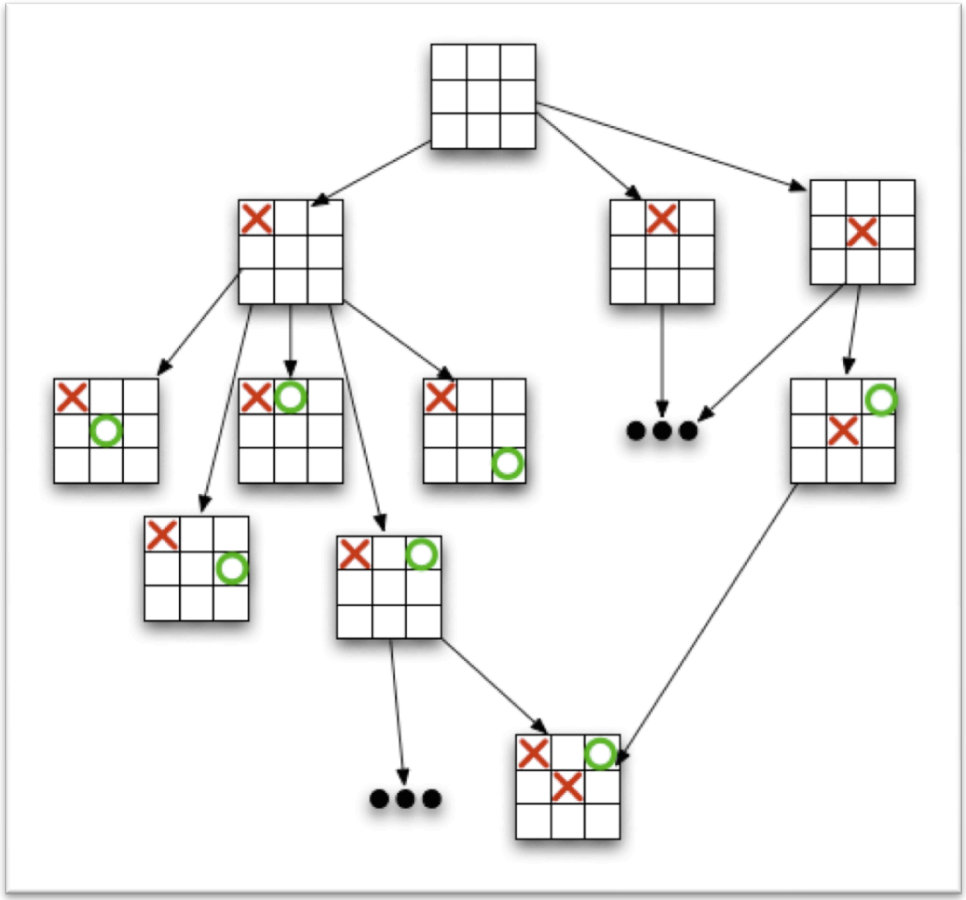
Family trees



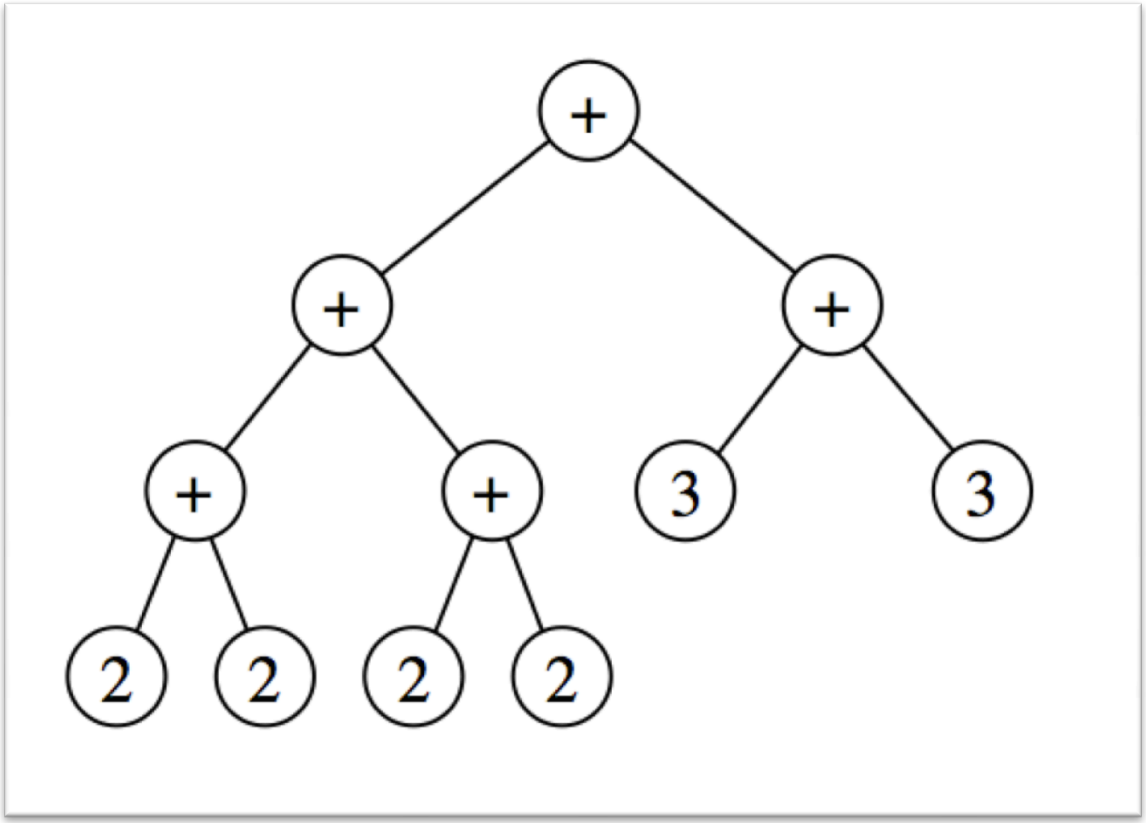
Organizational charts



Game trees



Expression trees



Trees as Containers

- Like lists, trees aggregate data
- Like lists, we can determine whether the data structure *contains* a particular element

- CHALLENGE: can we determine where a tree contains a particular element quickly?