A6

February 14, 2019

1 Induction on Trees

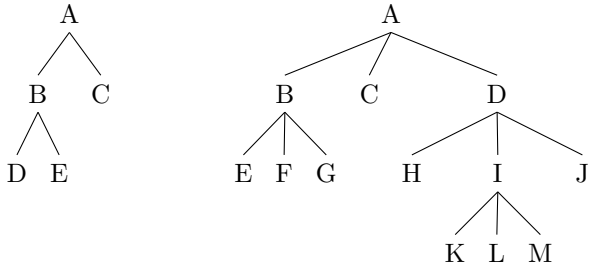
Definitions. As explained in class, a full binary tree is defined as follows:

```
abstract class Tree<E>

class Empty<E> extends Tree<E>

class Node<E> extends Tree<E>
  E data;
  Tree<E> left, right;
```

Empty trees will not count as nodes. A *leaf* is a node with empty trees as children. A node that is not a leaf is called an *internal node.* A tree is called an m-ary tree if every internal node has no more than m children. The tree is called a *full* m-ary tree if every internal node has exactly m children. For example, consider the following trees:



The tree on the left is a full 2-ary tree: it has 5 nodes, 3 of them D, E, and C are leaves and two of them A and B are internal nodes. In other words, we have $m = 2$, $l = 3$, $i = 2$, and $n = 5$. The tree on the right is a full 3-ary tree: it has 13 nodes, 9 leaves, and 4 internal nodes.

Proofs. To prove a property P of binary trees, the general recipe is:

- Prove $P(\mathit{Empty})$
- Assume $P(t_1)$ and $P(t_2)$ hold for an arbitrary trees t_1 and t_2 and prove $P(\mathit{Node}\ d\ t_1\ t_2)$ for the larger tree produced by attaching the subtrees t_1 and t_2 to a node with data d.

This establishes that for every tree t, we have $P(t)$. Sometimes the statement to prove is only valid for trees that have at least one node. In that case, the first case above is to prove $P(\mathit{Node}\ d\ \mathit{Empty}\ \mathit{Empty})$ for a leaf containing data d.

Example Let's prove that a full binary with n nodes has $\frac{n+1}{2}$ leaves. The statement is not true for an empty tree, so we will try to prove it for $n \geq 1$. Let $N(t)$ be the number of nodes for a tree t and $L(t)$ be the number of leaves for a tree t. Formally we want to prove that for every non-empty tree $L(t) = \frac{N(t)+1}{2}$.

- Base case: t has one node, i.e., $N(t) = 1$. We want to prove that $L(t) = \frac{N(t)+1}{2} = 1$. There are two constructors for trees. Since the tree is not empty, it must have been constructed using the constructor for nodes. Since we only have one node, it must be a leaf.
- Inductive case: the tree t is constructed using $Node(d, t_1, t_2)$ where t_1 and t_2 are two smaller trees with $N(t_1) = n_1$ and $N(t_2) = n_2$. We have $N(t) = n_1 + n_2 + 1$. By induction $L(t_1) = \frac{n_1+1}{2}$ and $L(t_2) = \frac{n_2+1}{2}$. The leaves of t are the leaves of both t_1 and t_2, i.e., $L(t) = \frac{n_1+1}{2} + \frac{n_2+1}{2}$. We need to prove:

  $$L(t) = \frac{N(t)+1}{2}$$

  The right hand side $\frac{N(t)+1}{2} = \frac{n_1+n_2+2}{2} = \frac{n_1+1}{2} + \frac{n_2+1}{2}$ which is equal to the left hand side.

Example. Now that we have proved that for non-empty full binary trees $L(t) = \frac{N(t)+1}{2}$, we can easily write a formula for the number of internal node $I(t)$. Indeed by definition $N(t) = I(t) + L(t)$. So:

$$I(t) = N(t) - L(t) = N(t) - \frac{N(t)+1}{2} = \frac{2N(t) - N(t) - 1}{2} = \frac{N(t)-1}{2}.$$

2 Exercises

- Prove that a full m-ary with n nodes has $(n-1)/m$ internal nodes and $((m-1)n+1)/m$ leaves.
- Prove that a full m-ary with i internal nodes has $mi+1$ nodes and $(m-1)i+1$ leaves.
- Prove that a full m-ary with l leaves has $(ml-1)/(m-1)$ nodes and $(l-1)/(m-1)$ internal nodes.
- Now suppose that someone starts a chain letter. Each person who receives the letter is asked to send it on to four other people. Some people do this, but others do not send any letters. Assume that everyone who forwarded the letter forwarded it to exactly four people and that all the people are distinct. Furthermore, assume that the chain letter ended after 100 people received it but did not send it out.
  - How many people have seen the letter, including the first person?
  - How many people sent out the letter?
- Now move on to the starter files and complete the implementation of the `Tree` class hierarchy.
- When you are done submit the solution to the exercises (as a PDF) along with your code to the autograder.