Chapter 7 outline:

- Recursively-defined sets (Monday)
- Structural induction (**Today**)
- Mathematical induction (next week Monday)
- Non-recursive programs—loops (next week Wednesday)
- Loop invariant proofs (next week Friday)
- Recursively-defined sets application: The Huffman encoding (week-after Monday)
- Leftover topic: Arrays, vectors, and intervals (week-after Wednesday)

Last time we saw

- A recursive definition of whole numbers
- A recursive definition of trees, particularly *full binary trees*; a full binary tree is either
  - a leaf, or
  - an internal node together with two children which are full binary trees.

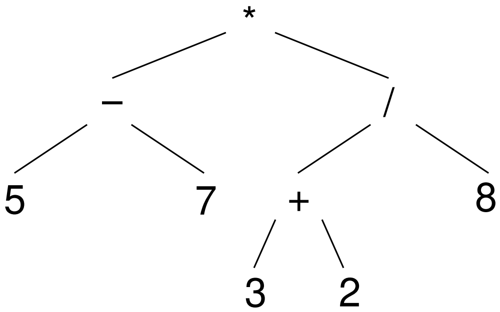
Today we see

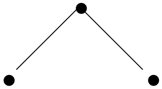
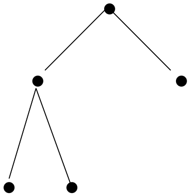
- Self-referential proofs

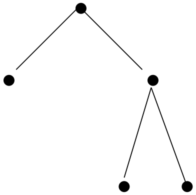
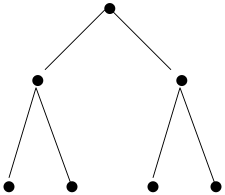
Expression trees:

$$\begin{aligned} \mathit{Expression} &\rightarrow \mathit{Variable} \mid \mathit{Constant} \\ &\quad \mid \mathit{Expression}\,\mathit{Operator}\,\mathit{Expression} \end{aligned}$$

$$\mathit{Operator} \rightarrow + \mid - \mid * \mid /$$



Tree	Nodes	Links
	1	0
	3	2
	5	4

Tree	Nodes	Links
	5	4
	7	6

While building bigger trees from smaller trees, *the number of nodes is (and remains) one more than the number of links*. (Invariant)

Theorem 7.1 *For any full binary tree T,* $\texttt{nodes}(T) = \texttt{links}(T) + 1$*.*

Let $\mathcal{T}$ be the set of full binary trees. Then, we're saying

$$\forall\ T \in \mathcal{T}, \texttt{nodes}(T) = \texttt{links}(T) + 1$$

Theorem 7.1 *For any full binary tree T, $\texttt{nodes}(T) = \texttt{links}(T) + 1$.*

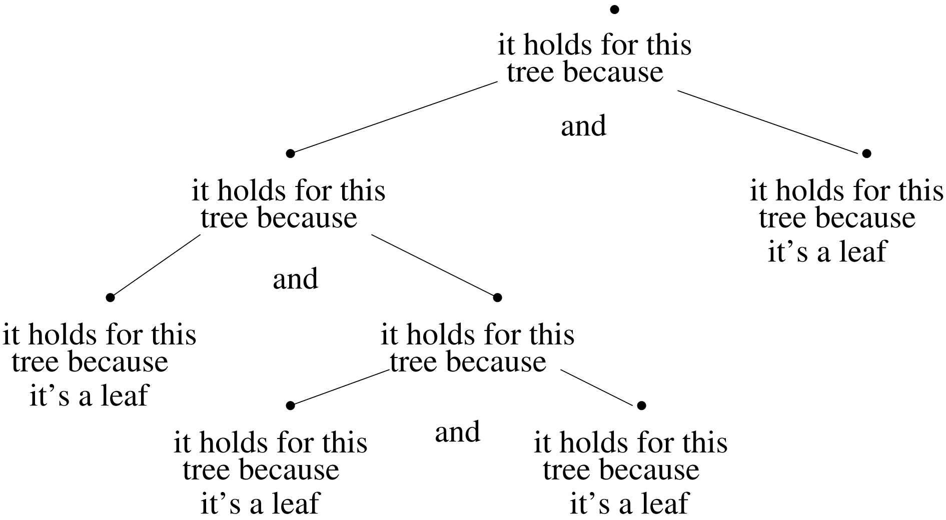
Proof. Suppose $T \in \mathcal{T}$. *[What is a tree? the definition says it's either a leaf or an internal with two subtrees. We can use division into cases.]*

Case 1. Suppose T is a leaf. Then, by how `nodes` and `links` are defined, $\texttt{nodes}(T) = 1$ and $\texttt{links}(T) = 0$. Hence $\texttt{nodes}(T) = \texttt{links}(T) + 1$.

Case 2. Suppose T is an internal node with links to subtrees T_1 and T_2. Moreover, by how `nodes` and `links` are defined, $\texttt{links}(T) = \texttt{links}(T_1) + \texttt{links}(T_2) + 2$. Then,

$\texttt{nodes}(T)$	$=$	$1 + \texttt{nodes}(T_1) + \texttt{nodes}(T_2)$	by the definition of `nodes`
	$=$	$1 + \texttt{links}(T_1) + 1 + \texttt{links}(T_2) + 1$	by Theorem 7.1
	$=$	$\texttt{links}(T_1) + \texttt{links}(T_2) + 2 + 1$	by algebra
	$=$	$\texttt{links}(T) + 1$	by substitution

Either way, $\texttt{nodes}(T) = \texttt{links}(T) + 1$. $\square$


it holds for this tree because
and
it holds for this tree because
and
it holds for this tree because it's a leaf
it holds for this tree because
and
it holds for this tree because it's a leaf
it holds for this tree because it's a leaf
it holds for this tree because it's a leaf

Theorem 7.1 *For any full binary tree* T, $\texttt{nodes}(T) = \texttt{links}(T) + 1$.

Proof. Suppose $T \in \mathcal{T}$.

Base case. Suppose T is a leaf. Then, by how nodes and links are defined, $\texttt{nodes}(T) = 1$ and $\texttt{links}(T) = 0$. Hence $\texttt{nodes}(T) = \texttt{links}(T) + 1$.

Inductive case Suppose T is an internal node with links to subtrees T_1 and T_2 such that $\texttt{nodes}(T_1) = \texttt{links}(T_1) + 1$ and $\texttt{nodes}(T_2) = \texttt{links}(T_2) + 1$. Moreover, by how nodes and links are defined, $\texttt{links}(T) = \texttt{links}(T_1) + \texttt{links}(T_2) + 2$. Then,

$$
\begin{aligned}
\texttt{nodes}(T) &= 1 + \texttt{nodes}(T_1) + \texttt{nodes}(T_2) && \text{by the definition of nodes} \\
&= 1 + \texttt{links}(T_1) + 1 + \texttt{links}(T_2) + 1 && \text{by the inductive hypothesis} \\
&= \texttt{links}(T_1) + \texttt{links}(T_2) + 2 + 1 && \text{by algebra} \\
&= \texttt{links}(T) + 1 && \text{by substitution}
\end{aligned}
$$

Either way, $\texttt{nodes}(T) = \texttt{links}(T) + 1$. $\square$

Let X be a a recursively defined set, and let $\{Y, Z\}$ be a partition of X, where Y is defined by a simple set of elements $Y = \{y_1, y_2, \ldots\}$ and Z is defined by a recursive rule.

Examples:

- X is the set of pizzas, $Y = Crusts$, and $Z = \{(top, bot) \mid top \in Toppings \ \wedge \ bot \in X\}$
- $X = \mathbb{W}$, $Y = \{0\}$, and $Z = \{\mathtt{succ}(n) \mid n \in \mathbb{W}\}$
- $X = \mathcal{T}$, Y is the set of leaves, and Z is the set of internals with children $T_1, T_2 \in \mathcal{T}$.

Let X be a a recursively defined set, and let $\{Y, Z\}$ be a partition of X, where Y is defined by a simple set of elements $Y = \{y_1, y_2, \ldots\}$ and Z is defined by a recursive rule.

To prove something in the form of $\forall\ x \in X, I(x)$, do this:

Base case: Suppose $x \in Y$.

$\vdots$

$I(x)$

Inductive case: Suppose $x \in Z$. *[Using x and the definition of Z, find components $a, b, \ldots \in X$.]*

Suppose $I(a), I(b), \ldots$. *[The* **inductive hypothesis***]*

$\vdots$

Use the inductive hypothesis

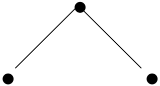
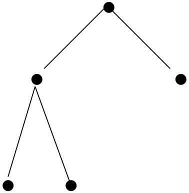
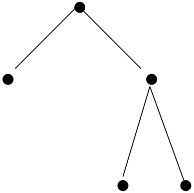
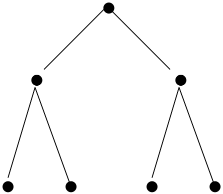
$\vdots$

$I(x)$ $\square$

7.2.1 For any full binary tree T, $\texttt{leaves}(T) = \texttt{internals}(T) + 1$.

Let the *height* of a full binary tree be 1 if the tree is a node by itself (leaf), or 1 more than the maximum height of its two children, if it is an internal node.

7.2.5 For any full binary tree T, $\texttt{nodes}(T) \leq 2^{\texttt{height}(T)} - 1$.

Tree	Nodes	Height
	1	1
	3	2
	5	3
	5	3
	7	3

For next time:

Do Exercises 7.2.(2,3,5).

Read Section 7.3

Take quiz