Chapter 6 roadmap:
- Recursive definitions and types (Monday)
- Structural induction (Today)
- Mathematical induction (Friday)
- Loop invariant proofs (next week Monday and 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
While building bigger trees from smaller trees, the number of nodes is (and remains) one more than the number of links. (Invariant)

**Theorem 6.1** For any full binary tree $T$, $\text{nodes}(T) = \text{links}(T) + 1$.

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

$$\forall T \in \mathcal{T}, \text{nodes}(T) = \text{links}(T) + 1$$
Theorem 6.1 For any full binary tree $T$, $\text{nodes}(T) = \text{links}(T) + 1$.

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, $\text{nodes}(T) = 1$ and $\text{links}(T) = 0$. Hence $\text{nodes}(T) = \text{links}(T) + 1$.

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, $\text{links}(T) = \text{links}(T_1) + \text{links}(T_2) + 2$. Then,

$$
\text{nodes}(T) = 1 + \text{nodes}(T_1) + \text{nodes}(T_2) \quad \text{by the definition of nodes}
$$
$$
= 1 + \text{links}(T_1) + 1 + \text{links}(T_2) + 1 \quad \text{by Theorem 6.1}
$$
$$
= \text{links}(T_1) + \text{links}(T_2) + 2 + 1 \quad \text{by algebra}
$$
$$
= \text{links}(T) + 1 \quad \text{by substitution}
$$

Either way, $\text{nodes}(T) = \text{links}(T) + 1$. □
Theorem 6.1 For any full binary tree $T$, $\text{nodes}(T) = \text{links}(T) + 1$.

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

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

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

$$
\text{nodes}(T) = 1 + \text{nodes}(T_1) + \text{nodes}(T_2) \quad \text{by the definition of nodes}
$$

$$
= 1 + \text{links}(T_1) + 1 + \text{links}(T_2) + 1 \quad \text{by the inductive hypothesis}
$$

$$
= \text{links}(T_1) + \text{links}(T_2) + 2 + 1 \quad \text{by algebra}
$$

$$
= \text{links}(T) + 1 \quad \text{by substitution}
$$

Either way, $\text{nodes}(T) = \text{links}(T) + 1$. $\square$
Let $X$ be 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 lists, $Y = \{[]\}$, and $Z = \{a :: \text{rest} \mid \text{rest} \in X\}$
- $X = \mathbb{W}$, $Y = \{0\}$, and $Z = \{\text{succ}(n) \mid n \in \mathbb{W}\}$
- $X = \mathcal{T}$, $Y$ 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 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 \( x \in Y \).

\[
I(x)
\]

**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]

Use the inductive hypothesis

\[
I(x) \square
\]
For next time:

*See Schoology for homework problems, based on problems from Section 6.4.*

*Skim 6.(5 & 6)*

*Take quiz*