4. Structural Induction

There are some interesting ways to extend Induction techniques to sets that are not as “simple” or “well-behaved” as our lovely naturals, $\mathbb{N}$. The two ingredients our examples and exercises have so far used are (1) the fact that $\mathbb{N}$ is well-ordered and (2) the presence of recursive structure. It turns out that the first property can be weakened, but as long as recursive structure is present a successful “induction-like” proof can be made.

A proof by **Structural Induction** is the name for this method, and here are the official “ingredients” needed to pull it off:

- A proposition about objects that feature recursive structure
- A way of talking about “sub-structures” (i.e. a **partial ordering**)
- Base Case(s)

Here is an outline:

Structural Induction Outline

Proposition. $\forall x \in S$, $P(x)$

Proof by Structrual Induction

Base Case

Show that $P(s)$ is true for the “smallest” elements $s \in S$

(These are sometimes called “base elements.”)

Recursive Step

Show being true at “smaller elements” implies being true at “larger ones”

Show $P(x) \Rightarrow P(\text{larger elements built from } x)$

Assume $P(x)$ (this is called the **inductive hypothesis**.)

Use S’s recursive structure to relate x to “larger elements built from x”

Use recursive structure to conclude $P(\text{larger elements built from } x)$ $\square$

Because the set S can be quite general, it is a bit difficult to talk about its recursive structure, and that’s why phrases like “larger elements” and “elements built from” are used above. As usual, a few examples will help.

Example 4.1. *Consider the set $S \subseteq \mathbb{Z} \times \mathbb{Z}$ defined by the following rules:*

(1) $(0,0) \in S$

(2) $(a,b) \in S \Rightarrow (a,b+1) \in S$

(3) $(a,b) \in S \Rightarrow (a+1,b+1) \in S$

(4) $(a,b) \in S \Rightarrow (a+2,b+1) \in S$

Rule (1) tells us a base element of S, while rules (2)-(4) define S recursively (just like recursively-defined sequences have rules that generate "new" elements from "old" ones).

Pause here to explore the set S. For instance, make certain you can determine whether or not the following ordered pairs are or are not members of S: $(0,1),(1,1),(2,1),(3,2),(2,3),(0,2)$. (Spoiler alert: all of these are elements of S.)

Example 4.2. *Let S be the recursively-defined set from the previous example.*

Proposition. $\forall\,(a,b) \in S,\ a \leq 2b$

Proof by Structural Induction

Base Case

We need only confirm the proposition is true for the element $(0,0)$, i.e. $a=0$ and $b=0$. It follows that $0 \leq 2 \cdot 0 = 0$ is true.

Recursive Step

Suppose the proposition is true for an arbitrary element (a,b). We need only confirm that the proposition is true for the "larger" elements $(a,b+1),(a+1,b+1)$ and $(a+2,b+1)$.

Our hypothesis means $a \leq 2b$, and this helps us show the proposition is true for all elements "built" from (a,b).

$$\text{(true for } (a,b+1)\text{)} : a \leq 2b < 2b+2 = 2(b+1).$$

$$\text{(true for } (a+1,b+1)\text{)} : a+1 \leq 2b+1 < 2b+2 = 2(b+1).$$

$$\text{(true for } (a+2,b+1)\text{)} : a+2 \leq 2b+2 = 2(b+1).$$

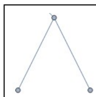
This completes our proof. □

Note that the recursive definition of S allows us to talk about "larger" or "more complicated" elements that are "built from" other ones.

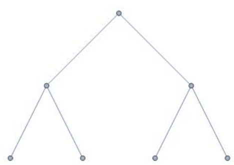
Our next example is a popular one that is discussed in many Computer Science courses, and it is about **full binary trees**. Recall that these trees contain a special node called **the root**, and that every other node is joined to either 0 or 2 "lower" nodes by an edge. Before exploring the next example, we need one more definition, and it is defined recursively! The **height of a full binary tree** T, denoted $h(T)$, is

$$h(t) = \begin{cases} 0 & \text{if } T \text{ consists of one node} \\ \max\{h(T_1),\, h(T_2)\} + 1 & \text{where } T \text{ is built from two} \\ & \text{full binary sub-trees } T_1 \text{ and } T_2 \end{cases}$$

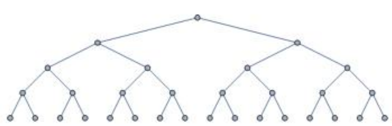
Example 4.3. *Check that you understand how to compute the heights of the following full binary trees.*



has height $h = 1$



has height $h = 2$



has height $h = 4$

Example 4.4.

Proposition. *If T is a full binary tree with n vertices then $n \leq 2^{h(T)+1} - 1$*

Proof (By Structural Induction)

Base Case
The full binary tree that consists of one node (i.e. $n = 1$) has height $h = 0$, and so it follows that $n = 1 = 2 - 1 = 2^{0+1} - 1 = 2^{h(T)+1} - 1$.

Recursive Step Suppose T has n vertices and that it can be decomposed into two full binary trees T_1 and T_2, each with n_1 and n_2 vertices, and that the proposition is true for both T_1 and T_2. (We will show that the proposition is true for the full binary tree T.) It follows that

$$n = 1 + n_1 + n_2 \tag{30}$$

$$\leq 1 + 2^{h(T_1)+1} - 1 + 2^{h(T_2)+1} - 1 \tag{31}$$

$$= 2^{h(T_1)+1} + 2^{h(T_2)+1} - 1 \tag{32}$$

$$\leq 2 \cdot \max\left\{2^{h(T_1)+1},\, 2^{h(T_2)+1}\right\} - 1 \tag{33}$$

$$= 2 \cdot 2^{\max\{h(T_1),\, h(T_2)\}+1} - 1 \tag{34}$$

$$= 2 \cdot 2^{h(T)} - 1 = 2^{h(T)+1} - 1. \; \square \tag{35}$$

In the above example Line (31) follows from the inductive hypothesis and Line (34) follows from the fact that the maximum of $2^a, 2^b$ equals $2^{\max\{a,b\}}$.