1 Weak Induction Introduction

Here are two hypothetical situations that can help communicate the idea of induction.

1.1 A Domino Argument

Suppose there are infinitely many dominoes labeled 1, 2, 3, ... standing up in such a way that when you push over domino $k$ it then pushes over domino $k + 1$. Of course nothing is happening yet. Suppose then you push over domino 1. What will happen? Well, domino 1 falls, causing domino 2 to fall, causing domino 3 to fall, and so on. For every $n$, domino $n$ will fall.

In this hypothetical there are two things that are true. Let’s say $F(n)$ is true iff domino $n$ falls.

(a) The first domino falls. That is, $F(n)$ is true.

(b) For every domino $k$, if domino $k$ falls then domino $k + 1$ falls. That is, $\forall k \geq 1, F(k) \rightarrow f(k + 1)$.

You can conclude that $\forall n \geq 1, F(n)$.

1.2 A Weather Argument

Suppose I told you two things:

(a) It will be sunny on January 1 2022.

(b) For any day from January 1 2022 onwards, if it is sunny on that day, it will be sunny on the next day.

What could you conclude?

You could conclude that it will be sunny every day from January 1 2022 onwards! The reason for this is that:

• First (a) tells you it will be sunny on January 1 2022.
• Then (b) tells you that because it is sunny on January 1 2022, it will also be sunny on January 2 2022.
• Then (b) tells you that because it is sunny on January 2 2022, it will also be sunny on January 3 2022.
• Then (b) tells you that because it is sunny on January 3 2022, it will also be sunny on January 4 2022.
• And so on.

The idea behind weak mathematical induction is just this. Strong mathematical induction is only slightly different.
2 Weak Mathematical Induction

2.1 Introduction

Weak mathematical induction is also known as the First Principle of Mathematical Induction and works as follows:

2.2 How it Works

Suppose some statement $P(n)$ is defined for all $n \geq n_0$ where $n_0$ is a nonnegative integer. Suppose that we want to prove that $P(n)$ is actually true for all $n \geq n_0$. We do this by proving two things:

(a) The Base Case: We prove that $P(n_0)$ is true.

(b) The Inductive Step: We prove that for any $k \geq n_0$, if $P(k)$ is true (this is called the inductive hypothesis) then $P(k+1)$ is also true.

More formally we have:

**Theorem 2.2.1.** Suppose $P(n)$ is a statement for all $n \geq n_0$ and suppose that:

(a) $P(n_0)$

(b) $\forall k \geq n_0, P(k) \rightarrow P(k+1)$

Then $\forall n \geq n_0, P(n)$.

2.3 Examples

**Example 2.1.** Let’s prove that $2^n - n^2 \geq 0$ for all $n \geq 4$.

Here $P(n)$ is the statement $2^n - n^2 \geq 0$ and the starting value is $n_0 = 4$.

(a) The Base Case:

We claim that $2^4 - 4^2 \geq 0$. Well $2^4 - 4^2 = 16 - 16 = 0 \geq 0$ so it is true.

(b) The Inductive Step:

We will prove that:

$$\forall k \geq 4, \text{ if } 2^k - k^2 \geq 0 \text{ then } 2^{k-1} - (k-1)^2 \geq 0$$

Suppose that $k \geq 4$ and $2^k - k^2 \geq 0$ (the inductive hypothesis).

Observe that:
\[2^{k+1} - (k + 1)^2 = 2 \cdot 2^k - (k + 1)^2 \]
\[\geq 2k^2 - (k + 1)^2 \]
\[= 2k^2 - k^2 - 2k - 1 \]
\[= k^2 - 2k - 1 \]
\[= k(k - 2) - 1 \geq 4(4 - 2) - 1 = 7 \geq 0 \]

Then we are done. \[\square\]
Example 2.2. Let’s prove that:

\[
\sum_{i=1}^{n} i = \frac{n(n + 1)}{2}
\]

for all \(n \geq 0\).

(a) The Base Case:

We claim that:

\[
\sum_{i=1}^{1} i = \frac{1(1 + 1)}{2}
\]

Since the left hand side is 1 and the right hand side is 1 this is true.

(b) The Inductive Step:

We will prove that:

\[
\forall k \geq 1, \text{ if } \sum_{i=1}^{k} i = \frac{k(k + 1)}{2} \text{ then } \sum_{i=1}^{k+1} i = \frac{(k + 1)((k + 1) + 1)}{2}
\]

Suppose that \(k \geq 1\) and \(\sum_{i=1}^{k} i = \frac{k(k+1)}{2}\) (the inductive hypothesis).

Observe that:

\[
\sum_{i=1}^{k+1} i = \frac{(k + 1)(k + 1 + 1)}{2} = \left[ \sum_{i=1}^{k} i \right] + (k + 1)
\]

\[
= \frac{k(k + 1)}{2} + k + 1
\]

\[
= \frac{k(k + 1) + 2(k + 1)}{2}
\]

\[
= \frac{(k + 2)(k + 1)}{2}
\]

Then we are done. \(\square\)
Example 2.3. Suppose that \( n \geq 1 \). Take a \( 2^n \times 2^n \) chessboard with a corner removed. Let’s prove the board can be covered with three-square L-shaped pieces.

(a) The Base Case:
This obviously true for a \( 2^1 \times 2^1 \) chessboard with a corner removed because it takes exactly one L-shaped piece to cover it.

(b) The Inductive Step:
We will prove that:
For all \( k \geq 1 \), if we can cover a \( 2^k \times 2^k \) chessboard with a corner removed then we can cover a \( 2^{k+1} \times 2^{k+1} \) chessboard with a corner removed.

Suppose we have (somehow) managed to cover a \( 2^k \times 2^k \) chessboard with a corner removed (the inductive hypothesis):

\[
\begin{array}{c}
\text{Done!} \\
\end{array}
\]

If we’re now given a \( 2^{k+1} \times 2^{k+1} \) chessboard with a corner removed we can cover it using three of the above arrangements (whatever they are) and one extra L-shaped piece. Notice that this new chessboard is twice as long and twice as wide as the previous one.

\[
\begin{array}{c c c c c}
\text{Done!} & \text{Done!} \\
\text{Done!} & \text{Done!} \\
\end{array}
\]

Then we are done.
Example 2.4. Let’s prove that:

\[
\sum_{j=1}^{n} j(j + 1) = \frac{n(n + 1)(n + 2)}{3}
\]

for all \( n \geq 1 \).

(a) The Base Case:

We claim that:

\[
\sum_{j=1}^{1} j(j + 1) = \frac{1(1 + 1)(1 + 2)}{3}
\]

Since the left hand side is 2 and the right hand side is 2 it is true.

(b) The Inductive Step:

We will prove that:

\[
\forall k \geq 1, \text{ if } \sum_{j=1}^{k} j(j + 1) = \frac{k(k + 1)(k + 2)}{3}
\]

then \( \sum_{j=1}^{k+1} j(j + 1) = \frac{(k + 1)((k + 1) + 1)((k + 1) + 2)}{3} \)

Suppose that \( k \geq 1 \) and \( \sum_{j=1}^{k} j(j + 1) = \frac{k(k+1)(k+2)}{3} \) (the inductive hypothesis).

Observe that:

\[
\sum_{j=1}^{k+1} j(j + 1) = \sum_{j=1}^{k} j(j + 1) + (k + 1)((k + 1) + 1)
\]

\[
= \frac{k(k + 1)(k + 2)}{3} + (k + 1)(k + 2)
\]

\[
= k(k + 1)(k + 2) + 3(k + 1)(k + 2)
\]

\[
= \frac{3}{3}
\]

Then we are done.
Example 2.5. Define a sequence recursively by:

\[
    a_n = \begin{cases} 
        a_0 = 3 \\
        a_n = 5a_{n-1} + 8 & \text{for } n \geq 1 
    \end{cases}
\]

Let’s prove that \( a_n \equiv 3 \mod 4 \) for all \( n \geq 0 \).

(a) The Base Case:
We claim that \( a_0 \equiv 3 \mod 4 \) but since \( a_0 = 3 \) this is clearly true.

(b) The Inductive Step:
We will prove that:

\[
    \forall k \geq 0, \text{ if } a_k \equiv 3 \mod 4 \text{ then } a_{k+1} \equiv 3 \mod 4
\]

Suppose that \( k \geq 0 \) and \( a_k \equiv 3 \mod 4 \) (the inductive hypothesis).
Observe that:

\[
    a_{k+1} = 5a_k + 8 \\
    \equiv 5(3) + 8 \mod 4 \\
    \equiv 23 \mod 4 \\
    \equiv 3 \mod 4
\]

Then we are done.
Example 2.6. Let’s prove that $3 | n^3 - n$ for all $n \geq 1$.

(a) The Base Case:

We claim that $3 | 1^3 - 1$ but since $1^3 - 1 = 0$ and $3 | 0$ this is clearly true.

(b) The Inductive Step:

We will prove that:

$$\forall k \geq 0, \text{ if } 3 | k^3 - k \text{ then } 3 | (k + 1)^3 - (k + 1)$$

Suppose that $3 | k^3 - k$ (the inductive hypothesis). This means that $k^3 - k = 3\alpha$ for some $\alpha \in \mathbb{Z}$.

Observe that:

$$(k + 1)^3 - (k + 1) = k^3 + 3k^2 + 3k + 1 - k - 1$$
$$= k^3 - k + 3k^2 + 3$$
$$= 3\alpha + 3(k^2 + k)$$
$$= 3(\alpha + k^2 + k)$$

And so $3 | (k + 1)^3 - (k + 1)$.

Then we are done. \qed
Example 2.7. Let’s prove that:

\[ 1(1!) + 2(2!) + \ldots + n(n!) = (n + 1)! - 1 \]

For all \( n \geq 0 \).

(a) The Base Case:
We claim that \( 1(1!) = (1 + 1)! - 1 \). Since the left hand side is 1 and the right hand side is 1 this is true.

(b) The Inductive Step:
We will prove that:

\[ \forall k \geq 0, \text{ if } 1(1!) + 2(2!) + \ldots + k(k!) = (k + 1)! - 1 \]

then \( 1(1!) + 2(2!) + \ldots + (k + 1)(k + 1)! = ((k + 1) + 1)! - 1 \)

Suppose that \( 1(1!) + 2(2!) + \ldots + k(k!) = (k + 1)! - 1 \) (the inductive hypothesis).
Then observe that:

\[
1(1!) + 2(2!) + \ldots + (k + 1)(k + 1)! = 1(1!) + 2(2!) + \ldots + k(k!) + (k + 1)(k + 1)!
\]

\[
= (k + 1)! - 1 + (k + 1)((k + 1)!) \\
= 1(k + 1)! + (k + 1)(k + 1)! - 1 \\
= (1 + k + 1)(k + 1)! - 1 \\
= ((k + 2)(k + 1)! - 1 \\
= (k + 2)! - 1
\]

Then we are done. □
3 Strong Mathematical Induction

3.1 Introduction

Let’s begin with an intuitive example. This is not a formal proof by strong induction (we haven’t even talked about what strong induction is!) but it hits some of the major ideas intuitively.

Example 3.1. Suppose that all we have are 3¢ and 10¢ stamps. Prove that we can make any postage of 18¢ or more.

The first thing to note is that if we tried to use weak induction the inductive step won’t help. This is because knowing that we can make exactly $k$¢ doesn’t give us any information about how to make $(k + 1)$¢.

Suppose we started doing it one by one:

\[
\begin{align*}
18 &= 3 + 3 + 3 + 3 + 3 + 3 \\
19 &= 10 + 3 + 3 + 3 \\
20 &= 10 + 10
\end{align*}
\]

At this point we might notice something. We might notice that to do 21¢ we can simply do 18¢ and add a 3¢ stamp. Then to do 22¢ we simply do 19¢ and add a 3¢ stamp. Since this idea continues see that we can in fact do any postage 18¢ or more.

Before discussing strong mathematical induction formally we will state that the three cases we did first are the three base cases and that the thing we notice is the inductive step.

Observe that all three base cases were necessary because we can’t try to do 20¢ by doing 17¢ and adding a 3¢ stamp because we haven’t done 17¢, and in fact 17¢ can’t be done!

\[\square\]

3.2 How it Works

The general idea behind strong mathematical induction is this.

(a) We prove some number of base cases $n_0, \ldots, n = ???$.

(b) We assume that the statement is true for $i = n_0, n_0+1, \ldots, k$ and we prove that the statement is true for $k + 1$.

The major point of confusion arises because it may not be clear how many base cases we need to prove. In light of this it is usually easier to prove the inductive step first and then check the inductive step to see how far back it references. That reference must be greater than or equal to $n_0$ and that will tell us how far our base cases need to go.

11
Confused? Let’s return to the stamp problem again and approach it from this angle.
3.3 Examples

Example 3.2. Suppose that all we have are 3¢ and 10¢ stamps. Prove that we can make any postage of 18¢ or more.

(a) The Inductive Step:

Assume we can make all amounts 18, 19, 20, ..., $k$ (the induction hypothesis). How can we make $k + 1$? Well, easy, we simply make $(k - 2)¢$ and add in another 3¢ stamp.

(b) The Base Case(s):

In order to do what we did, we must have $k - 2 \geq 18$ which means $k \geq 20$.

This means that the cases 18, 19, 20 must be done separately and induction gets us from 20 to 21, from 21 to 22, and so on.

So we have three base cases:

\[
\begin{align*}
18 &= 3 + 3 + 3 + 3 + 3 + 3 \\
19 &= 10 + 3 + 3 + 3 \\
20 &= 10 + 10
\end{align*}
\]

Then we are done. \qed

Note 3.3.1. I have seen very few resources which suggest doing the inductive step first and using it to analyze how many base cases are needed. Most resources just, somehow, produce the number of base cases out of thin air, which is confusing. \qed
Example 3.3. Define a sequence recursively by:

\[
\begin{align*}
    a_n &= \begin{cases} 
    a_1 = 1 \\
    a_2 = 4 \\
    a_n = 2a_{n-1} - a_{n-2} & \text{For } n \geq 3
    \end{cases}
\end{align*}
\]

Let’s prove that \(a_n = 3n - 2\) for all \(n \geq 1\).

(a) The Inductive Step:

We assume that \(a_i = 3i - 2\) for \(i = 1, 2, \ldots, k\) and we prove that \(a_{k+1} = 3(k+1) - 2\) (the induction hypothesis).

Observe that:

\[
    a_{k+1} = 2a_k - a_{k-1}
    \]

\[
    = 2(3k - 2) - (3(k - 1) - 2)
    \]

\[
    = 6k - 4 - 3k + 3 + 2
    \]

\[
    = 3k + 1
    \]

\[
    = 3(k + 1) - 2
    \]

(b) The Base Case(s):

When proving the statement for \(k+1\) we had to use the statement for \(k-1\). Because our claim starts with \(n = 1\) we have to ensure that \(k - 1 \geq 1\), so \(k \geq 2\).

This means that there are two base cases, \(a_1\) and \(a_2\). We check those:

\(a_1 = 1 = 3(1) - 2\) is true and \(a_2 = 4 = 3(2) - 2\) is true.

Then we are done. \(\square\)
**Example 3.4.** Let’s prove that every integer greater than 2 can be written as a product of (perhaps just one) primes.

(a) The Inductive Step:

Suppose every integer 2, 3, ..., \( k \) can be written as the product of primes (the induction hypothesis). We claim that \( k + 1 \) can be, also.

Either \( k + 1 \) is prime or it isn’t. If it’s prime, then it can be written as a product of itself.

If it isn’t prime then \( k + 1 = ab \) with \( 2 \leq a \leq k \) and \( 2 \leq b \leq k \). By the induction hypothesis we know \( a \) and \( b \) are each products of primes, and therefore \( k + 1 \) is, too.

(b) The Base Case(s):

The base cases are a bit sneaky here. In order to write \( k + 1 = ab \) with \( 2 \leq a \leq k \) and \( 2 \leq b \leq k \) we must have \( k + 1 \geq (2)(2) = 4 \). Thus we must have \( k \geq 3 \) and so our base cases are 2 and 3.

Each of these can certainly be written as the product of one prime.

Then we are done.
4 Structural Induction

4.1 Introduction

Weak and strong mathematical induction are both predicated on the fact that we are proving something for all \( n \geq n_0 \) for some \( n_0 \). This means that there is some organization of the items for \( n = n_0, n = n_0 + 1, n = n_0 + 2 \) and so on. However not all collections of objects are organized like this. Some examples:

- The set of binary trees cannot necessarily be organized this way.
- A recursively defined set cannot necessarily be organized this way.

However both of these things are defined by giving some elements in the set (sort of like base cases) and then some rule(s) for adding new elements to the sets. Structural induction is essentially a way of doing induction on these recursively defined sets.

4.2 How it Works

Suppose we want to prove some property is true for all items in a recursively defined set. We proceed as follows:

(a) Base Cases(s): We prove that the property is true for the original items in the set.

(b) Inductive Step: We prove that when the rules add new things to the set that the property is preserved.

4.3 Examples

Here are a bunch of examples.

Example 4.1. Suppose a set \( S \) is defined recursively as follows:

(a) \( 1 \in S \)

(b) If \( x \in S \) then \( 2x \in S \).

Let’s prove that all elements of \( S \) are integer powers of 2.

(a) Base Case: Observe that \( 1 = 2^0 \) so 1 is an integer power of 2.

(b) Inductive Step: Suppose \( x \in S \) and \( x \) is an integer power of 2. We claim that \( 2x \) is an integer power of 2 as well. Since \( x \) is a power of 2 we know \( x = 2^k \) for some \( k \in \mathbb{Z} \). Then \( 2x = 2(2^k) = 2^{k+1} \) and since \( k + 1 \in \mathbb{Z} \) we know that \( 2x \) is also a power of 2.

\[ \square \]

Take a few minutes to see that the inductive step is showing that the property is preserved. Whenever the original item in \( S \) has the property and we add a new item to \( S \), that new item will also have the property. It follows that all items have that property.
Example 4.2. Suppose a set $S$ is defined recursively as follows:

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

(b) If $(x, y) \in S$ then $(x, y + 1), (x + 1, y + 1), (x + 2, y + 1) \in S$.

Let’s prove that every element $(x, y) \in S$ has $x \leq 2y$.

(a) Base Case: Observe that $(0, 0)$ certainly satisfies $0 \leq 2(0)$.

(b) Inductive Step: Suppose $(x, y) \in S$ and $x \leq 2y$. We have three things to show since there are three rules for adding new elements:

- We need to prove that $(x, y + 1)$ satisfies $x \leq 2(y + 1)$. However since $x \leq 2y \leq 2y + 2 = 2(y + 1)$ this is true.
- We need to prove that $(x + 1, y + 1)$ satisfies $x + 1 \leq 2(y + 1)$. However since $x \leq 2y$ we have $x + 1 \leq 2y + 1 \leq 2y + 2 = 2(y + 1)$ this is true.
- We need to prove that $(x + 2, y + 1)$ satisfies $x + 2 \leq 2(y + 1)$. However since $x \leq 2y$ we have $x + 2 \leq 2y + 2 = 2(y + 1)$ this is true.

□

Example 4.3. Suppose a set $S$ of strings of $a$ and $b$ is defined recursively as follows:

(a) The empty string is in $S$.

(b) If $x \in S$ then $axb, bxa$ are in $S$ and if $x, y \in S$ then $xy \in S$.

Let’s prove that every string in $S$ has an equal number of $a$ and $b$.

(a) Base Case: Observe that the empty string has 0 of each, and $0 = 0$.

(b) Inductive Step: We have three things to show since there are three rules for adding new elements:

- We need to prove that if $x \in S$ has an equal number of $a$ and $b$ then $axb$ has an equal number of $a$ and $b$. But this is clear since we’re adding one of each.
- We need to prove that if $x \in S$ has an equal number of $a$ and $b$ then $bxa$ has an equal number of $a$ and $b$. But this is clear since we’re adding one of each.
- We need to prove that if $x, y \in S$ each has an equal number of $a$ and $b$ then $xy$ has an equal number of $a$ and $b$. But this is clear since if $x$ has $k$ of each and $y$ has $j$ of each then $xy$ has $j + k$ of each.

□
Example 4.4. Suppose a set $S$ is defined recursively as follows:

(a) $0 \in S$
(b) If $x \in S$ then $2x + 1 \in S$.

Let’s prove that $S = \{2^n - 1 \mid n \in \mathbb{Z}_{\geq 0}\}$.

To prove this we actually need to prove both $\subseteq$ and $\supseteq$. The first of these is by structural induction, the second by weak induction.

Let’s first show that $S \subseteq \{2^n - 1 \mid n \in \mathbb{Z}_{\geq 0}\}$. This means showing that everything in $S$ has the form $2^n - 1$ for some $n \in \mathbb{Z}_{\geq 0}$.

(a) Base Case: Observe that $0 = 2^0 - 1$ so it is true.

(b) Inductive Step: Suppose that $x \in S$ and $x = 2^n - 1$ for some $n \in \mathbb{Z}_{\geq 0}$.

We claim that $2x - 1$ also has this form. However observe that: $2x - 1 = 2(2^n - 1) - 1 = 2^{n+1} - 1$ and since $n + 1 \in \mathbb{Z}_{\geq 0}$ we are done.

Let’s next show that $\{2^n - 1 \mid n \in \mathbb{Z}_{\geq 0}\} \subseteq S$ using weak induction. This means proving that for all $n \geq 0$ we have $2^n - 1 \in S$.

(a) Base Case: We check $n = 0$. Since $2^0 - 1 = 0$ and we know $0 \in S$ was given, the base case is true.

(b) Inductive Step: Suppose $2^k - 1 \in S$ for $k \geq 0$. We claim $2^{k+1} - 1 \in S$.

Well observe that since $2^k - 1 \in S$ we know that $2(2^k - 1) + 1 \in S$ by the construction of $S$. However:

$$2(2^k - 1) + 1 = 2^{k+1} - 2 + 1 = 2^{k+1} - 1$$

and so $2^{k+1} - 1 \in S$ as desired.
Example 4.5. Recall how binary trees are defined recursively. Let’s prove that \( N(T) = E(T) + 1 \) for any binary tree \( T \).

(a) Base Case: If \( T \) is a single node then \( N(T) = 1 \) and \( E(T) = 0 \) and \( 1 = 0 + 1 \) is true.

(b) Inductive Step: There are two ways to create new binary trees during the recursive construction and we must examine both of them.

- Suppose \( T \) is a binary tree for which \( N(T) = E(T) + 1 \) and we create a new binary tree \( T' \) by creating a new root node and attaching \( T \) to it. We claim that \( N(T') = E(T') + 1 \).

  In creating this new binary tree we add one node and one edge and so \( N(T') = N(T) + 1 \) and \( E(T') = E(T) + 1 \) and so then:
  \[
  N(T') = N(T) + 1 = E(T) + 1 + 1 = E'(T) + 1
  \]

- Suppose \( T_1 \) and \( T_2 \) are binary trees for which \( N(T_1) = E(T_1) + 1 \) and \( N(T_2) = E(T_2) + 1 \) and we create a new binary tree \( T' \) by creating a new root node and attaching both \( T_1 \) and \( T_2 \) to it. We claim that \( N(T') = E(T') + 1 \).

  In creating this new binary tree we add one node and two edges and so \( N(T') = N(T_1) + N(T_2) + 1 \) and \( E(T') = E(T_1) + E(T_2) + 2 \) and so then:
  \[
  N(T') = N(T_1) + N(T_2) + 1 = E(T_1) + 1 + E(T_2) + 1 = E(T_1) + E(T_2) + 1 = E(T') + 1
  \]
**Example 4.6.** Let’s prove that \( L(T) \leq 2^{H(T)} \) for any binary tree \( T \).

(a) Base Case: If \( T \) is a single node then \( L(T) = 1 \) and \( H(T) = 0 \) and \( 1 \leq 2^0 \) is true.

(b) Inductive Step: Again there are two things to show:

- Suppose \( T \) is a binary tree for which \( L(T) \leq 2^{H(T)} \) and we create a new binary tree \( T' \) by creating a new root node and attaching \( T \) to it. We claim that \( L(T') \leq 2^{H(T')} \).

  In creating this new binary tree the number of leaves does not change and the height increases by 1, so \( L(T') = L(T) \) and \( H(T') = H(T) + 1 \) and so then:
  \[
  L(T') = L(T) \leq 2^{H(T)} = 2^{H(T)+1} \leq 2^{H(T)}
  \]

- Suppose \( T_1 \) and \( T_2 \) are binary trees for which \( L(T_1) \leq 2^{H(T_1)} \) and \( L(T_2) \leq 2^{H(T_2)} \) and we create a new binary tree \( T' \) by creating a new root node and attaching both \( T_1 \) and \( T_2 \) to it. We claim that \( L(T') \leq 2^{H(T')} \).

  In creating this new binary tree the number of leaves in \( T' \) is the sum of the number of leaves in \( T_1 \) and \( T_2 \), so:
  \[
  L(T') = L(T_1) + L(T_2)
  \]
  The height of \( T' \) will be 1 plus the maximum of the heights of \( T_1 \) and \( T_2 \). However it’s certainly the case that \( H(T') \geq H(T_1) + 1 \) and \( H(T') \geq H(T_2) + 1 \). Then we get:
  \[
  L(T') = L(T_1) + L(T_2) \leq 2^{H(T_1)} + 2^{H(T_2)} \\
  \leq 2^{H(T_1)+1} + 2^{H(T_2)-1} \\
  \leq 2 \cdot 2^{H(T')-1} \\
  \leq 2^{H(T')}
  \]

Note: Another approach to managing the height is to first look at the case where \( H(T_1) \geq H(T_2) \). In that case \( H(T') = H(T_1) + 1 \) and then:

  \[
  L(T') = L(T_1) + L(T_2) \leq 2^{H(T_1)} + 2^{H(T_2)} \\
  \leq 2^{H(T_1)} + 2^{H(T_1)} \\
  = 2^{H(T_1)+1} + 2^{H(T_1)-1} \\
  \leq 2 \cdot 2^{H(T_1)-1} \\
  \leq 2^{H(T')}
  \]

The case where \( H(T_2) \geq H(T_1) \) is exactly the same with the trees exchanged.

\[ \square \]
Example 4.7. Let’s prove that \( N(T) \leq 2^{H(T)} + 1 \) for any binary tree \( T \).

(a) Base Case: If \( T \) is a single node then \( N(T) = 1 \) and \( H(T) = 0 \) and \( 1 \leq 2^{0+1} - 1 \) is true.

(b) Inductive Step: Again there are two things to show:

- Suppose \( T \) is a binary tree for which \( N(T) \leq 2^{H(T)} + 1 \). and we create a new binary tree \( T' \) by creating a new root node and attaching \( T \) to it. We claim that \( N(T') \leq 2^{H(T')} + 1 \). It’s actually easier to show that \( 2^{H(T')} + 1 \geq N(T') \).

  In creating this new binary tree we add one node and the height increases by 1, so \( N(T') = N(T) + 1 \) and \( H(T') = H(T) + 1 \) and so then:

  \[
  2^{H(T')} + 1 = 2^{H(T) + 1} + 1 \\
  = 2 \cdot 2^{H(T) + 1} - 1 \\
  \geq 2(N(T) + 1) - 1 \\
  = 2N(T) + 1 \\
  \geq N(T) + 1 \\
  = N(T')
  \]

- Suppose \( T_1 \) and \( T_2 \) are binary trees for which \( N(T_1) \leq 2^{H(T_1)} + 1 \) and \( N(T_2) \leq 2^{H(T_2)} + 1 \) and we create a new binary tree \( T' \) by creating a new root node and attaching both \( T_1 \) and \( T_2 \) to it. We claim that \( N(T') \leq 2^{H(T')} + 1 \).

  In creating this new binary tree the number of nodes in \( T \) is the sum of the number of nodes in \( T_1 \) and \( T_2 \) plus 1 more so:

  \[
  N(T') = N(T_1) + N(T_2) + 1
  \]

  The height is tricky. The height of \( T' \) will be 1 plus the maximum of the heights of \( T_1 \) and \( T_2 \). However it’s certainly the case that \( H(T') \geq H(T_1) + 1 \) and \( H(T') \geq H(T_2) + 1 \). Then we get:

  \[
  N(T') = N(T_1) + N(T_2) + 1 \leq 2^{H(T_1)} + 1 + 2^{H(T_2)} + 1 - 1 + 1 \\
  \leq 2^{H(T)} + 2^{H(T)} + 1 - 1 \\
  \leq 2^{H(T)} + 2^{H(T)} - 1 \\
  \leq 2^{H(T)} + 1 - 1
  \]

Note: Another approach to managing the height is as in the previous example.

\[\square\]