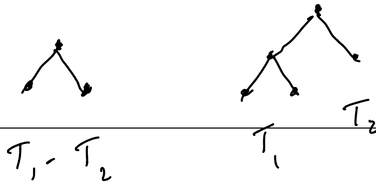
5.3 Structural Induction



A third type of induction, is useful for proving facts about objects defined recursively.

Examples of objects defined recursively:

- Full binary trees:
  I. **Base case**: There is a full binary tree consisting only of a single vertex r.
  II. **Recursion**: If T_1, T_2 are disjoint full binary trees, there is a full binary tree, denoted by $T_1 \cdot T_2$, consisting of a root r together with edges connecting the root to each of the roots of the left subtree T_1 and the right subtee T_2.
  III. **Restriction**: No full binary trees exist besides those derived from I. and II. above.

- The height of binary trees:
  I. **Base step**: The height of the full binary tree T consisting of only a single root r is $h(T) = 0$.
  II. **Recursion**: If T_1 and T_2 are full binary trees, then the full binary tree $T = T_1 \cdot T_2$ has height $h(T) = 1 + \max(h(T_1), h(T_2))$.

- The number of vertices for binary trees:

Base step: For T, a tree w/ just one node/root, let $n(T) = 1$

Recursion: Given trees T_1, T_2, define $n(T_1 \cdot T_2) = 1 + n(T_1) + n(T_2)$.

- Fibonacci numbers:
  I. **Base case**: $f_0 = 0, f_1 = 1$
  II. **Recursion**: for $n \geqslant 2$, $f_n = f_{n-1} + f_{n-2}$.

$f_0 = 0$
$f_1 = 1$
$f_2 = 1$
$f_3 = 2$
$f_4 = 3$

- Parenthesis structures P: Grammar of parentheses
  I. **Base step**: () is in P
  II. **Recursion**:
     * If E is in P, so is (E).
     * If E and F are in P, so is EF.

() ✓

(())((())) ✓

)(()() ✗

III. **Restriction**: No configurations of parentheses are in P besides those derived from I and II.

One can prove properties of recursively defined structures using a variation on induction called **structural induction**.

Definition 1 (Structural Induction): Let S be a set that has been defined recursively, and consider a property that objects in S may or may not satisfy. To prove that every object of S satisfies the property:

1. Show that each object in the BASE for S satisfies the property.
2. Show that for each rule in the RECURSION, if the rule is applied to objects in S that satisfy the property, then the objects defined by the rule also satisfy the property.

Examples:

$\doteq 1.618$ Golden ratio/section

$x = \frac{1}{2} \pm \frac{\sqrt{5}}{2}$ $\quad \phi^2 = \phi + 1$

1. Let $\phi = \frac{1}{2}(1 + \sqrt{5})$, which is a root of the equation $x^2 = x + 1$. Show that the Fibonacci numbers $f_n \geqslant \phi^{n-2}$ for $n \geqslant 3$.

Base case: f_3 compared w/ ϕ^1 $\quad f_3 = 2$ and $\phi^1 \doteq 1.618$ ✓

Let's assume our claim holds for $f_3, f_4, \ldots, f_k$. Show $f_{k+1} \geq \phi^{k-1}$

$$\phi^{k-1} = \phi^2 \cdot \phi^{k-3} = (\phi + 1)\phi^{k-3} = \phi^{k-2} + \phi^{k-3}$$

$$\leq f_k + f_{k-1} = f_{k+1}$$

2. Show that for all grammatical configurations of parentheses in P, there are an equal number of left- and right-parentheses.

Base case: ( ) has same number of left/right parens ✓.

Recursion step: Rule 1 $E \in P$ generates (E)

Since E is in P, the I.H. says it has equal no. of left/right parens

$\longrightarrow (E)$.

2^{nd} Rule: E, F w/ the claimed property generate EF.

m left/right parens

n left/right parens

has m+n left parens
also m+n right "
$\longrightarrow$ equal.

3. Show that for any full binary tree T, $n(T) \leqslant 2^{h(T)+1} - 1$.

Base step: Take T to be single-node tree

$n(T) = 1$, $h(T) = 0$ $\qquad 1 \leq 2^{0+1} - 1$. ✓

Recursion step: Take input trees T_1, T_2, supposing our claim holds for these — i.e.

$$n(T_1) \leq 2^{h(T_1)+1} - 1$$

$$n(T_2) \leq 2^{h(T_2)+1} - 1$$

Look at their product $T_1 \cdot T_2$

$$n(T_1 \cdot T_2) = 1 + \underline{n(T_1)} + \underline{n(T_2)}$$

$$\leq 1 + \left(2^{h(T_1)+1} - 1\right) + \left(2^{h(T_2)+1} - 1\right)$$

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

$$= 2 \cdot \left[2^{h(T_1)} + 2^{h(T_2)}\right] - 1$$

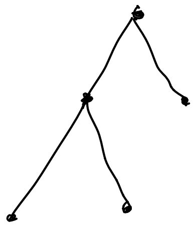
$$\leq 2 \cdot \left[2^{\max(h(T_1), h(T_2))} + 2^{\max(h(T_1), h(T_2))}\right] - 1$$

$= 2 \cdot 2^{\max(h(T_1), h(T_2))}$

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

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

$$\overset{\text{desired destination}}{=} 2^{\underline{h(T_1 \cdot T_2)} + 1} - 1$$

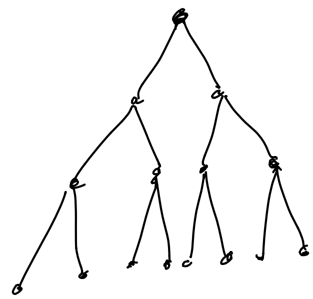


$n(T) = 5$

$h(T) = 2$

$n(T) \leq 2^{h(T)+1} - 1$

$5 \leq 2^{2+1} - 1 = 7$



T

$n(T) = 15$

$h(T) = 3$

$15 \leq 2^{3+1} - 1$

$15 \leq 15$.