Strong Induction cont.

Ex: Which dollar amounts can be formed using only \$2 and \$5 bills.

Prove using Strong Induction.

2, 4, 5, 6, 7, 8, 9, 10, ...

Claim: all amounts ≥ 4 can be formed

Basis Step: $P(4)$, $P(5)$

$P(4)$: $\$4 = \$2 + \$2$ ✓

$P(5)$: $\$5 = \$5$ ✓

Inductive Step:

Assume $P(j)$ for $j \le k$, some $k \ge 5$

show $P(k+1)$

Case k even:

$$k = 2 \cdot i + 5 \cdot j$$

$$k+1 = 2 \cdot i + 5j + 1$$

$$= 2(i-2) + 5(j+1)$$

$$= 2i - 4 + 5j + 5$$

$$= 2i + 5j + 1 \quad \checkmark$$

remove 2 \$2 bills
add 1 \$5 bill

case k odd:

$$k = 2i + 5j$$

$$k+1 = 2(i+3) + 5(j-1)$$

$$= 2i + 6 + 5j - 5$$

$$= \underbrace{2i + 5j}_{k} + 1 \quad \checkmark$$

remove 1 \$5 bill
add 3 \$2 bills

$\therefore P(n)$ holds for $\forall n \geq 4$

Recursion and Structural Induction

Recursive Definitions

Basis Step: Value @ zero
initial set

Recursive step: rule for creating
new values/elements

from previous
values/elements

Example:
well-formed formulae (WFF)
in propositional logic

Basis Step: T, F, s are wffs
s is a propositional
variable.

Recursive step:
If E, F are WFFs,
then $(\neg E)$ . . . .

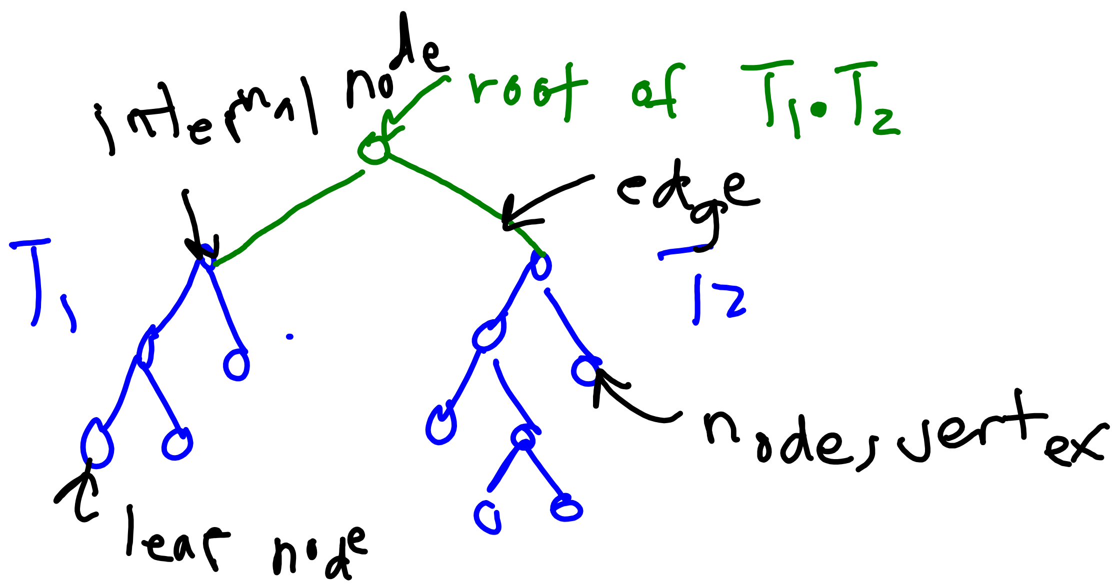
$(E \vee F)$ minimal required
$(E \wedge F)$
$(E \to F)$
$(E \leftrightarrow F)$
$(E \oplus F)$
are also WFFs.

Full Binary Trees

Basis Step: A single vertex r is a full binary tree

Recursive Step:
If T_1, T_2 are full binary trees

then $T_1 \cdot T_2$, consisting of a root r with left child T_1 and right child T_2, is a full binary tree.



Height of a full binary tree

Basis Step: the height of the

tree with only a
root is
$h(r) = 0$

Recursive Step:
if T_1, T_2 are full binary trees,
then $T_1 \cdot T_2 = T$ has height

$$h(T) = 1 + \max(h(T_1), h(T_2))$$

Number of vertices of a full binary tree

Basis Step: ... with only a root has
$n(T) = 1$ vertex

Recursive Step:
If T_1, T_2 are full binary trees, then $T = T_1 \cdot T_2$ has

$$n(T) = 1 + n(T_1) + n(T_2) \text{ vertices}$$

<u>Claim</u>: if T is a full binary tree, then

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

Structural Induction

Basis Step: Show that the result holds for the recursive basis step.

Recursive Step: Show that, if the result holds for each element used to recursively construct the new value/element then the result hold for the new value/element.

Proof that $n(T) \le 2^{h(T)+1} - 1$

Basis Step: Tree with 1 node

$n(T) = 1 \qquad 2^{0+1} - 1 = 2 - 1 = 1$

$h(T) = 0 \qquad 1 \le 1$ ✓

Recursive step:

Assume $n(T_i) \le 2^{h(T_i)+1} - 1$

T_1, T_2 are full binary Trees

show $T = T_1 \cdot T_2$ has

$h(T)+1$

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

$$n(T) = 1 + n(T_1) + n(T_2)$$ by rec def.

$$\le 1 + \underline{2^{h(T_1)+1}} - 1 + \underline{\underline{2^{h(T_2)+1}}} - 1$$ by #4

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

$$\xrightarrow{\text{bc clever}} \le 2\max\left(2^{h(T_1)+1}, 2^{h(T_2)+1}\right) - 1$$

$$a+b \le 2 \cdot \max(a, b)$$

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

$$\xrightarrow{\text{rec def of } h(T)} = 2 \cdot 2^{h(T)} - 1$$

$$h(T) = 1 + \max(h(T_1), h(T_2))$$

$$= 2^{h(T)+1} - 1$$

Claim: $n(T) \le 2^{h(T)+1} - 1$

Basis Step: let $T = r$

$n(T) = 1 \quad 2^{0+1} - 1 = 2 - 1 = 1$

$h(T) = 0$

$1 \le 1$

Recursive step:

Assume $n(T_i) \le 2^{h(T_i)+1} - 1$ for full binary trees T_1, T_2

Show $n(T) \le 2^{h(T)+1} - 1$ for $T = T_1 \cdot T_2$

$n(T) = 1 + n(T_1) + n(T_2)$

$$\leq 1 + 2^{h(T_1)+1} - 1 + 2^{h(T_2)+1} - 1 \quad \underline{\text{IH}}$$

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

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

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

$$= 2 \cdot 2^{h(T)} - 1$$

$$= 2^{h(T)+1} - 1 \quad \checkmark$$

$\square$

Strings

let Σ be a set of symbols called alphabet.

e.g. $\Sigma = \{a, b, c, \ldots, z\}$

$\Sigma = \{0, 1\}$

Σ^* is the set of all strings composed of symbols from Σ

recursive definition of Σ^*:

Basis step: $\varepsilon \in \Sigma^*$

ε is the empty string

Recursive step:

If $w \in \Sigma^*$

then $wa \in \Sigma^*$ for $a \in \Sigma$

wa is the concatenation of a onto w.

$w = a$

e.g. $x = \underline{c}\,a\,\underline{t} \Rightarrow w^x = cat$

Concatenation is recursively defined

Basis step: $w\varepsilon = w$, $w \in \Sigma^*$

Recursive step:

if $w_1 \in \Sigma^*$, $w_2 \in \Sigma^*$, $x \in \Sigma$

then $w_1(w_2 x) = (w_1 w_2)x$

length of a string $\ell(w)$

Basis step: $\ell(\varepsilon) = 0$

Recursive step:

if $w \in \Sigma^*$, $x \in \Sigma$

then $\ell(wx) = \ell(w) + 1$

Claim: $l(xy) = l(x) + l(y)$

Let $P(y) := l(xy) = l(x) + l(y)$

Basis step: let $y = \varepsilon$

$P(\varepsilon) := l(x\varepsilon) = l(x) + l(\varepsilon)$

$l(x\varepsilon) = l(x)$ $\quad l(x) + l(\varepsilon)$

$l(\varepsilon) = 0$

$= \checkmark$

Recursive step:

Assume $l(xy) = l(x) + l(y)$ $\quad P(y)$

for some $y \in \Sigma^*$

show $l(xya) = l(x) + l(ya)$ $\quad P(ya)$

for $y \in \Sigma^*$, $a \in \Sigma$

$l(xya) = l(xy) + 1$

$= \ell(x) + \ell(y) + 1$ by IH

$= \ell(x) + \ell(ya)$

✓

Solve $A_n = 2A_{n-1} + 3$, $A_0 = 1$

prove your answer.

$$A_n = 2A_{n-1} + 3$$

$$= 2(2A_{n-2} + 3) + 3 = 2^2 A_{n-2} + 2 \cdot 3 + 3$$

$$= 2^2(2A_{n-3} + 3) + 2 \cdot 3 + 3$$

$$= 2^3 A_{n-3} + 2^2 \cdot 3 + 2 \cdot 3 + 3 \cdot 2^0$$

$\vdots$

$$= 2^n A_0 + 3 \sum_{i=0}^{n-1} 2^i$$

$$= 2n + 3\frac{2^{n}-1}{2-1} = 2^{n} + 3(2^{n}-1)$$

$$= 2^{n} + 3 \cdot 2^{n} - 3$$

$$= 4 \cdot 2^{n} - 3$$

$$\boxed{A_{n} = 2^{n+2} - 3}$$

Proof by Induction

Basis case: $n = 0$

$A_0 = 1$ $\quad A^{0} = 2^{0+2} - 3 = 2^{2} - 3 = 4 - 3 = 1$

$1 = 1$ ✓

Inductive step

Assume $A_{k} = 2^{k+2} - 3$

Show $A_{k+1} = 2^{k+3} - 3$ ←

$A_{k+1} = 2A_{k-2} + 3$ by rec def

$= 2(2^{k-1} - 3) + 3$ by IH

$= 2^{k+3} - 6 + 3$

$= 2^{k+3} - 3$ ← ✓

$\square$