Lecture 9
Feb. 3rd, 2015

Recall: a binary tree is empty or a left binary tree and a right binary tree
Let $\mathbb{BT}$ be the set of Binary Trees
Define l on $\mathbb{BT}$ by:

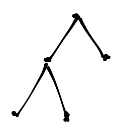
$$l(\text{empty}) = 0$$
$$l(\text{single node}) = 0$$

If $t \in \mathbb{BT}$, $t \neq$ empty, $t \neq$ single node, let t_l and t_r be the left and right subtree.

$$l(t) = l(t_l) + l(t_r)$$

note: l stands for the number of leaves of a given binary tree.

$$l(bt) = l(\text{left sub}) + l(\text{right sub}) = l(\text{dot}) + l(\text{dot}) + l(\text{dot})$$
$$= 1 + 1 + 1 = 3$$



e.g. For $t \in \mathbb{BT}$, $l(t) \leq 2^{h(t)-1}$

Proof by Structural Induction:

Base Case empty

$$l(\text{empty}) = 0 \quad (\text{by definition})$$
$$\leq \frac{1}{2} = 2^0 = 2^{h(\text{empty})-1} \quad (\text{by definition of } h)$$

Inductive Step
Let $t \in \mathbb{BT}$ with left and right subtrees t_l, t_r.
Case t_l, t_r are empty, i.e. t is single node.

(IH) Assume $l(t_l) \leq 2^{h(t_l)-1}$, $l(t_r) \leq 2^{h(t_r)-1}$

$l(t) = 1$ (by definition)
$h(t) = 1 + \max(h(t_l) + h(t_r))$ (by defn)
$= 1 + \max(h(\text{empty}) + h(\text{empty}))$
$= 1 + \max(0, 0)$ (by defn)
$= 1$
$l(t) = 1 = 2^0 = 2^{1-1} = 2^{h(t)-1} \leq 2^{h(t)-1}$

Case t not just a single node
(IH) Assume $l(t_l) \leq 2^{h(t_l)-1}$, $l(t_r) \leq 2^{h(t_r)-1}$

$$
\begin{aligned}
l(t) &= l(t_r) + l(t_l) \quad \text{(by defn)} \\
&\leq 2^{h(t_l)-1} + 2^{h(t_r)-1} \quad \text{(by IH)} \\
&\leq 2^{\max(h(t_l), h(t_r))-1} + 2^{\max(h(t_l), h(t_r))-1} = 2 \cdot 2^{\max(h(t_l), h(t_r))-1} = 2^{1+\max(h(t_l), h(t_r))-1} \\
&= 2^{h(t)} - 1
\end{aligned}
$$

Let $\mathbb{BT}$ be defined by:

empty $\in \mathbb{BT}$,
if left, right $\in \mathbb{BT}$, then total $\in \mathbb{BT}$.

$[P(\text{empty}), \forall \text{left}, \text{right} \in \mathbb{BT}, P(\text{left}) \wedge P(\text{right}) \Rightarrow P(\text{total})] \Rightarrow \forall \text{part} \in \mathbb{BT}, P(\text{part})$

Question: Nonempty full BTs have an odd number of nodes.
For $t \in \mathbb{BT}$, let $P(t)$ be: if t not empty tree and t is full, then t has an odd number of nodes.

Proof by structural induction:
P(empty): vacuously true
INDUCTIVE STEP:
Let $t \in \mathbb{BT}$ with left and right subtrees t_l, t_r.
(IH) If t_l is full and not empty, then it has odd number of nodes.
If t_r is full and not empty, then it has odd number of nodes
Assume t not empty and t is full, either t_l and t_r are both empty, or they are both not empty.
If t_l, t_r are empty, t is just single node, so odd number nodes.
If t_l, t_r are non-empty and full, they by (IH) they have an odd number of nodes
The number of nodes in t is $1 + \#$ in t_l + $\#$ in t_r is 1 + odd + odd = odd