Mathematical Induction and the Size of Game Trees

Each path from 'start' to a terminal node, labeled 'p' (for pass) in a game tree corresponds to a possible sequence of bids and vice-versa. We would like to determine the number terminal nodes in a game tree corresponding to bankroll b. That is, with a bankroll b how many different ways the bidding could progress. Note that the shape of the trees depends only on the bankroll b and not on the stakes s. The stakes are used to determine the payoff labels attached to the terminal nodes in the analysis of the game.

To start, look at some examples with small values of b. We will use the notation T_b to refer to the tree for bankroll b.

Example 1: For $b = 1$ there is only one possible bidding sequence. Player 1 bids 1 and player 2 passes. The tree T_1 is shown in figure 1.

Example 2: For $b = 2$ we have the tree in figure 2 with $4 = 2^2$ non-terminal nodes and with the $3 = 2^2 - 1$ terminal nodes corresponding to the bid sequences as follows. The node labeled $\{1\}$ has player 1 bidding 1 followed by player 2 passing; the node labeled $\{1, 2\}$ has player 1 bidding 1, followed by player 2 bidding 2 followed by player 1 passing, the node labeled $\{2\}$ has player 1 bidding 2 followed by player 2 passing. Thus the numbers represent the values bid, by player 1 and alternately player 2. (The bids are always increasing in an auction.)

Example 3: To construct the tree T_3 for $b = 3$ from the tree T_2 for $b = 2$ note that bidding sequences which *do not* use the bid 3 are already listed in T_2. Bidding sequences which *do* use the bid 3 end with the bids 3, pass. The bids preceding these last two bids do not use the bid 3, and thus are a sequence of bids in T_2 which do not end in pass. This sequence corresponds to a non-terminal node in T_2.

So, to construct T_3, copy down T_2 and for each non-terminal node, including start, in T_2 attach the two nodes 3, pass. (This is done in figure 3, with edges leading to attached nodes dashed.)

The number of nodes in T_3 can be determined from the description of the construction. (Since we begin with T_2 and for each non-terminal node in T_2 we create both a new terminal node and a new non-terminal node.)

$$\begin{aligned}
\#\text{Terminal nodes in } T_3 &= \#\text{Terminal nodes in } T_2 + \#\text{Non-terminal nodes in } T_2 \\
&= 2^2 + (2^2 - 1) = 2^3 - 1 \\
&\text{and} \\
\#\text{Non-terminal nodes in } T_3 &= \#\text{Non-terminal nodes in } T_2 + \#\text{Non-terminal nodes in } T_2 \\
&= 2^2 + 2^2 = 2^3
\end{aligned}$$

Example 4: To construct T_4, copy down T_3 and for each non-terminal node, including start, in T_3 attach the two nodes 4, pass. (This is done in figure 4, with edges leading to attached nodes dashed.)

The number of nodes in T_4 can then be determined by the description of the construction. (Since we begin with T_3 and for each non-terminal node in T_3 we create both a new terminal node and a new non-terminal node.)

$$\begin{aligned}
\#\text{Terminal nodes in } T_4 &= \#\text{Terminal nodes in } T_3 + \#\text{Non-terminal nodes in } T_3 \\
&= 2^3 + (2^3 - 1) = 2^4 - 1 \\
&\text{and} \\
\#\text{Non-terminal nodes in } T_4 &= \#\text{Non-terminal nodes in } T_3 + \#\text{Non-terminal nodes in } T_3 \\
&= 2^3 + 2^3 = 2^4
\end{aligned}$$

We start to see a pattern and see that to get T_5 we would copy down T_4 and for each non-terminal node in T_4 attach two new nodes, one non-terminal (labeled 5) and one terminal. Similarly we get T_6, T_7, T_8, $\cdots$. When we get to T_{100} we should have the following.

Example 100: To construct T_{100}, copy down T_{99} and for each non-terminal node, including start, in T_{99} attach the two nodes 100, pass. The number of nodes in T_{100} can then be determined by the description of the construction. (Since we begin with T_{99} and for each non-terminal node in T_{99} we create both a new terminal node and a new non-terminal node.)

$$\begin{aligned}
\#\text{Terminal nodes in } T_{100} &= \#\text{Terminal nodes in } T_{99} + \#\text{Non-terminal nodes in } T_{99} \\
&= 2^{99} + (2^{99} - 1) = 2^{100} - 1 \\
&\text{and} \\
\#\text{Non-terminal nodes in } T_{100} &= \#\text{Non-terminal nodes in } T_{99} + \#\text{Non-terminal nodes in } T_{99} \\
&= 2^{99} + 2^{99} = 2^{100}
\end{aligned}$$

Note that we have used the number of nodes in T_{99} in order to determine the number of nodes in T_{100}. So we really should write down examples 6,7,8,9,10,11, ... 96,97,98,99 *before* we write down example 100. This would take a long time and still would not accomplish the goal of showing that say T_{1996} has 2^{1996} non-terminal nodes. That would require writing down 1996 examples!

The key is to notice that the examples all follow the same pattern. So instead of actually writing down all 100 (or 1996 or $\cdots$) examples we will show that we *could* in principle write down all these examples, arriving at the conclusion that we want for T_{100} (or T_{1996} or $\cdots$). The key then is to write down a *template* for doing all the examples. By examining the first few examples we see that we get:

Example b: To construct T_b, copy down T_{b-1} and for each non-terminal node, including start, in T_{b-1} attach the two nodes b, pass. The number of nodes in T_b can then be determined by the description of the construction. (Since we begin with T_{b-1} and for each non-terminal node in T_{b-1} we create both a new terminal node and a new non-terminal node.)

$$\begin{aligned}
\#\text{Terminal nodes in } T_b &= \#\text{Terminal nodes in } T_{b-1} + \#\text{Non-terminal nodes in } T_{b-1} \\
&= 2^{b-1} + (2^{b-1} - 1) = 2^b - 1 \\
&\text{and} \\
\#\text{Non-terminal nodes in } T_b &= \#\text{Non-terminal nodes in } T_{b-1} + \#\text{Non-terminal nodes in } T_{b-1} \\
&= 2^{b-1} + 2^{b-1} = 2^b
\end{aligned}$$

What we have just done is a proof by a technique known as *mathematical induction*. We have shown that the game tree T_b for bankroll b has 2^b non-terminal and $2^b - 1$ non-terminal nodes. To really complete the proof we need a starting point. Our generic example does not work for T_1. So we must do that example by itself. But from example 1, we see that T_1 has $2^1 - 1 = 1$ terminal node and $2^1 = 2$ non-terminal nodes. So the node counts for T_1 are correct.

To reiterate what we do for the proof: we write down a generic template which could be used to count the number of nodes in T_b for *any* value of b. This template uses information about the number of nodes in T_{b-1}. (Called the inductive hypothesis.) We then see that for any particular value of b we *could* write down examples $1, 2, 3, \ldots b-2, b-1, b$ arriving at the desired result for b. *But*, because the template is generic and we see that it will work for any example we do not actually have to write all these examples down. Note that we do not even need to write down examples 2,3 and 4 as we did above. These were done for motivation so that we would believe the generic example b but would not be needed in a formal proof. We need only our starting point, example 1 and the generic example.

As a final note, observe that the sets labeled on the pass nodes for T_3 correspond to subsets of $\{1, 2, 3\}$ and on T_4 to subsets of $\{1, 2, 3, 4\}$. We have listed all possible subsets except the empty set. Thus our proof also shows that the number of different nonempty subsets (corresponding in our case to a list of all values bid) of $\{1, 2, \ldots, b-1, b\}$ is $2^b - 1$. We could have also used this result about subsets to get the correct counts on terminal nodes in the game tree.