CSC236 Week 3 Tutorial:

Complete and Structural Induction

Slides Credit: Charlene Chu

Preliminary

Here is a recursive definition for some set T of non-empty binary trees.

- A single node is in T
- If t_1 and t_2 are in T, then the bigger tree with root r connected to the roots of t_1 and t_2 is in T
- Nothing else is in T

Use structural induction to prove the following property for each of the elements in T: there are m nodes that have two children and $m+1$ nodes that have no children, *i.e.*, the number of nodes with no children is one plus the number of nodes with two children.

Exercise 1

A *full binary tree* is a binary tree where every node has either 0 or 2 children. Prove that every non-empty full binary tree has an odd number of nodes.

Exercise 2

Consider binary strings that start with a 1. By interpreting the 1's as specifying powers of 2, these strings are called *binary representations* of positive integers. For example:

$$\begin{aligned} 10 &= 2^1 = 2 \\ 1011 &= 2^3 + 2^1 + 2^0 = 11 \\ 110001 &= 2^5 + 2^4 + 2^0 = 49 \end{aligned}$$

Prove that every natural number $n \geq 1$ has a binary representation. For a challenge, also prove that the binary representation is unique.