CSC236 – Tutorial 2: Complete and Structural Induction

Exercise 1: Powers of 2

Prove that every natural number $n \geq 1$ can be written as the sum of distinct powers of 2. For example, $9 = 2^3 + 2^0$, $15 = 2^3 + 2^2 + 2^1 + 2^0$. (This is a useful result to prove, as it shows that all natural numbers have at least one binary representation.)
(Bonus: prove that there is exactly one binary representation for $n \geq 1$ whose most-significant bit is a 1.)

Exercise 2: Full Binary Trees

A *full binary tree* is a binary tree where every nonleaf node has 2 children. Prove — first by complete induction, second by structural induction — that every non-empty full binary tree has an odd number of nodes.

Exercise 3: Well-Formed Formulas

The XOR ("exclusive or") boolean operator $\oplus$ has the following truth table:

P	Q	$P \oplus Q$
T	T	F
T	F	T
F	T	T
F	F	F

That is, $\oplus$ returns true if and only if exactly one of its arguments is true.
We can define the set of *well-formed XOR formulas* recursively by:

1. Any propositional variable P is a well-formed XOR formula.
2. If S_1, S_2 are two well-formed XOR formulas, then so is $(S_1 \oplus S_2)$.

Prove that every well-formed XOR formula is true if and only if it has an odd number of variable occurrences that are true. (Note: if P were true, then $P \oplus P$ would have **two** variable occurrences that are true, not one!)