Assignment 4

due Wednesday, September 21

Problem 1 (2 points)

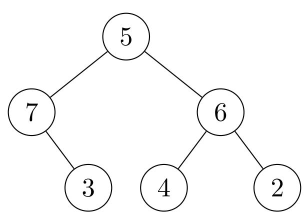
Consider a variation of the Towers of Hanoi puzzle where we assume the pegs A, B, and C are in a row, and we are only allowed to transfer a disk to an *adjacent* peg, which is to say, moves from A to C or vice-versa are ruled out. Convince yourself that the following algorithm works:

procedure HANOIADJ(n, A, B, C)
  if $n = 0$ **then**
    return
  else
    move $n - 1$ disks from A to C
    move the last disk from A to B
    move $n - 1$ disks from C to A
    move the last disk from B to C
    move $n - 1$ disks from A to C
  end if
end procedure

Write a Lean program to output the list of moves required to move n disks. (For an extra challenge, try to figure out how many steps it takes.)

Problem 2 (4 points)

A binary tree with nodes labeled from a datatype α is just what it sounds like. For example, the following is a binary tree with nodes labeled by natural numbers:



As in the textbook, by "binary tree" we really mean "extended binary tree," which means that we count the empty tree.

Do the following:

1. Define a datatype LBinTree α in Lean. It should be similar to BinTree, as defined in the textbook, but the node constructor should include the label, like the cons constructor for List.

2. Define myTree : LBinTree Nat corresponding to the example above.

3. Define a function addNodes : LBinTree Nat $\to$ LBinTree Nat that adds up the nodes of a tree with labels from Nat. On the example above, it should return 27.

4. Define a function toListInorder that creates a list with an *inorder* traversal (left subtree first, then the node, then the right subtree). On the example above, it should return $[7, 3, 5, 4, 6, 2]$.

Problem 3 (3 points)

Write a Lean procedure pascal which, on input n, outputs the first n rows of Pascal's triangle. We adopt the convention that the row numbers start with 0, so the row numbered n should have $n+1$ entries, $\binom{n}{0}, \binom{n}{1}, \ldots, \binom{n}{n}$. For example, on input 6, the program should output the following:

```
0: 1
1: 1 1
2: 1 2 1
3: 1 3 3 1
4: 1 4 6 4 1
5: 1 5 10 10 5 1
```

Problem 4 (3 points)

Use the definition of "A is a subformula of B" in Section 4.1 to prove that if A, B, and C are any propositional formulas, A is a subformula of B, and B is a subformula of C, then A is a subformula of C.

(Hint. This is tricky. Remember that, by definition, "A is a subformula of B" means $A \in \mathrm{subformulas}(B)$. Show by induction C that for every A and B, if A is a subformula of B and B is a subformula of C, then A is a subformula of C.)

Problem 5 (2 points)

Prove the following carefully, using the semantic definitions in Section 4.2: let Γ and Γ' be sets of propositional formulas and let A be a propositional formula. If $\Gamma \models A$ and $\Gamma' \supseteq \Gamma$, then $\Gamma' \models A$.

Problem 6 (2 points)

Prove the following carefully: $\{A \to B, A\} \models B$.

Problem 7 (2 points)

Use an algebraic calculation to prove $(A \to B) \wedge A \equiv A \wedge B$, using the equivalences in Section 4.3 and the equivalence $U \to V \equiv \neg U \vee V$.

Problem 8 (2 points)

Use an algebraic calculation to prove $(A \to B) \vee \neg A \equiv A \to B$. You can carry out multiple applications of associativity and commutativity of $\vee$ in one step.