The Towers of Hanoi

We are given 3 pegs $A$, $B$ and $C$, and $n$ disks with a hole, such that no two disks have the same diameter. Initially all $n$ disks rest on peg $A$, ordered according to their size, with the largest one at the bottom. The aim is to transfer all $n$ disks from $A$ to $C$ by a sequence of single-disk moves such that we never place a larger disk on top of a smaller one. Peg $B$ may be used for intermediate storage.

The pegs and moves can be modelled as follows:

```isabelle
datatype peg = A | B | C

type_synonym move = "peg * peg"
```

Define a primitive recursive function

```isabelle
consts
  move :: "nat => peg => peg => move list"
```

such that $\text{move } n \ a \ b$ returns a list of (legal) moves that transfer $n$ disks from peg $a$ to peg $c$.

Show that this requires $2^n - 1$ moves:

```isabelle
theorem "length (move n a b) = 2^n - 1"
```

Hint: You need to strengthen the theorem for the induction to go through. Beware: subtraction on natural numbers behaves oddly: $n - m = 0$ if $n \leq m$. 

Correctness

In the last section we introduced the towers of Hanoi and defined a function \texttt{move} to generate the moves to solve the puzzle. Now it is time to show that \texttt{move} is correct. This means that

- when executing the list of moves, the result is indeed the intended one, i.e. all disks are moved from one peg to another, and
- all of the moves are legal, i.e. never is a larger disk placed on top of a smaller one.

Hint: This is a non-trivial undertaking. The complexity of your proofs will depend crucially on your choice of model, and you may have to revise your model as you proceed with the proof.