Workshop Testing and Formal Methods, Week 5, With Answers

```
module Workshop7Answers where

import Data.List
```

This workshop is about proving things by induction, and about the connection between recursion and induction. More precisely, this workshop is about how you prove things by induction on data-types defined by recursion. Background reading for this topic is "The Haskell Road", Chapter 7.

For further instruction on this topic from YouTube, have a look at `http://www.khanacademy.org/math/algebra/algebra-functions/v/proof-by-induction`. You should watch this video.

The simplest case of proof by (mathematical) induction is induction on the natural numbers. The task is to prove that a property P holds of all natural number. The proof has two steps:

1. Show that the property P holds for the number 0.
2. Show that if the property P holds for the number n, then it will also hold for $n+1$.

In the second case, the assumption that P holds for n is called the *induction hypothesis*. The best way to learn proof by induction is by practice. Here we go. (And again: please watch the video mentioned above.)

Question 1 Prove by induction that it holds for all natural numbers n that

$$1 + 2 + \cdots + n = \frac{n(n+1)}{2}.$$

Answer: To be proved: for all $n \in \mathbb{N} : 1 + 2 + \cdots + n = \frac{n(n+1)}{2}$.
Proof by induction.
Base case: for $n = 0$ the property holds.
Induction step.
Induction hypothesis: $1 + 2 + \cdots + n = \frac{n(n+1)}{2}$.
To be proved: $1 + 2 + \cdots + n + (n + 1) = \frac{(n+1)(n+2)}{2}$.
Note: $\frac{(n+1)(n+2)}{2}$ is the result of substituting $n + 1$ for n in $\frac{n(n+1)}{2}$.

Proof (of the induction step):

$$
\begin{aligned}
1+2+\cdots+n+(n+1) &\stackrel{ih}{=} \frac{n(n+1)}{2}+(n+1) \\
&= \frac{n(n+1)}{2}+\frac{2(n+1)}{2} \\
&= \frac{(n+2)(n+1)}{2} \\
&= \frac{(n+1)(n+2)}{2}
\end{aligned}
$$

Note: step $\stackrel{ih}{=}$ uses the induction hypothesis.

Question 2 Prove by induction that it holds for all natural numbers n that

$$1^2+2^2+\cdots+n^2=\frac{n(n+1)(2n+1)}{6}.$$

Answer: To be proved: for all $n \in \mathbb{N} : 1^2+2^2+\cdots+n^2=\frac{n(n+1)(2n+1)}{6}$.
Proof by induction.
Base case: for $n=0$ the property holds.
Induction step.
Induction hypothesis: $1^2+2^2+\cdots+n^2=\frac{n(n+1)(2n+1)}{6}$.
To be proved: $1^+2^2+\cdots+n^2+(n+1)^2=\frac{(n+1)(n+2)(2n+3)}{6}$.
Proof:

$$
\begin{aligned}
1^2+2^2+\cdots+n^2+(n+1)^2 &\stackrel{ih}{=} \frac{n(n+1)(2n+1)}{6}+(n+1)^2 \\
&= \frac{n(n+1)(2n+1)}{6}+\frac{6(n+1)^2}{6} \\
&= \frac{(2n^2+n)(n+1)}{6}+\frac{(6n+6)(n+1)}{6} \\
&= \frac{(2n^2+7n+6)(n+1)}{6} \\
&= \frac{(n+2)(2n+3)(n+1)}{6} \\
&= \frac{(n+1)(n+2)(2n+3)}{6}
\end{aligned}
$$

Question 3 Prove by induction that it holds for all natural numbers n that

$$1^3+2^3+\cdots+n^3=\left(\frac{n(n+1)}{2}\right)^2.$$

Answer: To be proved: for all $n \in \mathbb{N} : 1^3 + 2^3 + \cdots + n^3 = \left(\frac{n(n+1)}{2}\right)^2$.
Proof by induction.
Base case: for $n = 0$ the property holds.
Induction step.
Induction hypothesis: $1^3 + 2^3 + \cdots + n^3 = \left(\frac{n(n+1)}{2}\right)^2$.
To be proved: $1^3 + 2^3 + \cdots + n^3 + (n+1)^3 = \left(\frac{(n+1)(n+2)}{2}\right)^2$.
Proof:

$$
\begin{aligned}
1^3 + 2^3 + \cdots + n^3 + (n+1)^3 &\stackrel{ih}{=} \left(\frac{n(n+1)}{2}\right)^2 + (n+1)^3 \\
&= \frac{n^2(n+1)^2}{4} + \frac{4(n+1)^3}{4} \\
&= \frac{n^2(n+1)^2}{4} + \frac{(4n+4)(n+1)^2}{4} \\
&= \frac{(n+1)^2(n^2+4n+4)}{4} \\
&= \frac{(n+1)^2(n+2)^2}{4} \\
&= \left(\frac{(n+1)(n+2)}{2}\right)^2.
\end{aligned}
$$

Question 4 Prove by induction that if A is a finite set with $|A| = n$, then $|\mathcal{P}(A)| = 2^n$.

Answer: The empty set has 0 members and $\mathcal{P}(\emptyset) = \{\emptyset\}$ has $1 (= 2^0)$ member, so in the base case the assertion holds.

Suppose that if $|A| = n$, then $|\mathcal{P}(A)| = 2^n$. Now let A' be such that $A \subseteq A'$ and $|A'| = n + 1$. Then there is exactly one object x with $x \notin A$ and $A' = A \cup \{x\}$. Now consider the subsets B' of A'. How many of those are there? The subsets B' of A' are of two kinds: those with $x \in B'$ and those with $x \notin B'$. In fact, for any subset B of A, $B \subseteq A'$ and $B \cup \{x\} \subseteq A'$, and $B \neq B \cup \{x\}$. Thus A' has twice as many subsets as A. and we have, by induction hypothesis, that $|\mathcal{P}(A')| = 2 \cdot 2^n = 2^{n+1}$.

Question 5 A permutation of a list is a reordering of the members of a list. Here is a Haskell implementation:

```
perms :: [a] ->[[a]]
perms [] = [[]]
perms (x:xs) = concat (map (insrt x) (perms xs)) where
  insrt x [] = [[x]]
  insrt x (y:ys) = (x:y:ys) : map (y:) (insrt x ys)
```

Find a formula (closed form) for the number of permutations of a list of n distinct objects, and proof your guess by induction.

Answer: There are $n!$ permutations for a list of n distinct objects. The empty list has a single permutation, and indeed, $0! = 1$ (by the convention for an empty product).

Suppose a list of n distinct objects has $n!$ permutations. Then there are $n + 1$ ways to insert a new object into one of these. Together this gives $(n + 1) \times n! = (n + 1)!$ permutations of a list of $n + 1$ distinct elements.

Question 6 Prove by induction that it holds for all natural numbers n that

$$3^{2n+3} + 2^n \text{ is divisible by } 7.$$

Answer: To be proved: for all $n \in \mathbb{N}$ there is an $A \in \mathbb{N}$ such that $3^{2n+3} + 2^n = 7A$.
Proof by induction.
Base case: for $n = 0$ we have $3^3 + 2^0 = 27 + 1 = 28 = 7 \cdot 4$, so the property holds.
Induction step.
Induction hypothesis: for some $A \in \mathbb{N}$ it holds that $3^{2n+3} + 2^n = 7A$.
To be proved: for some $B \in \mathbb{N}$ it holds that $3^{2n+5} + 2^{n+1} = 7B$.
Proof:

$$\begin{aligned}
3^{2(n+1)+3} + 2^{n+1} &= 3^{2n+2+3} + 2 \cdot 2^n \\
&= 3^2 \cdot 3^{2n+3} + 3^2 \cdot 2^n - (3^2 - 2) \cdot 2^n \\
&= 3^2(3^{2n+3} + 2^n) - 7 \cdot 2^n \\
&\overset{ih}{=} 3^2(7A) - 7 \cdot 2^n = 7(3^2 \cdot A - 2^n).
\end{aligned}$$

It is not necessary to have 0 as base case. Here are some examples where the base case is a different number.

Question 7 Show by induction that for all natural numbers n with $n \geq 3$ it holds that $n^2 > 2n$.

Answer: To be proved: for all $n \in \mathbb{N}$ with $n \geq 3$ it holds that $n^2 > 2n$.
Proof by induction.
Base case: for $n = 3$ we have $3^2 = 9 > 2 \cdot 3 = 6$, so the property holds.
Induction step.
Induction hypothesis: $n^2 > 2n$.
To be proved: $(n+1)^2 > 2(n+1)$.
Proof:
From $n^2 > 2n$, which is true by induction hypothesis, it follows that $n^2 + 2n + 1 > 2n + 2n + 1$. Since we can assume that $n \geq 3$, we also have $2n > 1$. Combining these inequalities, we get $(n+1)^2 = n^2 + 2n + 1 > 2n + 2n + 1 > 2n + 2 = 2(n+1)$.

Question 8 Show by induction that for all natural numbers n with $n \geq 5$ it holds that $2^n > n^2$.

Answer: To be proved: for all $n \in \mathbb{N}$ with $n \geq 5$ it holds that $2^n > n^2$.
Proof by induction.
Base case: for $n = 5$ we have $2^5 = 32 > 5^2 = 25$, so the property holds.
Induction step.
Induction hypothesis: $2^n > n^2$.
To be proved: $2^{n+1} > (n+1)^2$.
Proof:
We need to show that $2^{n+1} = 2 \cdot 2^n = 2^n + 2^n > (n+1)^2 = n + 2n + 1$. From the induction hypothesis we get that $2^n > n^2$. From the previous exercise we have that $n^2 \geq 2n + 1$ for all $n \geq 3$. Since $n \geq 5$ this result applies, so we get: $2^{n+1} = 2 \cdot 2^n = 2^n + 2^n \overset{ih}{>} n^2 + n^2 \geq n^2 + 2n + 1 = (n+1)^2$.

Question 9 Consider the following game for two players. Starting situation: a number of matches is on a stack. The players take turns. A move consists in removing 1, 2 or 3 matches from the stack. The player who can make the last move (the move that leaves the stack empty) has won the game.

Suppose there are 4N matches on the stack, and the other player moves. How should you respond? Prove by induction that your strategy assures that you will win the game.

Answer: To be proved: for all $N \geq 1$ it is the case that if there are $4N$ matches on the stack and player B moves, then player A wins.

Base case: there are 4 matches left. Player B can take 1,2 or 3 matches, so there are 3, 2, or 1 matches left. Player A takes them all and wins.

Induction step. Assume that if there are $4N$ matches left and player B moves, then player A wins. To show: if there are $4(N+1)$ matches left and player B moves, then player A wins.

Let there be $4(N+1) = 4N+4$ matches left. Player B can take 1, 2 or 3 matches. Player A responds with taking 3, 2, or 1 matches, and there $4N$ matches left with player B moving. According to the induction hypothesis this is a win for player A.

A useful generalisation of mathematical induction on the natural numbers is *structural induction*, where we wish to prove that some property P holds of all members of a recursively defined datatype such as trees, lists or formulas.

Question 10 Recall the earlier definition of formulas of propositional logic:

$$\varphi ::= p \mid (\neg\varphi) \mid (\varphi \wedge \varphi) \mid (\varphi \vee \varphi) \mid (\varphi \rightarrow \varphi) \mid (\varphi \leftrightarrow \varphi)$$

Give a recursive definition of the number of occurrences of connectives in a propositional formula.

Answer: Here is the definition: $C(p) = 0$, $C(\neg\varphi) = C(\varphi)+1$, $C(\varphi_1\wedge\varphi_2) = C(\varphi_1\vee\varphi_2) = C(\varphi_1 \rightarrow \varphi_2) = C(\varphi_1 \leftrightarrow \varphi_2) = C(\varphi_1) + C(\varphi_2) + 1$.

Question 11 Give a recursive definition of the number of occurrences of atoms (atomic subformulas) in a formula of propositional logic.

Answer: $A(p) = 1$, $A(\neg\varphi) = A(\varphi)$, $A(\varphi_1 \wedge \varphi_2) = A(\varphi_1 \vee \varphi_2) = A(\varphi_1 \rightarrow \varphi_2) = A(\varphi_1 \leftrightarrow \varphi_2) = A(\varphi_1) + A(\varphi_2)$.

Question 12 Let $S(\varphi)$ be the number of occurrences of subformulas of a propositional formula φ, let $A(\varphi)$ be the number of atoms of φ, and let $C(\varphi)$ be the number of connectives of φ. Prove by structural induction:

$$S(\varphi) = A(\varphi) + C(\varphi), \text{ for all propositional formulas } \varphi.$$

Answer: Proof by structural induction.
Base case: if φ is an atom p, then $S(p) = 1 = 1+0 = A(p)+C(p)$, so the property holds. Induction step. Suppose the property holds for φ_1 and φ_2. We have to show that it also holds for $\neg\varphi$ and for $\varphi_1 \odot \varphi_2$, where $\odot$ is one of $\wedge, \vee, \rightarrow, \leftrightarrow$.

First the case of $\neg\varphi_1$. We have $S(\neg\varphi_1) = S(\varphi_1) + 1$, $A(\neg\varphi_1) = A(\varphi_1)$, and $C(\neg\varphi_1) = C(\varphi_1) + 1$, and the required equality $S(\neg\varphi_1) = A(\neg\varphi_1) + C(\neg\varphi_1)$ follows from these equalities plus the induction hypothesis.

Next the case of $\varphi_1 \odot \varphi_2$ (we can lump all binary connectives together, as the definitions of S, A, C coincide for all of them). We have: $S(\varphi_1 \odot \varphi_2) = S(\varphi_1) + S(\varphi_2) + 1$, $A(\varphi_1 \odot \varphi_2) = A(\varphi_1) + A(\varphi_2)$, and $C(\varphi_1 \odot \varphi_2) = C(\varphi_1) + C(\varphi_2) = 1$. The required equality $S(\varphi_1 \odot \varphi_2) = A(\varphi_1 \odot \varphi_2) + C(\varphi_1 \odot \varphi_2)$ follows from these equalities plus the induction hypothesis.

Consider the following definition of binary trees:

```
data Btree a = Leaf a | Node (Btree a) (Btree a) deriving (Eq,Show)
```

The *depth* of a binary tree is given by:

1. $\text{depth}(i) = 0$

2. $\text{depth}(\underset{B_1\ B_2}{\overset{\bullet}{\wedge}}) = \max(\text{depth}(B_1), \text{depth}(B_2)) + 1.$

In Haskell:

```
depth :: Btree a -> Int
depth (Leaf _) = 0
depth (Node t1 t2) = max (depth t1) (depth t2) + 1
```

Question 13 What is the *minimum* number of internal nodes (non leaf nodes) that can occur in a binary tree of depth n? First guess, by looking at examples. Next prove your guess by induction.

Answer: A reasonable guess is: a tree of depth n must have at least n internal nodes.

Proof of this guess by induction: a tree of depth 0 is a single leaf, so it has no internal nodes. Suppose a tree of depth n has at least n nodes. A minimal tree of depth $n+1$ can be constructed from a minimal tree of depth n by replacing a deepest leaf by an internal node with two leafs. This gives $n+1$ internal nodes.

Question 14 What is the *minimum* number of leaf nodes that can occur in a binary tree of depth n? First guess, by looking at examples. Next prove your guess by induction.

Answer: A reasonable guess is: a tree of depth n must have at least $n+1$ leaf nodes. Surely, a tree of depth 0 is a single leaf, so it has 1 leaf node. If a tree of depth n has at least $n+1$ leaf nodes, then a tree of depth $n+1$ has to have at least $n+2$ leaf nodes: replace a deepest leaf by an internal node with two leafs. This removes one leaf and adds two, giving $n+2$ leaf nodes altogether.

Question 15 What is the *maximum* number of internal nodes (non leaf nodes) that can occur in a binary tree of depth n. First guess, by looking at examples. Next prove your guess by induction.

Answer: Draw some example trees and notice the following: to get the maximum number of nodes, the tree needs to be balanced. A balanced tree of depth 0 has 0 internal nodes. A balanced tree of depth 1 has 1 internal node. A balanced tree of depth 2 has 3 internal nodes. A balanced tree of depth 3 has 7 internal nodes. A balanced tree of depth 4 has 15 internal nodes. This suggests that a balanced tree of depth n has $2^n - 1$ internal nodes. Next, we prove by induction that this form is correct.

Base case: if $n = 0$ then the form gives $2^0 - 1 = 0$ internal nodes. This is indeed the number of internal nodes of a tree of depth 0.
Induction step: assume that a tree of depth n has $2^n - 1$ internal nodes. We have to show that a tree of depth $n + 1$ has $2^{n+1} - 1$ internal nodes.
To construct a balanced tree of depth $n + 1$ from one of depth n, just replace each leaf node of the old tree by a new branch with two leaf nodes. This means that the number of internal nodes that gets added equals the number of leaf nodes of a tree of dept n, which, as we have seen in the answer to the previous question, equals 2^n.

This gives: the number of internal nodes of a tree of depth $n + 1$ equals the number of internal nodes of a tree of depth n plus 2^n. By the induction hypothesis we know that a tree of depth n has $2^n - 1$ internal nodes. The number of internal nodes of a tree of depth $n + 1$ is therefore

$$2^n - 1 + 2^n = 2 \times 2^n - 1 = 2^{n+1} - 1.$$

Done.

Question 16 What is the *maximum* number of leaf nodes that can occur in a binary tree of depth n. First guess, by looking at examples. Next prove your guess by induction.

Answer: Look at the example trees again and notice that a largest tree of depth n is in fact a balanced tree of depth n. Inspection of some balanced trees yields: A balanced tree of depth 0 has 1 leaf node. A balanced tree of depth 1 has 2 leaf nodes. A balanced tree of depth 2 has 4 leaf nodes. A balanced tree of depth 3 has 8 leaf nodes. This suggests that a balanced tree of depth n has 2^n leaf nodes. Next, we prove by induction that this form is correct.

Base case: if $n = 0$ then the form gives $2^0 = 1$ leaf nodes. This is indeed the number of leaf nodes of a tree of depth 0.
Induction step: assume that a tree of depth n has 2^n leaf nodes. We have to show that a tree of depth $n + 1$ has 2^{n+1} leaf nodes.
To construct a balanced tree of depth $n + 1$ from one of depth n, just replace each leaf node of the old tree by a new branch with two leaf nodes. This doubles the number of

leaf nodes. Using the induction hypthesis, we see that the new number of leaf nodes is $2 \times 2^n = 2^{n+1}$. This proves the induction step. Done.

Question 17 Consider the following program for merging two lists:

```
merge :: Ord a => [a] -> [a] -> [a]
merge xs [] = xs
merge [] ys = ys
merge (x:xs) (y:ys) = if x <= y
                      then x : merge xs (y:ys)
                      else y : merge (x:xs) ys
```

Show with induction that if `xs` and `ys` are finite and sorted (ordered), then `merge xs ys` is sorted.

Answer: Structural induction on pairs $(\mathrm{xs}, \mathrm{ys})$, following the pattern of the program. Two base cases:

If `xs` is ordered and `ys= []`, the result is `xs`, which is ordered.

If `xs=[]` is ordered and `ys` is ordered, the result is `ys`, which is ordered.

Assume that `(x:xs)` and `(y:ys)` are ordered and that `x <= y`. Then the merge of `xs` and `(y:ys)` is ordered by induction hypothesis, and since `(x:xs)` is ordered and `x <= y`, `x` is less than or equal to the first member of the merge of `xs` and `(y:ys)`. So putting `x` in front of this merge creates an ordered list.

Then case where `x > y` is similar.