Haskell Exercises 10: Proofs by Induction

Antoni Diller
26 July 2011

1 Mathematical Induction Proofs

(1) Prove, using mathematical induction, that, for all \( n \geq 1 \),
\[
\sum_{i=0}^{n-1} 2i + 1 = n^2.
\]

(2) Prove, using mathematical induction, that, for all \( n \geq 1 \),
\[
\sum_{i=0}^{n-1} 2^i = 2^n - 1.
\]

(3) Prove, using mathematical induction, that, for all \( n \geq 1 \),
\[
\sum_{i=0}^{n-1} 3^i = \frac{3^n - 1}{2}.
\]

(4) Prove, using mathematical induction, that, for all \( n \geq 1 \),
\[
\sum_{i=0}^{n-1} i^2 = \frac{n(n + 1)(2n + 1)}{6}.
\]

(5) Prove, using mathematical induction, that, for all \( n \geq 1 \),
\[
\sum_{i=0}^{n-1} i \times 2^i = (n - 1) \times 2^{n+1} + 2.
\]

(6) Prove, using mathematical induction, that, for all \( n \geq 3 \), \( 2n + 1 < 2^n \).
2 Structural Induction Proofs

In the following proofs use the following definitions:

\[ (++): [a] \to [a] \to [a] \]
\[\[] \; ++ \; ys \; = \; ys\]
\[(x:xs) \; ++ \; ys \; = \; x \; : \; (xs \; ++ \; ys)\]

\[\text{concat} \; : \; [[[a]]] \to [a] \]
\[\text{concat} \; [] \; = \; []\]
\[\text{concat} \; (xs:xss) \; = \; xs \; ++ \; \text{concat} \; xss\]

\[\text{reverse} \; : \; [a] \to [a] \]
\[\text{reverse} \; [] \; = \; []\]
\[\text{reverse} \; (x:xs) \; = \; \text{reverse} \; xs \; ++ \; [x]\]

\[\text{map} \; : \; (a \to b) \to [a] \to [b] \]
\[\text{map} \; f \; [] \; = \; []\]
\[\text{map} \; f \; (x:xs) \; = \; f \; x \; : \; \text{map} \; f \; xs\]

\[\; : \; \; (b \to c) \to (a \to b) \to a \to c\]
\[(f.g) \; x \; = \; f \; (g \; x)\]

\[\text{filter} \; : \; (a \to \text{Bool}) \to [a] \to [a] \]
\[\text{filter} \; p \; [] \; = \; []\]
\[\text{filter} \; p \; (x:xs)\]
\[\; | \; p \; x \; = \; x \; : \; \text{filter} \; p \; xs\]
\[\; | \; \text{otherwise} \; = \; \text{filter} \; p \; xs\]

\[\text{foldr} \; : \; (a \to b \to b) \to b \to [a] \to b\]
\[\text{foldr} \; f \; e \; [] \; = \; e\]
\[\text{foldr} \; f \; e \; (x:xs) \; = \; f \; x \; (\text{foldr} \; f \; e \; xs)\]

\[\text{foldl} \; : \; (b \to a \to b) \to b \to [a] \to b\]
\[\text{foldl} \; f \; e \; [] \; = \; e\]
\[\text{foldl} \; f \; e \; (x:xs) \; = \; \text{foldl} \; f \; (f \; e \; x) \; xs\]

\[\text{flip} \; : \; (b \to a \to c) \to a \to b \to c\]
\[\text{flip} \; f \; x \; y \; = \; f \; y \; x\]
(7) Prove, using structural induction, that
\[ xs ++ [] = xs \]

(8) Prove, using structural induction, that
\[ \text{concat} (xss ++ yss) = \text{concat} xss ++ \text{concat} yss \]

(9) Prove, using structural induction, that
\[ \text{reverse} (xs ++ ys) = \text{reverse} ys ++ \text{reverse} xs \]

(10) Prove, using structural induction, that
\[ \text{map} (f.g) = \text{map} f . \text{map} g \]

(11) Prove, using structural induction, that
\[ \text{filter} p . \text{concat} = \text{concat} . \text{map} (\text{filter} p) \]

(12) Prove, using structural induction, that
\[ \text{foldr} f e xs = \text{foldl} (\text{flip} f) e (\text{reverse} xs) \]

(13) Prove, using structural induction, that
\[ \text{foldr} f e . \text{concat} = \text{foldr} (\text{flip} (\text{foldr} f)) e \]

(14) Prove, using structural induction, that
\[ \text{foldr} f e . \text{concat} = \text{foldr} f a . \text{map} (\text{foldr} f e) \]