Fun With Functions

Tobias Nipkow

May 28, 2015

Abstract

This is a collection of cute puzzles of the form “Show that if a function satisfies the following constraints, it must be . . .” Please add further examples to this collection!

Apart from the one about factorial, they all come from the delightful booklet by Terence Tao [1] but go back to Math Olympiads and similar events.

Please add further examples of this kind, either directly or by sending them to me. Let us make this a growing body of fun!

theory FunWithFunctions imports Complex-Main begin
declare implies-True-equals[simp] False-implies-equals[simp]

See [1]. Was first brought to our attention by Herbert Ehler who provided a similar proof.

theorem identity1: fixes f :: nat ⇒ nat
assumes fff: \( \forall n. f(f(n)) < f(Suc(n)) \)
shows f(n) = n
⟨proof⟩

See [1]. Possible extension: Should also hold if the range of f is the reals!

lemma identity2: fixes f :: nat ⇒ nat
assumes f(k) = k and k ≥ 2
and f-times: \( \forall m n. f(m*n) = f(m)*f(n) \)
and f-mono: \( \forall m n. m<n \Rightarrow f m < f n \)
shows f(n) = n
⟨proof⟩

One more from Tao’s booklet. If f is also assumed to be continuous, \( f x = x + 1 \) holds for all reals, not only rationals. Extend the proof!

theorem plus1:
fixes f :: real ⇒ real
assumes 0: \( f 0 = 1 \) and f-add: \( \forall x y. f(x+y+1) = f x + f y \)
assumes r : Q shows f(r) = r + 1

1
The only total model of a naive recursion equation of factorial on integers is 0 for all negative arguments. Probably folklore.

\begin{verbatim}
theorem ifac-neg0: fixes ifac :: int ⇒ int
assumes ifac-rec: \(\forall i. \text{ifac} \ i = (\text{if } \ i = 0 \ \text{then } 1 \ \text{else } i*\text{ifac}(i - 1))\)
shows \(i < 0 \implies \text{ifac} \ i = 0\)
\end{verbatim}

References