Leanduction from scratch?

Cheni Yang, Nicolas Arnold

26.07.2023

Table of Contents

Induction

In type theory, each inductive type can be assigned to an related induction principle.

- The natural numbers
- The binary tree
- A context-free grammar

Induction Principle

The principle for N is easy to guess, but what is the principle for Tree?
The generation of induction principles is computable! This leads us to the tactic "induction".

Induction Tactic

The induction tactic only do two things:

- apply the principle on certain variables
- split the context into more cases

We can write our own induction tactic with some programming skills(OCaml).

Proof without Induction

Given an easy example: $\forall n : \mathbb{N}, n + 1 > n$.

A long Journey of Treeduction

Given following proposition: $\forall t : \text{Tree}, |t| \leq 2^{H(t)+1}$

Combinators

Combinators combine tactics and cases together to save the redundant codes.

- all_goals and any_goals
- repeat
- try

Higher Tactics

Some user-defined tactics(especially in Mathlib) help us solve some trivial goals easily.

- ring
- omega
- split_ifs

Table of Contents

Types of Equality

- In Martin-Löf type theory $\equiv$
- Leibniz equality $\doteq$
- In Homotopy type theory $\cong$

$\doteq \simeq \equiv$

The Martin-Löf equality is equivalent to the Leibniz equality, which should be hold in every type system!
$\forall (A : \text{Type})(x\ y : A), x \doteq y \iff x \equiv y.$

The Reverse of Leibniz?

The function extensionality can be proven on the quotient of setoid of functions. It also works when we set it just as a new axiom of the system.

$\forall (A : \text{Type})(B : A \to \text{Type})(f_1\ f_2 : \Pi x : A, Bx), \forall (x : A), f_1(x) = f_2(x) \implies f_1 = f_2.$

Table of Contents

References

Abel et al.
Leibniz equality is isomorphic to Martin-Löf identity, parametrically.

LeanCommunity
Tutorial: tactic writing in Lean.

J, Avigad.
A proof of function extensionality from quotients.