The Coq proof assistant : principles and practice

J.-F. Monin

Université Grenoble Alpes

2016

Lecture 8

Outline

Outline

Properties of constructors

Constructors make distinguishable values

Constructors with different names

Tactic discriminate

Same constructor applied to different arguments

Each constructor is injective

Proof: using appropriate projections

See coq file

Automated using tactic injection

Inversion

```
Inductive even : nat -> Prop :=
  | E0 : even 0
  | E2:  forall n:nat, even n -> even (S (S n)).
```

Problem 1

Given a goal containing an assumption `even 1`, conclude because such an assumption is inconsistent

Problem 2

Given a goal containing an assumption `e : even S (S x)`, get an assumption `even x`, because only `E2 x` can make the type of `e`

Inversion: contents

```
Inductive even : nat -> Prop :=
  | E0 : even 0
  | E2:  forall n:nat, even n -> even (S (S n)).
```

Why this name

The above reasoning looks like a reading of constructors in the opposite way.

Warning

Nothing to do with induction, just case analysis.

But technically more involved than expected

Basically, `destruct` or `case` works well when the conclusion contains occurrences of X, if X is the argument of the hypothesis to be exploited `even` X

Inversion techniques


```
Inductive even : nat -> Prop :=
  | E0 : even 0
  | E2:  forall n:nat, even n -> even (S (S n)).
```

By hand

See example in coq file

Automated

Tactic `inversion` and variants

Outline

Partial functions

... have to be represented either by total functions, or by inductive predicates.

Example

On colors: see coq file

Outline

A small development

Finding the min of a list

See coq file