# The Coq proof assistant : principles and practice

J.-F. Monin

Université Grenoble Alpes

2016

Lecture 8

# Outline

Structural induction

Induction on a inductive predicate

Well-founded induction

# Outline

Structural induction

Induction on a inductive predicate

Well-founded induction

# Structural induction



## A very natural generalisation of induction

### On lists

$$\frac{P\ nil \qquad \forall n \forall l, P\ l \Rightarrow P\ (n :: l)}{\forall l, P\ l}$$

Examples: stuttering list, associativity of append, reverse

### On binary trees

$$\frac{P\ leaf \qquad \forall n \forall t_l t_r, P\ t_l \Rightarrow P\ t_r \Rightarrow P\ (Node\ t_l\ n\ t_r)}{\forall t, P\ t}$$

Examples: number of keys and of leaves, algorithms on binary search trees

# Outline

# Induction on a inductive predicate





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

We expect the following induction principle:

$$\frac{P\ 0 \qquad \forall n, \mathit{even}\ n \Rightarrow P\ n \Rightarrow P\ (S\ (S\ n))}{\forall n, \mathit{even}\ n \Rightarrow P\ n}$$

# Lists of consecutive even numbers



```
Inductive natlist: Set :=
  | E : natlist
  | C :  nat -> natlist -> natlist.
```

$$\frac{P\ E \qquad \forall n \forall l, P\ l \Rightarrow P\,(C\ n\ l)}{\forall l, P\ l}$$

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

$$\frac{P\ E0 \qquad \forall n \forall l, P\ l \Rightarrow P\,(E2\ n\ l)}{\forall l, P\ l}$$

$$\frac{P\ 0\ E0 \qquad \forall n \forall l, P\ n\ l \Rightarrow P\,(S\,(S\ n))\,(E2\ n\ l)}{\forall n l, P\ n\ l}$$

# Lists of consecutive even numbers (cont'd)



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

$$\frac{P\ 0\ E0 \qquad \forall n \forall l, P\ n\ l \Rightarrow P\ (S\ (S\ n))\ (E2\ n\ l)}{\forall n l, P\ n\ l}$$

Take for P a predicate which does not depend on its second argument: $P\ n\ l \stackrel{\text{def}}{=\!=} Q\ n$

$$\frac{Q\ 0 \qquad \forall n\, \forall (l : evl\ n), Q\ n \Rightarrow Q\ (S\ (S\ n))}{\forall n (l : evl\ n), Q\ n}$$

$$\frac{Q\ 0 \qquad \forall n, evl\ n \Rightarrow Q\ n \Rightarrow Q\ (S\ (S\ n))}{\forall n, evl\ n \Rightarrow Q\ n}$$

Now, evl reads just *even*

# Functional interpretation



```
Inductive list : Set :=
  | E : list
  | C :  nat -> list -> list.
```

$$\frac{P\ E \qquad \forall n \forall l, P\ l \Rightarrow P(C\ n\ l)}{\forall l, P\ l}$$

Lists of consecutive even numbers
typed according to the value of the expected next head

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

$$\frac{P\ E0 \qquad \forall n \forall l, P\ l \Rightarrow P(E2\ n\ l)}{\forall l, P\ l}$$

$$\frac{P\ 0\ E0 \qquad \forall n \forall l, P\ n\ l \Rightarrow P\ (S\ (S\ n))\ (E2\ n\ l)}{\forall n l, P\ n\ l}$$

# Booleans and inductively defined predicates



```
Fixpoint evenb (n:nat) : bool :=
  match n with
  | O        => true
  | S O      => false
  | S (S n') => evenb n'
  end.
```

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

```
Theorem even_evenb : ∀ n, even n -> evenb n = true.
```

By induction on the structure of the proof of even n

```
Theorem evenb_even : ∀ n, evenb n = true -> even n.
```

By induction on $n$

# Booleans and inductively defined predicates



```
Theorem even_evenb :
  ∀ n, even n -> evenb n = true.
```

By induction on the structure of the proof of even n
Don't have to bother about odd numbers

```
Theorem evenb_even :
   ∀ n, evenb n = true -> even n.
```

By induction on $n$: need for strengthening and discrimination.

## Inversion

Issue: getting the possible ways of constructing a hypothesis
Easier for evenb than for even, see *even_inversion.v*

This issue cannot be avoided for non-deterministic relations

# Outline

Structural induction

Induction on a inductive predicate

Well-founded induction

# Stronger induction principles



$$\frac{P\ 0 \qquad P\ 1 \qquad \forall n, P\ n\ \wedge P\ (S\ n) \Rightarrow P\ (S\ (S\ n))}{\forall n, P\ n}$$

$$\frac{P\ 0 \qquad \forall n, (\forall m, m \leq n \Rightarrow P\ m) \Rightarrow P\ (S\ n)}{\forall n, P\ n}$$

By (basic) induction on $Q\ n \stackrel{\text{def}}{=\!=} \forall m, m \leq n \Rightarrow P\ m$

Rephrasing

$$\frac{\forall n, (\forall m, m < n \Rightarrow P\ m) \Rightarrow P\ n}{\forall n, P\ n}$$

Well-founded induction on $(nat, <)$

# Well-founded induction



Material:

- $S$: a set, called the domain of the induction
- $R$: a relation on $S$
- $R$ is well-founded (see below)

Then we have the following induction principle:

$$\frac{\forall x, (\forall y, R\ y\ x \Rightarrow P\ y) \Rightarrow P\ x}{\forall x, P\ x}$$

Two definitions on *well-founded* (equivalent in classical logic)

- any decreasing chain eventually stops
- all elements of $S$ are accessible

An element is accessible $\stackrel{\text{def}}{=\!=}$ all its predecessors are accessible

# Well-founded relation



- $R$ is well-founded if
  all elements of $S$ are accessible for $R$

```
Variable A : Type.
Variable R : A -> A -> Prop.

Inductive Acc (x: A) : Prop :=
  Acc_intro : (∀ y:A, R y x -> Acc y) -> Acc x.
```

# Important application



## Theorem of chocolate tablets

### Statement

Let us take a tablet containing $n$ tiles
and cut it into pieces along grooves

How many shots are needed for reducing the tablet into tiles?

### Answer

$n - 1$

It does not depend on successive choices of grooves!

### Proof

By well-founded induction on $(\mathtt{nat}, <)$

# Construction of well-founded relations



E.g. the lexicographic ordering of two well-founded relations is well-founded.