

Review of some basic constructs

Jean-François Monin

2013-08-07



<http://sts.thss.tsinghua.edu.cn/Coqschool2013>



Case analysis and decomposition of a tuple4

Assume `t` a tuple4

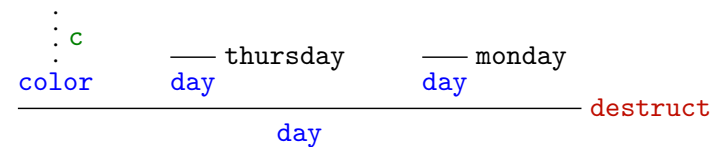
Interactive definition

```
destruct t as [c1 c2 c3 c4 | d1 d2 d3 d4 | x1 x2 x3 x4].
- commands using c1 c2 c3 c4.
- commands using d1 d2 d3 d4.
- commands using x1 x2 x3 x4.
```

Explicit definition

```
match t with
| Mk4color c1 c2 c3 c4 => ... c1 c2 c3 c4 ...
| Mk4day d1 d2 d3 d4 => ... d1 d2 d3 d4 ...
| Mk4t4 x1 x2 x3 x4 => ... x1 x2 x3 x4 ...
end.
```

Case analysis on a color



Interactive and explicit definition in Coq

```
Definition day_of_c: day.
destruct c.
- apply thursday.
- apply monday.
Defined.

Definition day_of_c: day :=
match c with
| white => thursday
| black => monday
end.
```

Explicit definition of functions

```
Definition day_of: forall (c: color), day :=
match c with
| white => thursday
| black => monday
end.
```

Application: by juxtaposition without parenthesis

day_of black

Parentheses can be used for grouping

More functions

```
Definition Set_of : forall (c: color), Set :=  
  fun (c: color) =>  
    match c with  
    | white => color  
    | black => day  
    end.
```

```
Definition funny : forall (c: color), Set_of c :=  
  fun (c: color) =>  
    match c with  
    | white => black  
    | black => wednesday  
    end.
```

Interactive definition of a function

Use `intro`

Special case: arrow

The type of the result does not depend on the argument

$T \rightarrow U$ is a shorthand for

$\forall x : T, U$

Application to logic and proof trees

Universal quantification

Let T be a type and Q a predicate on T

$Q : T \rightarrow \text{Prop}$

A **proof** q of $\forall x : T, Q_x$

is a **function** which maps any value a in T to a proof of Q_a

In order to **use** q , we **apply** it to a , notation: $q\ a$

In order to **prove** q , we can start with **intro** x .

Application to logic and proof trees

Implication

Let P and Q be propositions

P : Prop

Q : Prop

A **proof** r of $P \rightarrow Q$

is a **function** which maps any proof p of P to a proof of Q

In order to **use** r , we **apply** it to p , notation: $r p$

In order to **prove** r , we can start with **intro** hp .

Rules (general)

$$\frac{\frac{\frac{\vdots f}{\forall x : T, U_x} \quad \frac{\vdots a}{T}}{U_a} \text{ apply}}{\frac{[x : T] \quad \vdots u_x}{U_x} \text{ intro } x} \text{ apply}$$

Warning: this x makes sense only in u_x ,
i.e. is available only from $x : T$ to U_x

Products and functions

Consider an environment containing $x : T$ where we define

a term $u_x : U$

But in general, U may depend on x .

Then: consider an environment containing $x : T$ where we define

▶ a type U_x

▶ a term $u_x : U_x$

Then $\text{fun } x \Rightarrow u_x$ is a function defined **for all** x , and returning u_x each time it is **applied** to some argument for x .

$$\text{fun } x : T \Rightarrow u_x : \forall x : T, U_x$$

Application

If $f : \forall x : T, U_x$ and if $a : T$

then f can be applied to a and the type of the result is U_a

When the type of the result does not depend on x

$$\frac{\frac{\frac{\vdots f}{\forall x : T, U} \quad \frac{\vdots a}{T}}{U} \text{ apply}}{\frac{[x : T] \quad \vdots u_x}{U} \text{ intro } x} \text{ apply}$$

Warning: this x makes sense only in u_x ,
i.e. is available only from $x : T$ to U

Other syntax: $T \rightarrow U$ instead of $\forall x : T, U$

$$\frac{\begin{array}{c} \vdots f \\ T \rightarrow U \end{array} \quad \begin{array}{c} \vdots a \\ T \end{array}}{U} \text{ apply}$$

$$\frac{\begin{array}{c} [x : T] \\ \vdots u \\ U \end{array}}{T \rightarrow U} \text{ intro } x$$

Warning: this x makes sense only in u_x ,
i.e. is available only from $x : T$ to U