

# Coq - Fiche

October 26, 2023

THEVENET Louis

## Table des matières

1. Dédution naturelle .....	1
2. Spécificité de Coq .....	1
3. Logique des prédicats .....	2
4. Preuves de programmes fonctionnels .....	2
4.1. Types inductifs .....	2
5. Documentation utile .....	3

## 1. Dédution naturelle

*Exemple* : Exemples utiles

Nom	Tactique	Utilité
$I_{\rightarrow}$	intro H.	
$I_{\forall}$	intro x.	Se débarrasser du $\forall$ , $x$ devient une hypothèse, marche aussi pour les $\rightarrow$
	intros.	Fait des intro. jusqu'à ne plus pouvoir, à faire au début d'une preuve
$E_{\forall}$	generalize y.	généraliser une formule
$I_{=}$	reflexivity.	Axiome égalité
$E_{=}$	rewrite $\rightarrow$ H	Si on a $a = b$ en hypothèse, on peut remplacer $a$ par $b$

## 2. Spécificité de Coq

**Définition 2.1**: Coq permet de travailler à la fois sur les conclusions (ce qu'on prouve) et sur les hypothèses. Par exemple on *casse* l'hypothèse :

*Exemple* :

`destruct H as (Hpsi, Hpsi)` permet de réaliser

$$\frac{\Gamma, \text{Hpsi} : \varphi \vdash \chi, \text{Hpsi} : \psi \vdash \chi}{\Gamma, H : \varphi \wedge \psi \vdash \chi}$$

`destruct H as [Hpsi | Hpsi]` permet de réaliser une disjonction de cas, on obtient deux choses à prouver.

$$\frac{\Gamma, \text{Hpsi} : \varphi \vdash \chi, \text{Hpsi} : \psi \vdash \chi}{\Gamma, H : \varphi \vee \psi \vdash \chi}$$

*Exemple :* `cut (φ)`

Pour prouver  $\psi$ , on peut montrer  $\varphi \wedge \varphi \rightarrow \psi$

$$\frac{\Gamma \vdash \varphi \rightarrow \psi \quad \Gamma \vdash \varphi}{\Gamma \vdash \psi} (\text{cut } (\varphi))$$

### 3. Logique des prédicats

Ici on commence à quantifier et à utiliser des règles comme `generalize x`, voir `coq_formulaire.pdf`

**Définition 3.1:** `pose proof H as (x_1, H1)`

Si l'hypothèse  $H$  est du type  $\exists x_1 : A, \varphi$ , on applique cette hypothèse et on obtient  $x_1 : A$  et  $H1 : \varphi$  en hypothèses

Si  $H$  est du type  $\forall x : A, \exists y : A, \varphi$  :

- `pose proof H as (y, H1)` va prendre un  $x$  qui n'existe pas forcément
- `pose proof (H x) as (y, H1)` on a donné le  $x$

**Définition 3.2:** `exists x1`

Si on veut prouver un `exists x : A, P x y`, et qu'on a un `x1`, on peut l'exhiber pour n'avoir plus qu'à prouver `P x1 y`. On dit à Coq « voilà le  $x$  que tu veux »

**Définition 3.3:** `apply H` ou `apply (H x y)`

On applique un lemme avec ou sans hypothèses. Par exemple, si on a une équivalence, on peut passer de d'un côté à l'autre en fournissant si besoin les variables à utiliser

### 4. Preuves de programmes fonctionnels

**Définition 4.1:** `rewrite H`

Applique une hypothèse de la forme  $G = D$  en remplaçant les termes, de gauche à droite (`rewrite <- H` pour l'autre sens)

#### 4.1. Types inductifs

**Définition 4.1.1:** induction  $x$  démarre une preuve par induction sur  $x$

## 5. Documentation utile

- <https://le.qun.ch/en/blog/coq/>