

Logique dans Coq

1 Utilisation de Coq

Sur turing, Coq peut être utilisé directement en ligne de commande (`coqtop`). L'exécutable `/usr/bin/coqtop` permet de lancer la boucle d'interprétation, `/usr/bin/coqc` de compiler des théories. Néanmoins, on privilégiera l'utilisation de l'interface graphique `coqide` plus conviviale.

La documentation complète du système est disponible en ligne <http://coq.inria.fr>, notamment le manuel de référence (en anglais).

2 Démonstrations en logique intuitionniste

On reprend les exemples vus en cours. Le tableau ci-dessous rappelle la correspondance entre les règles d'introduction et d'élimination et les tactiques de Coq. On rappelle que $\perp \equiv \mathbf{False}$ et $\neg A \equiv A \rightarrow \perp$.

	règle	tactique	règle	tactique
\rightarrow	$\frac{\Gamma, H : A \rightarrow B \vdash A}{\Gamma, H : A \rightarrow B \vdash B}$	<code>apply H</code>	$\frac{\Gamma, A \vdash B}{\Gamma \vdash A \rightarrow B}$	<code>intros</code>
\rightarrow	$\frac{\Gamma \vdash A \quad \Gamma, A \vdash P}{\Gamma \vdash P}$	<code>assert A</code>		
\wedge	$\frac{\Gamma, A, B \vdash P}{\Gamma, A \wedge B \vdash P}$	<code>destruct H</code>	$\frac{\Gamma \vdash A \quad \Gamma \vdash B}{\Gamma \vdash A \wedge B}$	<code>split</code>
\vee	$\frac{\Gamma, A \vdash P \quad \Gamma, B \vdash P}{\Gamma, H : A \vee B \vdash P}$	<code>destruct H</code>	$\frac{\Gamma \vdash A}{\Gamma \vdash A \vee B} \quad \frac{\Gamma \vdash B}{\Gamma \vdash A \vee B}$	<code>left ou right</code>
\perp	$\frac{\Gamma \vdash \perp}{\Gamma \vdash A}$	<code>exfalso</code>		
\neg	$\frac{\Gamma \vdash A \quad \Gamma \vdash \neg A}{\Gamma \vdash \perp}$	<code>absurd A</code>	$\frac{\Gamma, A \vdash \perp}{\Gamma \vdash \neg A}$	<code>intro</code>
\exists	$\frac{\Gamma, x : A, P x \vdash Q}{\Gamma, H : (\exists x : A, P x) \vdash Q}$	<code>destruct H</code>	$\frac{\Gamma, v : A \vdash P v}{\Gamma, v : A \vdash \exists x : A, P x}$	<code>exists v</code>
\forall	$\frac{\Gamma \vdash P t}{\Gamma, H : (\forall x : A, P x \rightarrow Q x) \vdash Q t}$	<code>apply H</code>	$\frac{\Gamma, x : A \vdash P x}{\Gamma \vdash \forall x : A, P x}$	<code>intros</code>
	$\frac{}{\Gamma \vdash A} A \in \Gamma$	<code>assumption</code>		

1- Démontrer en Coq les énoncés suivants :

- 10: $\forall A B C : \mathbf{Prop}, ((A \rightarrow B) \wedge (B \rightarrow C)) \rightarrow (A \rightarrow C)$
- 11: $\forall A B : \mathbf{Prop}, A \vee B \rightarrow B \vee A$
- 12: $\forall A B C : \mathbf{Prop}, ((A \wedge B) \rightarrow C) \rightarrow A \rightarrow B \rightarrow C$
- 13: $\forall A B C : \mathbf{Prop}, (A \rightarrow B \rightarrow C) \rightarrow (A \wedge B) \rightarrow C$
- 14: $\forall A B C : \mathbf{Prop}, (A \wedge (B \vee C)) \rightarrow ((A \wedge B) \vee (A \wedge C))$
- 15: $\forall A B C : \mathbf{Prop}, ((A \wedge B) \vee (A \wedge C)) \rightarrow (A \wedge (B \vee C))$

Ne pas oublier de faire *valider* la démonstration par le système avec la commande `Qed`.

2- Observer les termes de preuve construits par le système avec la commande `Print` suivi du nom d'un théorème, par exemple `Print 11`. On peut également observer les étapes de construction du terme de preuve grâce à la commande `Show Proof`. à n'importe quel moment de la démonstration.

3- Ajouter l'axiome du tiers-exclu de la logique classique par la commande suivante :

```
Axiom tiers_exclu : forall A:Prop, A \/ ~A.
```

Démontrer les propriétés suivantes en logique classique :

— Contraposée : $\forall A B : \text{Prop}, (A \rightarrow B) \longleftrightarrow (\neg B \rightarrow \neg A)$

— Double-négation : $\forall A : \text{Prop}, \neg(\neg A) \longleftrightarrow A$

— Loi de Pierce : $\forall A B : \text{Prop}, ((A \rightarrow B) \rightarrow A) \rightarrow A$

4- Démontrer dans Coq l'énoncé suivant :

```
forall (A : Set) (B : A -> A -> Prop),  
(exists y : A, (forall x : A, B x y)) -> (forall x : A, exists y : A, B x y).
```

Essayer de démontrer sa réciproque et expliquer pourquoi cela est impossible.