

# Initiation à la recherche

The Curry Howard Correspondence:  
A Gentle Introduction

---

Jolan PHILIPPE

24 Novembre 2025

Université d'Orléans

L'assistant de preuve Rocq: Overview

Un peu de Correspondance Curry-Howard

Rocq en Pratique

## ACM SIGPLAN Software Award 2013

L'assistant de preuve Rocq fournit un environnement riche pour le développement interactif de raisonnements formels vérifiés par machine. Rocq a un impact profond sur la recherche en langages de programmation et en systèmes [...] Il a été largement adopté comme outil de recherche par la communauté travaillant sur les langages de programmation [...] Enfin, et ce n'est pas le moindre, ces succès ont contribué à susciter un large intérêt pour la théorie des types dépendants, la logique de base, richement expressive, sur laquelle Rocq est fondé.



## ACM SIGPLAN Software Award 2013

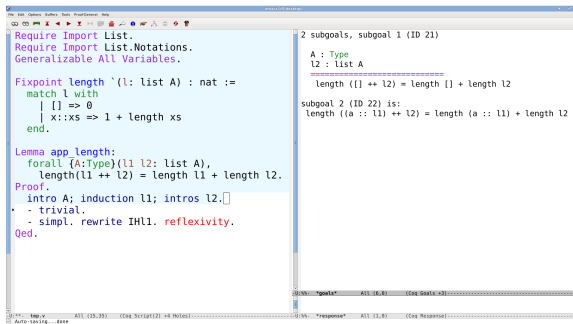
[...] L'équipe de Coq continue de développer le système, apportant à chaque nouvelle version des améliorations significatives en expressivité et en facilité d'utilisation.

En bref, Coq joue un rôle essentiel dans notre transition vers une nouvelle ère de garantie formelle en mathématiques, en sémantique et en vérification de programmes.



# Foundations

- Calcul des constructions inductives
- Correspondance Curry-Howard



```
Require Import List.
Require Import List.Notations.
Generalizable All Variables.

Fixpoint length `(l: list A) : nat :=
  match l with
  | [] => 0
  | x::xs => 1 + length xs
  end.

Lemma app length:
  forall {A:Type}(l1 l2: list A),
    length(l1 ++ l2) = length l1 + length l2.
Proof.
  intro A; induction l1; intros l2.
  - trivial.
  - simpl. rewrite IHl1. reflexivity.
Qed.
```

2 subgoals, subgoal 1 (ID 21)

A : Type  
l2 : list A

length ([] ++ l2) = length [] + length l2

subgoal 2 (ID 22) is:

length ((a :: l1) ++ l2) = length (a :: l1) + length l2

Goal: "goal1" All (6,8) (Coq Goals +3)

Goal: "response" All (1,8) (Coq Response)

L'assistant de preuve Rocq: Overview

Un peu de Correspondance Curry-Howard

Rocq en Pratique

# Notations

## Formules logiques

- $A, B, \dots$ : propositions atomiques
- if  $F_1$  et  $F_2$  sont des formules logiques,  $F_1 \rightarrow F_2$  est une formule logique lue " $F_1$  implique  $F_2$ "

## Programmes

On considère un ensemble  $\mathcal{X}$  de variables

- si  $x \in \mathcal{X}$ ,  $x$  est un programme
- si  $e_1$  et  $e_2$  sont des programmes,  $e_1 \ e_2$  est un programme (application)
- si  $x \in \mathcal{X}$  et  $e$  est un programme,  $\lambda x.e$  est un programme (abstraction de fonction)

En Scheme:  $x, (e_1 \ e_2), \text{lambda}(x)(e)$

En OCaml:  $x, e_1 \ e_2, \text{fun } x \rightarrow e$

## Formules logiques

- $A, B, \dots$ : propositions atomiques
- if  $F_1$  et  $F_2$  sont des formules logiques,  $F_1 \rightarrow F_2$  est une formule logique lue “ $F_1$  implique  $F_2$ ”



# Correspondance de Curry-Howard

## Déduction naturelle

- $A, B$ : formules avec:
  - propositions atomiques
  - $\rightarrow$  (implication)
- $\Gamma$ : ensemble d'hypothèses

$$(v) \frac{A \in \Gamma}{\Gamma \vdash A}$$

$$(i) \frac{\Gamma \cup \{A\} \vdash B}{\Gamma \vdash A \rightarrow B}$$

$$(a) \frac{\Gamma \vdash A \rightarrow B \quad \Gamma \vdash A}{\Gamma \vdash B}$$

## $\lambda$ -Calculus simplement typé

- $A, B$ : types
- $x$ : variables
- $e$ : programmes (variable, abstraction, application)
- $\Gamma$ : ensemble de paires (variable, type)

$$(V) \frac{x : A \in \Gamma}{\Gamma \vdash x : A}$$

$$(L) \frac{\Gamma \cup \{x : A\} \vdash e : B}{\Gamma \vdash (\lambda x : A. e) : A \rightarrow B}$$

$$(A) \frac{\Gamma \vdash e : A \rightarrow B \quad \Gamma \vdash e' : A}{\Gamma \vdash (e \ e') : B}$$

En Python:  $\lambda x:A. e \equiv \text{lambda } x : e$  et  $x$  ont le type  $A$

## Déduction naturelle – Example 1

$$\frac{}{\vdash A \rightarrow B \rightarrow (A \rightarrow C) \rightarrow (B \rightarrow C) \rightarrow C}$$

## Déduction naturelle – Exemple 1

$$(i) \frac{\frac{}{A \vdash B \rightarrow (A \rightarrow C) \rightarrow (B \rightarrow C) \rightarrow C}}{\vdash A \rightarrow B \rightarrow (A \rightarrow C) \rightarrow (B \rightarrow C) \rightarrow C}}$$

## Déduction naturelle – Example 1

$$\begin{array}{c} \frac{}{A, B \vdash (A \rightarrow C) \rightarrow (B \rightarrow C) \rightarrow C} \\ (i) \frac{}{A \vdash B \rightarrow (A \rightarrow C) \rightarrow (B \rightarrow C) \rightarrow C} \\ (i) \frac{}{\vdash A \rightarrow B \rightarrow (A \rightarrow C) \rightarrow (B \rightarrow C) \rightarrow C} \end{array}$$

## Déduction naturelle – Example 1

$$\begin{array}{c} \frac{}{A, B, A \rightarrow C \vdash (B \rightarrow C) \rightarrow C} \\ (i) \frac{}{A, B \vdash (A \rightarrow C) \rightarrow (B \rightarrow C) \rightarrow C} \\ (i) \frac{}{A \vdash B \rightarrow (A \rightarrow C) \rightarrow (B \rightarrow C) \rightarrow C} \\ (i) \frac{}{\vdash A \rightarrow B \rightarrow (A \rightarrow C) \rightarrow (B \rightarrow C) \rightarrow C} \end{array}$$

## Déduction naturelle – Exemple 1

$$\begin{array}{c} \overline{\Gamma \equiv A, B, A \rightarrow C, B \rightarrow C \vdash C} \\ (i) \frac{}{A, B, A \rightarrow C \vdash (B \rightarrow C) \rightarrow C} \\ (i) \frac{}{A, B \vdash (A \rightarrow C) \rightarrow (B \rightarrow C) \rightarrow C} \\ (i) \frac{}{A \vdash B \rightarrow (A \rightarrow C) \rightarrow (B \rightarrow C) \rightarrow C} \\ (i) \frac{}{\vdash A \rightarrow B \rightarrow (A \rightarrow C) \rightarrow (B \rightarrow C) \rightarrow C} \end{array}$$

## Déduction naturelle – Exemple 1

$$\begin{array}{c}
 (a) \frac{\overline{\Gamma \vdash A \rightarrow C} \quad \overline{\Gamma \vdash A}}{\Gamma \equiv A, B, A \rightarrow C, B \rightarrow C \vdash C} \\
 (i) \frac{}{A, B, A \rightarrow C \vdash (B \rightarrow C) \rightarrow C} \\
 (i) \frac{}{A, B \vdash (A \rightarrow C) \rightarrow (B \rightarrow C) \rightarrow C} \\
 (i) \frac{}{A \vdash B \rightarrow (A \rightarrow C) \rightarrow (B \rightarrow C) \rightarrow C} \\
 (i) \frac{}{\vdash A \rightarrow B \rightarrow (A \rightarrow C) \rightarrow (B \rightarrow C) \rightarrow C}
 \end{array}$$

## Déduction naturelle – Exemple 1

$$\begin{array}{c}
 \frac{\frac{(v) \frac{A \rightarrow C \in \Gamma}{\Gamma \vdash A \rightarrow C}}{(a) \frac{}{\Gamma \equiv A, B, A \rightarrow C, B \rightarrow C \vdash C}} \quad \frac{(v) \frac{A \in \Gamma}{\Gamma \vdash A}}{(i) \frac{}{A, B, A \rightarrow C \vdash (B \rightarrow C) \rightarrow C}} \\
 \frac{}{(i) \frac{}{A, B \vdash (A \rightarrow C) \rightarrow (B \rightarrow C) \rightarrow C}} \\
 \frac{}{(i) \frac{}{A \vdash B \rightarrow (A \rightarrow C) \rightarrow (B \rightarrow C) \rightarrow C}} \\
 \frac{}{(i) \frac{}{\vdash A \rightarrow B \rightarrow (A \rightarrow C) \rightarrow (B \rightarrow C) \rightarrow C}}
 \end{array}$$



## Déduction naturelle – Example 2

$$\frac{}{\vdash A \rightarrow B \rightarrow (A \rightarrow C) \rightarrow (B \rightarrow C) \rightarrow C}$$

## Déduction naturelle – Exemple 2

$$(i) \frac{\frac{}{A \vdash B \rightarrow (A \rightarrow C) \rightarrow (B \rightarrow C) \rightarrow C}}{\vdash A \rightarrow B \rightarrow (A \rightarrow C) \rightarrow (B \rightarrow C) \rightarrow C}}$$

## Déduction naturelle – Example 2

$$\begin{array}{c} \frac{}{A, B \vdash (A \rightarrow C) \rightarrow (B \rightarrow C) \rightarrow C} \\ (i) \frac{}{A \vdash B \rightarrow (A \rightarrow C) \rightarrow (B \rightarrow C) \rightarrow C} \\ (i) \frac{}{\vdash A \rightarrow B \rightarrow (A \rightarrow C) \rightarrow (B \rightarrow C) \rightarrow C} \end{array}$$

## Déduction naturelle – Exemple 2

$$\begin{array}{c} \frac{}{A, B, A \rightarrow C \vdash (B \rightarrow C) \rightarrow C} \\ (i) \frac{}{A, B \vdash (A \rightarrow C) \rightarrow (B \rightarrow C) \rightarrow C} \\ (i) \frac{}{A \vdash B \rightarrow (A \rightarrow C) \rightarrow (B \rightarrow C) \rightarrow C} \\ (i) \frac{}{\vdash A \rightarrow B \rightarrow (A \rightarrow C) \rightarrow (B \rightarrow C) \rightarrow C} \end{array}$$

## Déduction naturelle – Exemple 2

$$\begin{array}{c} (i) \frac{\overline{\Gamma \equiv A, B, A \rightarrow C, B \rightarrow C \vdash C}}{A, B, A \rightarrow C \vdash (B \rightarrow C) \rightarrow C} \\ (i) \frac{(i) \frac{A, B \vdash (A \rightarrow C) \rightarrow (B \rightarrow C) \rightarrow C}{A \vdash B \rightarrow (A \rightarrow C) \rightarrow (B \rightarrow C) \rightarrow C}}{(i) \frac{A \vdash B \rightarrow (A \rightarrow C) \rightarrow (B \rightarrow C) \rightarrow C}{\vdash A \rightarrow B \rightarrow (A \rightarrow C) \rightarrow (B \rightarrow C) \rightarrow C}} \end{array}$$

## Déduction naturelle – Exemple 2

$$\begin{array}{c}
 (a) \frac{\overline{\Gamma \vdash B \rightarrow C} \quad \overline{\Gamma \vdash B}}{\Gamma \equiv A, B, A \rightarrow C, B \rightarrow C \vdash C} \\
 (i) \frac{}{A, B, A \rightarrow C \vdash (B \rightarrow C) \rightarrow C} \\
 (i) \frac{}{A, B \vdash (A \rightarrow C) \rightarrow (B \rightarrow C) \rightarrow C} \\
 (i) \frac{}{A \vdash B \rightarrow (A \rightarrow C) \rightarrow (B \rightarrow C) \rightarrow C} \\
 (i) \frac{}{\vdash A \rightarrow B \rightarrow (A \rightarrow C) \rightarrow (B \rightarrow C) \rightarrow C}
 \end{array}$$

## Déduction naturelle – Exemple 2

$$\begin{array}{c}
 (v) \frac{B \rightarrow C \in \Gamma}{\Gamma \vdash B \rightarrow C} \quad (v) \frac{B \in \Gamma}{\Gamma \vdash B} \\
 (a) \frac{}{\Gamma \equiv A, B, A \rightarrow C, B \rightarrow C \vdash C} \\
 (i) \frac{}{A, B, A \rightarrow C \vdash (B \rightarrow C) \rightarrow C} \\
 (i) \frac{}{A, B \vdash (A \rightarrow C) \rightarrow (B \rightarrow C) \rightarrow C} \\
 (i) \frac{}{A \vdash B \rightarrow (A \rightarrow C) \rightarrow (B \rightarrow C) \rightarrow C} \\
 (i) \frac{}{\vdash A \rightarrow B \rightarrow (A \rightarrow C) \rightarrow (B \rightarrow C) \rightarrow C}
 \end{array}$$

# Correspondance de Curry-Howard

$\lambda$ -calcul: trouver un terme du type donné

$\vdash ?$

$: A \rightarrow B \rightarrow (A \rightarrow C) \rightarrow (B \rightarrow C) \rightarrow C$



# Correspondance de Curry-Howard

$\lambda$ -calcul: trouver un terme du type donné

$$(L) \frac{x:A \vdash ?}{\vdash \lambda x:A. ?} : B \rightarrow (A \rightarrow C) \rightarrow (B \rightarrow C) \rightarrow C$$
$$: A \rightarrow B \rightarrow (A \rightarrow C) \rightarrow (B \rightarrow C) \rightarrow C$$

# Correspondance de Curry-Howard

$\lambda$ -calcul: trouver un terme du type donné

$$\begin{array}{c} \text{(L)} \frac{x:A, y:B \vdash ?}{x:A \vdash \lambda y:B. ?} \quad : (A \rightarrow C) \rightarrow (B \rightarrow C) \rightarrow C \\ \text{(L)} \frac{x:A \vdash \lambda y:B. ?}{\vdash \lambda x:A. \lambda y:B. ?} \quad : B \rightarrow (A \rightarrow C) \rightarrow (B \rightarrow C) \rightarrow C \\ \vdash \lambda x:A. \lambda y:B. ? \quad : A \rightarrow B \rightarrow (A \rightarrow C) \rightarrow (B \rightarrow C) \rightarrow C \end{array}$$

# Correspondance de Curry-Howard

$\lambda$ -calcul: trouver un terme du type donné

$$\begin{array}{c} \text{(L)} \frac{\overline{x:A, y:B, f:A \rightarrow C \vdash ?} \quad : (B \rightarrow C) \rightarrow C}{x:A, y:B \vdash \lambda f:A \rightarrow C. ? \quad : (A \rightarrow C) \rightarrow (B \rightarrow C) \rightarrow C} \\ \text{(L)} \frac{x:A \vdash \lambda y:B. \lambda f:A \rightarrow C. ? \quad : B \rightarrow (A \rightarrow C) \rightarrow (B \rightarrow C) \rightarrow C}{\vdash \lambda x:A. \lambda y:B. \lambda f:A \rightarrow C. ? \quad : A \rightarrow B \rightarrow (A \rightarrow C) \rightarrow (B \rightarrow C) \rightarrow C} \end{array}$$

# Correspondance de Curry-Howard

$\lambda$ -calcul: trouver un terme du type donné

$$\begin{array}{c}
 (L) \frac{\overline{\Gamma \equiv x:A, y:B, f:A \rightarrow C, g:B \rightarrow C \vdash ? : C}}{x:A, y:B, f:A \rightarrow C \vdash \lambda g:B \rightarrow C. ? : (B \rightarrow C) \rightarrow C} \\
 (L) \frac{(L) \frac{x:A, y:B \vdash \lambda f:A \rightarrow C. \lambda g:B \rightarrow C. ? : (A \rightarrow C) \rightarrow (B \rightarrow C) \rightarrow C}{x:A \vdash \lambda y:B. \lambda f:A \rightarrow C. \lambda g:B \rightarrow C. ? : B \rightarrow (A \rightarrow C) \rightarrow (B \rightarrow C) \rightarrow C}}{(L) \frac{\vdash \lambda x:A. \lambda y:B. \lambda f:A \rightarrow C. \lambda g:B \rightarrow C. ? : A \rightarrow B \rightarrow (A \rightarrow C) \rightarrow (B \rightarrow C) \rightarrow C}{}
 \end{array}$$

# Correspondance de Curry-Howard

$\lambda$ -calcul: trouver un terme du type donné

$$\begin{array}{c}
 (A) \frac{\overline{\Gamma \vdash f:A \rightarrow C} \quad \overline{\Gamma \vdash x:A}}{\overline{\Gamma \equiv x:A, y:B, f:A \rightarrow C, g:B \rightarrow C \vdash (f\ x) : C}} \\
 (L) \frac{}{x:A, y:B, f:A \rightarrow C \vdash \lambda g:B \rightarrow C. (f\ x) : (B \rightarrow C) \rightarrow C} \\
 (L) \frac{}{x:A, y:B \vdash \lambda f:A \rightarrow C. \lambda g:B \rightarrow C. (f\ x) : (A \rightarrow C) \rightarrow (B \rightarrow C) \rightarrow C} \\
 (L) \frac{}{x:A \vdash \lambda y:B. \lambda f:A \rightarrow C. \lambda g:B \rightarrow C. (f\ x) : B \rightarrow (A \rightarrow C) \rightarrow (B \rightarrow C) \rightarrow C} \\
 (L) \frac{}{\vdash \lambda x:A. \lambda y:B. \lambda f:A \rightarrow C. \lambda g:B \rightarrow C. (f\ x) : A \rightarrow B \rightarrow (A \rightarrow C) \rightarrow (B \rightarrow C) \rightarrow C}
 \end{array}$$

# Correspondance de Curry-Howard

$\lambda$ -calcul: trouver un terme du type donné

$$\begin{array}{c}
 (V) \frac{f:A \rightarrow C \in \Gamma}{\Gamma \vdash f:A \rightarrow C} \quad (V) \frac{x:A \in \Gamma}{\Gamma \vdash x:A} \\
 (A) \frac{}{\Gamma \equiv x:A, y:B, f:A \rightarrow C, g:B \rightarrow C \vdash (f \ x) : C} \\
 (L) \frac{}{x:A, y:B, f:A \rightarrow C \vdash \lambda g:B \rightarrow C. (f \ x) : (B \rightarrow C) \rightarrow C} \\
 (L) \frac{}{x:A, y:B \vdash \lambda f:A \rightarrow C. \lambda g:B \rightarrow C. (f \ x) : (A \rightarrow C) \rightarrow (B \rightarrow C) \rightarrow C} \\
 (L) \frac{}{x:A \vdash \lambda y:B. \lambda f:A \rightarrow C. \lambda g:B \rightarrow C. (f \ x) : B \rightarrow (A \rightarrow C) \rightarrow (B \rightarrow C) \rightarrow C} \\
 (L) \frac{}{\vdash \lambda x:A. \lambda y:B. \lambda f:A \rightarrow C. \lambda g:B \rightarrow C. (f \ x) : A \rightarrow B \rightarrow (A \rightarrow C) \rightarrow (B \rightarrow C) \rightarrow C}
 \end{array}$$

$\lambda x:A. \lambda y:B. \lambda f:A \rightarrow C. \lambda g:B \rightarrow C. (f \ x)$

st une façon d'encoder l'arbre de preuve de

$A \rightarrow B \rightarrow (A \rightarrow C) \rightarrow (B \rightarrow C) \rightarrow C$

Pour toute formule, il existe une preuve de cette formule en déduction naturelle si et seulement s'il existe un  $\lambda$ -terme qui a cette formule comme type.

- Théorème  $\Leftrightarrow$  Type
- Preuve  $\Leftrightarrow$  Programme

# Correspondance de Curry-Howard

Questions :

- Étant donnée une logique, quelles sont les constructions de langage de programmation correspondantes ?
- Étant donnée une construction de langage de programmation, quelle est son interprétation dans le monde logique ?

Vue statique et vue dynamique :

- Le monde de la programmation ne se limite pas au typage, les programmes **s'exécutent**.
- Que signifie l'exécution d'un programme dans le monde logique ?



L'assistant de preuve Rocq: Overview

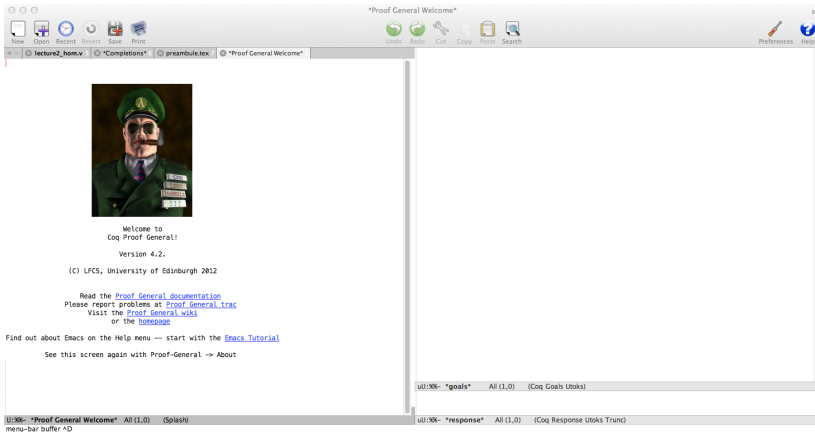
Un peu de Correspondance Curry-Howard

Rocq en Pratique

- Langage de programmation fonctionnel
- Système de types riche : permet d'exprimer des propriétés logiques
- Langage pour construire des preuves (c-à-d des termes de preuve)
- Extraction de programmes

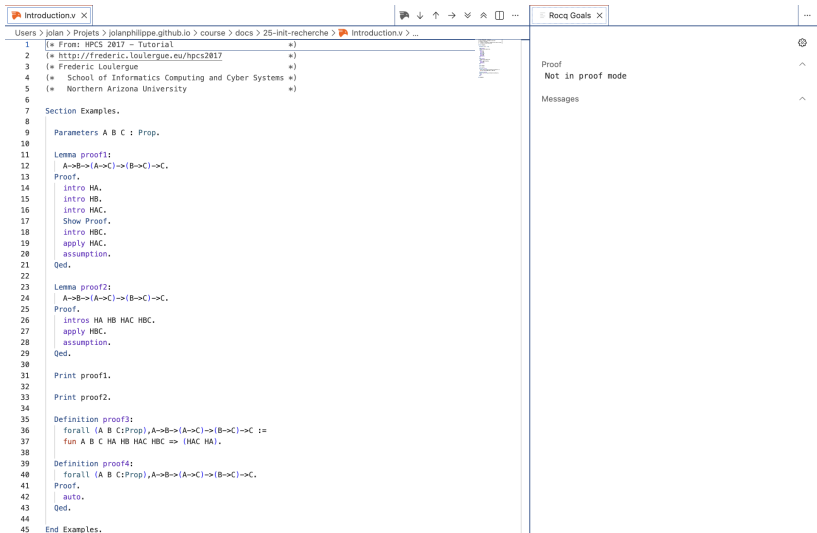
# Exemples précédents dans Rocq

## Le mode Proof General pour Emacs . . .



# Exemples précédents dans Rocq

... ou VsRocq dans Visual Studio Code ...

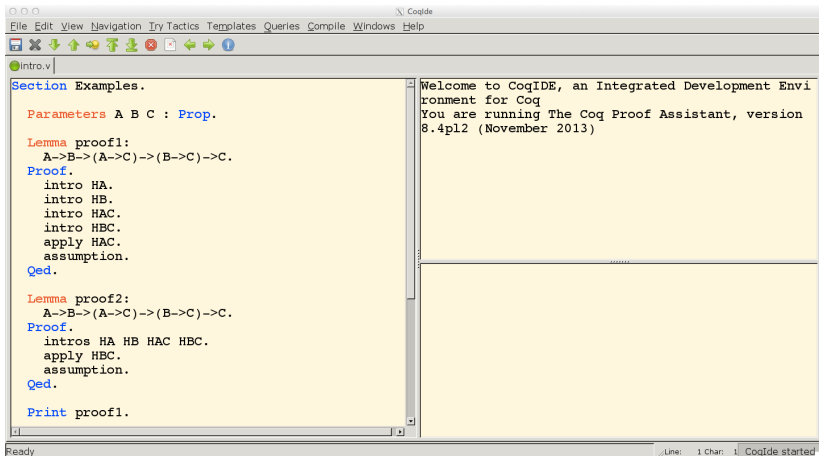


```
1 (* From: HPCS 2017 - Tutorial *)
2 (* http://frederic.loulergue.eu/hpcs2017 *)
3 (* Frederic Loulergue *)
4 (* School of Informatics Computing and Cyber Systems *)
5 (* Northern Arizona University *)
6
7 Section Examples.
8
9 Parameters A B C : Prop.
10
11 Lemma proof1:
12   A->B->(A->C)->(B->C)->C.
13 Proof.
14   intro HA.
15   intro HB.
16   intro HAC.
17   Show Proof.
18   intro HBC.
19   apply HAC.
20   assumption.
21 Qed.
22
23 Lemma proof2:
24   A->B->(A->C)->(B->C)->C.
25 Proof.
26   intros HA HB HAC HBC.
27   apply HBC.
28   assumption.
29 Qed.
30
31 Print proof1.
32
33 Print proof2.
34
35 Definition proof3:
36   forall (A B C:Prop), A->B->(A->C)->(B->C)->C :=
37   fun A B C HA HB HAC HBC => (HAC HA).
38
39 Definition proof4:
40   forall (A B C:Prop), A->B->(A->C)->(B->C)->C.
41 Proof.
42   auto.
43 Qed.
44
45 End Examples.
```

Proof  
Not in proof mode  
Messages

# Exemples précédents dans Rocq

...ou RocqIDE



The screenshot shows the RocqIDE application window. The title bar reads 'CocqIde'. The menu bar includes 'File', 'Edit', 'View', 'Navigation', 'Try Tactics', 'Templates', 'Queries', 'Compile', 'Windows', and 'Help'. Below the menu bar is a toolbar with various icons. The main editor area is split into two panes. The left pane, titled 'intro.v', contains the following Coq code:

```
Section Examples.  
  
Parameters A B C : Prop.  
  
Lemma proof1:  
  A->B->(A->C)->(B->C)->C.  
Proof.  
  intro HA.  
  intro HB.  
  intro HAC.  
  intro HBC.  
  apply HAC.  
  assumption.  
Qed.  
  
Lemma proof2:  
  A->B->(A->C)->(B->C)->C.  
Proof.  
  intros HA HB HAC HBC.  
  apply HBC.  
  assumption.  
Qed.  
  
Print proof1.
```

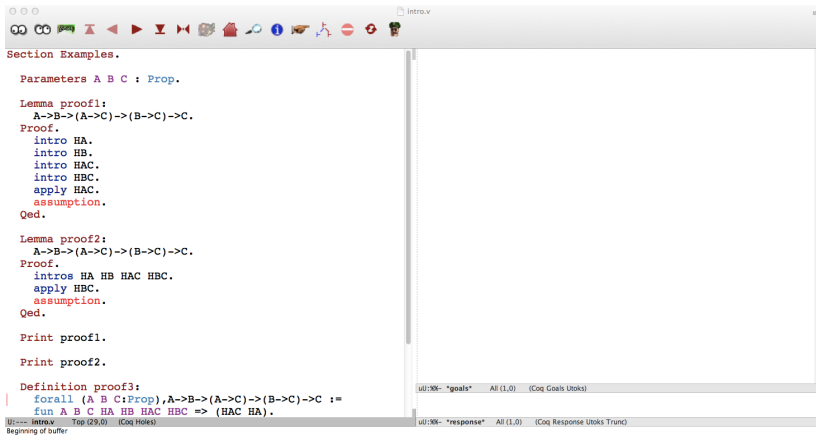
The right pane displays a welcome message:

```
Welcome to CoqIDE, an Integrated Development Environment for Coq  
You are running The Coq Proof Assistant, version 8.4pl2 (November 2013)
```

At the bottom of the window, a status bar shows 'Ready' on the left and 'Line: 1 Char: 1 CocqIde started' on the right.

# Exemples précédents dans Rocq

On ouvre le fichier `Introduction.v`<sup>1</sup>:



```
Section Examples.

Parameters A B C : Prop.

Lemma proof1:
  A->B->(A->C)->(B->C)->C.
Proof.
  intro HA.
  intro HB.
  intro HAC.
  intro HBC.
  apply HAC.
  assumption.
Qed.

Lemma proof2:
  A->B->(A->C)->(B->C)->C.
Proof.
  intros HA HB HAC HBC.
  apply HBC.
  assumption.
Qed.

Print proof1.

Print proof2.

Definition proof3:
  forall (A B C:Prop), A->B->(A->C)->(B->C)->C :=
  fun A B C HA HB HAC HBC => (HAC HA).

|
U:--- intro.v   Top (29,0)   (Coq Holes)
Beginning of buffer
```

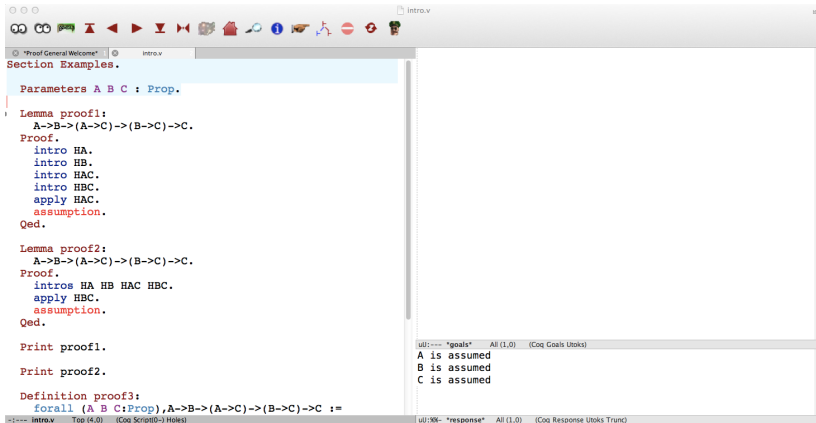
ul:90%~ "goals" All (1,0) (Coq Goals Utkols)

ul:90%~ "response" All (1,0) (Coq Response Utkols Trunc)

<sup>1</sup>disponible sur <https://jolanphilippe.github.io/course/docs/25-init-recherche/Introduction.v>

# Exemples précédents dans Rocq

Nous commençons à alimenter Rocq avec des commandes :



The screenshot shows the Rocq IDE interface. The main editor displays a Coq script with the following content:

```
Section Examples.  
  
Parameters A B C : Prop.  
  
Lemma proof1:  
  A->B->(A->C)->(B->C)->C.  
Proof.  
  intro HA.  
  intro HB.  
  intro HAC.  
  intro HBC.  
  apply HAC.  
  assumption.  
Qed.  
  
Lemma proof2:  
  A->B->(A->C)->(B->C)->C.  
Proof.  
  intros HA HB HAC HBC.  
  apply HBC.  
  assumption.  
Qed.  
  
Print proof1.  
  
Print proof2.  
  
Definition proof3:  
  forall (A B C:Prop), A->B->(A->C)->(B->C)->C :=
```

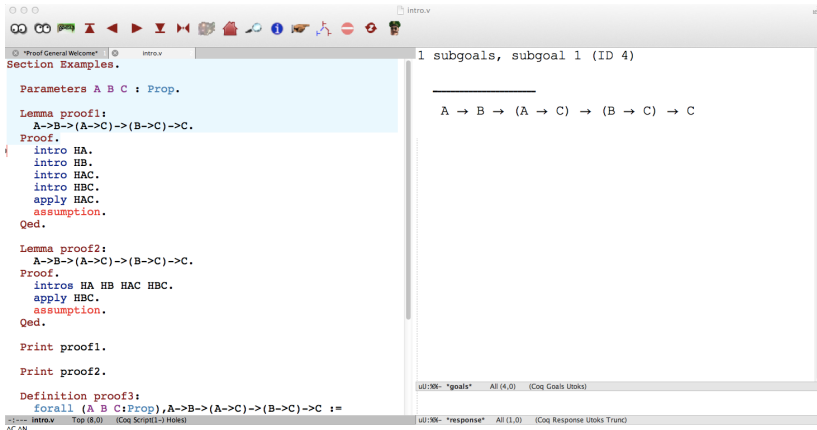
The bottom right panel shows the current goals:

```
ul:--- "goals" All (1,0) (Coq Goals Utoks)  
A is assumed  
B is assumed  
C is assumed
```

The bottom status bar shows the current file is `Intro.v` at line 4, column 0, and the Coq version is 8.10.0.

# Exemples précédents dans Rocq

On énonce un lemme et entrons dans le mode de preuve interactif :



The screenshot shows the Rocq IDE interface. The left pane displays a Coq script with the following content:

```
Section Examples.  
  
Parameters A B C : Prop.  
  
Lemma proof1:  
  A->B->(A->C)->(B->C)->C.  
Proof.  
  intro HA.  
  intro HB.  
  intro HAC.  
  intro HBC.  
  apply HAC.  
  assumption.  
Qed.  
  
Lemma proof2:  
  A->B->(A->C)->(B->C)->C.  
Proof.  
  intros HA HB HAC HBC.  
  apply HBC.  
  assumption.  
Qed.  
  
Print proof1.  
Print proof2.  
  
Definition proof3:  
  forall (A B C:Prop), A->B->(A->C)->(B->C)->C :=
```

The right pane shows the interactive proof state for the first subgoal (ID 4). The goal is:

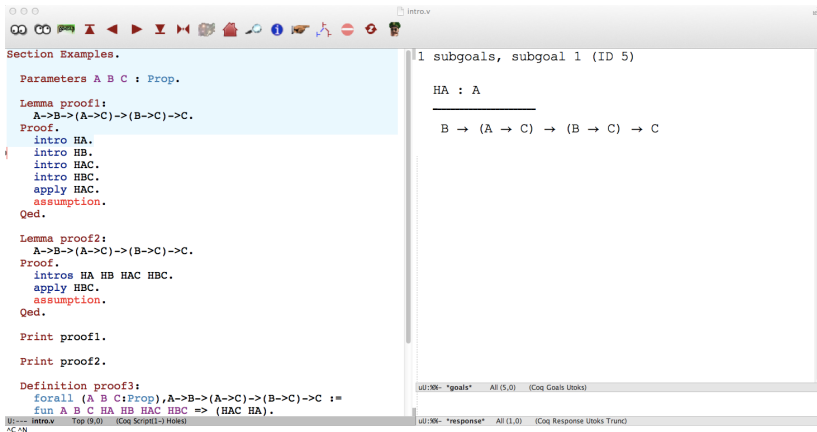
$$A \rightarrow B \rightarrow (A \rightarrow C) \rightarrow (B \rightarrow C) \rightarrow C$$

The bottom status bar indicates the current state: "1 subgoals, subgoal 1 (ID 4)". The bottom-most status bar shows the current file and line numbers: "Intro.v Top (8,0) (Coq Script11-) Holes".



# Exemples précédents dans Rocq

La tactique intro « applique » la règle (i) :



The screenshot shows the Rocq IDE interface. The left pane displays a Coq script with the following content:

```
Section Examples.  
  
Parameters A B C : Prop.  
  
Lemma proof1:  
  A->B->(A->C)->(B->C)->C.  
Proof.  
  intro HA.  
  intro HB.  
  intro HAC.  
  intro HBC.  
  apply HAC.  
  assumption.  
Qed.  
  
Lemma proof2:  
  A->B->(A->C)->(B->C)->C.  
Proof.  
  intros HA HB HAC HBC.  
  apply HBC.  
  assumption.  
Qed.  
  
Print proof1.  
Print proof2.  
  
Definition proof3:  
  forall (A B C:Prop), A->B->(A->C)->(B->C)->C :=  
  fun A B C HA HB HAC HBC => (HAC HA).
```

The right pane shows the proof state for the first subgoal (ID 5):

1 subgoals, subgoal 1 (ID 5)

HA : A

---

$B \rightarrow (A \rightarrow C) \rightarrow (B \rightarrow C) \rightarrow C$

The bottom status bar shows the current position in the script: U:--- intro.v Top (9,0) (Coq Script(1-) Holes) and the response status: ul:%%- "response" All (1,0) (Coq Response Utoks Trunc).

# Exemples précédents dans Rocq

Le contexte est maintenant similaire à  $\Gamma$  :

The screenshot shows the Rocq IDE interface. The left pane displays a Coq script with the following content:

```
Section Examples.  
  
Parameters A B C : Prop.  
  
Lemma proof1:  
  A->B->(A->C)->(B->C)->C.  
Proof.  
  intro HA.  
  intro HB.  
  intro HAC.  
  intro HBC.  
  apply HAC.  
  assumption.  
Qed.  
  
Lemma proof2:  
  A->B->(A->C)->(B->C)->C.  
Proof.  
  intros HA HB HAC HBC.  
  apply HBC.  
  assumption.  
Qed.  
  
Print proof1.  
Print proof2.  
  
Definition proof3:  
  forall (A B C:Prop), A->B->(A->C)->(B->C)->C :=  
  fun A B C HA HB HAC HBC => (HAC HA).
```

The right pane shows the proof state for the first subgoal (ID 8):

```
1 subgoals, subgoal 1 (ID 8)  
  
HA : A  
HB : B  
HAC : A → C  
HBC : B → C  
-----  
C
```

The status bar at the bottom indicates the current goal is "goals" (ID 8,0) and the response is "response" (ID 1,0).

# Exemples précédents dans Rocq

Nous appliquons la règle (a) en nommant la partie implication :

The screenshot shows the Rocq IDE interface. The left pane displays a Coq script with the following content:

```
Section Examples.  
  
Parameters A B C : Prop.  
  
Lemma proof1:  
  A->B->(A->C)->(B->C)->C.  
Proof.  
  intro HA.  
  intro HB.  
  intro HAC.  
  intro HBC.  
  apply HAC.  
  assumption.  
Qed.  
  
Lemma proof2:  
  A->B->(A->C)->(B->C)->C.  
Proof.  
  intros HA HB HAC HBC.  
  apply HBC.  
  assumption.  
Qed.  
  
Print proof1.  
Print proof2.  
  
Definition proof3:  
  forall (A B C:Prop), A->B->(A->C)->(B->C)->C :=  
  fun A B C HA HB HAC HBC => (HAC HA).
```

The right pane shows the current goal state:

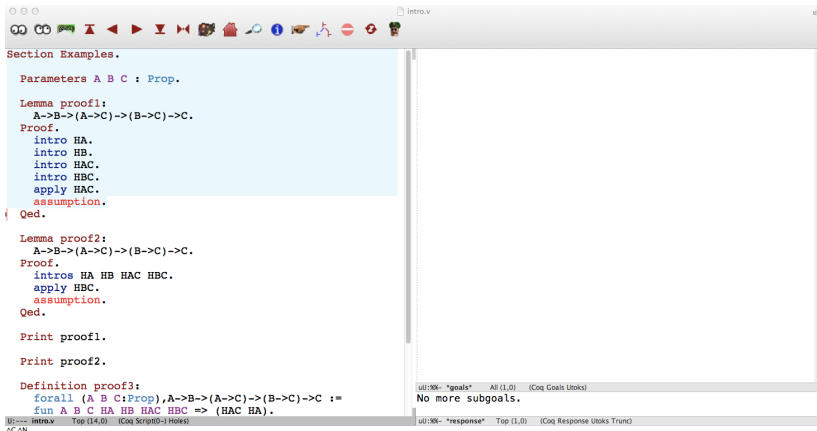
```
1 subgoals, subgoal 1 (ID 9)  
  
HA : A  
HB : B  
HAC : A → C  
HBC : B → C  
-----  
A
```

At the bottom, the status bar indicates the current goal is "goals" (ID 9) and the response is "response" (ID 1).

et nous n'avons donc plus qu'à traiter  $A$  ...

# Exemples précédents dans Rocq

... qui est une hypothèse, nous utilisons la règle ( $\nu$ ) :



```
Section Examples.

Parameters A B C : Prop.

Lemma proof1:
  A->B->(A->C)->(B->C)->C.
Proof.
  intro HA.
  intro HB.
  intro HAC.
  intro HBC.
  apply HAC.
  assumption.
Qed.

Lemma proof2:
  A->B->(A->C)->(B->C)->C.
Proof.
  intros HA HB HAC HBC.
  apply HBC.
  assumption.
Qed.

Print proof1.

Print proof2.

Definition proof3:
  forall (A B C:Prop), A->B->(A->C)->(B->C)->C :=
  fun A B C HA HB HAC HBC => (HAC HA).

[0] intro.v Top (14,0) (Coq Script0-1) Holes
```

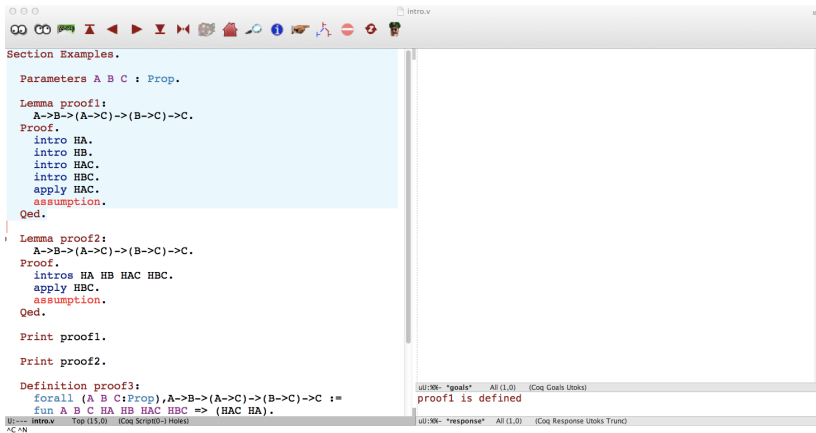
ul:306- \*goals\* All (1,0) (Coq Goals Utkols)  
No more subgoals.

ul:306- \*response\* Top (1,0) (Coq Response Utkols Trunc)

“No more subgoals”  $\equiv$  preuve terminée  $\equiv$   $\lambda$ -terme construit

# Exemples précédents dans Rocq

Qed vérifie le typage du terme par rapport à l'énoncé du lemme :



```
Section Examples.

Parameters A B C : Prop.

Lemma proof1:
  A->B->(A->C)->(B->C)->C.
Proof.
  intro HA.
  intro HB.
  intro HAC.
  intro HBC.
  apply HAC.
  assumption.
Qed.

Lemma proof2:
  A->B->(A->C)->(B->C)->C.
Proof.
  intros HA HB HAC HBC.
  apply HBC.
  assumption.
Qed.

Print proof1.

Print proof2.

Definition proof3:
  forall (A B C:Prop), A->B->(A->C)->(B->C)->C :=
  fun A B C HA HB HAC HBC => (HAC HA).

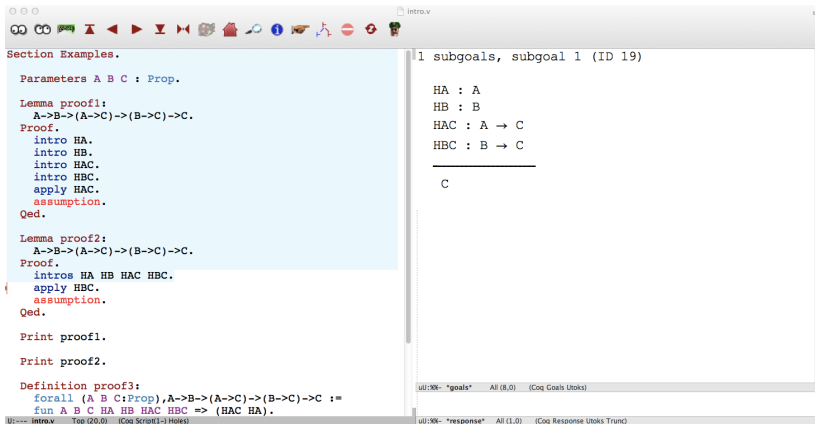
U:--- Intro.v Top (15,0) (Coq Script0-) Holes)
^C ^N
```

ul:30%- "goals" All (1,0) (Coq Goals Utkols)  
proof1 is defined

ul:30%- "response" All (1,0) (Coq Response Utkols Trunc)

# Exemples précédents dans Rocq

Deuxième version, nous faisons plusieurs intro :



The screenshot shows the Rocq IDE interface. The left pane displays a Coq script with three lemmas and a definition. The right pane shows the current goal state after several `intro` commands.

```
Section Examples.  
  
Parameters A B C : Prop.  
  
Lemma proof1:  
  A->B->(A->C)->(B->C)->C.  
Proof.  
  intro HA.  
  intro HB.  
  intro HAC.  
  intro HBC.  
  apply HAC.  
  assumption.  
Qed.  
  
Lemma proof2:  
  A->B->(A->C)->(B->C)->C.  
Proof.  
  intros HA HB HAC HBC.  
  apply HBC.  
  assumption.  
Qed.  
  
Print proof1.  
Print proof2.  
  
Definition proof3:  
  forall (A B C:Prop), A->B->(A->C)->(B->C)->C :=  
  fun A B C HA HB HAC HBC => (HAC HA).  
end
```

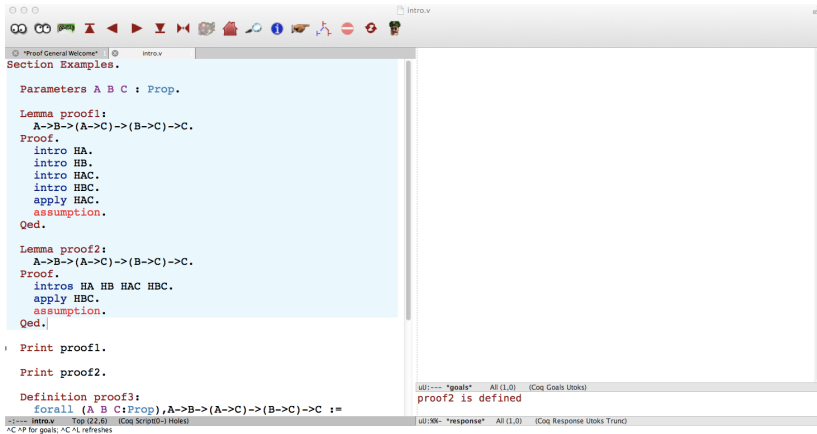
The right pane shows the goal state:

```
1 subgoals, subgoal 1 (ID 19)  
  
HA : A  
HB : B  
HAC : A → C  
HBC : B → C  
-----  
C
```

The status bar at the bottom indicates the current file is `Intro.v` and the current goal is `Top (20,0) (Coq Script1-1) Holes`.

# Exemples précédents dans Rocq

et apply HBC au lieu de apply HAC :



```
Section Examples.

Parameters A B C : Prop.

Lemma proof1:
  A->B->(A->C)->(B->C)->C.
Proof.
  intro HA.
  intro HB.
  intro HAC.
  intro HBC.
  apply HAC.
  assumption.
Qed.

Lemma proof2:
  A->B->(A->C)->(B->C)->C.
Proof.
  intros HA HB HAC HBC.
  apply HBC.
  assumption.
Qed.

Print proof1.

Print proof2.

Definition proof3:
  forall (A B C:Prop), A->B->(A->C)->(B->C)->C :=
```

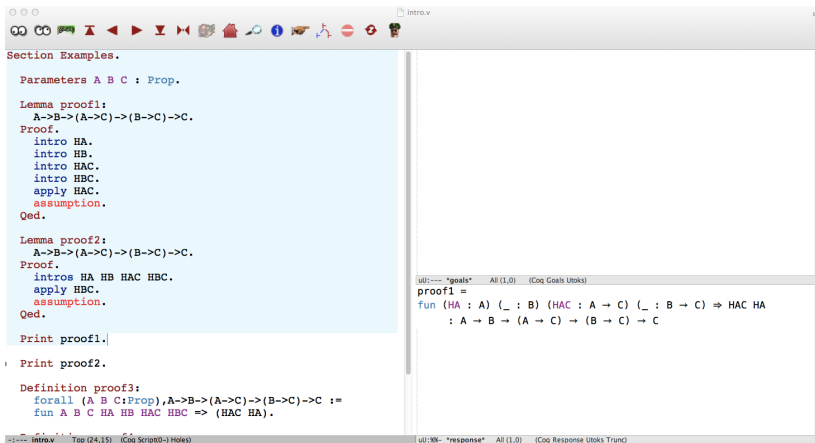
ul:--- "goals" All (1,0) (Coq Goals Utoks)  
proof2 is defined

ul:%%- "response" All (1,0) (Coq Response Utoks Trunc)

intro.v Top (22,6) (Coq Script0-) Holes  
^C ^P for goals; ^C ^L refreshes

# Exemples précédents dans Rocq

Print t. affiche le terme t :



```
Section Examples.  
  
Parameters A B C : Prop.  
  
Lemma proof1:  
  A->B->(A->C)->(B->C)->C.  
Proof.  
  intro HA.  
  intro HB.  
  intro HAC.  
  intro HBC.  
  apply HAC.  
  assumption.  
Qed.  
  
Lemma proof2:  
  A->B->(A->C)->(B->C)->C.  
Proof.  
  intros HA HB HAC HBC.  
  apply HBC.  
  assumption.  
Qed.  
  
Print proof1.  
  
Print proof2.  
  
Definition proof3:  
  forall (A B C:Prop), A->B->(A->C)->(B->C)->C :=  
  fun A B C HA HB HAC HBC => (HAC HA).  
  
Print proof3.
```

ul:--- "goals" All (1,0) (Coq Goals Utoks)  
proof1 =  
fun (HA : A) (\_ : B) (HAC : A → C) (\_ : B → C) => HAC HA  
: A → B → (A → C) → (B → C) → C

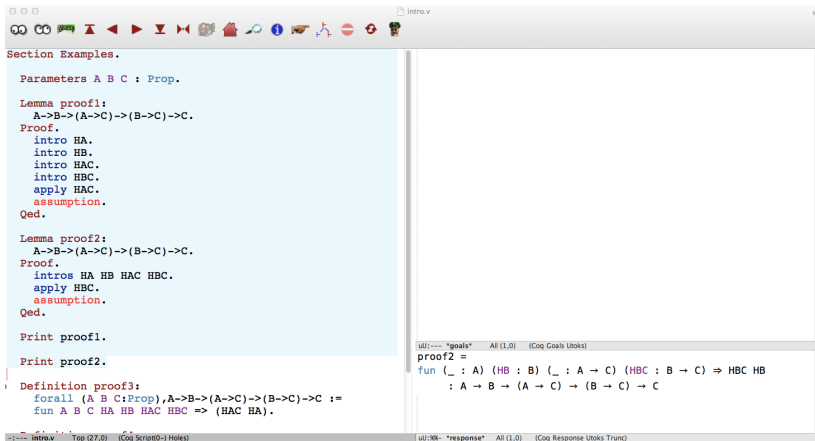
ul:--- "response" All (1,0) (Coq Response Utoks Trunc)

C'est le  $\lambda$ -terme que nous avons construit "à la main"



# Exemples précédents dans Rocq

Le  $\lambda$ -terme pour la seconde preuve est :



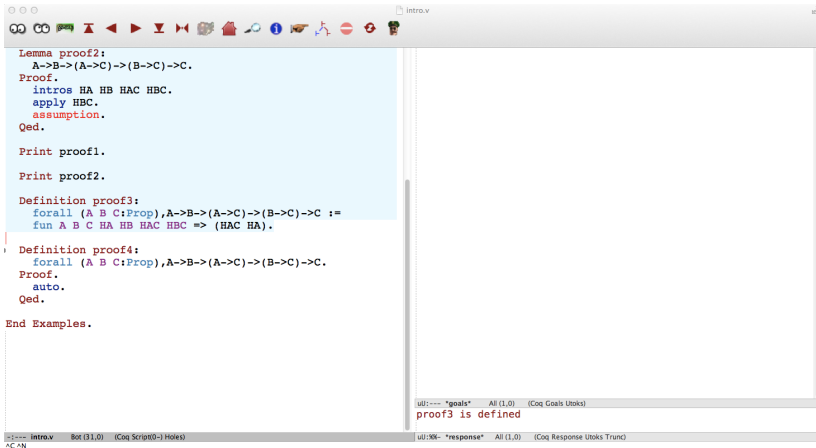
```
Section Examples.  
  
Parameters A B C : Prop.  
  
Lemma proof1:  
  A->B->(A->C)->(B->C)->C.  
Proof.  
  intro HA.  
  intro HB.  
  intro HAC.  
  intro HBC.  
  apply HAC.  
  assumption.  
Qed.  
  
Lemma proof2:  
  A->B->(A->C)->(B->C)->C.  
Proof.  
  intros HA HB HAC HBC.  
  apply HBC.  
  assumption.  
Qed.  
  
Print proof1.  
  
Print proof2.  
  
Definition proof3:  
  forall (A B C:Prop), A->B->(A->C)->(B->C)->C :=  
  fun A B C HA HB HAC HBC => (HAC HA).  
end
```

ui:--- \*goals\* All (1,0) (Coq Goals Utoks)  
proof2 =  
fun (\_ : A) (HB : B) (\_ : A → C) (HBC : B → C) => HBC HB  
: A → B → (A → C) → (B → C) → C

ui:--- intro.v Top (27,0) (Coq Script0-1 Holes)  
ui:--- \*response\* All (1,0) (Coq Response Utoks Trunc)

# Exemples précédents dans Rocq

Nous pourrions donner directement la preuve sous la forme d'un  $\lambda$ -terme :



```
Lemma proof2:
  A->B->(A->C)->(B->C)->C.
Proof.
  intros HA HB HAC HBC.
  apply HBC.
  assumption.
Qed.

Print proof1.

Print proof2.

Definition proof3:
  forall (A B C:Prop), A->B->(A->C)->(B->C)->C :=
  fun A B C HA HB HAC HBC => (HAC HA).

Definition proof4:
  forall (A B C:Prop), A->B->(A->C)->(B->C)->C.
Proof.
  auto.
Qed.

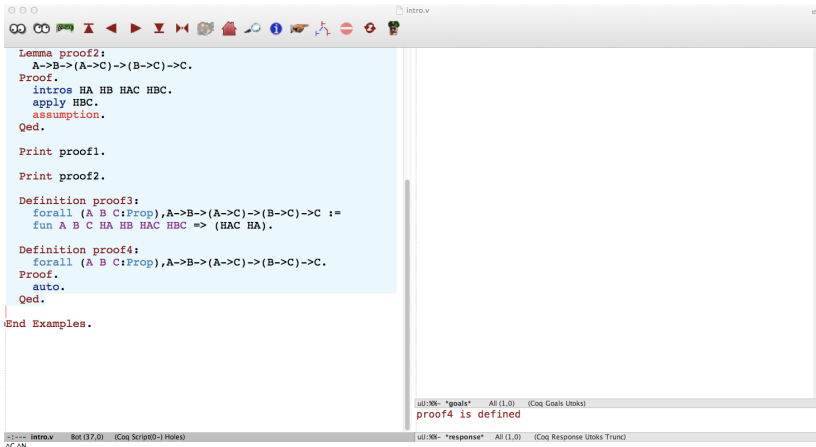
End Examples.
```

ul: --- "goals" All (1,0) (Coq Goals Utoks)  
proof3 is defined

ul: --- "response" All (1,0) (Coq Response Utoks Trunc)

# Exemples précédents dans Rocq

... ou utiliser des tactiques plus puissantes de Coq :



```

Lemma proof2:
  A->B->(A->C)->(B->C)->C.
Proof.
  intros HA HB HAC HBC.
  apply HBC.
  assumption.
Qed.

Print proof1.

Print proof2.

Definition proof3:
  forall (A B C:Prop), A->B->(A->C)->(B->C)->C :=
  fun A B C HA HB HAC HBC => (HAC HA).

Definition proof4:
  forall (A B C:Prop), A->B->(A->C)->(B->C)->C.
Proof.
  auto.
Qed.

End Examples.

```

ui:~X%- "goals" All (1,0) (Coq Goals Utoks)

proof4 is defined

ui:~X%- "response" All (1,0) (Coq Response Utoks Trunc)

intro.v Bot (37,0) (Coq Script(0-) Holes)

AC ^N