

Initiation à la recherche

The Curry Howard Correspondence:
A Gentle Introduction

Jolan PHILIPPE

24 Novembre 2025

Université d'Orléans

Plan

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

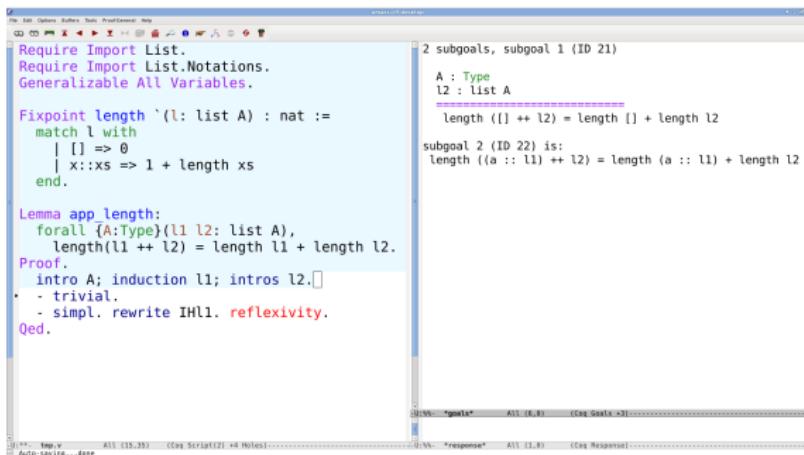
[...] L'équipe de Coq continue de développer le système, apportant à chaque nouvelle version des améliorations significatives en expressivité et en facilité d'utilisation.

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



The screenshot shows the Coq proof assistant interface. The left pane displays a Coq script for defining the length of a list. The right pane shows the proof state with two subgoals.

```
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

Plan

L'assistant de preuve Rocq: Overview

Un peu de Correspondance Curry-Howard

Rocq en Pratique

Notations

Formules logiques

- A, B, \dots : propositions atomiques
- si 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 , $(e_1\ e_2)$, $\lambda x. e$

En OCaml: x , $e_1\ e_2$, $\lambda x\rightarrow e$

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)
- Γ : ensemble d'hypothèses

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

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

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

Logical World

λ -Calculus simplement typé

- A, B : types
- x : variables
- e : programmes (variable, abstraction, application)
- Γ : 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

Programming World

Déduction naturelle – Example 1

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

Déduction naturelle – Example 1

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

Déduction naturelle – Example 1

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

Déduction naturelle – Example 1

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

Déduction naturelle – Example 1

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

Déduction naturelle – Example 1

$$\frac{(i) \frac{(i) \frac{(i) \frac{(a) \frac{\Gamma \vdash A \rightarrow C}{\Gamma \equiv A, B, A \rightarrow C, B \rightarrow C \vdash C} \quad \Gamma \vdash A}{A, B, A \rightarrow C \vdash (B \rightarrow C) \rightarrow C} \quad A, B \vdash (A \rightarrow C) \rightarrow (B \rightarrow C) \rightarrow C}{A \vdash B \rightarrow (A \rightarrow C) \rightarrow (B \rightarrow C) \rightarrow C}}{\vdash A \rightarrow B \rightarrow (A \rightarrow C) \rightarrow (B \rightarrow C) \rightarrow C}}$$

Correspondance de Curry-Howard

Déduction naturelle – Example 1

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

Déduction naturelle – Example 2

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

Déduction naturelle – Example 2

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

Déduction naturelle – Example 2

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

Déduction naturelle – Example 2

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

Déduction naturelle – Example 2

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

Déduction naturelle – Example 2

$$\frac{(i) \frac{(i) \frac{(i) \frac{(a) \frac{\Gamma \vdash B \rightarrow C}{\Gamma \equiv A, B, A \rightarrow C, B \rightarrow C \vdash C} \quad \Gamma \vdash B}{A, B, A \rightarrow C \vdash (B \rightarrow C) \rightarrow C} \quad \Gamma \vdash B}{A, B \vdash (A \rightarrow C) \rightarrow (B \rightarrow C) \rightarrow C} \quad \Gamma \vdash B}{A \vdash B \rightarrow (A \rightarrow C) \rightarrow (B \rightarrow C) \rightarrow C} \quad \Gamma \vdash B}{\vdash A \rightarrow B \rightarrow (A \rightarrow C) \rightarrow (B \rightarrow C) \rightarrow C}$$

Correspondance de Curry-Howard

Déduction naturelle – Example 2

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

Correspondance de Curry-Howard

λ -calculus: trouver un terme du type donné

$\vdash ?$

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

Correspondance de Curry-Howard

λ -calculus: trouver un terme du type donné

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

Correspondance de Curry-Howard

λ -calculus: trouver un terme du type donné

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

Correspondance de Curry-Howard

λ -calculus: trouver un terme du type donné

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

Correspondance de Curry-Howard

λ -calculus: trouver un terme du type donné

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

Correspondance de Curry-Howard

λ -calculus: trouver un terme du type donné

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

Correspondance de Curry-Howard

λ -calculus: trouver un terme du type donné

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

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

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

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

Correspondance de Curry-Howard

Pour toute formule, il existe une preuve de cette formule en déduction naturelle si et seulement s'il existe un λ -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 ?

Plan

L'assistant de preuve Rocq: Overview

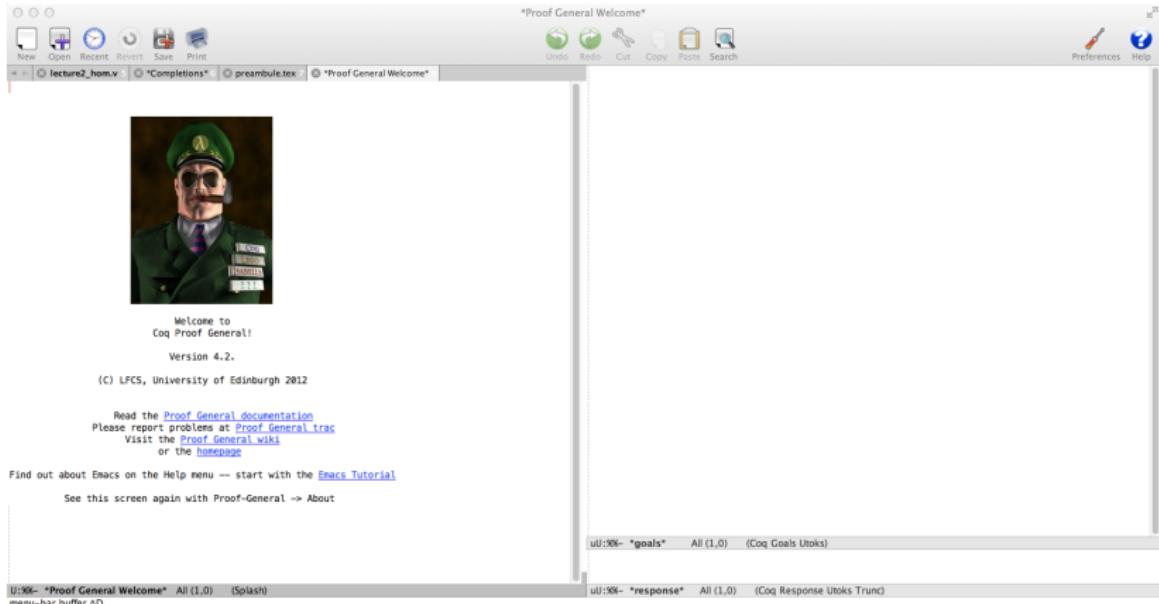
Un peu de Correspondance Curry-Howard

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

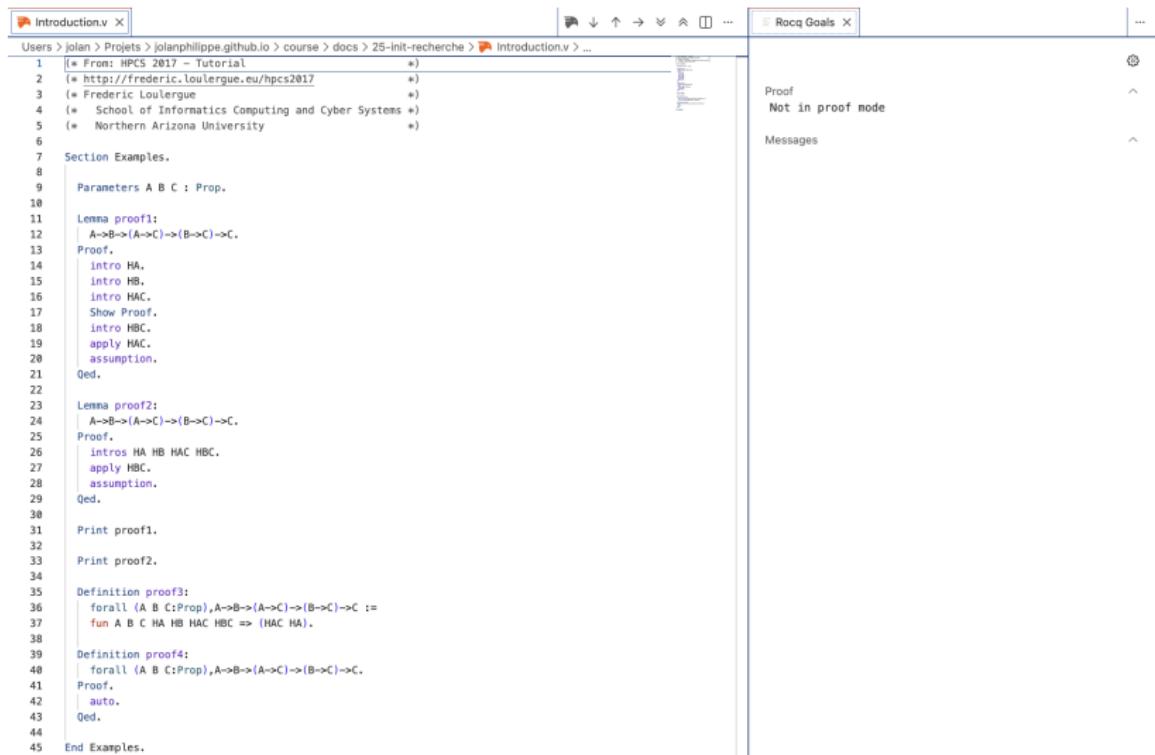
Exemples précédents dans Rocq

Le mode Proof General pour Emacs . . .



Exemples précédents dans Rocq

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



The screenshot shows the VsRocq extension in Visual Studio Code. On the left, the code editor displays a file named 'Introduction.v' containing a Coq proof script. The script includes definitions for 'HPC5 2017' and 'Introduction', and several lemmas ('proof1', 'proof2', 'proof3', 'proof4') with their respective proofs. On the right, a 'Rocq Goals' pane shows the proof state: 'Not in proof mode'. The 'Messages' section is empty.

```
Users > jolan > Projets > jolanphilippe.github.io > course > docs > 25-init-recherche > Introduction.v > ...
```

```
1 (* From: HPC5 2017 - Tutorial *)
2 (* http://ffrederic.louergue.eu/hpcs2017 *)
3 (* Frederic Louergue *)
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 HBC.
20 | assumption.
21 Qed.
```

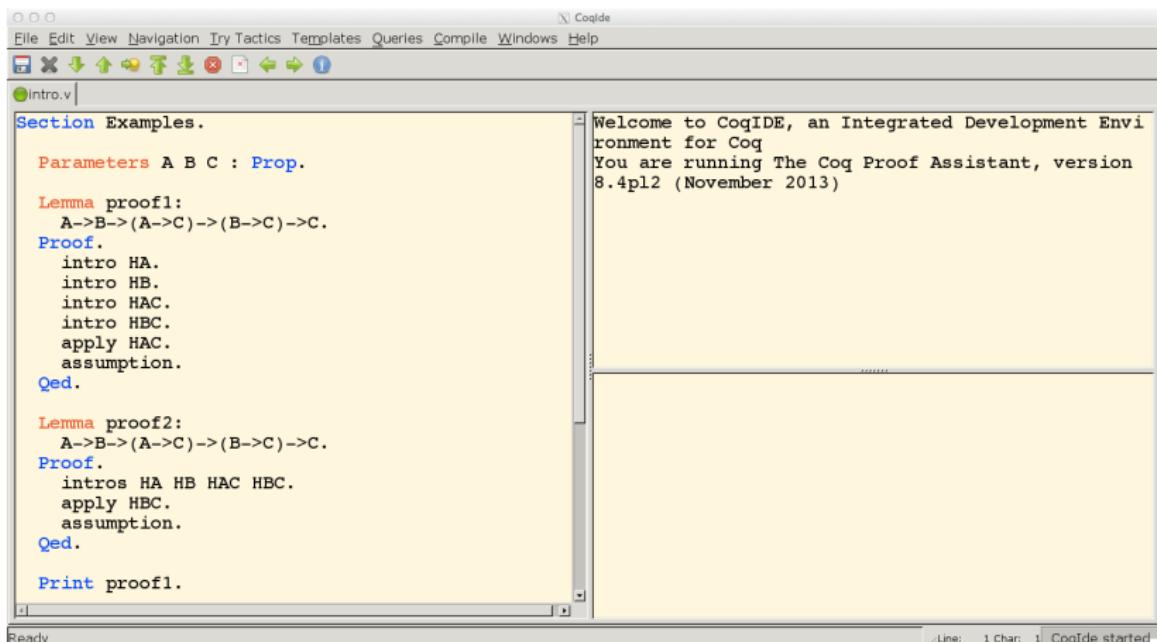
```
22
23 Lemma proof2:
24 | A->B->(A->C)->(B->C)->C.
25 Proof.
26 | intros HA HB HAC HBC.
27 | apply HBC.
28 | assumption.
29 Qed.
```

```
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.
```

Exemples précédents dans Rocq

... ou RocqIDE



The screenshot shows the CoqIDE interface with a proof script in the left pane and a welcome message in the right pane.

Left Pane (intro.v):

```
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.

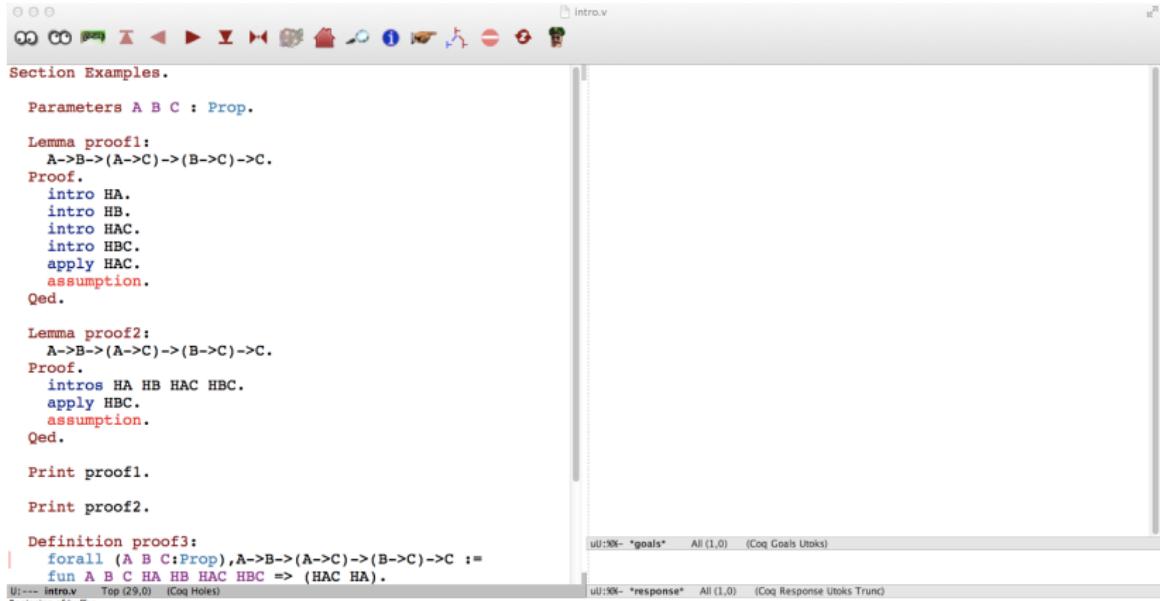
Print proof1.
```

Right Pane:

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

Exemples précédents dans Rocq

On ouvre le fichier `Introduction.v`¹:



```
intro.v

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
```

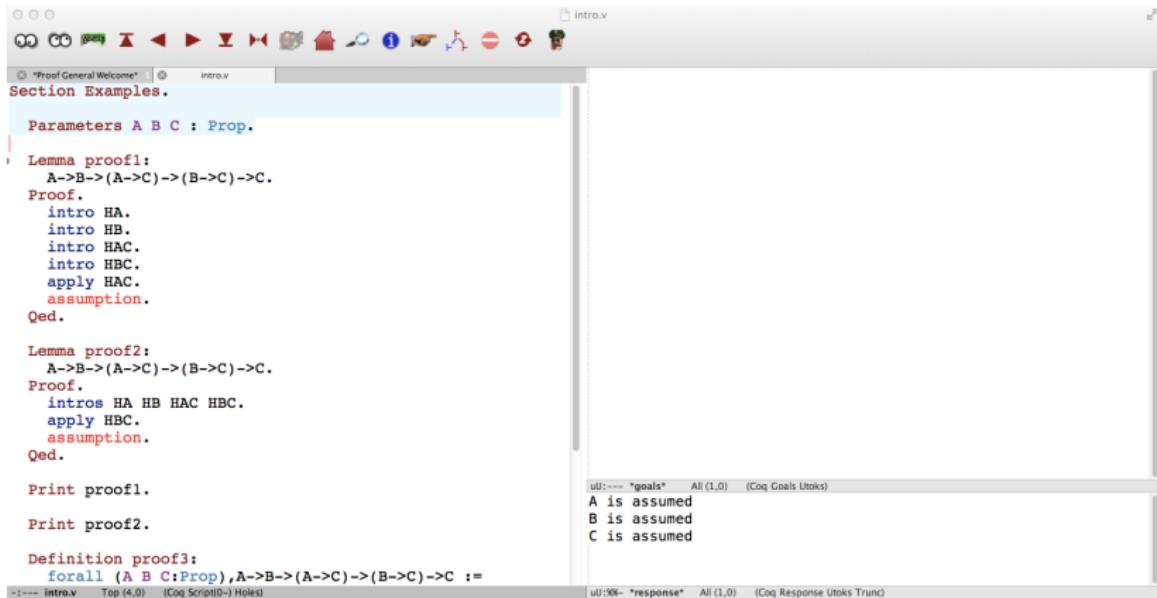
uU:XX- *goals* All (1,0) (Coq Goals Utoks)

uU:XX- *response* All (1,0) (Coq Response Utoks Trunc)

¹disponible 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 interface with a Coq script in the left pane and its execution results in the right pane.

Left pane (Script):

```
intro.v
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 :=

```

Right pane (Execution Results):

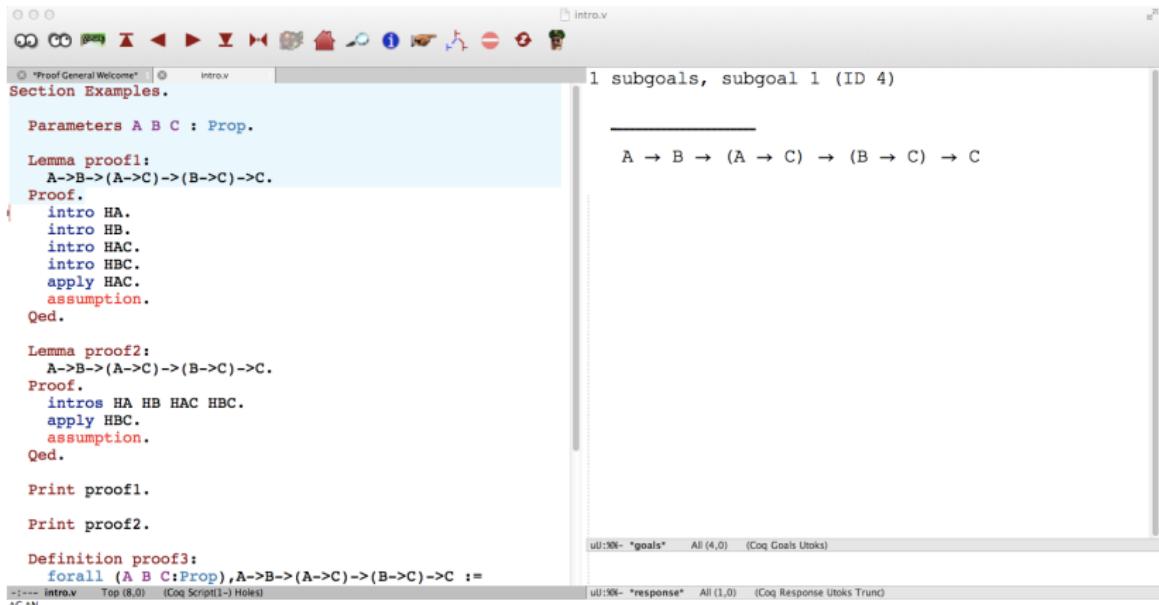
```
uU:--- "goals"  All (1,0)  (Coq Goals Utoks)
A is assumed
B is assumed
C is assumed

uU:--- intro.v  Top (4,0)  (Coq ScriptUtoks Holes)

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

Exemples précédents dans Rocq

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



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

```
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 proof state with one subgoal:

```
1 subgoals, subgoal 1 (ID 4)
```

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

At the bottom, the status bar shows:

- Left: intro.v Top (8,0) (Coq Script(1-)> Holes)
- Center: uU:%%- *goals* All (4,0) (Coq Goals Utoks)
- Right: uU:%%- *response* All (1,0) (Coq Response Utoks Trunc)

Exemples précédents dans Rocq

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

The screenshot shows the Rocq proof assistant interface. On the left, a Coq script is displayed with various tactics and definitions. On the right, the proof state is shown with subgoals and the current tactic application.

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).

intro.v Top (9,0) (Coq Script(1-) Holes)

1 subgoals, subgoal 1 (ID 5)

HA : A

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

uU:5%- *goals* All (5,0) (Coq Goals Utoks)

uU:5%- *response* All (1,0) (Coq Response Utoks Trunc)

Exemples précédents dans Rocq

Le contexte est maintenant similaire à Γ :

The screenshot shows the Rocq proof assistant interface. On the left, the script file `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.

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 state of the proof. It displays the context and a goal:

```
1 subgoals, subgoal 1 (ID 8)

HA : A
HB : B
HAC : A → C
HBC : B → C

C
```

At the bottom, two tabs are visible:

- `uU:%%- *goals*` All (8,0) (Coq Goals Utocks)
- `uU:%%- *response*` All (1,0) (Coq Response Utocks Trunc)

Exemples précédents dans Rocq

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

The screenshot shows the Rocq proof assistant interface. On the left, the script file `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.

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:

1 subgoals, subgoal 1 (ID 9)

```
HA : A
HB : B
HAC : A → C
HBC : B → C
_____
A
```

At the bottom, the status bar shows:

uU:--- intro.v Top (13,0) (Coq Script(1-) Holes)

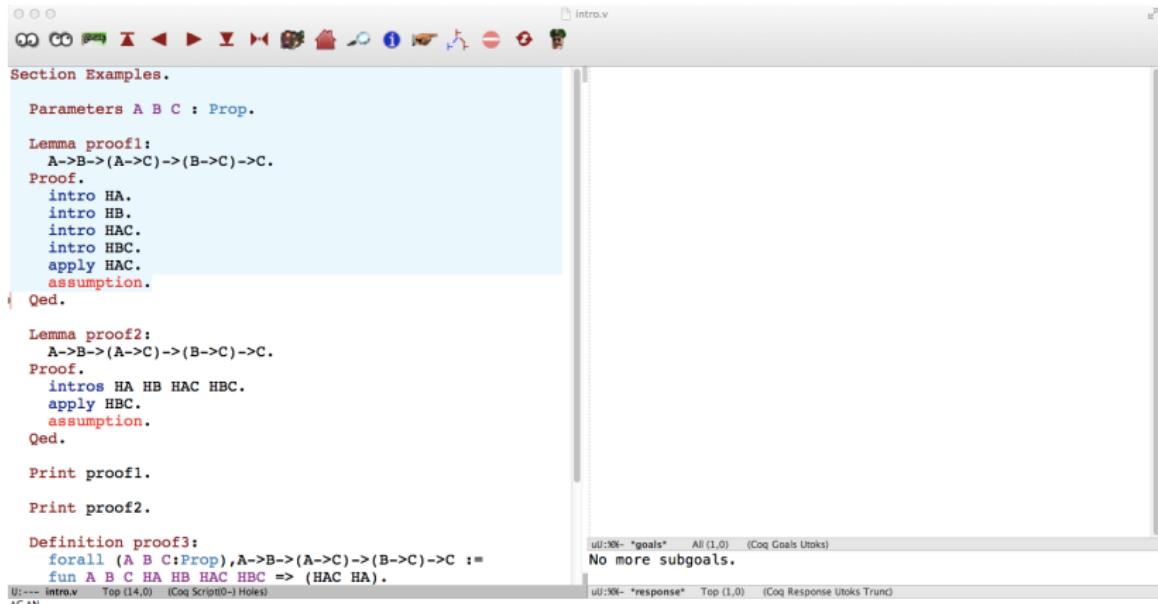
uU:--- *goals* All (8,0) (Coq Goals Utoks)

uU:--- *response* All (1,0) (Coq Response Utoks Trunc)

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

Exemples précédents dans Rocq

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



The screenshot shows the CoqIDE interface with a proof script in the file `intro.v`. The script defines three lemmas: `proof1`, `proof2`, and `proof3`. `proof1` and `proof2` are proofs by induction on A , while `proof3` is a definition of a function `forall` that takes $(A B C : \text{Prop})$ and returns $(A \rightarrow B \rightarrow (A \rightarrow C) \rightarrow (B \rightarrow C) \rightarrow C)$. The `Print` command is used to display the definitions of `proof1` and `proof2`. The status bar at the bottom shows the file name `intro.v`, the top level `Top`, and the number of holes `(Coq Script)(0-- Holes)`. The bottom status bar shows the command `uU:%%- "goals"`, the top level `Top`, and the number of goals `(Coq Goals Utoks)`. The message `No more subgoals.` is displayed, indicating that the proof is complete.

```
intro.v
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).
```

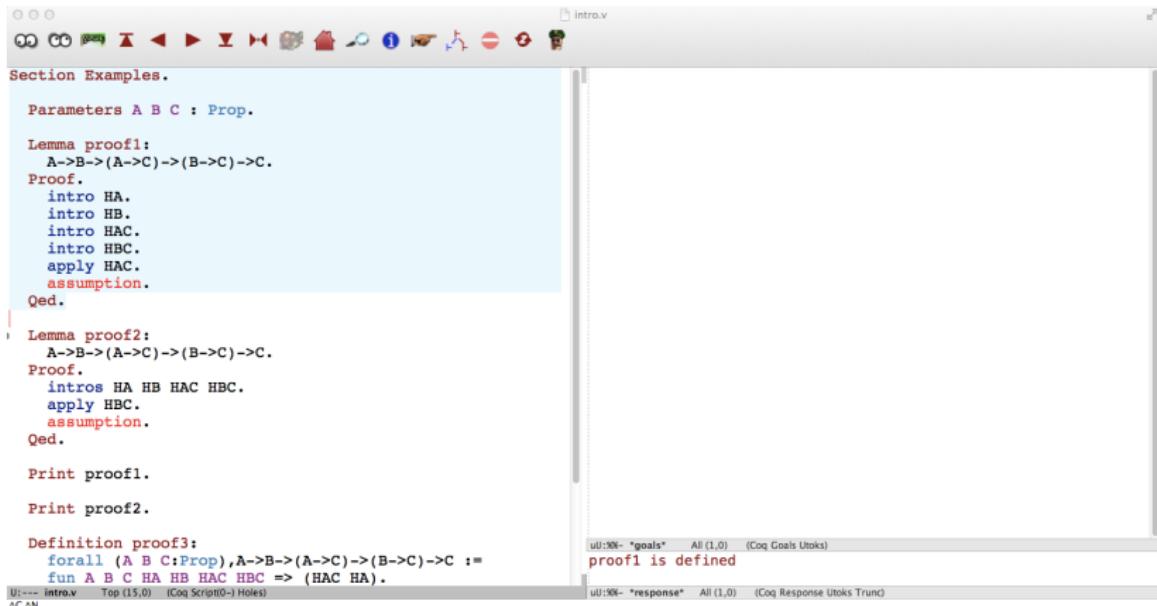
uU:%%- "goals" All (1,0) (Coq Goals Utoks)
No more subgoals.

uU:%%- "response" Top (1,0) (Coq Response Utoks 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 :



The screenshot shows the Rocq interface with a Coq script in the left pane and its execution results in the right pane.

Coq Script (intro.v):

```
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).
```

Execution Results:

uU:OK- "goals" All (1,0) (Coq Goals Utoks)

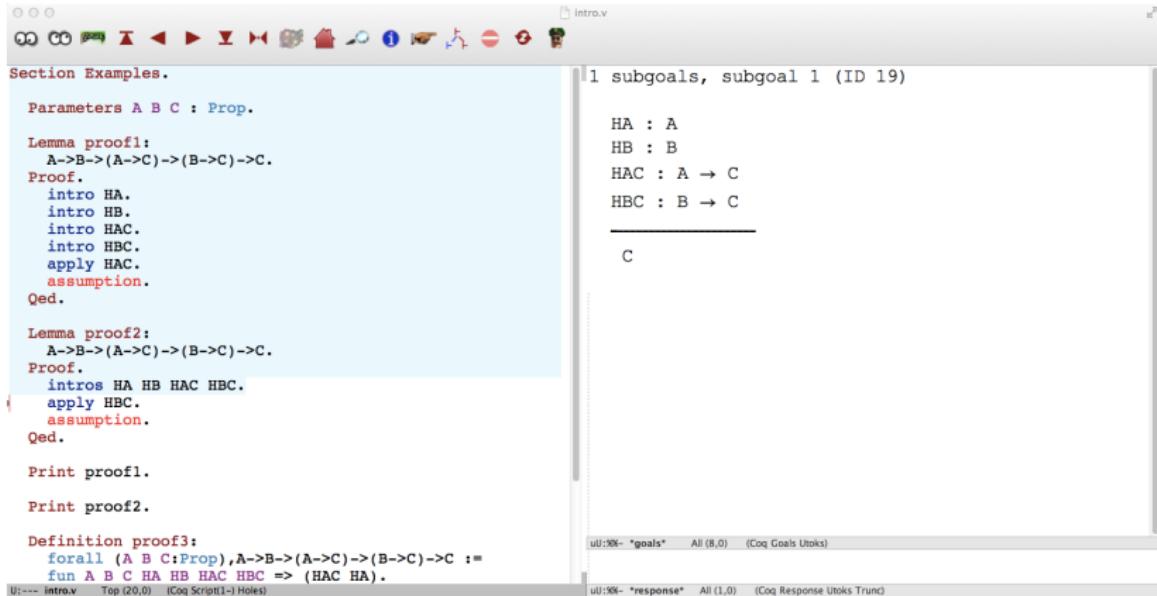
proof1 is defined

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

U:--- intro.v Top (15,0) (Coq Script(0-- Holes))
AC ^N

Exemples précédents dans Rocq

Deuxième version, nous faisons plusieurs intro :



The screenshot shows the Rocq proof assistant interface. The left pane displays a Coq script with several definitions and proofs. The right pane shows the state of the proof, including subgoals and the current goal. The bottom pane shows the history of the proof process.

```
intro.v
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).
```

1 subgoals, subgoal 1 (ID 19)

HA : A
HB : B
HAC : A → C
HBC : B → C

C

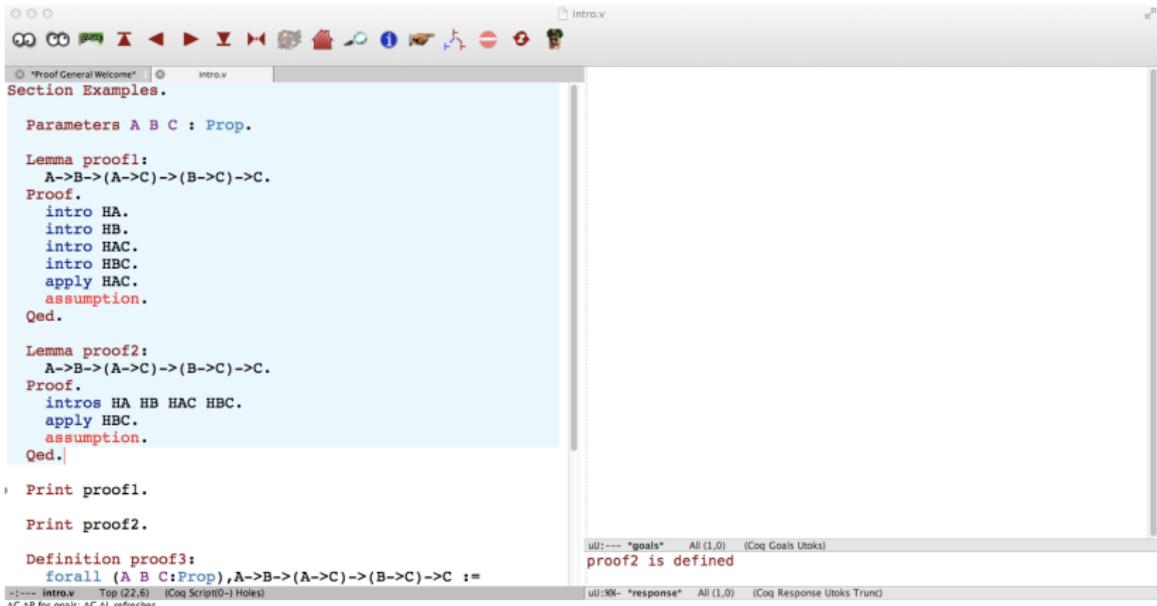
uU:0%- *goals* All (8,0) (Coq Goals Utoks)

uU:0%- *response* All (1,0) (Coq Response Utoks Trunc)

U:--- intro.v Top (20,0) (Coq Script(1-) Holes)

Exemples précédents dans Rocq

et apply HBC au lieu de apply HAC :



The screenshot shows the Rocq proof assistant interface. The top bar displays the title "intro.v" and various toolbar icons. The main workspace contains a proof script:

```
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 :=
```

Below the script, the status bar shows:

```
|-:-- intro.v  Top (22,6)  (Coq Script(0--> Holes))
  ^C ^P for goals; ^C ^L refreshes
```

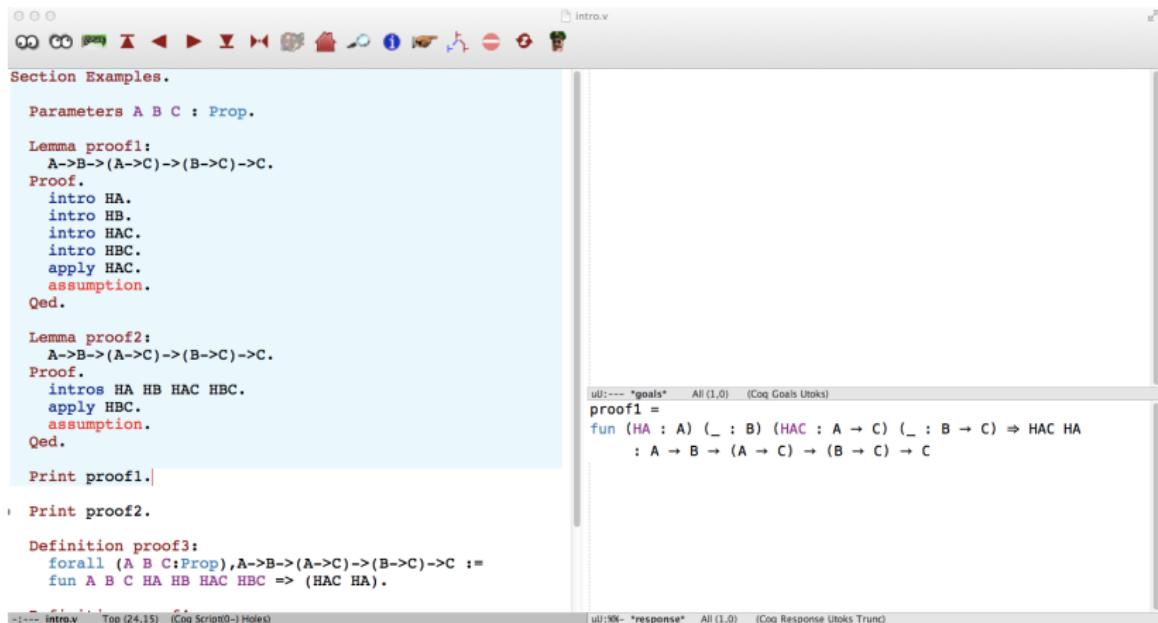
The right side of the interface shows the execution results:

```
uU:--- *goals*  All (1,0)  (Coq Goals Utoks)
proof2 is defined
```

```
uU:%%- *response*  All (1,0)  (Coq Response Utoks Trunc)
```

Exemples précédents dans Rocq

Print t . affiche le terme t :



The screenshot shows the CoqIDE interface with a proof script in the left pane and its term output in the right pane.

Proof Script (intro.v):

```
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).
```

Term Output (right pane):

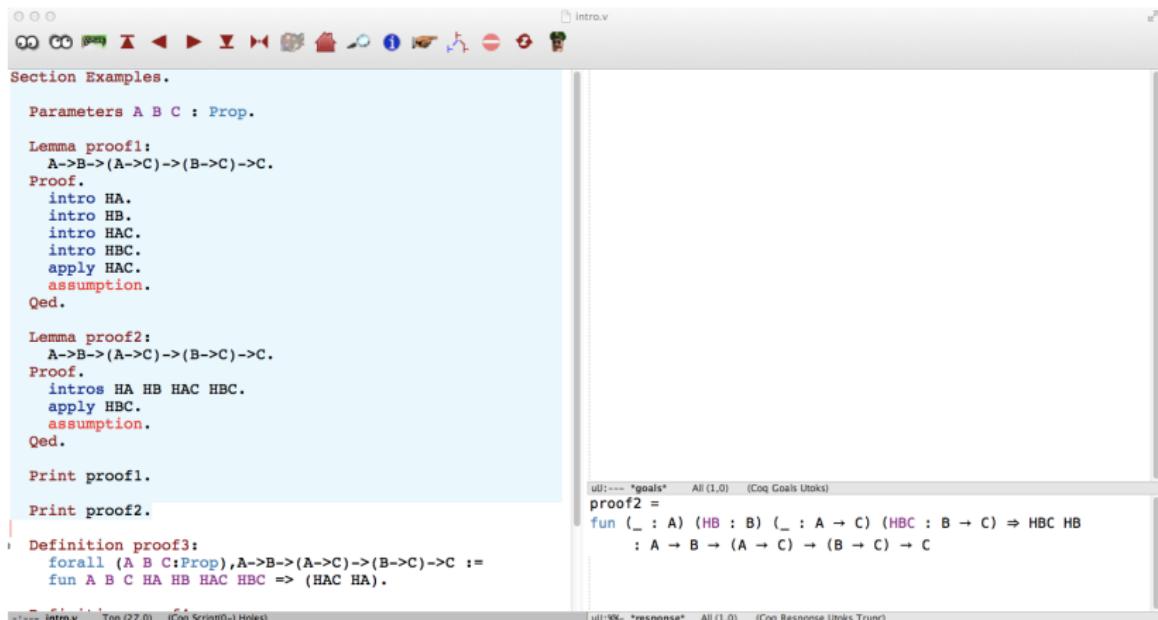
```
uU:--- "goals"  All (1,0)  (Coq Goals Utoks)
proof1 =
fun (HA : A) (_ : B) (HAC : A → C) (_ : B → C) ⇒ HAC HA
  : A → B → (A → C) → (B → C) → C
```

Bottom status bar: Top (24,15) (Coq Script 0-1 Holes) uU:%%- *response* All (1,0) (Coq Response Utoks Trunc)

C'est le λ -terme que nous avons construit “à la main”

Exemples précédents dans Rocq

Le λ -terme pour la seconde preuve est :



The screenshot shows the Rocq proof assistant interface. The top bar has icons for file operations and a file named "intro.v". The main window is divided into two panes. The left pane contains 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).
```

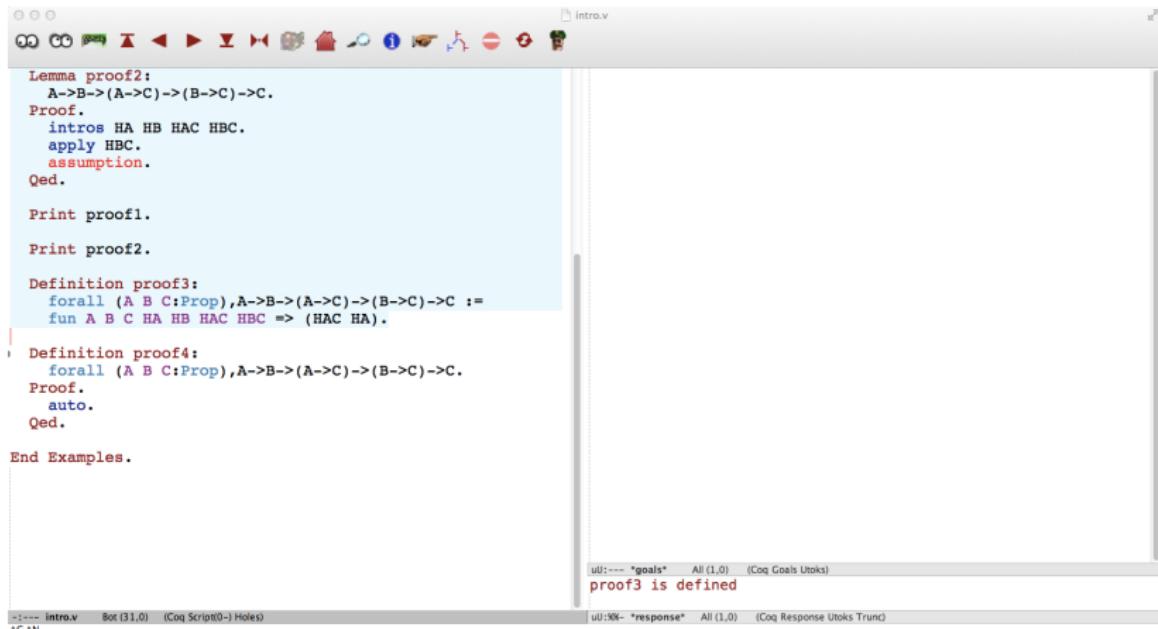
The right pane shows the corresponding λ -terms:

```
uU:--- "goals"  All (1,0)  (Coq Goals UtoIs)
proof2 =
  fun (_ : A) (HB : B) (_ : A → C) (HBC : B → C) ⇒ HBC HB
    : A → B → (A → C) → (B → C) → C
  : A → B → (A → C) → (B → C) → C
```

At the bottom, the status bar shows "intro.v" and "Top (27,0) (Coq Script 0- Holes)" on the left, and "uU:--- "response" All (1,0) (Coq Response UtoIs Trunc)" on the right.

Exemples précédents dans Rocq

Nous pourrions donner directement la preuve sous la forme d'un λ -terme :



The screenshot shows the CoqIDE interface with a proof script in the left pane and its execution results in the right pane.

Left pane (Proof Script):

```
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.
```

Right pane (Execution Results):

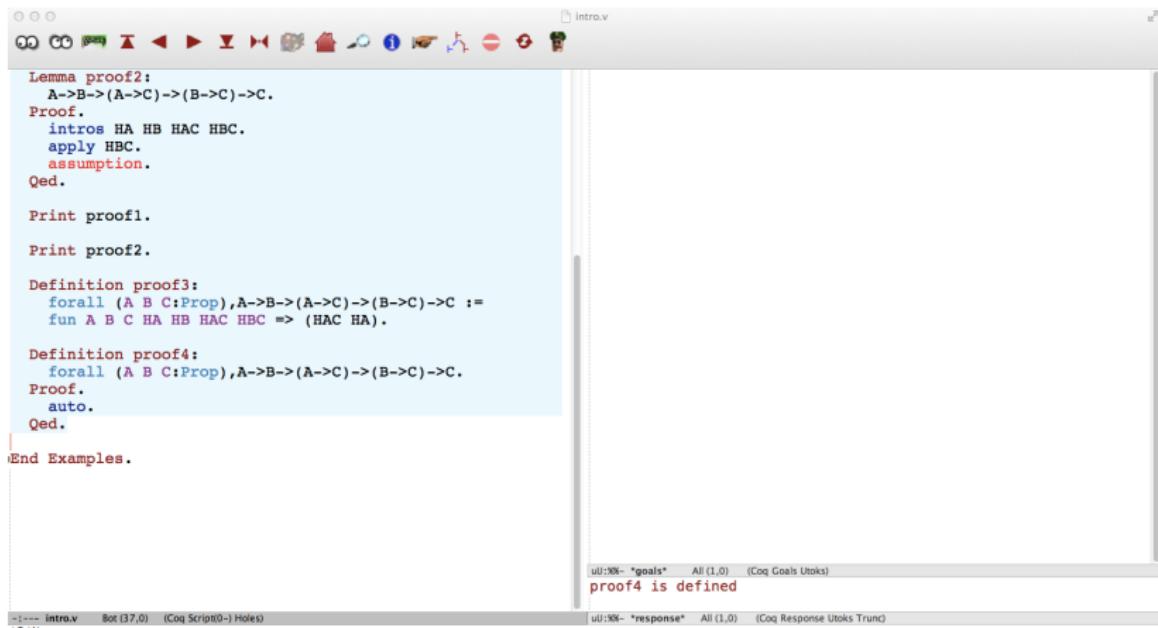
```
uU:--- "goals"    All (1,0)  (Coq Goals Utols)
proof3 is defined
```

At the bottom of the interface, status bars show:

- Left: `introduction.v` Bot (31,0) (Coq Script(0-) Holes)
- Right: `uU:--- "response" All (1,0) (Coq Response Utols Trunc)`

Exemples précédents dans Rocq

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



The screenshot shows the Rocq interface with a Coq script in the left pane and its execution results in the right pane.

Left pane (Script):

```
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.
```

Right pane (Execution Results):

```
uU:XX- *goals*  All (1,0)  (Coq Goals Utoks)
proof4 is defined
uU:XX- *response* All (1,0)  (Coq Response Utoks Trunc)
```

At the bottom left, the status bar shows: `-:--- intro.v Bot (37,0) (Coq Script(0-) Holes)` and `AC^N`. At the bottom right, it shows: `uU:XX- *response* All (1,0) (Coq Response Utoks Trunc)`.