

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. On the left, the code for defining the length function and proving its additivity is displayed:

```
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.
```

On the right, the proof state is shown with two subgoals:

```
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
```

The bottom status bar indicates the file is "tmp.v", has 15 lines, and contains 4 holes.

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

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)
- 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

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

Correspondance de Curry-Howard

Déduction naturelle – Example 1

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

Correspondance de Curry-Howard

Déduction naturelle – Example 1

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

Correspondance de Curry-Howard

Déduction naturelle – Example 1

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

Correspondance de Curry-Howard

Déduction naturelle – Example 1

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

Correspondance de Curry-Howard

Déduction naturelle – Example 1

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

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

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

Correspondance de Curry-Howard

Déduction naturelle – Example 2

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

Correspondance de Curry-Howard

Déduction naturelle – Example 2

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

Correspondance de Curry-Howard

Déduction naturelle – Example 2

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

Correspondance de Curry-Howard

Déduction naturelle – Example 2

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

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

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

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

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

Correspondance de Curry-Howard

λ -calculus: trouver un terme du type donné

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

Correspondance de Curry-Howard

λ -calculus: trouver un terme du type donné

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

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

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

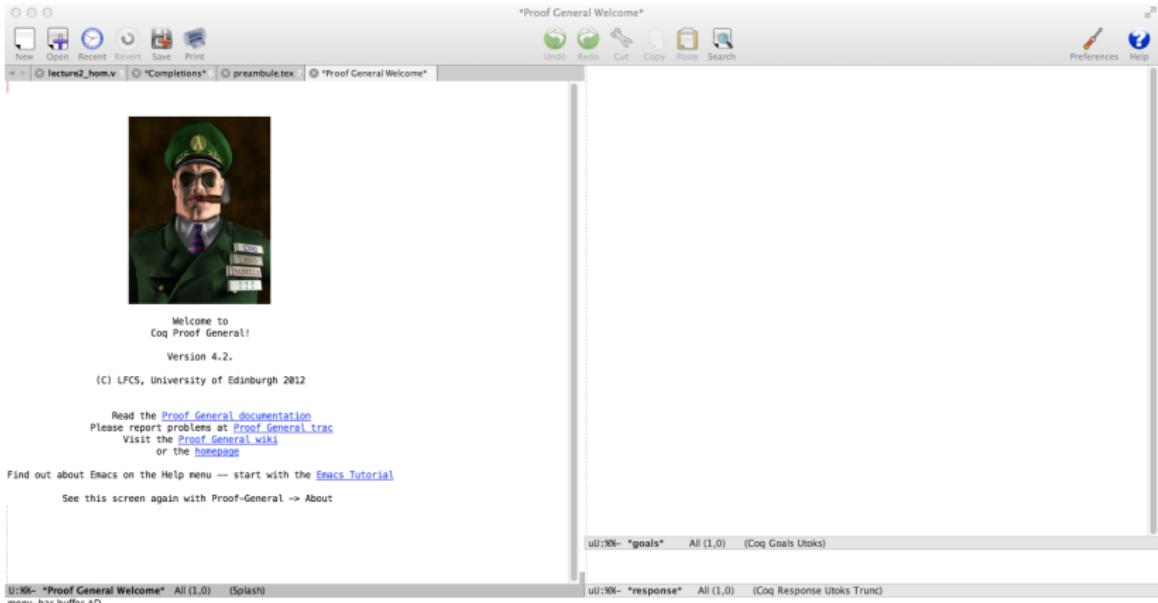
Rocq en Pratique

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, there is a code editor window titled "Introduction.v" containing a Coq script. The script includes definitions like "HPC5 2017", parameters A, B, C, lemmas proof1 and proof2, and definitions proof3 and proof4. On the right, there is a panel titled "Rocq Goals" which displays the message "Not in proof mode".

```
Users > jolan > Projets > jolanphilippe.github.io > course > docs > 25-init-recherche > Introduction.v > ...
1 (* From: HPC5 2017 - Tutorial *)
2 (* http://frédéric.loulergue.eu/hpcs2017 *)
3 (* Frédéric 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 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 file named `intro.v` open. The code contains a series of Coq tactics and annotations:

```
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.
```

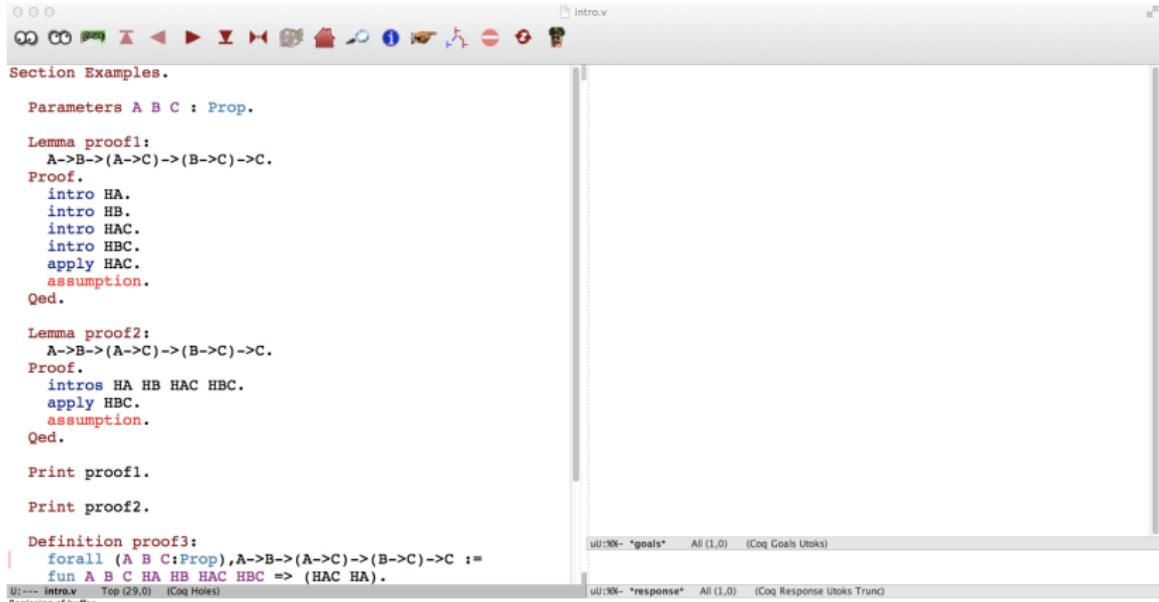
In the top right pane, there is a welcome message:

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

The status bar at the bottom right indicates "CoqIDE started".

Exemples précédents dans Rocq

On ouvre le fichier `Introduction.v`¹:



The screenshot shows the Rocq proof assistant interface. At the top, there is a toolbar with various icons. Below the toolbar, the title bar displays "intro.v". The main area contains the Coq code for the `Introduction.v` file. The code includes definitions for `proof1` and `proof2`, a proof script for `proof1`, and a definition for `proof3`. At the bottom, there are two status bars: one for the current buffer ("U:--- intro.v Top (29,0) (Coq Holes)") and one for the response buffer ("uU:XX- *goals* All (1,0) (Coq Goals Utoks)").

```
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).
```

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 output in the right pane.

Left Pane (Script):

```
intro.v
Proof General Welcome! 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 (Output):

```
uU:--- "goals"  All (1,0)  (Coq Goals Utoks)
A is assumed
B is assumed
C is assumed
uU:--- intro.v  Top (4,0)  (Coq Script(0-) 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. On the left, there is a code editor window titled "intro.v" containing a Coq script. The script defines parameters A, B, and C of type Prop, states two lemmas (proof1 and proof2) involving implications between them, and prints their definitions. Below the code editor, the status bar shows "intro.v Top (8,0) (Coq Script(1-> Holes))". On the right, there is a proof state window titled "intro.v" showing one subgoal: $A \rightarrow B \rightarrow (A \rightarrow C) \rightarrow (B \rightarrow C) \rightarrow C$. The status bar below it shows "uU:%%- *goals* All (4,0) (Coq Goals Utoks)". At the bottom, there is a response window titled "intro.v" showing the result of the proof, which is a definition of proof3. The status bar below it shows "uU:%%- *response* All (1,0) (Coq Response Utoks Trunc)". The interface includes a toolbar at the top with various icons for file operations and help.

```
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 :=
```

Exemples précédents dans Rocq

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

The screenshot shows the Rocq proof assistant interface. On the left, a code editor displays a Coq script named `intro.v`. The script contains several parts:

- A section titled "Section Examples".
- Parameters `A B C : Prop.`
- A lemma `proof1` with its proof script:

```
Lemma proof1:
  A->B->(A->C)->(B->C)->C.
Proof.
  intro HA.
  intro HB.
  intro HAC.
  intro HBC.
  apply HAC.
  assumption.
Qed.
```
- A lemma `proof2` with its proof script:

```
Lemma proof2:
  A->B->(A->C)->(B->C)->C.
Proof.
  intros HA HB HAC HBC.
  apply HBC.
  assumption.
Qed.
```
- Print statements for `proof1` and `proof2`.
- A definition `proof3` with its implementation:

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

On the right, the proof state is shown in a tree-like structure. The root node is labeled "1 subgoals, subgoal 1 (ID 5)". Below it is a horizontal line with the hypothesis `HA : A` above it. Further down the line is the goal `B → (A → C) → (B → C) → C`.

At the bottom of the interface, two tabs are visible:

- "U:-- intro.v Top (9,0) (Coq Script(1-) Holes)"
- "uU:%%- *goals* All (5,0) (Coq Goals Utoks)"
- "uU:%%- *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, there is a toolbar with various icons for file operations, navigation, and proof steps. Below the toolbar is a section titled "Section Examples." containing Coq code. The code defines parameters A, B, and C of type Prop, and two lemmas, proof1 and proof2, which prove the same logical statement: $A \rightarrow B \rightarrow (A \rightarrow C) \rightarrow (B \rightarrow C) \rightarrow C$. Both proofs use an "intro" tactic to introduce assumptions HA, HB, HAC, and HBC, followed by an "apply" tactic to apply HAC or HBC respectively, and an "assumption" tactic to close the proof. The code concludes with "Qed." for each lemma. After the lemmas, there are "Print" statements for proof1 and proof2, and a definition proof3 that provides a more abstract view of the proof structure. At the bottom of the code editor, there is a status bar showing the file name "intro.v", the top level (12,0), and the Coq version (8.0). On the right side of the interface, there is a proof state window. It shows the goal "1 subgoals, subgoal 1 (ID 8)" and the context "HA : A", "HB : B", "HAC : A → C", and "HBC : B → C". Below the context, there is a horizontal line and the term "C" on a new line, indicating the goal to be proved. At the very bottom of the interface, there are two tabs: "uU:%%- *goals*" (All (8,0) (Coq Goals Utoks)) and "uU:%%- *response*" (All (1,0) (Coq Response Utoks Trunc)).

```
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 (8,0) (Coq Goals Utoks)

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

```
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:%%- *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 Coq proof assistant interface. The top part displays the file `intro.v` 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 :=
  fun A B C HA HB HAC HBC => (HAC HA).
```

The bottom part shows the output in the `response` tab:

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

At the bottom left, the status bar shows: `intro.v Top (14,0) (Coq Script|0-) Holes`. At the bottom right, it shows: `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 proof assistant interface. At the top, there is a toolbar with various icons. Below the toolbar, the title bar says "intro.v". The main area contains a Coq 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 :=  
    fun A B C HA HB HAC HBC => (HAC HA).
```

At the bottom left, the status bar shows "U:--- intro.v Top (15,0) (Coq Script)(0-- Holes)" and "AC ^N". On the right side, there are two small windows showing the state of the proof. The top window is titled "uU:XK- *goals*" and shows the message "proof1 is defined". The bottom window is titled "uU:XK- *response*" and shows the message "All (1,0) (Coq Response Utols Trunc)".

Exemples précédents dans Rocq

Deuxième version, nous faisons plusieurs intro :

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 side of the interface shows the proof state and the generated code. The proof state is:

```
1 subgoals, subgoal 1 (ID 19)

HA : A
HB : B
HAC : A → C
HBC : B → C

C
```

The generated code in the `proof3` definition is:

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

At the bottom, the status bar shows:

```
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 has tabs for "intro.v" and "intro.v". The left pane displays two proofs:

```
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.
```

The right pane shows the state of the proof:

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

At the bottom, status bars show:

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

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

Exemples précédents dans Rocq

Print t . affiche le terme t :

The screenshot shows the Rocq proof assistant interface. The top bar has various icons for file operations and help. The main area is divided into two panes. The left pane contains Coq script code:

```
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 and its reduction steps:

```
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
```

At the bottom, status bars show the current file (intro.v), top level (Top (24,15)), and response level (uU:%%- *response* All (1,0)).

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 status bar showing "intro.v". The main window is divided into two panes. The left pane contains a Coq script with the following code:

```
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).
```

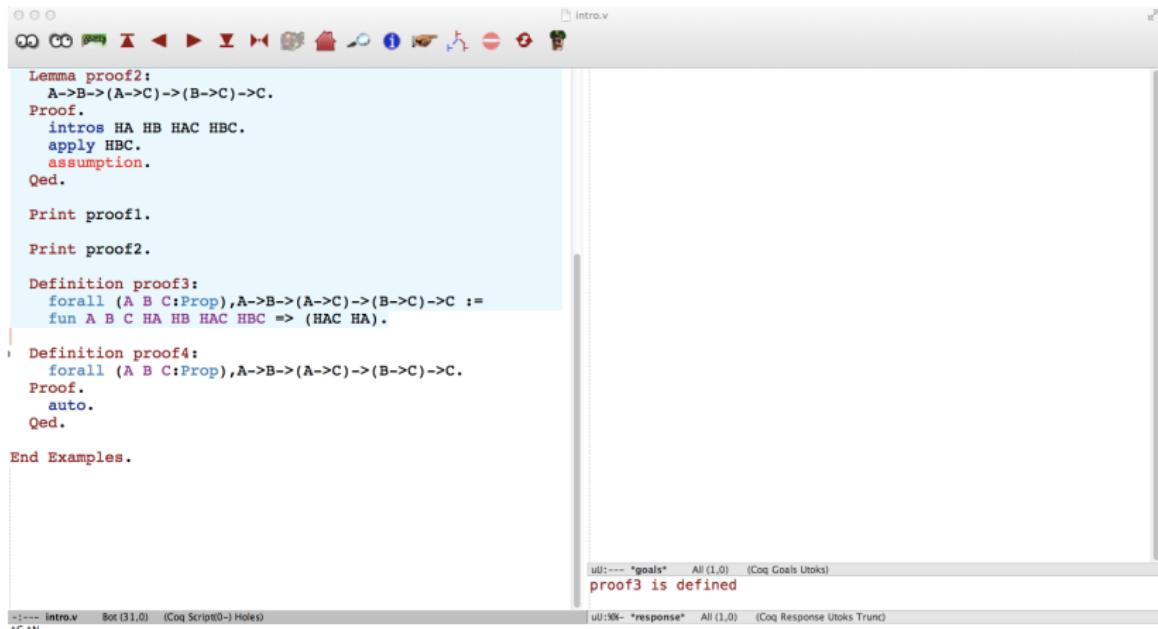
The right pane shows the term derivation for proof2:

```
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
```

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

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 named `intro.v`. The script contains the following code:

```
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.
```

The `proof3` definition is highlighted in blue. The status bar at the bottom shows the file name `intro.v`, the number of lines (31,0), and the fact that it is a Coq Script (0 holes). The bottom right corner displays the message "proof3 is defined".

Exemples précédents dans Rocq

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

The screenshot shows the Rocq interface with a toolbar at the top and a main workspace below. The workspace contains a Coq script with several definitions and proofs:

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

At the bottom, two tabs are visible: "intro.v" and "intro.v - response". The "intro.v - response" tab shows the output of the "proof4" definition:

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