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

L'assistant de preuve Rocq (anciennement Coq) i

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.

L’assistant de preuve Rocq (anciennement Coq) i

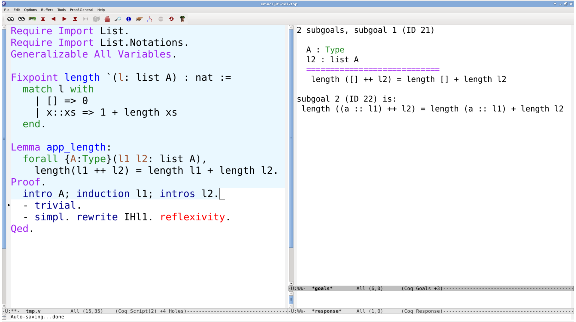
ACM SIGPLAN Software Award 2013

[. . .] 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



Plan

Notations

Formules logiques

- A, B, $\ldots$: propositions atomiques
- if F_1 et F_2 sont des formules logiques, $F_1 \to F_2$ 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`, `(e1 e2)`, `lambda(x)(e)`

En OCaml: `x`, `e1 e2`, `fun x -> e`

Notations

Formules logiques

- A, B, $\ldots$: 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

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

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

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

Logical World

λ-Calculus simplement typé

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

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

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

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

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

Programming World

Correspondance de Curry-Howard

Déduction naturelle – Example 1

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

Déduction naturelle – Example 1

$$(i)\,\dfrac{\dfrac{}{A \vdash B \to (A \to C) \to (B \to C) \to C}}{\vdash A \to B \to (A \to C) \to (B \to C) \to C}$$

Déduction naturelle – Example 1

$$
(i)\dfrac{(i)\dfrac{\overline{A,\ B \vdash (A \to C) \to (B \to C) \to C}}{A \vdash B \to (A \to C) \to (B \to C) \to C}}{\vdash A \to B \to (A \to C) \to (B \to C) \to C}
$$

Déduction naturelle – Example 1

$$\dfrac{(i)\;\dfrac{(i)\;\dfrac{(i)\;\dfrac{\overline{A,\ B,\ A \to C \vdash (B \to C) \to C}}{A,\ B \vdash (A \to C) \to (B \to C) \to C}}{A \vdash B \to (A \to C) \to (B \to C) \to C}}{\vdash A \to B \to (A \to C) \to (B \to C) \to C}}{}$$

Déduction naturelle – Example 1

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\Gamma \equiv A,\ B,\ A \to C,\ B \to C \vdash C}}{A,\ B,\ A \to C \vdash (B \to C) \to C}(i)}{A,\ B \vdash (A \to C) \to (B \to C) \to C}(i)}{A \vdash B \to (A \to C) \to (B \to C) \to C}(i)}{\vdash A \to B \to (A \to C) \to (B \to C) \to C}(i)
$$

Déduction naturelle – Example 1

$$
(i)\dfrac{(i)\dfrac{(i)\dfrac{(i)\dfrac{(a)\dfrac{\overline{\Gamma \vdash A \to C} \qquad \overline{\Gamma \vdash A}}{\Gamma \equiv A,\ B,\ A \to C,\ B \to C \vdash C}}{A,\ B,\ A \to C \vdash (B \to C) \to C}}{A,\ B \vdash (A \to C) \to (B \to C) \to C}}{A \vdash B \to (A \to C) \to (B \to C) \to C}}{\vdash A \to B \to (A \to C) \to (B \to C) \to C}
$$

Déduction naturelle – Example 1

$$
(i)\dfrac{(i)\dfrac{(i)\dfrac{(i)\dfrac{(a)\dfrac{(v)\dfrac{A \to C \in \Gamma}{\Gamma \vdash A \to C} \qquad (v)\dfrac{A \in \Gamma}{\Gamma \vdash A}}{\Gamma \equiv A,\ B,\ A \to C,\ B \to C \vdash C}}{A,\ B,\ A \to C \vdash (B \to C) \to C}}{A,\ B \vdash (A \to C) \to (B \to C) \to C}}{A \vdash B \to (A \to C) \to (B \to C) \to C}}{\vdash A \to B \to (A \to C) \to (B \to C) \to C}
$$

Déduction naturelle – Example 2

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

Déduction naturelle – Example 2

$$
(i)\,\dfrac{\dfrac{}{A \vdash B \to (A \to C) \to (B \to C) \to C}}{\vdash A \to B \to (A \to C) \to (B \to C) \to C}
$$

Correspondance de Curry-Howard

Déduction naturelle – Example 2

$$
(i)\dfrac{(i)\dfrac{\overline{A,\ B \vdash (A \to C) \to (B \to C) \to C}}{A \vdash B \to (A \to C) \to (B \to C) \to C}}{\vdash A \to B \to (A \to C) \to (B \to C) \to C}
$$

Correspondance de Curry-Howard

Déduction naturelle – Example 2

$$
\dfrac{\dfrac{\dfrac{\dfrac{}{A,\ B,\ A \to C \vdash (B \to C) \to C}}{A,\ B \vdash (A \to C) \to (B \to C) \to C}(i)}{A \vdash B \to (A \to C) \to (B \to C) \to C}(i)}{\vdash A \to B \to (A \to C) \to (B \to C) \to C}(i)
$$

Déduction naturelle – Example 2

$$
\cfrac{\cfrac{\cfrac{\cfrac{\overline{\Gamma \equiv A,\ B,\ A \to C,\ B \to C \vdash C}}{A,\ B,\ A \to C \vdash (B \to C) \to C}(i)}{A,\ B \vdash (A \to C) \to (B \to C) \to C}(i)}{A \vdash B \to (A \to C) \to (B \to C) \to C}(i)}{\vdash A \to B \to (A \to C) \to (B \to C) \to C}(i)
$$

Correspondance de Curry-Howard

Déduction naturelle – Example 2

$$
(i)\dfrac{(i)\dfrac{(i)\dfrac{(i)\dfrac{(a)\dfrac{\dfrac{}{\Gamma \vdash B \to C} \qquad \dfrac{}{\Gamma \vdash B}}{\Gamma \equiv A,\ B,\ A \to C,\ B \to C \vdash C}}{A,\ B,\ A \to C \vdash (B \to C) \to C}}{A,\ B \vdash (A \to C) \to (B \to C) \to C}}{A \vdash B \to (A \to C) \to (B \to C) \to C}}{\vdash A \to B \to (A \to C) \to (B \to C) \to C}
$$

Correspondance de Curry-Howard

Déduction naturelle – Example 2

$$
(i)\dfrac{(i)\dfrac{(i)\dfrac{(i)\dfrac{(a)\dfrac{(v)\dfrac{B \to C \in \Gamma}{\Gamma \vdash B \to C} \qquad (v)\dfrac{B \in \Gamma}{\Gamma \vdash B}}{\Gamma \equiv A,\ B,\ A \to C,\ B \to C \vdash C}}{A,\ B,\ A \to C \vdash (B \to C) \to C}}{A,\ B \vdash (A \to C) \to (B \to C) \to C}}{A \vdash B \to (A \to C) \to (B \to C) \to C}}{\vdash A \to B \to (A \to C) \to (B \to C) \to C}
$$

Correspondance de Curry-Howard

λ-calculus: trouver un terme du type donné

$$\dfrac{}{\vdash\ ?\ : A \to B \to (A \to C) \to (B \to C) \to C}$$

Correspondance de Curry-Howard

λ-calculus: trouver un terme du type donné

$$
(L)\,\frac{\dfrac{}{x{:}A \vdash \;? \quad : \; B\to(A\to C)\to(B\to C)\to C}}{\vdash \lambda x{:}A.\; ? \quad : \; A\to B\to(A\to C)\to(B\to C)\to C}
$$

Correspondance de Curry-Howard

λ-calculus: trouver un terme du type donné

$$
\dfrac{\dfrac{\overline{x{:}A,\ y{:}B \vdash\ ?\ :\ (A \to C) \to (B \to C) \to C}}{x{:}A \vdash \lambda y{:}B.?\ :\ B{\to}(A{\to}C){\to}(B{\to}C){\to}C}\,(L)}{\vdash \lambda x{:}A.\lambda y{:}B.\ ?\ :\ A{\to}B{\to}(A{\to}C){\to}(B{\to}C){\to}C}\,(L)
$$

Correspondance de Curry-Howard

λ-calculus: trouver un terme du type donné

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

Correspondance de Curry-Howard

λ-calculus: trouver un terme du type donné

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

Correspondance de Curry-Howard

λ-calculus: trouver un terme du type donné

$$
(L)\dfrac{(L)\dfrac{(L)\dfrac{(L)\dfrac{(A)\dfrac{\overline{\Gamma \vdash f{:}A \to C} \qquad \overline{\Gamma \vdash x{:}A}}{\Gamma \equiv x{:}A,\ y{:}B,\ f{:}A \to C,\ g{:}B \to C \vdash (f\ x)\ :\ C}}{x{:}A,\ y{:}B,\ f{:}A \to C \vdash \lambda g{:}B{\to}C.(f\ x)\ :\ (B \to C) \to C}}{x{:}A,\ y{:}B \vdash \lambda f{:}A{\to}C.\lambda g{:}B{\to}C.(f\ x)\ :\ (A \to C) \to (B \to C) \to C}}{x{:}A \vdash \lambda y{:}B.\lambda f{:}A{\to}C.\lambda g{:}B{\to}C.(f\ x)\ :\ B{\to}(A{\to}C){\to}(B{\to}C){\to}C}}{\vdash \lambda x{:}A.\lambda y{:}B.\lambda f{:}A{\to}C.\lambda g{:}B{\to}C.(f\ x)\ :\ A{\to}B{\to}(A{\to}C){\to}(B{\to}C){\to}C}
$$

Correspondance de Curry-Howard

λ-calculus: trouver un terme du type donné

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

$\lambda x{:}A.\lambda y{:}B.\lambda f{:}A{\to}C.\lambda g{:}B{\to}C.(f\ x)$

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

$A{\to}B{\to}(A{\to}C){\to}(B{\to}C){\to}C$

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

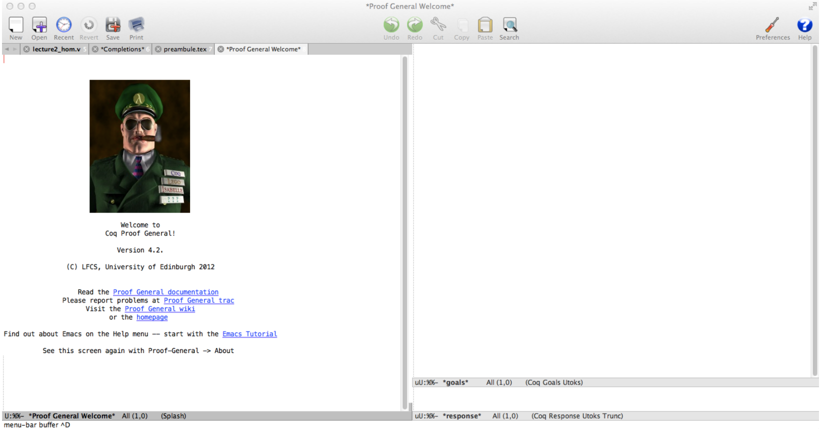
Rocq en Pratique

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 Visal Studio Code . . .

Introduction.v

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

```
(* From: HPCS 2017 - Tutorial                            *)
(* http://frederic.loulergue.eu/hpcs2017                 *)
(* Frederic Loulergue                                    *)
(*   School of Informatics Computing and Cyber Systems *)
(*   Northern Arizona University                         *)

Section Examples.

  Parameters A B C : Prop.

  Lemma proof1:
    A->B->(A->C)->(B->C)->C.
  Proof.
    intro HA.
    intro HB.
    intro HAC.
    Show Proof.
    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).

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

End Examples.
```

Rocq Goals

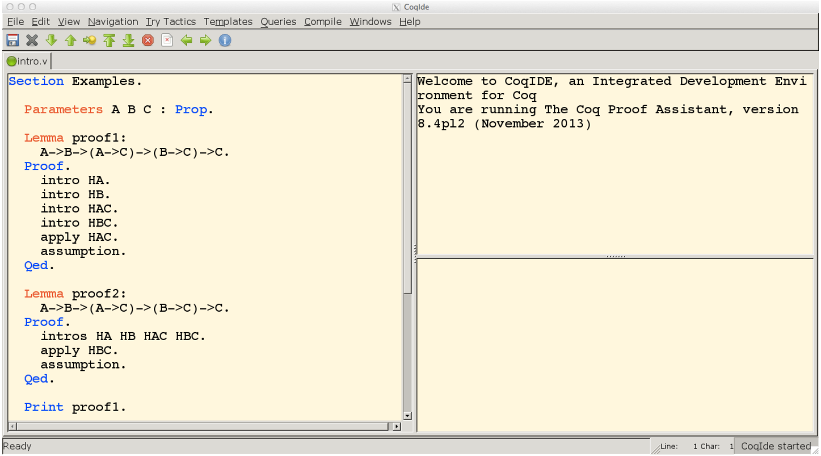
Proof

```
Not in proof mode
```

Messages

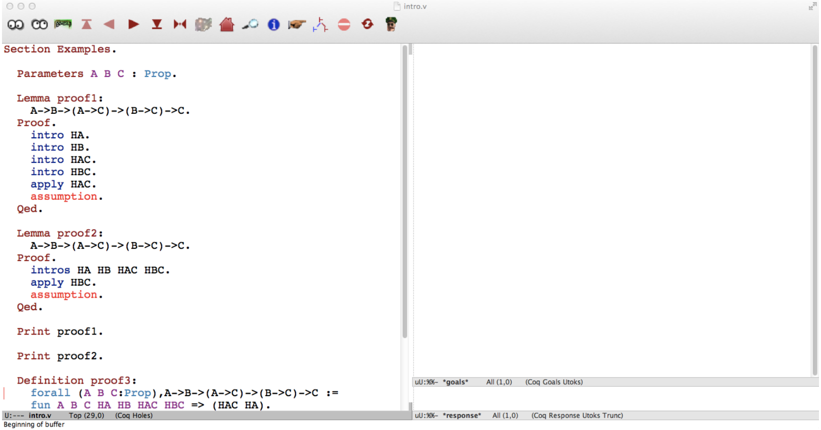
Exemples précédents dans Rocq

. . . ou RocqIDE



Exemples précédents dans Rocq

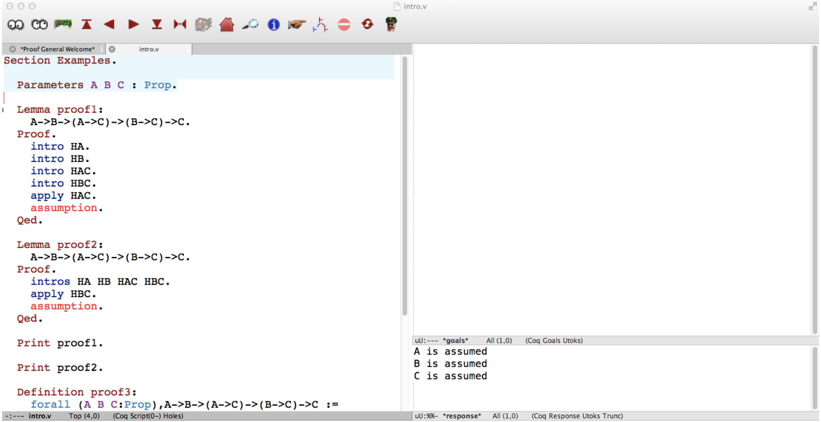
On ouvre le fichier `Introduction.v`[1]:



[1]disponible sur https://jolanphilippe.github.io/course/docs/25-init-recherche/Introduction.v

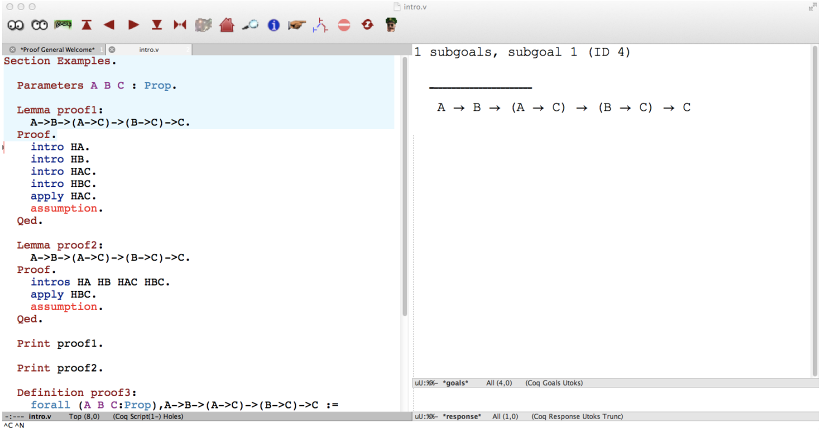
Exemples précédents dans Rocq

Nous commençons à alimenter Rocq avec des commandes :



Exemples précédents dans Rocq

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



Exemples précédents dans Rocq

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

```
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 (9,0)     (Coq Script(1-) Holes)
^C ^N
```

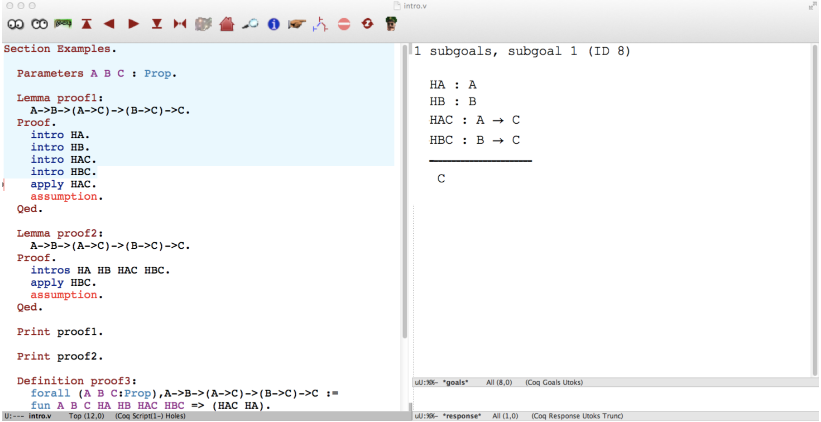
```
1 subgoals, subgoal 1 (ID 5)

  HA : A
  ============================
   B → (A → C) → (B → C) → C

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

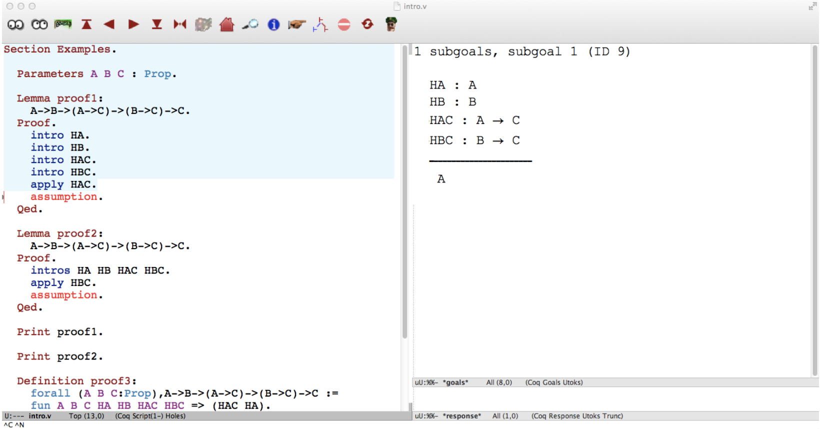
Exemples précédents dans Rocq

Le contexte est maintenant similaire à Γ :



Exemples précédents dans Rocq

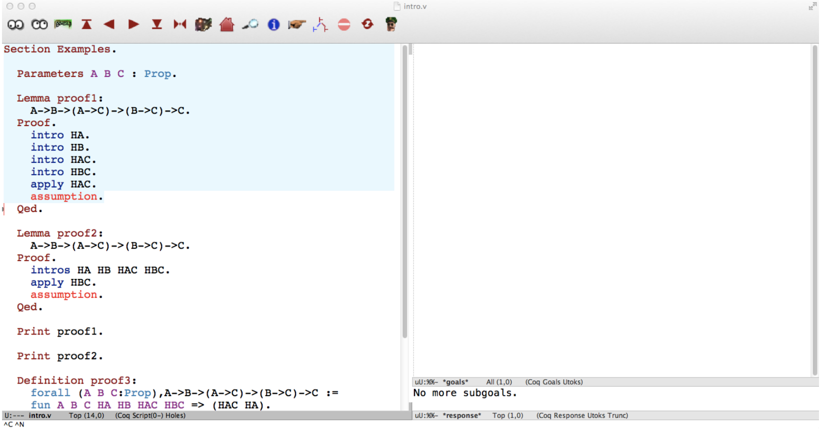
Nous appliquons la règle (a) en nommant la partie implication :



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

Exemples précédents dans Rocq

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



"No more subgoals" $\equiv$ preuve terminée $\equiv$ λ-terme construit

Exemples précédents dans Rocq

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

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 (15,0) (Coq Script(0-) Holes)

^C ^N

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

```
proof1 is defined
```

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

Exemples précédents dans Rocq

Deuxième version, nous faisons plusieurs `intro` :

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 (20,0)    (Coq Script(1-) Holes)
```

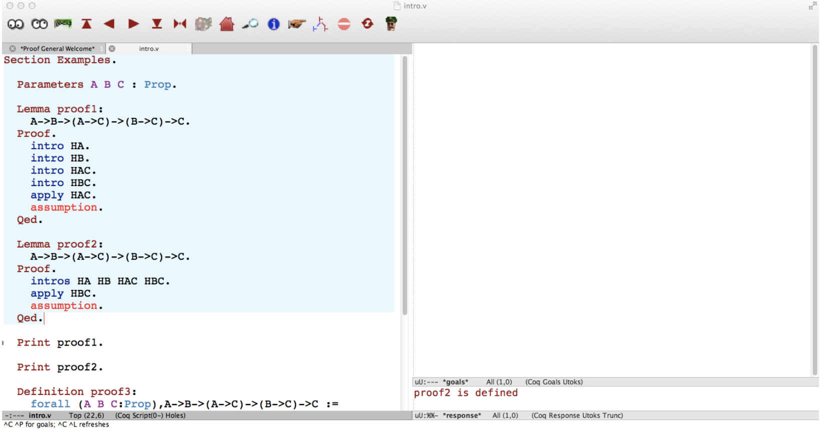
```
1 subgoals, subgoal 1 (ID 19)

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

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

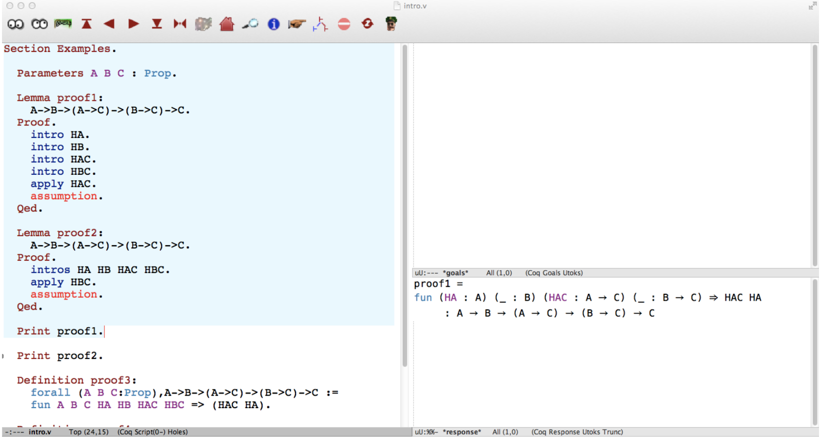
Exemples précédents dans Rocq

et `apply HBC` au lieu de `apply HAC` :



Exemples précédents dans Rocq

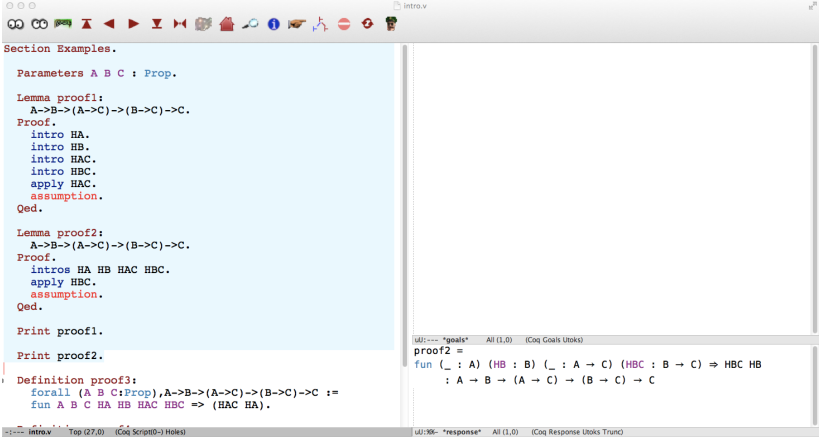
`Print t.` affiche le terme t :



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

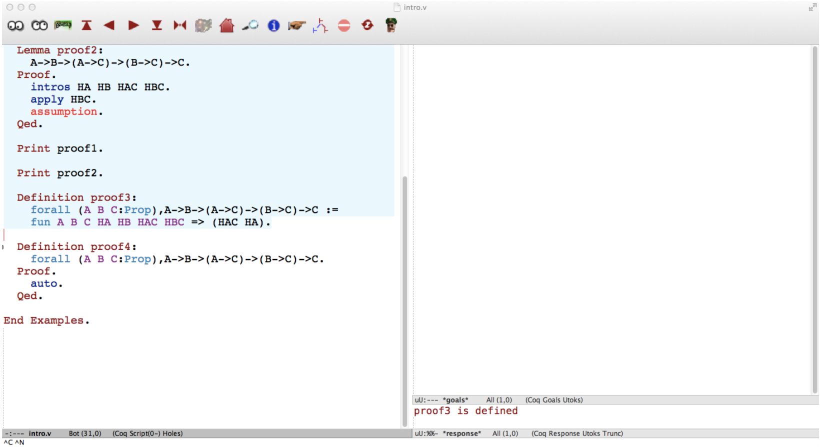
Exemples précédents dans Rocq

Le λ-terme pour la seconde preuve est :



Exemples précédents dans Rocq

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



Exemples précédents dans Rocq

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