MDV : cours 1

Systèmes de déduction

Initiation à Coq

Qu'est ce qu'un assistant de preuve ?

Un système informatique pour

- écrire des énoncés logiques (spécifier),
- puis les prouver (vérifier).

L'outil informatique apporte

- de la rigueur,
- de l'automatisation.

Un tel système repose sur une logique formelle

Logique formelle

Un logique formelle comprend

- un langage pour écrire des formules,
- une interprétation (sémantique) pour donner du sens aux formules (définir les formules valides),
- un système de déduction pour construire des preuves de formules (définir les formules prouvables).

Méta-théorie

- contradictoire : toute formule est-elle valide ? (sinon la logique est dite cohérente)
- correction (*soundness*) : toute formule prouvable est elle valide ?
- complétude : toute formule valide est elle prouvable ?
- décidabilité : existe-t-il un algorithme pour décider si une formule est valide ou non ?

Plan

Plan

Plan

Plan

Logique propositionnelle

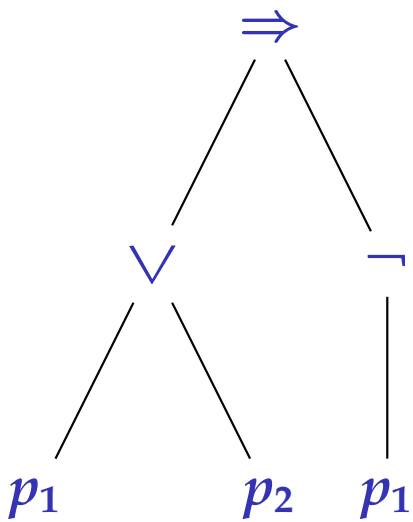
- Syntaxe
- Sémantique
- Systèmes de déduction
- Exercices en Coq

Syntaxe

$$\phi \quad ::= \quad \bot \quad | \quad x \quad | \quad \neg\phi' \quad | \quad \phi' \vee \phi'' \quad | \quad \phi' \wedge \phi'' \quad | \quad \phi' \Rightarrow \phi''$$

avec x une variable propositionnelle choisie dans un ensemble (dénombrable) $\{p_1, p_2, \ldots\}$.

Exemple :



$$(p_1 \vee p_2) \Rightarrow \neg p_1$$

Sémantique

Domaine d'interprétation :

$$\mathbb{B} = \{\mathbf{t}, \mathbf{f}\}$$

Interprétation des connecteurs : on associe à chaque opérateur élémentaire o, une interprétation $[\![o]\!] \in \mathbb{B} \times \mathbb{B} \longrightarrow \mathbb{B}$ (ou $\mathbb{B} \longrightarrow \mathbb{B}$) décrite par une table de vérité.

x	$[\![\neg]\!]\, x$
$\mathbf{t}$	$\mathbf{f}$
$\mathbf{f}$	$\mathbf{t}$

x	y	$x\,[\![\wedge]\!]\,y$	$x\,[\![\vee]\!]\,y$	$x\,[\![\Rightarrow]\!]\,y$
$\mathbf{t}$	$\mathbf{t}$	$\mathbf{t}$	$\mathbf{t}$	$\mathbf{t}$
$\mathbf{t}$	$\mathbf{f}$	$\mathbf{f}$	$\mathbf{t}$	$\mathbf{f}$
$\mathbf{f}$	$\mathbf{t}$	$\mathbf{f}$	$\mathbf{t}$	$\mathbf{t}$
$\mathbf{f}$	$\mathbf{f}$	$\mathbf{f}$	$\mathbf{f}$	$\mathbf{t}$

Sémantique (2)

Interprétation des formules : dépend d'une valuation $\rho \in P \to \mathbb{B}$

$$[\![\phi]\!] : (P \to \mathbb{B}) \to \mathbb{B}$$

$$\begin{aligned}
[\![\bot]\!]_\rho &= \mathfrak{f} \\
[\![x]\!]_\rho &= \rho(x) \\
[\![\neg\phi]\!]_\rho &= [\![\neg]\!][\![\phi]\!]_\rho \\
[\![\phi_1\, o\, \phi]\!]_\rho &= [\![\phi_1]\!]_\rho [\![o]\!] [\![\phi_2]\!]_\rho, \quad o \in \{\vee, \wedge, \Rightarrow\}
\end{aligned}$$

Exemple : si $\rho = [p_1 \mapsto \mathfrak{t}, p_2 \mapsto \mathfrak{f}]$

$$\begin{aligned}
[\![(p_1 \vee p_2) \Rightarrow \neg p_1]\!]_\rho &= (\mathfrak{t}[\![\vee]\!]\mathfrak{f})[\![\Rightarrow]\!]([\![\neg]\!]\mathfrak{t}) \\
&= \mathfrak{t}[\![\Rightarrow]\!]\mathfrak{f} \\
&= \mathfrak{f}
\end{aligned}$$

Sémantique (3)

Quelques définitions

- Une formule ϕ est satisfaite par une valuation $\rho \in P \to \mathbb{B}$ si

$$[\![\phi]\!]_\rho = t$$

- Une formule ϕ est valide si elle est satisfaite par toute valuation :

$$\forall \rho \in P \to \mathbb{B},\ [\![\phi]\!]_\rho = t$$

Nous le notons :

$$\models \phi$$

- Une formule ϕ est conséquence logique d'un ensemble de formules Γ si elle est satisfaite par toutes les valuations qui satisfont les formules de Γ :

$$\forall \rho \in P \to \mathbb{B},\ (\forall \phi' \in \Gamma,\ [\![\phi']\!]_\rho = t) \Rightarrow [\![\phi]\!]_\rho = t$$

Nous le notons :

$$\Gamma \models \phi$$

Système de déduction

Un système de déduction comprend

- des axiomes (feuilles de l'arbre de preuve),
- des règles de déduction (feuilles ou embranchements de l'arbre de preuve).

Plusieurs systèmes pour la logique propositionnelle

- *à la Hilbert* (avec des variantes)
- déduction naturelle
- calcul des séquents

Preuve à la Hilbert

- 12 familles d'axiomes (pour toutes formules A, B et C)

$$A \Rightarrow (B \Rightarrow A) \qquad (A \Rightarrow (B \Rightarrow C)) \Rightarrow ((A \Rightarrow B) \Rightarrow (A \Rightarrow C))$$

$$\bot \Rightarrow A \qquad A \Rightarrow (\neg A \Rightarrow \bot) \qquad (A \Rightarrow \bot) \Rightarrow \neg A$$

$$(A \wedge B) \Rightarrow A \qquad (A \wedge B) \Rightarrow B \qquad A \Rightarrow (B \Rightarrow (A \wedge B))$$

$$A \vee (\neg A) \qquad A \Rightarrow (A \vee B)) \qquad B \Rightarrow (A \vee B))$$

$$(A \vee B) \Rightarrow ((A \Rightarrow C) \Rightarrow ((B \Rightarrow C) \Rightarrow C))$$

- 2 règles de déduction

$$\frac{A \Rightarrow B \quad A}{B}\, MP \qquad (\textit{modus ponens})$$

$$\frac{}{A}\, Ax \quad \text{avec } A \text{ un axiome} \qquad (\textit{axiome contextuel})$$

Preuve à la Hilbert (2)

Une preuve (à la Hilbert) d'une formule ϕ est un arbre

- dont la racine est étiqueté par ϕ,
- les sous-arbres sont des preuves de formules $\phi_1, \ldots, \phi_n$ tels que

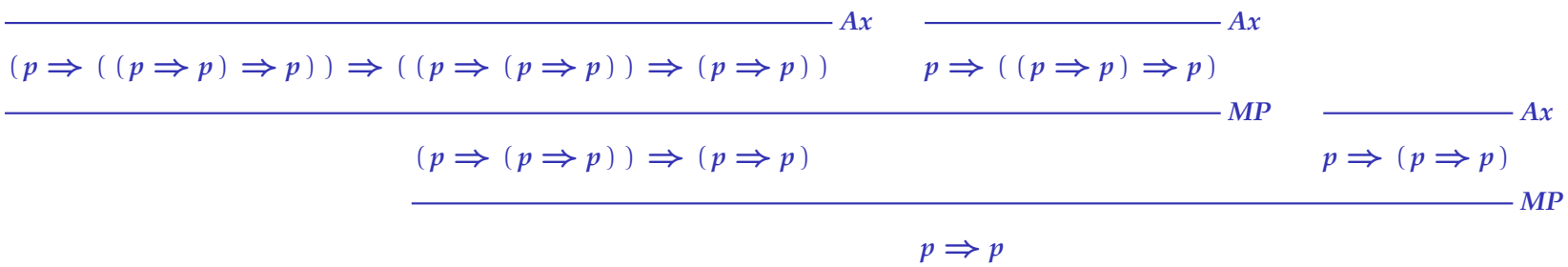
$$\frac{\phi_1 \ \ldots \ \phi_n}{\phi}$$

est une règle de déduction.

Quand une telle preuve existe, nous notons

$$\vdash_H \phi$$

Preuve à la Hilbert : exemple



Un peu fastidieux...

Preuve à la Hilbert : propriétés

Théorème de correction :

$$\vdash_H \phi \text{ implique } \models \phi$$

Théorème de complétude :

$$\models \phi \text{ implique } \vdash_H \phi$$

Déduction naturelle

On veut pouvoir prouver $A \Rightarrow B$ en admettant A et en prouvant B.
Il faut alors manipuler un contexte dans les règles de déduction : notion de séquent.

$$\Gamma \vdash \phi$$

Γ représente un ensemble de formules $\{\phi_1, \ldots, \phi_p\}$.

Remarques

- $\Gamma \cup \{\phi\}$ est noté Γ, ϕ.
- En général, un séquent est de la forme $\phi_1, \ldots, \phi_p \vdash \psi_1, \ldots, \psi_n$, mais en déduction naturelle $n = 1$.

Déduction naturelle

Une preuve (en déduction naturelle) d'un séquent $\Gamma \vdash \phi$ est un arbre

- dont la racine est étiquette par $\Gamma \vdash \phi$,
- les sous-arbres sont des preuves des séquents $\Gamma_1 \vdash \phi_1, \ldots, \Gamma_n \vdash \phi_n$ tels que

$$\frac{\Gamma_1 \vdash \phi_1 \ \ldots \ \Gamma_n \vdash \phi_n}{\Gamma \vdash \phi}$$

est une instance de l'une des règles de déduction suivantes

Déduction naturelle

$$\frac{}{\Gamma, A \vdash A}\ \text{Ax}$$

$$\frac{}{\Gamma \vdash A \vee (\neg A)}\ \text{Tiers Exclu}$$

$$\frac{\Gamma \vdash \bot}{\Gamma \vdash A}\ \text{élim}_{\bot}$$

$$\frac{\Gamma, A \vdash \bot}{\Gamma \vdash \neg A}\ \text{intro}_{\neg}$$

$$\frac{\Gamma \vdash \neg A \quad \Gamma \vdash A}{\Gamma \vdash \bot}\ \text{élim}_{\neg}$$

$$\frac{\Gamma, A \vdash B}{\Gamma \vdash A \Rightarrow B}\ \text{intro}_{\Rightarrow}$$

$$\frac{\Gamma \vdash A \quad \Gamma \vdash A \Rightarrow B}{\Gamma \vdash B}\ \text{élim}_{\Rightarrow}$$

$$\frac{\Gamma \vdash A \quad \Gamma \vdash B}{\Gamma \vdash A \wedge B}\ \text{intro}_{\wedge}$$

$$\frac{\Gamma \vdash A \wedge B}{\Gamma \vdash A}\ \text{élim}^1_{\wedge}$$

$$\frac{\Gamma \vdash A \wedge B}{\Gamma \vdash B}\ \text{élim}^2_{\wedge}$$

$$\frac{\Gamma \vdash A}{\Gamma \vdash A \vee B}\ \text{intro}^1_{\vee}$$

$$\frac{\Gamma \vdash B}{\Gamma \vdash A \vee B}\ \text{intro}^2_{\vee}$$

$$\frac{\Gamma \vdash A \vee B \quad \Gamma, A \vdash C \quad \Gamma, B \vdash C}{\Gamma \vdash C}\ \text{élim}_{\vee}$$

Déduction naturelle : exemple

$$\dfrac{\dfrac{\dfrac{}{A \wedge (A \Rightarrow B) \vdash A \wedge (A \Rightarrow B)}\,Ax}{A \wedge (A \Rightarrow B) \vdash A \Rightarrow B}\,i_{\wedge} \qquad \dfrac{\dfrac{}{A \wedge (A \Rightarrow B) \vdash A \wedge (A \Rightarrow B)}\,Ax}{A \wedge (A \Rightarrow B) \vdash A}\,i_{\wedge}}{\dfrac{A \wedge (A \Rightarrow B) \vdash B}{\vdash (A \wedge (A \Rightarrow B)) \Rightarrow B}\,i_{\Rightarrow}}\,e_{\Rightarrow}$$

Exercice

Prouver en déduction naturelle les axiomes du système de Hilbert.

Déduction naturelle : propriétés

Théorème de correction :

$$\Gamma \vdash \phi \text{ prouvable implique } \Gamma \models \phi$$

Théorème de complétude :

$$\models \phi \text{ implique } \Gamma \vdash \phi \text{ prouvable}$$

Déduction naturelle : preuve constructive

Une preuve est constructive si elle n'utilise pas la règle du Tiers Exclu.

$$\frac{}{\Gamma \vdash A \vee (\neg A)}\ \text{Tiers Exclu}$$

La déduction naturelle constructive est correcte (mais pas complète) pour la logique propositionnelle.

Exemple de tautologie non démontrable en déduction naturelle constructive : formule de Peirce

$$(((A \Rightarrow B) \Rightarrow A) \Rightarrow A)$$

Coq : un assistant de preuve en logique constructive

Nous allons découvrir les commandes de base de Coq, en nous restreignant à la logique propositionnelle.

Nous nous interdirons d'utiliser l'automatisation dans cette séance.

Il faut faire ses gammes !

Notations

Notation :

$\Rightarrow$ est noté $\rightarrow$ (->)

Parenthésage :

`A→(B→C)` est simplement noté `A→B→C`.

But courant :

un séquent $A_1, \ldots, A_n \vdash A$ s'affiche

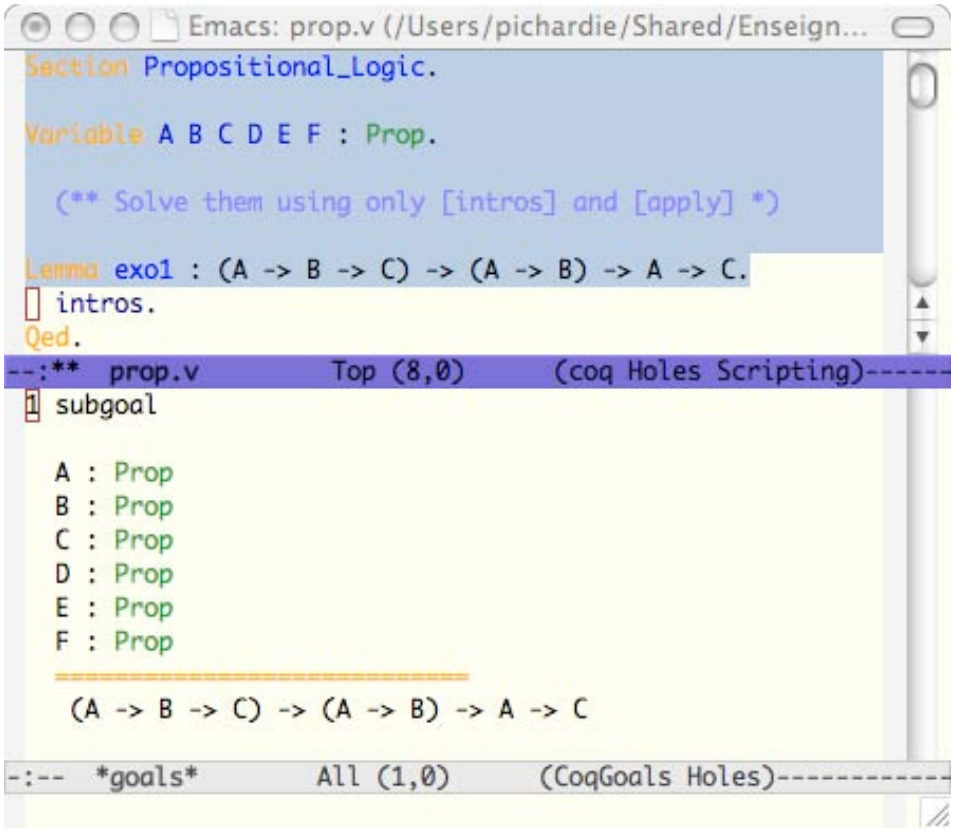
```
H1 : A1
     ...
Hn : An
============================
 A
```

L'environnement de travail

Deux fenêtres : le fichier courant + les réponses de Coq.

L'évaluation du fichier se fait linéairement.

Utilisez les raccourcis `Ctrl`+flèches pour faire avancer/reculer la zone d'évaluation.



Règle intro$_\Rightarrow$

$$\frac{\Gamma, A \vdash B}{\Gamma \vdash A \Rightarrow B}\ \text{intro}_\Rightarrow$$

```
============================
A → B
```

```
H : A
============================
B
```

Commande : **intros** H.

Remarques

- on peut enchainer plusieurs introductions : **intros** H1 H2.
- on peut laisser le système nommer les hypothèses : **intros**.

Règle élim$_\Rightarrow$

$$\frac{\Gamma \vdash A \Rightarrow B \qquad \Gamma \vdash A}{\Gamma \vdash B}\ \text{élim}_\Rightarrow$$

```
  H : A → B
============================
  B
```

```
H : A → B
============================
 A
```

Commande : `apply H.`

Remarques

- contrairement à élim$_\Rightarrow$, il y un seul sous arbre de preuve : il faut avoir déjà une preuve de $A \Rightarrow B$ dans son contexte
- pour forcer Coq à générer un sous-but pour $A \Rightarrow B$, on peut taper `assert (A→B).`

Règle Ax

$$\frac{}{\Gamma, A \vdash A}\ \text{Ax}$$

```
H : A
============================
 A
```

Commande : `apply H.`

Remarque

- on peut éviter de nommer l'hypothèse en utilisant la commande `assumption`.

Règle intro$_\wedge$

$$\frac{\Gamma \vdash A \qquad \Gamma \vdash B}{\Gamma \vdash A \wedge B}\ \text{intro}_\wedge$$

```
H : A
H0 : B
============================
 A ∧ B
```

```
H : A
H0 : B
============================
 A
```

```
H : A
H0 : B
============================
 B
```

Commande : **`split`**.

Règle élim$_\wedge$

$$\frac{\Gamma \vdash A \wedge B}{\Gamma \vdash A}\ \text{élim}^1_\wedge \qquad \frac{\Gamma \vdash A \wedge B}{\Gamma \vdash B}\ \text{élim}^2_\wedge$$

```
H : A ∧ B
============================
 C
```

```
H : A
H0 : B
============================
 C
```

Commande : `destruct H.`

Remarque

- on peut nommer les hypothèses introduites avec la syntaxe
  `destruct H as [H H0].`

Règle intro$_\vee$

$$\frac{\Gamma \vdash A}{\Gamma \vdash A \vee B}\ \text{intro}^1_\vee \qquad \frac{\Gamma \vdash B}{\Gamma \vdash A \vee B}\ \text{intro}^2_\vee$$

```
============================
A ∨ B
```

```
============================
A
```

Commande : **`left`**.

Remarque

- intro$^2_\vee$ correspond à la commande **`right`**.

Règle élim$_\vee$

$$\frac{\Gamma \vdash A \vee B \qquad \Gamma, A \vdash C \qquad \Gamma, B \vdash C}{\Gamma \vdash C}\ \text{élim}_\vee$$

```
H : A ∨ B
============================
 C
```

```
H : A
============================
 C
```

```
H : B
============================
 C
```

Commande : **`destruct H.`**

Règle élim$_\bot$

$$\frac{\Gamma \vdash \bot}{\Gamma \vdash A}\ \text{élim}^{1}_{\bot}$$

```
H : False
============================
 A
```

Commande : **elim** H.

Remarque

- **elim** H peut aussi s'utiliser quand H:A1 → ... → An → False (mais il faut alors décharger n sous-buts)

Règle intro¬

$$\frac{\Gamma, A \vdash \bot}{\Gamma \vdash \neg A}\ \text{intro}_\neg$$

```
============================          H : A
 ¬ A                                  ============================
                                       False
```

Commande : **intros** H.

Remarque

- ¬ A est en fait du sucre syntaxique pour A → False.
- **intros**. ne marche pas ici.

Règle élim¬

$$\frac{\Gamma \vdash A \qquad \Gamma \vdash \neg A}{\Gamma \vdash \bot}\ \text{élim}_\neg$$

```
============================
 ¬ A
```

```
============================
 B
```

```
============================
 A
```

Commande : **absurd** `A.`

Remarque

- la commande est utile même quand `B` n'est pas égal à `False`.

Règle du tiers exclu

$$\frac{}{\Gamma \vdash A \vee \neg A}\ \text{Tiers Exclu}$$

Il faut ajouter un axiome (*excluded middle*) !

```
Variable em : ∀ P:Prop, P ∨ ¬P.
```

```
H : A
============================
 C
```

```
============================
 C
```

```
H : ¬ A
============================
 C
```

Commande : `destruct (em A).`

Exercices

Prouver les lemmes du fichier `prop.v` en remplaçant à chaque fois la tactique d'automatisation **`tauto`** présente.

Plan

Logique du premier ordre

- Syntaxe
- Sémantique
- Systèmes de déduction
- Exercices en Coq

Syntaxe

Définition des termes

Soit $\mathcal{F}$ un ensemble de symboles chacun munis d'une arité (nombre de paramètres). Soit $\mathcal{X}$ un ensemble de variables.
L'ensemble des termes $\mathcal{T}(\mathcal{F},\mathcal{X})$ est défini par

$$t \quad ::= \quad x \quad | \quad f(t_1,\ldots,t_n)$$

avec $x \in \mathcal{X}, f \in \mathcal{F}$ un symbole de fonction d'arité n et $t_1,\ldots,t_n$ des termes de $\mathcal{T}(\mathcal{F},\mathcal{X})$

Exemple : Soient $\mathcal{F} = \{f : 1,\ g : 2,\ a : 0\}$ et $\mathcal{X} = \{x,y,z\}$.

$f(x) \quad a \quad z \quad f(g(a(x),f(a))) \quad g(x,x)$ sont des termes de $\mathcal{T}(\mathcal{F},\mathcal{X})$.

Syntaxe

Définition des formules

Soit $\mathcal{P}$ un ensemble de symboles de prédicats munis d'une arité. L'ensemble des formules sur $\mathcal{F}$, $\mathcal{X}$ et $\mathcal{P}$ est défini par

$$\begin{aligned}\phi \quad ::= \quad & \bot \mid p(t_1, \ldots, t_n) \\ & \mid \neg\phi' \mid \phi' \vee \phi'' \mid \phi' \wedge \phi'' \mid \phi' \Rightarrow \phi'' \\ & \mid \exists x\, \phi' \mid \forall x\, \phi'\end{aligned}$$

avec $x \in \mathcal{X}$, $t_1, \ldots, t_n \in \mathcal{T}(\mathcal{F}, \mathcal{X})$ et $p \in \mathcal{P}$ un symbole de prédicat d'arité n.

Exemple : si $P = \{p : 1,\ q : 2, \leq : 2\}$, les expressions suivantes sont des formules

$$p(f(a)) \qquad q(g(f(a), x), y) \qquad \forall x\, \exists y\, f(y) \leq x$$

Sémantique

Une interprétation I de P et F comprend :

- un ensemble non vide D_I (*domaine*),
- une interprétation $[\![f]\!]_I \in D_I^n \to D_I$ pour chaque symbole $f \in \mathcal{F}$ d'arité n,
- une interprétation $[\![p]\!]_I \in D_I^n \to \mathbb{B}$ pour chaque symbole de prédicat $p \in P$ d'arité n.

Une valuation sur un ensemble de variables $\mathcal{X}$ est une fonction $\mathcal{X} \to D_I$ avec D_I un domaine d'interprétation.

Notation : pour $\rho \in \mathcal{X} \to D_I$, $x \in \mathcal{X}$ et $d \in D_I$, la valuation $\rho[x \mapsto d]$ est définie par

$$\rho[x \mapsto d](y) = \begin{cases} d & \text{si } x = y \\ \rho(y) & \text{si } x \neq y \end{cases}$$

Sémantique (2)

Étant donné une interprétation I de P et F, l'interprétation $[\![t]\!]_I \in (\mathcal{X} \to D_I) \to D_I$ d'un terme t est définie par

$$\begin{aligned} [\![x]\!]_I \rho &= \rho(x) \\ [\![f(t_1, \ldots, t_n)]\!]_I \rho &= [\![f]\!]_I \left([\![t_1]\!]_I \rho, \ldots, [\![t_n]\!]_I \rho\right) \end{aligned}$$

L'interprétation $[\![f]\!]_I \in (\mathcal{X} \to D_I) \to \mathbb{B}$ d'une formule f est définie par

$$\begin{aligned} [\![p(t_1, \ldots, t_n)]\!]_I \rho &= [\![p]\!]_I \left([\![t_1]\!]_I \rho, \ldots, [\![t_n]\!]_I \rho\right) \\ [\![\bot]\!]_I \rho &= \mathfrak{f} \\ [\![\neg \phi]\!]_I \rho &= \ldots \\ &\ldots \\ [\![\forall x\ \phi]\!]_I \rho &= \begin{cases} \mathfrak{t} & \text{si pour } d \in D_I,\ [\![\phi]\!]_I \rho[x \mapsto d] = \mathfrak{t} \\ \mathfrak{f} & \text{sinon} \end{cases} \\ [\![\exists x\ \phi]\!]_I \rho &= \begin{cases} \mathfrak{t} & \text{s'il existe } d \in D_I \text{ tel que } [\![\phi]\!]_I \rho[x \mapsto d] = \mathfrak{t} \\ \mathfrak{f} & \text{sinon} \end{cases} \end{aligned}$$

Sémantique (3)

Quelques définitions

- Une formule ϕ est valide dans une interprétation I ssi :

$$\forall \rho \in \mathcal{X} \to \mathbb{B},\ [\![\phi]\!]_I \rho = t$$

- Une formule ϕ est satisfiable ssi elle est valide dans au moins une interprétation I.
- Une formule ϕ est universellement valide ssi elle est valide dans toute interprétation I.
- Une interprétation I est un modèle d'un ensemble de formules Γ (appelé théorie) ssi toutes les formules de Γ sont valides dans I.

Exercice

$$\mathcal{F} = \{c : 0,\ f : 1,\ \circ : 2,\ \bullet : 2\} \qquad \mathcal{X} = \{x,\ y\} \qquad P = \{z : 1,\ \bowtie : 2\}$$

$$\Phi = z(c) \wedge \forall x\, \forall y\ \ (f(x \circ y) \bowtie (f(x) \bullet f(y)))$$

- Montrer que Φ est valide dans plusieurs interprétations sur $\mathbb{R}$
- Montrer que Φ n'est pas universellement valide

Déduction naturelle en logique du premier ordre

On rajoute 4 règles de déduction

$$\frac{\Gamma \vdash A}{\Gamma \vdash \forall x\, A}\ \text{intro}_\forall \qquad \frac{\Gamma \vdash \forall x\, A}{\Gamma \vdash A[t/x]}\ \text{élim}_\forall$$

$$\frac{\Gamma \vdash A[t/x]}{\Gamma \vdash \exists x\, A}\ \text{intro}_\exists \qquad \frac{\Gamma \vdash \exists x\, A \qquad \Gamma, A \vdash B}{\Gamma \vdash B}\ \text{élim}_\exists$$

Pour la règle $\text{intro}_\forall$, x ne doit pas être libre dans A. Pour la règle $\text{élim}_\exists$, x ne doit pas être libre dans B et Γ.

$A[t/x]$ représente la substitution de la variable x par le terme t.

Exercice : définir formellement les notions de variable libre et de substitution (attention aux captures de variable).

Propriétés

Théorème de correction :

Si $\Gamma \vdash \phi$ est prouvable alors ϕ est valide dans tous les modèles de Γ.

Théorème de complétude :

Si ϕ est valide dans tous les modèles de Γ alors $\Gamma \vdash \phi$ est prouvable.

Si on retire la règle du tiers exclu le système de déduction est correct mais pas complet.

Lecture à la maison : lire introduction, chapitre 1 et 4 du poly[1] de Gilles Dowek pour en apprendre plus sur les propriétés des preuves constructives.

[1] `http://www.lix.polytechnique.fr/~dowek/Cours/proof.ps.gz`

Décidabilité, vérification de preuve

Logique propositionnelle :

- $\models^? \phi$ est décidable,
- on peut vérifier $\vdash \phi$ (et donc $\models \phi$) en vérifiant la cohérence de l'arbre de preuve,
- on peut vérifier $\models \phi$ en faisant une table de vérité.

Logique du premier ordre :

- $\models^? \phi$ est indécidable,
- on peut vérifier $\vdash \phi$ (et donc $\models \phi$) en vérifiant la cohérence de l'arbre de preuve,
- on ne peut pas vérifier (mécaniquement) $\models \phi$ en se basant uniquement sur la notion sémantique (circularité)

Logique du premier ordre en Coq

Notations :

- $\forall$ se note `forall`,
- $\exists$ se note `exists`

Règle intro$_\forall$

$$\frac{\Gamma \vdash A}{\Gamma \vdash \forall x\, A}\ \text{intro}_\forall$$

```
============================
 ∀ x:A, P x
```

```
a : A
============================
 P a
```

Commande : **intros** a.

Règle élim$_\forall$

$$\frac{\Gamma \vdash \forall x\, A}{\Gamma \vdash A[t/x]}\ \text{élim}_\forall$$

```
H : ∀ x:A, P x
============================
 P a
```

Commande : **`apply H.`**

Remarque

- si `H` est de la forme `∀ x y, Q y → P x`, il faut aider un peu Coq en donnant les substitutions à utiliser : **`apply`** `(H a b)` (ici `x=a` et `y=b`).

Règle intro$_\exists$

$$\frac{\Gamma \vdash A[t/x]}{\Gamma \vdash \exists x\, A}\ \text{intro}_\exists$$

```
a : A
============================
 ∃ x:A, P x
```

```
a : A
============================
 P a
```

Commande : **`exists`** `a.`

Règle élim$_\exists$

$$\frac{\Gamma \vdash \exists x\, A \qquad \Gamma, A \vdash B}{\Gamma \vdash B}\ \text{élim}_\exists$$

```
H : ∃ x:A, P x
============================
 B
```

```
a : A
H : P a
============================
 B
```

Commande : **`destruct`** `H.`

Remarque

- **`destruct`** `H as [a H].` permet de nommer les objets générés.

Exercice

Prouver les lemmes du fichier `pred.v` en remplaçant à chaque fois la tactique d'automatisation `firstorder` présente.

La prochaine fois...

Nous aborderons la notion de définition inductive...

... que nous avons déjà largement utilisée tout au long de ce cours !