

Correction des exercices de TD

Table des matières

1	TD/TP1 : Preuves en logique du premier ordre	1
2	TD/TP2 : Preuves en logique équationnelle	4
3	TD/TP3 : Types inductifs	7

1 TD/TP1 : Preuves en logique du premier ordre

Exercice 1 (Logique propositionnelle)

Démontrer les propositions suivantes dans LJ et LK :

1. $A \Rightarrow B \Rightarrow A$
2. $(A \Rightarrow B \Rightarrow C) \Rightarrow (A \Rightarrow B) \Rightarrow A \Rightarrow C$
3. $A \wedge B \Rightarrow B$
4. $B \Rightarrow A \vee B$
5. $(A \vee B) \Rightarrow (A \Rightarrow C) \Rightarrow (B \Rightarrow C) \Rightarrow C$
6. $A \Rightarrow \perp \Rightarrow \neg A$
7. $\perp \Rightarrow A$
8. $(A \Leftrightarrow B) \Rightarrow A \Rightarrow B$
9. $(A \Leftrightarrow B) \Rightarrow B \Rightarrow A$
10. $(A \Rightarrow B) \Rightarrow (B \Rightarrow A) \Rightarrow (A \Leftrightarrow B)$

1.	$\frac{\frac{\overline{A, B \vdash A} \text{ ax}}{A \vdash B \Rightarrow A} \Rightarrow_{\text{right}}}{\vdash A \Rightarrow B \Rightarrow A} \Rightarrow_{\text{right}}$	Ici, la preuve en LJ est la même que la preuve en LK.
2.	$\frac{\frac{\frac{\overline{B, A \vdash B} \text{ ax}}{A \Rightarrow B, A \vdash B} \Rightarrow_{\text{left}} \quad \frac{\overline{A \vdash A} \text{ ax}}{B \Rightarrow C, A \Rightarrow B, A \vdash C} \Rightarrow_{\text{left}} \quad \frac{\overline{C, A \Rightarrow B, A \vdash C} \text{ ax}}{A \Rightarrow B, A \vdash A} \Rightarrow_{\text{left}}}{\vdash (A \Rightarrow B \Rightarrow C) \Rightarrow (A \Rightarrow B) \Rightarrow A \Rightarrow C} \Rightarrow_{\text{right}} \times 3}$ <p>Preuve dans LJ</p>	$\frac{\frac{\frac{\overline{B, A \vdash B, C} \text{ ax}}{A \Rightarrow B, A \vdash B, C} \Rightarrow_{\text{left}} \quad \frac{\overline{A \vdash A, B, C} \text{ ax}}{B \Rightarrow C, A \Rightarrow B, A \vdash C} \Rightarrow_{\text{left}} \quad \frac{\overline{C, A \Rightarrow B, A \vdash C} \text{ ax}}{A \Rightarrow B, A \vdash C, A} \Rightarrow_{\text{left}}}{\vdash (A \Rightarrow B \Rightarrow C) \Rightarrow (A \Rightarrow B) \Rightarrow A \Rightarrow C} \Rightarrow_{\text{right}} \times 3}$ <p>Preuve dans LK</p>
3.	$\frac{\frac{\overline{A, B \vdash B} \text{ ax}}{A \wedge B \vdash B} \wedge_{\text{left}}}{\vdash A \wedge B \Rightarrow B} \Rightarrow_{\text{right}}$	Ici, la preuve en LJ est la même que la preuve en LK.
4.	$\frac{\frac{\overline{B \vdash B} \text{ ax}}{B \vdash A \vee B} \vee_{\text{right}}}{\vdash B \Rightarrow A \vee B} \Rightarrow_{\text{right}}$ <p>Preuve dans LJ</p>	$\frac{\frac{\overline{B \vdash A, B} \text{ ax}}{B \vdash A \vee B} \vee_{\text{right}}}{\vdash B \Rightarrow A \vee B} \Rightarrow_{\text{right}}$ <p>Preuve dans LK</p>

5.	$\frac{\frac{\frac{B, A \Rightarrow C, C \vdash C}{B, A \Rightarrow C, B \Rightarrow C \vdash C} \text{ax} \quad \frac{\frac{B, A \Rightarrow C \vdash B}{A, C, B \Rightarrow C \vdash C} \text{ax}}{\frac{A \vee B, A \Rightarrow C, B \Rightarrow C \vdash C}{\vdash (A \vee B) \Rightarrow (A \Rightarrow C) \Rightarrow (B \Rightarrow C) \Rightarrow C} \Rightarrow_{\text{right}} \times 3} \vee_{\text{left}} \Rightarrow_{\text{left}} \text{ax}$ <p>Preuve dans LJ</p>	$\frac{\frac{\frac{B, A \Rightarrow C, C \vdash C}{B, A \Rightarrow C, B \Rightarrow C \vdash C} \text{ax} \quad \frac{\frac{B, A \Rightarrow C \vdash B}{A, C, B \Rightarrow C \vdash C} \text{ax}}{\frac{A \vee B, A \Rightarrow C, B \Rightarrow C \vdash C}{\vdash (A \vee B) \Rightarrow (A \Rightarrow C) \Rightarrow (B \Rightarrow C) \Rightarrow C} \Rightarrow_{\text{right}} \times 3} \vee_{\text{left}} \Rightarrow_{\text{left}} \text{ax}$ <p>Preuve dans LK</p>
6.	$\frac{\frac{A, \perp \vdash \neg A}{\vdash A \Rightarrow \perp \Rightarrow \neg A} \perp_{\text{left}}}{\Rightarrow_{\text{right}} \times 2}$	Ici, la preuve en LJ est la même que la preuve en LK.
7.	$\frac{\frac{\perp \vdash A}{\vdash \perp \Rightarrow A} \perp_{\text{left}}}{\Rightarrow_{\text{right}}}$	Ici, la preuve en LJ est la même que la preuve en LK.
8.	$\frac{\frac{\frac{A \vdash A}{A \Leftrightarrow B, A \vdash B} \text{ax} \quad \frac{B, A \vdash B}{A \Leftrightarrow B, A \vdash B} \text{ax}}{\vdash (A \Leftrightarrow B) \Rightarrow A \Rightarrow B} \Leftrightarrow_{\text{left}} \Rightarrow_{\text{right}} \times 2$ <p>Preuve dans LJ</p>	$\frac{\frac{\frac{A \vdash A, B}{A \Leftrightarrow B, A \vdash B} \text{ax} \quad \frac{A, B \vdash B}{A \Leftrightarrow B, A \vdash B} \text{ax}}{\vdash (A \Leftrightarrow B) \Rightarrow A \Rightarrow B} \Leftrightarrow_{\text{left}} \Rightarrow_{\text{right}} \times 2$ <p>Preuve dans LK</p>
9.	$\frac{\frac{\frac{B \vdash B}{A \Leftrightarrow B, B \vdash A} \text{ax} \quad \frac{A, B \vdash A}{A \Leftrightarrow B, B \vdash A} \text{ax}}{\vdash (A \Leftrightarrow B) \Rightarrow B \Rightarrow A} \Leftrightarrow_{\text{left}} \Rightarrow_{\text{right}} \times 2$ <p>Preuve dans LJ</p>	$\frac{\frac{\frac{B \vdash A, B}{A \Leftrightarrow B, B \vdash A} \text{ax} \quad \frac{A, B \vdash A}{A \Leftrightarrow B, B \vdash A} \text{ax}}{\vdash (A \Leftrightarrow B) \Rightarrow B \Rightarrow A} \Leftrightarrow_{\text{left}} \Rightarrow_{\text{right}} \times 2$ <p>Preuve dans LK</p>
10.	$\frac{\frac{\frac{A \Rightarrow B, A, B \vdash A}{A \Rightarrow B, B \Rightarrow A, B \vdash A} \text{ax} \quad \frac{A \Rightarrow B, B \vdash B}{A \Rightarrow B, B \Rightarrow A, B \vdash A} \text{ax}}{\frac{A \Rightarrow B, B \Rightarrow A \vdash A \Leftrightarrow B}{\vdash (A \Rightarrow B) \Rightarrow (B \Rightarrow A) \Rightarrow (A \Leftrightarrow B)} \Rightarrow_{\text{right}} \times 2} \Rightarrow_{\text{left}} \text{ax}$ <p>Preuve dans LJ</p>	$\frac{\frac{\frac{A \Rightarrow B, A, B \vdash A}{A \Rightarrow B, B \Rightarrow A, B \vdash A} \text{ax} \quad \frac{A \Rightarrow B, B \vdash B}{A \Rightarrow B, B \Rightarrow A, B \vdash A} \text{ax}}{\frac{A \Rightarrow B, B \Rightarrow A \vdash A \Leftrightarrow B}{\vdash (A \Rightarrow B) \Rightarrow (B \Rightarrow A) \Rightarrow (A \Leftrightarrow B)} \Rightarrow_{\text{right}} \times 2} \Rightarrow_{\text{left}} \text{ax}$ <p>Preuve dans LK</p>

Exercice 2 (Logique du premier ordre)

Démontrer les propositions suivantes dans LJ et LK (si la proposition n'admet pas de preuve intuitionniste, démontrer la proposition dans LJ_(em)) :

1. $\forall x.P(x) \Rightarrow \exists y.P(y) \vee Q(y)$
2. $(\exists x.P(x) \vee Q(x)) \Rightarrow (\exists x.P(x)) \vee (\exists x.Q(x))$
3. $(\forall x.P(x)) \wedge (\forall x.Q(x)) \Rightarrow \forall x.P(x) \wedge Q(x)$
4. $(\forall x.P(x) \wedge Q(x)) \Rightarrow (\forall x.P(x)) \wedge (\forall x.Q(x))$
5. $(\forall x.\neg P(x)) \Rightarrow \neg(\exists x.P(x))$
6. $\neg(\forall x.P(x)) \Rightarrow \exists x.\neg P(x)$

1.	$\frac{\frac{\frac{P(x) \vdash P(x)}{P(x) \vdash P(x) \vee Q(x)} \text{ax}}{P(x) \vdash \exists y.P(y) \vee Q(y)} \vee_{\text{right}} \Rightarrow_{\text{right}} \exists_{\text{right}} \Rightarrow_{\text{right}} \forall_{\text{right}}$ <p>Preuve dans LJ</p>	$\frac{\frac{\frac{P(x) \vdash P(x), Q(x)}{P(x) \vdash P(x) \vee Q(x)} \text{ax}}{P(x) \vdash \exists y.P(y) \vee Q(y)} \vee_{\text{right}} \Rightarrow_{\text{right}} \exists_{\text{right}} \Rightarrow_{\text{right}} \forall_{\text{right}}$ <p>Preuve dans LK</p>
----	--	--

<p>2.</p> $\frac{\frac{\frac{Q(x) \vdash Q(x)}{Q(x) \vdash (\exists x.Q(x))} \text{ax} \quad \frac{P(x) \vdash P(x)}{P(x) \vdash (\exists x.P(x))} \text{ax}}{\frac{Q(x) \vdash (\exists x.P(x)) \vee (\exists x.Q(x))}{P(x) \vdash (\exists x.P(x)) \vee (\exists x.Q(x))} \vee_{\text{right}}} \vee_{\text{left}} \quad \frac{\frac{P(x) \vee Q(x) \vdash (\exists x.P(x)) \vee (\exists x.Q(x))}{\exists x.P(x) \vee Q(x) \vdash (\exists x.P(x)) \vee (\exists x.Q(x))} \exists_{\text{left}}}{\vdash (\exists x.P(x) \vee Q(x)) \Rightarrow (\exists x.P(x)) \vee (\exists x.Q(x))} \Rightarrow_{\text{right}}$ <p>Preuve dans LJ</p>	$\frac{\frac{\frac{Q(x) \vdash P(x), Q(x)}{P(x) \vee Q(x) \vdash P(x), Q(x)} \text{ax} \quad \frac{P(x) \vdash P(x), Q(x)}{P(x) \vee Q(x) \vdash \exists x.P(x), \exists x.Q(x)} \vee_{\text{left}}}{\frac{P(x) \vee Q(x) \vdash \exists x.P(x), \exists x.Q(x)}{P(x) \vee Q(x) \vdash (\exists x.P(x)) \vee (\exists x.Q(x))} \exists_{\text{right}} \times 2} \vee_{\text{right}} \quad \frac{\frac{P(x) \vee Q(x) \vdash (\exists x.P(x)) \vee (\exists x.Q(x))}{\exists x.P(x) \vee Q(x) \vdash (\exists x.P(x)) \vee (\exists x.Q(x))} \exists_{\text{left}}}{\vdash (\exists x.P(x) \vee Q(x)) \Rightarrow (\exists x.P(x)) \vee (\exists x.Q(x))} \Rightarrow_{\text{right}}$ <p>Preuve dans LK</p>
<p>3.</p> $\frac{\frac{\frac{P(x), Q(x) \vdash Q(x)}{P(x), Q(x) \vdash P(x) \wedge Q(x)} \text{ax} \quad \frac{P(x), Q(x) \vdash P(x)}{\forall x.P(x), \forall x.Q(x) \vdash P(x) \wedge Q(x)} \wedge_{\text{right}}}{\frac{\frac{\forall x.P(x), \forall x.Q(x) \vdash P(x) \wedge Q(x)}{\forall x.P(x), \forall x.Q(x) \vdash \forall x.P(x) \wedge Q(x)} \forall_{\text{left}} \times 2}{(\forall x.P(x)) \wedge (\forall x.Q(x)) \vdash \forall x.P(x) \wedge Q(x)} \wedge_{\text{left}} \Rightarrow_{\text{right}} \quad \vdash (\forall x.P(x)) \wedge (\forall x.Q(x)) \Rightarrow \forall x.P(x) \wedge Q(x)}$	<p>Ici, la preuve en LJ est la même que la preuve en LK.</p>
<p>4.</p> $\frac{\frac{\frac{P(x), Q(x) \vdash Q(x)}{P(x) \wedge Q(x) \vdash Q(x)} \text{ax} \quad \frac{P(x), Q(x) \vdash P(x)}{P(x) \wedge Q(x) \vdash P(x)} \wedge_{\text{left}}}{\frac{\forall x.P(x) \wedge Q(x) \vdash Q(x)}{\forall x.P(x) \wedge Q(x) \vdash \forall x.Q(x)} \forall_{\text{left}} \quad \frac{\frac{P(x), Q(x) \vdash P(x)}{P(x) \wedge Q(x) \vdash P(x)} \wedge_{\text{left}} \quad \frac{P(x) \wedge Q(x) \vdash P(x)}{\forall x.P(x) \wedge Q(x) \vdash P(x)} \forall_{\text{left}}}{\frac{\forall x.P(x) \wedge Q(x) \vdash \forall x.Q(x)}{\vdash (\forall x.P(x) \wedge Q(x)) \Rightarrow (\forall x.P(x)) \wedge (\forall x.Q(x))} \Rightarrow_{\text{right}}}$	<p>Ici, la preuve en LJ est la même que la preuve en LK.</p>
<p>5.</p> $\frac{\frac{\frac{P(x) \vdash P(x)}{\neg P(x), P(x) \vdash} \text{ax} \quad \neg_{\text{left}}}{\forall x. \neg P(x), P(x) \vdash} \forall_{\text{left}} \quad \frac{\forall x. \neg P(x), P(x) \vdash}{\forall x. \neg P(x), \exists x.P(x) \vdash} \exists_{\text{left}}}{\frac{\forall x. \neg P(x), \exists x.P(x) \vdash}{\forall x. \neg P(x) \vdash \neg(\exists x.P(x))} \neg_{\text{right}} \Rightarrow_{\text{right}} \quad \vdash (\forall x. \neg P(x)) \Rightarrow \neg(\exists x.P(x))}$	<p>Ici, la preuve en LJ est la même que la preuve en LK.</p>
<p>6.</p> $\frac{\frac{\frac{\neg P(x) \vdash \neg P(x)}{\neg P(x) \vdash \exists x. \neg P(x)} \text{ax} \quad \exists_{\text{right}}}{\neg \exists x. \neg P(x), \neg P(x) \vdash} \neg_{\text{left}} \quad \frac{\neg \exists x. \neg P(x), \neg P(x) \vdash}{\neg \exists x. \neg P(x) \vdash P(x)} \neg_{\text{right}} \text{em}}{\frac{\neg \exists x. \neg P(x) \vdash P(x)}{\neg \exists x. \neg P(x) \vdash \forall x.P(x)} \forall_{\text{right}} \quad \frac{\neg \exists x. \neg P(x) \vdash \forall x.P(x)}{\neg(\forall x.P(x)), \neg \exists x. \neg P(x) \vdash} \neg_{\text{left}} \quad \frac{\neg(\forall x.P(x)), \neg \exists x. \neg P(x) \vdash}{\neg(\forall x.P(x)) \vdash \neg \neg \exists x. \neg P(x)} \neg_{\text{right}} \text{em}}{\frac{\neg(\forall x.P(x)) \vdash \neg \neg \exists x. \neg P(x)}{\neg(\forall x.P(x)) \vdash \exists x. \neg P(x)} \Rightarrow_{\text{right}} \quad \vdash \neg(\forall x.P(x)) \Rightarrow \exists x. \neg P(x)}$ <p>Preuve dans LJ_(em) (pas de preuve dans LJ possible)</p>	$\frac{\frac{\frac{P(x) \vdash P(x)}{\vdash \neg P(x), P(x)} \text{ax} \quad \neg_{\text{right}}}{\vdash \exists x. \neg P(x), P(x)} \exists_{\text{right}} \quad \frac{\vdash \exists x. \neg P(x), P(x)}{\vdash \exists x. \neg P(x), \forall x.P(x)} \forall_{\text{right}} \quad \frac{\vdash \exists x. \neg P(x), \forall x.P(x)}{\vdash \neg(\forall x.P(x)) \vdash \exists x. \neg P(x)} \neg_{\text{left}} \Rightarrow_{\text{right}} \quad \vdash \neg(\forall x.P(x)) \Rightarrow \exists x. \neg P(x)$ <p>Preuve dans LK</p>

Exercice 3 (Preuves en Coq)

Démontrer les propositions des exercices 1 et 2 en Coq.

On rappelle que pour lancer Coq, il suffit de se mettre dans un terminal et de taper la commande `coqide`, qui lance l'IDE de Coq.

► Voir TP1

2 TD/TP2 : Preuves en logique équationnelle

Exercice 1 (Preuves dans les CCC dans LJ_{EQ})

Dans les Catégories Cartésiennes Closes (CCC), tous les isomorphismes de types sont capturés par une théorie équationnelle démontrée complètement par Sergei Soloviev en 1983. Cette théorie est la suivante :

1. $\forall x, y. x \times y \doteq y \times x$
2. $\forall x, y, z. x \times (y \times z) \doteq (x \times y) \times z$
3. $\forall x, y, z. ((x \times y) \rightarrow z) \doteq (x \rightarrow (y \rightarrow z))$
4. $\forall x, y, z. (x \rightarrow (y \times z)) \doteq (x \rightarrow y) \times (x \rightarrow z)$
5. $\forall x. x \times \mathbb{1} \doteq x$
6. $\forall x. (x \rightarrow \mathbb{1}) \doteq \mathbb{1}$
7. $\forall x. (\mathbb{1} \rightarrow x) \doteq x$

où $\mathbb{1}$ est une constante.

Démontrer dans cette théorie en utilisant LJ_{EQ} les propositions suivantes :

1. $\forall x, y. x \times (y \rightarrow \mathbb{1}) \doteq x$
2. $\forall x, y. ((x \times \mathbb{1}) \times y) \doteq (y \times (\mathbb{1} \times x))$
3. $\forall x, y, z. ((x \times \mathbb{1}) \rightarrow (y \times (z \times \mathbb{1}))) \doteq (((x \times \mathbb{1}) \rightarrow (z \rightarrow \mathbb{1}) \times z) \times (\mathbb{1} \rightarrow (x \rightarrow y)))$

1. Preuve :

$$\frac{\frac{\overline{\vdash x \doteq x} \text{ refl}}{\vdash x \times \mathbb{1} \doteq x} =_{\text{right}_2} (5)}{\vdash x \times (y \rightarrow \mathbb{1}) \doteq x} =_{\text{right}_2, (6), \sigma = [y/x]} \frac{}{\vdash \forall x, y. x \times (y \rightarrow \mathbb{1}) \doteq x} \forall_{\text{right}} \times 2$$

2. Preuve :

$$\frac{\frac{\frac{\overline{\vdash (x \times y) \doteq (x \times y)} \text{ refl}}{\vdash (x \times y) \doteq (y \times x)} =_{\text{right}_2, (1)} (5)}{\vdash ((x \times \mathbb{1}) \times y) \doteq (y \times (x \times \mathbb{1}))} =_{\text{right}_2, (1)} \frac{}{\vdash \forall x, y. ((x \times \mathbb{1}) \times y) \doteq (y \times (\mathbb{1} \times x))} \forall_{\text{right}} \times 2$$

3. Preuve :

$$\begin{array}{c}
\frac{}{\vdash (x \rightarrow y) \times (x \rightarrow z) \doteq (x \rightarrow y) \times (x \rightarrow z)} \text{refl} \\
\frac{}{\vdash (x \rightarrow y) \times (x \rightarrow z) \doteq ((x \rightarrow z) \times (x \rightarrow y))} =_{\text{right}_2, (1), \sigma = [(x \rightarrow z)/x, (x \rightarrow y)/y]} \\
\frac{}{\vdash (x \rightarrow y) \times (x \rightarrow z) \doteq ((x \rightarrow z \times \mathbb{1}) \times (x \rightarrow y))} =_{\text{right}_2, (5), \sigma = [z/x]} \\
\frac{}{\vdash (x \rightarrow y) \times (x \rightarrow z) \doteq ((x \rightarrow \mathbb{1} \times z) \times (x \rightarrow y))} =_{\text{right}_2, (1), \sigma = [\mathbb{1}/x, z/y]} \\
\frac{}{\vdash (x \rightarrow y) \times (x \rightarrow z) \doteq ((x \rightarrow \mathbb{1} \times z) \times (\mathbb{1} \rightarrow (x \rightarrow y)))} =_{\text{right}_2, (7), \sigma = [x \rightarrow y/x]} \\
\frac{}{\vdash (x \rightarrow y) \times (x \rightarrow z) \doteq ((x \rightarrow (z \rightarrow \mathbb{1}) \times z) \times (\mathbb{1} \rightarrow (x \rightarrow y)))} =_{\text{right}_2, (6), \sigma = [z/x]} \\
\frac{}{\vdash (x \rightarrow (y \times z)) \doteq ((x \rightarrow (z \rightarrow \mathbb{1}) \times z) \times (\mathbb{1} \rightarrow (x \rightarrow y)))} =_{\text{right}_2, (4)} \\
\frac{}{\vdash (x \rightarrow (y \times (z \times \mathbb{1}))) \doteq ((x \rightarrow (z \rightarrow \mathbb{1}) \times z) \times (\mathbb{1} \rightarrow (x \rightarrow y)))} =_{\text{right}_2, (5), \sigma = [z/x]} \\
\frac{}{\vdash ((x \times \mathbb{1}) \rightarrow (y \times (z \times \mathbb{1}))) \doteq (((x \times \mathbb{1}) \rightarrow (z \rightarrow \mathbb{1}) \times z) \times (\mathbb{1} \rightarrow (x \rightarrow y)))} =_{\text{right}_2, (5)} \\
\vdash \forall x, y, z. ((x \times \mathbb{1}) \rightarrow (y \times (z \times \mathbb{1}))) \doteq (((x \times \mathbb{1}) \rightarrow (z \rightarrow \mathbb{1}) \times z) \times (\mathbb{1} \rightarrow (x \rightarrow y))) \quad \forall_{\text{right}} \times 3
\end{array}$$

Exercice 2 (Preuves en arithmétique de Peano dans LJ_{EQ})

À la fin du 19ème siècle, Giuseppe Peano a proposé un ensemble d'axiomes (pour la plupart équationnels) pour l'arithmétique. Ces axiomes sont les suivants (nous avons omis un axiome qui est en fait un schéma d'axiome au premier ordre et qui représente la récurrence) :

1. $\forall x. \neg(s(x) \doteq o)$
2. $\forall x. \exists z. \neg(x \doteq o) \Rightarrow s(z) \doteq x$
3. $\forall x, y. s(x) \doteq s(y) \Rightarrow x \doteq y$
4. $\forall x. x + o \doteq x$
5. $\forall x, y. x + s(y) \doteq s(x + y)$
6. $\forall x. x \times o \doteq o$
7. $\forall x, y. x \times s(y) \doteq (x \times y) + x$

où o est une constante, s , $+$ et \times des fonctions.

Démontrer dans cette théorie et en utilisant LJ_{EQ} les propositions suivantes :

- $1 + 2 \doteq 3$
- $2 + 2 \doteq 4$
- $2 \times 2 \doteq 4$

NB : $2 \equiv s(s(o))$.

— Preuve :

$$\begin{array}{c}
\frac{}{\vdash s(s(s(o))) \doteq s(s(s(o)))} \text{refl} \\
\frac{}{\vdash s(s(s(o) + o)) \doteq s(s(s(o)))} =_{\text{right}_2, (4), \sigma = [s(o)/x]} \\
\frac{}{\vdash s(s(o) + s(o)) \doteq s(s(s(o)))} =_{\text{right}_2, (5), \sigma = [s(o)/x, o/y]} \\
\vdash s(o) + s(s(o)) \doteq s(s(s(o))) =_{\text{right}_2, (5), \sigma = [s(o)/x, s(o)/y]}
\end{array}$$

— Preuve :

$$\begin{array}{c}
\frac{}{\vdash s(s(s(s(o)))) \doteq s(s(s(s(o))))} \text{refl} \\
\frac{}{\vdash s(s(s(s(o) + o))) \doteq s(s(s(s(o))))} =_{\text{right}_2, (4), \sigma = [s(o)/x]} \\
\frac{}{\vdash s(s(s(o) + s(o))) \doteq s(s(s(s(o))))} =_{\text{right}_2, (5), \sigma = [s(s(o))/x, o/y]} \\
\vdash s(s(o) + s(s(o))) \doteq s(s(s(s(o)))) =_{\text{right}_2, (5), \sigma = [s(s(o))/x, s(o)/y]}
\end{array}$$

— Preuve :

$$\begin{array}{c}
\overline{\vdash s(s(s(o))) \doteq s(s(s(o)))} \text{ refl} \\
\overline{\vdash s(s(s(o) + o)) \doteq s(s(s(o)))} =_{\text{right}_2, (4), \sigma = [s(o)/x]} \\
\overline{\vdash s(s(s(o)) + s(o)) \doteq s(s(s(o)))} =_{\text{right}_2, (5), \sigma = [s(s(o))/x, o/y]} \\
\overline{\vdash s(s(o)) + s(s(o)) \doteq s(s(s(o)))} =_{\text{right}_2, (5), \sigma = [s(s(o))/x, s(o)/y]} \\
\overline{\vdash s(s(o) + s(s(o))) \doteq s(s(s(o)))} =_{\text{right}_2, (4), \sigma = [o/x]} \\
\overline{\vdash s(s(o + o)) + s(s(o)) \doteq s(s(s(o)))} =_{\text{right}_2, (5), \sigma = [o/x, o/y]} \\
\overline{\vdash s(o + s(o)) + s(s(o)) \doteq s(s(s(o)))} =_{\text{right}_2, (5), \sigma = [o/x, s(o)/y]} \\
\overline{\vdash (o + s(s(o))) + s(s(o)) \doteq s(s(s(o)))} =_{\text{right}_2, (6), \sigma = [s(s(o))/x]} \\
\overline{\vdash ((s(s(o)) \times o) + s(s(o))) + s(s(o)) \doteq s(s(s(o)))} =_{\text{right}_2, (7), \sigma = [s(s(o))/x, o/y]} \\
\overline{\vdash (s(s(o)) \times s(o)) + s(s(o)) \doteq s(s(s(o)))} =_{\text{right}_2, (7), \sigma = [s(s(o))/x, s(o)/y]} \\
\overline{\vdash s(s(o)) \times s(s(o)) \doteq s(s(s(o)))}
\end{array}$$

Exercice 3 (Isomorphismes de types dans les CCC en Coq)

En Coq, la théorie équationnelle correspondant aux isomorphismes de types dans les CCC peut être implémentée comme suit :

`Open Scope type_scope.`

`Section iso_axioms.`

`Variables A B C : Set.`

`Axiom Com : A * B = B * A.`

`Axiom Ass : A * (B * C) = A * B * C.`

`Axiom Cur : (A * B -> C) = (A -> B * C).`

`Axiom Dis : (A -> B * C) = (A -> B) * (A -> C).`

`Axiom P_unit : A * unit = A.`

`Axiom AR_unit : (A -> unit) = unit.`

`Axiom AL_unit : (unit -> A) = A.`

`End iso_axioms.`

1. Démontrer les lemmes suivants dans cette théorie en Coq :

`Lemma isos_ex1 : forall A B : Set, A * (B -> unit) = A.`

`Lemma isos_ex2 : forall A B : Set, A * unit * B = B * (unit * A).`

`Lemma isos_ex3 : forall A B C : Set,`
`(A * unit -> B * (C * unit)) =`
`(A * unit -> (C -> unit) * C) * (unit -> A -> B).`

2. Écrire une tactique qui normalise les expressions selon les axiomes de la théorie (attention, tous les axiomes ne sont pas bons à prendre).
3. Démontrer les propositions précédentes à l'aide de cette tactique.

► Voir TP2

Exercice 4 (Arithmétique de Peano en Coq)

En Coq, la théorie de l'arithmétique de Peano peut être implémentée comme suit :

```
Section Peano.

Parameter N : Set.
Parameter o : N.
Parameter s : N -> N.
Parameters plus mult : N -> N -> N.
Variables x y : N.
Axiom ax1 : ~(s z) = o.
Axiom ax2 : exists z, ~(x = o) -> (s z) = x.
Axiom ax3 : (s x) = (s y) -> x = y.
Axiom ax4 : (plus x o) = x.
Axiom ax5 : (plus x (s y)) = (s (plus x y)).
Axiom ax6 : (mult x o) = o.
Axiom ax7 : (mult x (s y)) = (plus (mult x y) x).

End Peano.
```

1. Démontrer les propositions suivantes dans cette théorie en Coq :
 - $1 + 2 = 3$
 - $2 + 2 = 4$
 - $2 \times 2 = 4$
2. Écrire une tactique qui calcule automatiquement dans cette théorie.
3. Démontrer les propositions précédentes à l'aide de cette tactique.
4. Même question en utilisant la tactique `autorewrite` (voir la documentation).

► Voir TP2

3 TD/TP3 : Types inductifs

Exercice 1 (Fonctions et preuves inductives sur les entiers)

1. Écrire la fonction `mult` sur les entiers naturels \mathcal{N} .
2. Démontrer que : $\forall n \in \mathcal{N}. \text{mult}(2, n) = \text{plus}(n, n)$.
3. Démontrer que : $\forall n \in \mathcal{N}. \text{mult}(n, 2) = \text{plus}(n, n)$.

1. Soit $\text{mult} : \mathcal{N} \times \mathcal{N} \rightarrow \mathcal{N}$.
 - (m_1) $\forall n \in \mathcal{N}. \text{mult}(0, n) = 0$.
 - (m_2) $\forall p, n \in \mathcal{N}. \text{mult}((S\ p), n) = \text{plus}(\text{mult}(p, n), n)$.

2. On sait que $2 = (S (S 0))$. Ainsi :

$$\begin{aligned}
mult((S (S 0)), n) &= plus(mult((S 0), n), n) && \text{par } m_2 \\
&= plus(plus(mult(0, n), n), n) && \text{par } m_2 \\
&= plus(plus(0, n), n) && \text{par } m_1 \\
&= plus(n, n) && \text{par } plus(0, n) = n
\end{aligned}$$

3. On rappelle la définition de $plus$:

$$\begin{aligned}
(p_1) \quad \forall n \in \mathcal{N}. plus(0, n) &= n \\
(p_2) \quad \forall p, n \in \mathcal{N}. plus((S p), n) &= (S plus(p, n))
\end{aligned}$$

Lemme 1. $\forall n \in \mathcal{N}. plus(n, 0) = n$

Démonstration. On prouve le lemme 1 par induction structurelle :

Base Pour $n = 0$, on a $plus(0, 0) = 0$ par p_1 . Par réflexivité, la propriété est vérifiée pour le cas de base.

Induction On suppose que $plus(n, 0) = n$. Montrons que $plus((S n), 0) = (S n)$.

$$\begin{aligned}
plus((S n), 0) &= (S plus(n, 0)) && \text{par } p_2 \\
&= (S n) && \text{par hypothèse d'induction}
\end{aligned}$$

Par réflexivité, la propriété est vérifiée $\forall n \in \mathcal{N}$. □

Lemme 2. $\forall n, m \in \mathcal{N}. plus(m, (S n)) = (S plus(m, n))$

Démonstration. On prouve le lemme 2 par induction structurelle :

Base Pour $m = 0$, on a $plus(0, (S n)) = (S n)$ par p_1 , et $(S plus(0, n)) = (S n)$. Par réflexivité, $(S n) = (S n)$ et donc la propriété est vérifiée pour le cas de base.

Induction On suppose que la propriété est vraie pour un m quelconque, c'est à dire que $plus(m, (S n)) = (S plus(m, n))$. Montrons que $plus((S m), (S n)) = (S (S plus(m, n)))$.

$$\begin{aligned}
plus((S m), (S n)) &= (S plus(m, (S n))) && \text{par } p_2 \\
&= (S (S plus(m, n))) && \text{par hypothèse d'induction}
\end{aligned}$$

Par réflexivité, on a bien $plus((S m), (S n)) = (S (S plus(m, n)))$, donc la propriété est vérifiée $\forall n, m \in \mathcal{N}$. □

On peut maintenant montrer $\forall n \in \mathcal{N}. mult(n, 2) = plus(n, n)$ par induction :

Base Pour $n = 0$, on a $mult(0, 2) = 0$ par m_1 , et $plus(0, 0) = 0$ par p_1 . Par réflexivité, $mult(0, 2) = plus(0, 0)$ et ainsi la propriété est vérifiée pour le cas de base.

Induction On suppose que $\forall n \in \mathcal{N}. mult(n, 2) = plus(n, n)$. Montrons que $\forall n \in \mathcal{N}. mult((S n), 2) = plus((S n), (S n))$. On sait que $2 = (S (S 0))$.

$$\begin{aligned}
mult((S n), (S (S 0))) &= plus(mult(n, 2), (S (S 0))) && \text{par } m_2 \\
&= plus(plus(n, n), (S (S 0))) && \text{par hypothèse d'induction} \\
&= (S plus(plus(n, n), (S 0))) && \text{par le lemme 2} \\
&= (S (S plus(plus(n, n), 0))) && \text{par le lemme 2} \\
&= (S (S plus(n, n))) && \text{par le lemme 1}
\end{aligned}$$

De plus :

$$\begin{aligned} plus((S\ n), (S\ n)) &= (S\ plus(n, (S\ n))) && \text{par } p_2 \\ &= (S\ (S\ plus(n, n))) && \text{par le lemme 2} \end{aligned}$$

Par réflexivité, on conclut que $mult((S\ n), 2) = plus((S\ n), (S\ n))$ et donc la propriété est vérifiée $\forall n \in \mathcal{N}$.

Exercice 2 (Fonctions et preuves inductives sur les listes)

1. Écrire la fonction *rev* qui inverse les éléments d'une liste.
2. Démontrer que : $\forall l \in \mathcal{L}, \forall e \in \mathcal{A}. rev(app(l, [e])) = e :: rev(l)$.
3. Démontrer que : $\forall l \in \mathcal{L}. rev(rev(l)) = l$.

1. On rappelle que l'ensemble \mathcal{A} est le type des éléments de la liste. On peut définir $rev : \mathcal{L} \rightarrow \mathcal{L}$:

$$\begin{aligned} (l_1) \quad rev([]) &= [] \\ (l_2) \quad \forall e \in \mathcal{A}, \forall l \in \mathcal{L}. rev(e :: l) &= app(rev(l), [e]). \end{aligned}$$

2. On rappelle la définition de *app* :

$$\begin{aligned} (a_1) \quad \forall l \in \mathcal{L}. app([], l) &= l \\ (a_2) \quad \forall e \in \mathcal{A}, \forall l_1, l_2 \in \mathcal{L}. app(e :: l_1, l_2) &= e :: app(l_1, l_2). \end{aligned}$$

Lemme 3. $\forall l \in \mathcal{L}, \forall e \in \mathcal{A}. rev(app(l, [e])) = e :: rev(l)$

Démonstration. On le démontre par induction.

Base Pour $l = []$ et $e \in \mathcal{A}$, on a $rev(app([], [e])) = rev([e]) = [e]$ et $e :: rev([]) = e :: [] = [e]$. Par réflexivité, $rev(app([], e)) = e :: rev([])$.

Induction On suppose que $\forall l \in \mathcal{L}, \forall e_1 \in \mathcal{A}. rev(app(l, [e_1])) = e_1 :: rev(l)$. Montrons que $\forall l \in \mathcal{L}, \forall e_1, e_2 \in \mathcal{A}. rev(app(e_2 :: l, [e_1])) = e_1 :: rev(e_2 :: l)$.

$$\begin{aligned} rev(app(e_2 :: l, [e_1])) &= rev(e_2 :: app(l, [e_1])) && \text{par } a_2 \\ &= app(rev(app(l, [e_1])), [e_2]) && \text{par } l_2 \\ &= app(e_1 :: rev(l), [e_2]) && \text{par hypothèse d'induction} \\ &= e_1 :: app(rev(l), [e_2]) && \text{par } a_2 \\ &= e_1 :: rev(e_2 :: l) && \text{par } l_2 \end{aligned}$$

Par réflexivité, $rev(app(e_2 :: l, [e_1])) = e_1 :: rev(e_2 :: l)$, donc la propriété est vérifiée $\forall e \in \mathcal{A}$ et $\forall l \in \mathcal{L}$. □

3. On démontre $\forall l \in \mathcal{L}. rev(rev(l)) = l$ par induction.

Base Pour $l = []$, on a $rev(rev([])) = rev([]) = []$. Par réflexivité, la propriété est vérifiée pour le cas de base.

Induction On suppose que $\forall l \in \mathcal{L}. rev(rev(l)) = l$. Montrons que $\forall e \in \mathcal{A}, \forall l \in \mathcal{L}. rev(rev(e :: l)) = e :: l$.

$$\begin{aligned} rev(rev(e :: l)) &= rev(app(rev(l), [e])) && \text{par } l_2 \\ &= e :: rev(rev(l)) && \text{par le lemme 3} \\ &= e :: l && \text{par hypothèse d'induction} \end{aligned}$$

Par réflexivité, $rev(rev(e :: l)) = e :: l$, donc la propriété est vérifiée $\forall l \in \mathcal{L}$.

Exercice 3 (Type inductif des formules en logique)

1. Définir le type des formules en logique propositionnelle.
2. Écrire la fonction sub , qui rend l'ensemble des sous-formules d'une formule F .
3. Écrire la fonction nbc , qui rend l'ensemble des connecteurs d'une formule F .
4. Écrire le schéma d'induction structurelle des formules.
5. Démontrer que : $|sub(F)| \leq 2 \times nbc(F) + 1$ pour toute formule F .

1. Soit \mathcal{V} l'ensemble des variables de propositions. Soit \mathcal{F} l'ensemble des formules de logique propositionnelle. On a \mathcal{F} :

- Si $A \in \mathcal{V}$, alors $A \in \mathcal{F}$,
- $\top, \perp \in \mathcal{F}$,
- $\forall \phi \in \mathcal{F}, \neg \phi \in \mathcal{F}$,
- $\forall \phi_1, \phi_2 \in \mathcal{F}, \phi_1 \text{ c } \phi_2 \in \mathcal{F}$ (avec $c \in \{\vee, \wedge, \Rightarrow, \Leftrightarrow\}$)

2. $sub : \mathcal{F} \rightarrow 2^{\mathcal{F}}$:

- $sub(\top) = \{\top\}$; $sub(\perp) = \{\perp\}$,
- $\forall A \in \mathcal{V}. sub(A) = \{A\}$,
- $\forall \phi \in \mathcal{F}. sub(\neg \phi) = \{\neg \phi\} \cup sub(\phi)$,
- $\forall \phi_1, \phi_2 \in \mathcal{F}. sub(\phi \text{ c } \phi_2) = \{\phi_1 \text{ c } \phi_2\} \cup sub(\phi_1) \cup sub(\phi_2)$ (avec $c \in \{\vee, \wedge, \Rightarrow, \Leftrightarrow\}$).

3. $nbc : \mathcal{F} \rightarrow \mathcal{N}$:

- $nbc(\top) = 0$; $nbc(\perp) = 0$,
- $\forall A \in \mathcal{V}. nbc(A) = 0$,
- $\forall \phi \in \mathcal{F}. nbc(\neg \phi) = 1 + nbc(\phi)$,
- $\forall \phi_1, \phi_2 \in \mathcal{F}. nbc(\phi \text{ c } \phi_2) = 1 + nbc(\phi_1) + nbc(\phi_2)$ (avec $c \in \{\vee, \wedge, \Rightarrow, \Leftrightarrow\}$).

4. $\forall P \in (\mathcal{F} \rightarrow Prop). (\forall A \in \mathcal{V}. P(A)) \Rightarrow (\forall \phi \in \mathcal{F}. P(\phi) \Rightarrow P(\neg \phi)) \Rightarrow (\forall \phi_1, \phi_2 \in \mathcal{F}. (P(\phi_1) \wedge P(\phi_2) \Rightarrow P(\phi_1 \wedge \phi_2) \wedge (P(\phi_1) \vee P(\phi_2) \Rightarrow P(\phi_1 \vee \phi_2)) \wedge (P(\phi_1) \Rightarrow P(\phi_2) \Rightarrow P(\phi_1 \Rightarrow \phi_2)) \wedge (P(\phi_1) \Leftrightarrow P(\phi_2) \Rightarrow P(\phi_1 \Leftrightarrow \phi_2)))) \Rightarrow \forall \phi \in \mathcal{F}. P(\phi)$

5. On démontre $\forall F \in \mathcal{F}. |sub(F)| \leq 2 \times nbc(F) + 1$ par induction :

Base $F \in \mathcal{V}$ ou $F \in \{\top, \perp\}$, alors $sub(F) = \{F\}$ et $nbc(F) = 2 \times 0 + 1 = 1$. Par réflexivité, $|\{F\}| = 1$, donc la propriété est vérifiée pour le cas de base.

Induction On suppose que $\forall \phi, \phi_1, \phi_2 \in \mathcal{F}. |sub(\phi)| \leq 2 \times nbc(\phi) + 1$.

- $F = \neg \phi$, dans ce cas, $sub(F) = \{F\} \cup sub(\phi)$. On a $|sub(F)| = 1 + |sub(\phi)| \leq 1 + 2 \times nbc(\phi) + 1 = 2nbc(\phi) + 2$ et $nbc(F) = 1 + nbc(\phi)$, on a $2(nbc(F)) + 1 = 2nbc(\phi) + 2$, on a donc bien $|sub(F)| \leq 2(nbc(F)) + 1$.
- $F = \phi_1 \text{ c } \phi_2$ avec $c \in \{\vee, \wedge, \Rightarrow, \Leftrightarrow\}$. Dans ce cas, $sub(F) = \{F\} \cup sub(\phi_1) \cup sub(\phi_2)$. On a $|sub(F)| = 1 + |sub(\phi_1)| + |sub(\phi_2)| \leq 1 + 2nbc(\phi_1) + 1 + 2nbc(\phi_2) + 1 = 2(nbc(\phi_1) + nbc(\phi_2)) + 3$ et $nbc(F) = 1 + nbc(\phi_1) + nbc(\phi_2)$, on a $2(nbc(F)) + 1 = 2(nbc(\phi_1) + nbc(\phi_2)) + 3$. On a donc bien $|sub(F)| \leq 2(nbc(F)) + 1$.

Exercice 4 (Relations inductives sur les listes)

1. Spécifier la relation « être une permutation de » pour deux listes.
2. Démontrer que la liste $[1; 2; 3]$ est une permutation de $[3; 2; 1]$.
3. Spécifier la relation « être triée » pour une liste.
4. Démontrer que la liste $[1; 2; 3]$ est triée.

1. On définit la relation inductive $is_perm : \mathcal{L} \times \mathcal{L} \rightarrow Prop$ de la façon suivante :
 - ($perm_1$) On a : $is_perm([], [])$,
 - ($perm_2$) $\forall l_1, l_2 \in \mathcal{L}, \forall e \in \mathcal{A}. is_perm(l_1, l_2) \Rightarrow is_perm(e :: l_1, e :: l_2)$.
 - ($perm_3$) $\forall l_1, l_2 \in \mathcal{L}, \forall e \in \mathcal{A}. is_perm(l_1, l_2) \Rightarrow is_perm(app(l_1, [e]), e :: l_2)$.
 - ($perm_4$) $\forall l_1, l_2 \in \mathcal{L}, \forall e \in \mathcal{A}. is_perm(l_1, l_2) \Rightarrow is_perm(e :: l_1, app(l_2, [e]))$.
 - ($perm_5$) $\forall l_1, l_2 \in \mathcal{L}, \forall e \in \mathcal{A}. is_perm(l_1, l_2) \Rightarrow is_perm(app(l_1, [e]), app(l_2, [e]))$.
2. On veut montrer que $is_perm([1; 2; 3], [3; 2; 1])$.
 - Par $perm_4$, on veut montrer $is_perm([2; 3], [3; 2])$.
 - Par $perm_4$, on veut montrer que $is_perm([2], [2])$.
 - Par $perm_2$, on veut montrer que $is_perm([], [])$.
 - Démontré en utilisant le cas de base.
3. On définit la relation inductive $is_sorted : \mathcal{L} \rightarrow Prop$ de la façon suivante :
 - ($sort_1$) On a $is_sorted([])$,
 - ($sort_2$) On a $\forall n \in \mathcal{N}. is_sorted([n])$,
 - ($sort_3$) $\forall n_1, n_2 \in \mathcal{N} \forall l \in \mathcal{L}. n_1 < n_2 \Rightarrow is_sorted(app(l + + [n_1], [n_2]))$.
4. On veut montrer que $is_sorted([1; 2; 3])$.
 - Par $sort_3$, comme $2 < 3$, on veut montrer $is_sorted([1; 2])$.
 - Par $sort_3$, comme $1 < 2$, on veut montrer $is_sorted([1])$.
 - Démontré en utilisant $sort_2$.

Exercice 5 (Preuves en Coq)

Faire les exercices 1, 2 et 3 en Coq.

► Voir TP3

Exercice 6 (Preuves en Coq)

1. Écrire la relation inductive is_even (vue en cours).
2. Écrire une tactique qui démontre des buts de la forme $is_even(n)$.
3. Écrire une tactique qui démontre les buts de la forme $\neg is_even(n)$.
4. Écrire une tactique qui démontre les buts précédents indifféremment.
5. Écrire la fonction f_{is_even} qui teste si un entier est pair.
6. Démontrer que la fonction f_{is_even} est correcte vis à vis de la relation is_even .

► Voir TP3