Université de Lorraine | Preuves assistées par ordinateur
2024-2025

Aide-memoire - Tactiques de LEAN

Version du 17 mars 2025

Comment utiliser cet aide-mémoire. Dans les pages qui suivent, vous allez trouver une liste d'encadrés, qui présentent chacun une étape-clé pouvant apparaître dans une démonstration, et le code LEAN qui lui correspond. Quand vous essaierez de démontrer un énoncé, vous pourrez étape par étape vous demander :

1. De quelle forme est l'objectif à démontrer actuellement ?
2. Quel est l'encadré pertinent ?
3. Dois-je modifier l'objectif, ou bien puis-je modifier le contexte en utilisant les hypothèses déjà introduites ?

En écrivant ligne à ligne les textes en bleu dans les encadrés qui suivent, cela vous permettra d'écrire des preuves complètes et correctes.

Avertissement. Cet aide-mémoire est amené à évoluer au fur et à mesure que des tactiques seront introduites dans les séances de TP. Vous pouvez l'imprimer si vous le souhaitez, mais prenez garde au fait qu'il s'agit ici de la version du **17 mars 2025**. N'utilisez pas cette version en fin de semestre, **l'aide-mémoire n'est pas encore complet**.

Important

Avant de pouvoir utiliser les commandes qui suivront dans cet aide-mémoire, vous devrez taper

```
import Mathlib.Tactic
```

au début de votre fichier.

1 Tactiques pour modifier l'objectif

Si l'objectif est de la forme...

Objectif : `P→Q`

---

1. En LEAN, on écrit [1] :

   `intro hypothese_P`

   Cela ajoute l'hypothèse `hypothese_P` au contexte (cette hypothèse affirme que `P` est vraie), et remplace l'objectif par `Q`.

2. Dans la démonstration, on écrit :

   Supposons que P est vraie, et appelons « `hypothese_P` » cette hypothèse. Démontrons que Q est vraie.

[1] On peut remplacer « `hypothese_P` » par le nom que l'on souhaite.

Objectif : `∀ x : E, P(x)`

---

1. En LEAN, on écrit [1] :

   `intro x`

   Cela ajoute un objet `x` de type `E` au contexte, et remplace l'objectif en cours par `P(x)`.

2. Dans la démonstration, on écrit :

   Soit $x \in E$. Montrons que $P(x)$ est vraie.

[1] On peut remplacer « `x` » par le nom que l'on souhaite.

Variante avec plusieurs objets.

Objectif : `∀ x1 x2 ... xn : E, P(x1, x2, ..., xn)`

---

1. En LEAN, on écrit [1] :

   `intro x1 x2 ... xn`

   Cela ajoute des objets `x1, x2, ..., xn` de type `E` au contexte, et remplace l'objectif en cours par `P(x1, x2, ..., xn)`.

2. Dans la démonstration, on écrit :

   Soient $x_1, x_2, \dots, x_n \in E$. Montrons que $P(x_1, x_2, \dots, x_n)$ est vraie.

[1] On peut remplacer « `x1, x2, ..., xn` » par les noms que l'on souhaite.

Objectif : `∃ x : E, P(x)`

---

1. En LEAN, on écrit [1] :

   `use <FORMULE>`

   Cela remplace l'objectif en cours par `P(<FORMULE>)`.

2. Dans la démonstration, on écrit :

   On doit montrer qu'il existe $x \in E$ tel que $P(x)$ soit vraie. Montrons que

   $$x = \text{<FORMULE>}$$

   convient.

Attention : pour pouvoir effectuer les étapes présentées ci-dessus, il faut choisir une formule adéquate pour x. Il n'est possible de taper cette formule que si le contexte contient déjà tous les éléments apparaissant dans la formule.

La manière de modifier le contexte est expliquée à la section suivante.

[1] Il faut remplacer « `<FORMULE>` » par la formule adéquate, en fonction de la situation.

Objectif : $\boxed{\texttt{P} \land \texttt{Q}}$

---

1. En LEAN, on écrit [1] :

```
apply And.intro
{
  --Inserer ici une preuve de P
}
{
  - -Inserer ici une preuve de Q
}
```

   Cela crée *deux objectifs* `P` et `Q` à montrer successivement. Entre les deux premières accolades, on doit écrire les arguments d'une preuve de `P`, et entre les deux accolades suivantes, les arguments d'une preuve de `Q`.

2. Dans la démonstration, on écrit :

   Montrons d'abord que P est vraie. *(Ecrire ici une preuve de P)*
   A présent, montrons que Q est vraie. *(Ecrire ici une preuve de Q)*

Objectif : $\boxed{\texttt{P} \lor \texttt{Q}}$

Il y a deux cas : ou bien on choisit de montrer que P est vraie, ou bien on choisit de montrer que Q est vraie. Savoir laquelle des affirmations on choisira de démontrer dépendra du contexte.

1. En LEAN, **si on choisit de démontrer l'affirmation de gauche**, on écrit [1] :

   ```
   apply Or.inl
   ```

   Cela remplace l'objectif par `P`.
   A l'inverse, **si on choisit de démontrer l'affirmation de droite**, on écrit

   ```
   apply Or.inr
   ```

   Cela remplace l'objectif par `Q`.

2. Suivant les cas, dans la démonstration, on écrit **ou bien :**

   On doit montrer que « P ou Q » est vraie. Montrons en fait que P est vraie. *(Ecrire ici une preuve de P)*

   ou bien :

   On doit montrer que « P ou Q » est vraie. Montrons en fait que Q est vraie. *(Ecrire ici une preuve de Q)*

---

[1] Le « `l` » de `Or.inl` signifie « left », et le « `r` » de `Or.inr` signifie « right ».

Objectif : $\boxed{\texttt{P} \leftrightarrow \texttt{Q}}$

---

1. En LEAN, on écrit [1] :

```
apply Iff.intro
{
  --Inserer ici une preuve de P→Q
}
{
  - -Inserer ici une preuve de Q→P
}
```

   Cela crée *deux objectifs* `P→Q` et `Q→P` à montrer successivement. Entre les deux premières accolades, on doit écrire les arguments d'une preuve de `P→Q`, et entre les deux accolades suivantes, les arguments d'une preuve de `Q→P`.

2. Dans la démonstration, on écrit :

   Montrons d'abord que P implique Q. *(Ecrire ici une preuve de $P \Rightarrow Q$)*
   A présent, montrons que Q implique P. *(Ecrire ici une preuve de $Q \Rightarrow P$)*

2 Utiliser des hypothèses du contexte

Si le contexte contient :

— une **hypothèse**[1] :

`hypo_P` : $\forall$ `x : E, P(x)`

— un **objet** `z`, de type `E`.

---

1. En LEAN, on a le droit d'écrire [2] :

```
have hypo_z := hypo_P z
```

Cela ajoute une hypothèse `hypo_z` au contexte, qui affirme que `P(z)` est vraie.

2. Dans la démonstration, on écrit :

On applique l'hypothèse `hypo_P` à z. On en déduit que $P(z)$ est vraie : notons `hypo_z` cette nouvelle hypothèse. .

---

[1] Les noms `hypo_P`, `z` pourront être remplacés par d'autres noms, en fonction de la situation.

[2] On peut choisir le nom que l'on veut pour l'hypothèse `hypo_z`.

Variante avec plusieurs objets. Si le contexte contient :

— une **hypothèse**[1] :

`hypo_P` : $\forall$ `x1 x2 ... xn : E, P(x1, x2,... , xn)`

— des **objets** `z1`, `z2`,..., `zn` de type `E`.

---

1. En LEAN, on a le droit d'écrire [2] :

```
have hypo_z := hypo_P z1 z2 ... zn
```

Cela ajoute une hypothèse `hypo_z` au contexte, qui affirme que `P(z1, z2,..., zn)` est vraie.

2. Dans la démonstration, on écrit :

On applique l'hypothèse `hypo_P` à $z_1, z_2, \ldots, z_n$. On en déduit que $P(z_1, z_2, \ldots, z_n)$ est vraie : notons `hypo_z` cette nouvelle hypothèse. .

Si le contexte contient :

— une **hypothèse** [1] :

hypo_P : $\exists$ x : E, P(x)

---

1. En LEAN, on a le droit d'écrire [2] [3] :

   `rcases hypo_P with ⟨ z, hypo_z ⟩.`

   Cela ajoute un objet `z` de type `E` au contexte, ainsi qu'une hypothèse `hypo_z`, qui affirme que `P(z)` est vraie.

2. Dans la démonstration, on écrit :

   D'après l'hypothèse `hypo_P`, il existe $z \in E$ tel que $P(z)$ soit vraie. Notons `hypo_z` l'hypothèse que $P(z)$ est vraie.
   .

---

[1] Le nom `hypo_P` devra être remplacé par un autre nom, en fonction de la situation.

[2] On peut choisir les noms que l'on veut pour `z` et `hypo_z`.

[3] Tapez `\<` et `\>` pour les chevrons « ⟨ » et « ⟩ ».

Si le contexte contient :

— une **hypothèse 1**[1] :

`hypo_implique :    P→Q`

— une **hypothèse 2** :

`hypo_P :    P`

---

1. En LEAN, on a le droit d'écrire [2] :

```
have hypo_Q := hypo_implique hypo_P
```

   Cela ajoute une hypothèse `hypo_Q` au contexte, qui affirme que `Q` est vraie.

2. Dans la démonstration, on écrit :

   Puisque P est vraie d'après `hypo_P` et que $P \Rightarrow Q$ d'après `hypo_implique`, alors Q est vraie aussi. Notons `hypo_Q` cette nouvelle hypothèse.

---

[1] Les noms `hypo_implique` et `hypo_P` pourront être remplacés par d'autres noms, en fonction de la situation.

[2] On peut choisir le nom que l'on veut pour l'hypothèse `hypo_Q`.

Variante. Implication avec hypothèse évidente.

Si le contexte contient :

— une **hypothèse 1**[1] :

`hypo_implique :    P→Q`

où `P` est une proposition *évidemment vraie* (par exemple une tautologie comme `x = x` ou un calcul immédiat comme `2 + 2 = 4`)

---

1. En LEAN, on a le droit d'écrire [2] :

```
have hypo_Q := hypo_implique (by simp)
```

   Cela ajoute une hypothèse `hypo_Q` au contexte, qui affirme que `Q` est vraie.

2. Dans la démonstration, on écrit :

   L'hypothèse de `hypo_implique` est évidemment satisfaite, donc `Q` est vraie. Notons `hypo_Q` cette nouvelle hypothèse.

Si le contexte contient :

— une **hypothèse**[1] :

$$H : \quad P \wedge Q$$

1. En LEAN, on a le droit d'écrire [2] :

```
have hypo_P := H.left
have hypo_Q := H.right
```

   Cela ajoute une hypothèse `hypo_P` au contexte, qui affirme que `P` est vraie, et une hypothèse `hypo_Q`, qui affirme que `Q` est vraie.

2. Dans la démonstration, on écrit :

   D'après `H`, P et Q sont vraies. Notons `hypo_P` l'hypothèse « P » et `hypo_Q` l'hypothèse « Q ».

[1] Le nom `H` pourra être remplacé par un autre nom, en fonction de la situation.

[2] On peut choisir le nom que l'on veut pour les hypothèses `hypo_P` et `hypo_Q`.

Si le contexte contient :

— une **hypothèse**[1] :

$$\texttt{H} : \quad \texttt{P} \vee \texttt{Q}$$

1. En LEAN, on a le droit d'écrire [2] :

```
rcases H with H_cas_1 | H_cas_2
{
  --Inserer ici une preuve de l'objectif en supposant P vraie
}
{
  - -Inserer ici une preuve de l'objectif en supposant Q vraie
}
```

Cela demande à LEAN de procéder à une **disjonction de cas**.
Tout d'abord, on doit démontrer l'objectif en remplaçant l'hypothèse `H` par l'hypothèse `H_cas_1`, qui affirme que `P` est vraie. Une fois cela fait, on doit démontrer à nouveau l'objectif en remplaçant `H` par l'hypothèse `H_cas_2`, qui affirme que `Q` est vraie.

2. Dans la démonstration, on écrit :

D'après `H`, P est vraie, ou Q est vraie. Supposons d'abord que P est vraie, et notons `H_cas_1` cette hypothèse. Démontrons le résultat dans ce cas. *(Ecrire maintenant une preuve de l'objectif sous cette hypothèse.)*

Supposons maintenant que Q est vraie, et notons `H_cas_2` cette hypothèse. Démontrons le résultat dans ce cas. *(Ecrire maintenant une preuve de l'objectif sous cette hypothèse.)*

[1] Le nom `H` pourra être remplacé par un autre nom, en fonction de la situation.

[2] On peut choisir le nom que l'on veut pour les hypothèses `H_cas_1` et `H_cas_2`.

Si le contexte contient :

— une **hypothèse**[1] :

H : $\quad$ P↔Q

1. En LEAN, on a le droit d'écrire [2] [3] :

```
have hypo_P_implique_Q := H.mp
have hypo_Q_implique_P := H.mpr
```

   Cela ajoute une hypothèse `hypo_P_implique_Q` au contexte, qui affirme que `P` implique `Q`, et une hypothèse `hypo_Q_implique_P`, qui affirme que `Q` implique `P`.

2. Dans la démonstration, on écrit :

   D'après `H`, P et Q sont équivalentes. Notons `hypo_P_implique_Q` l'hypothèse « $P \Rightarrow Q$ » et `hypo_Q_implique_P` l'hypothèse « $Q \Rightarrow P$ ».

[1] Le nom `H` pourra être remplacé par un autre nom, en fonction de la situation.

[2] On peut choisir le nom que l'on veut pour les hypothèses `hypo_P_implique_Q` et `hypo_Q_implique_P`.

[3] Les lettres « `mp` » désignent le « *modus ponens* », tandis que « `mpr` » désigne le « *modus ponens* récriproque ».

Si le contexte contient :

— une **hypothèse**[1] :

hypo_egalite : expression_1 = expression_2.

---

1. a) En LEAN, on a le droit d'écrire :

   ```
   rw [hypo_egalite]
   ```

   Dans l'objectif, cela remplace une occurence de l'expression `expression_1` par `expression_2`.

   b) Si à la place, on tape :

   ```
   rw [← hypo_egalite]
   ```

   alors dans l'objectif, cela remplace une occurence de l'expression `expression_2` par `expression_1`.

   c) Si `autre_hypo` est une autre hypothèse du contexte dans laquelle `expression_1` apparaît, alors on a le droit de taper :

   ```
   rw [hypo_egalite] at autre_hypo
   ```

   Dans l'hypothèse `autre_hypo`, cela remplace une occurence de l'expression `expression_1` par `expression_2`.

2. a) Dans la démonstration, on écrit :

   > Puisque `expression_1 = expression_2` d'après l'hypothèse `hypo_egalite`, on doit en fait montrer que :
   >
   > *(Ecrire la proposition à démontrer en remplaçant « `expression_1` » par « `expression_2` »).*

   b) On peut aussi écrire la même rédaction que ci-dessus, en remplaçant `expression_2` par `expression_1`.

   c) Dans le dernier cas, on peut écrire :

   > Puisque `expression_1 = expression_2` d'après l'hypothèse `hypo_egalite`, alors l'hypothèse `autre_hypo` est équivalente à
   >
   > *(réécrire l'hypothèse `autre_hypo` en remplaçant « `expression_1` » par « `expression_2` »).*

---

[1] Les noms `hypo_egalite` et les expressions précises de l'égalité dépendront des cas de figure.

3 Négations, preuves par l'absurde et par contraposée.

Pour que les commandes vues dans cette section fonctionnent correctement, vous devrez importer le module adéquat. Tapez

```
import Mathlib.Logic.Basic
```

au début de votre fichier.

Si l'objectif est de la forme...

Objectif : `¬ P`

1. En LEAN, on a le droit d'écrire [1] :

   ```
   intro hypothese_P
   ```

   Cela ajoute l'hypothèse `hypothese_P` au contexte (cette hypothèse affirme que `P` est vraie), et remplace l'objectif par `False`.

2. Dans la démonstration, on a le droit d'écrire :

   On doit montrer que P est fausse. Supposons par l'absurde que P est vraie, et appelons « `hypothese_P` » cette hypothèse. On va en déduire une contradiction.

[1] On n'est pas obligé de raisonner par l'absurde ! Voir l'encadré suivant.

Objectif : `¬ P`

(où `P` est une assertion écrite avec des quantificateurs)

1. En LEAN, on a le droit d'écrire [1] :

   ```
   push_neg
   ```

   Cela réécrit l'objectif en utilisant les règles classiques

   $$\neg(\exists x \in E, P(x)) \Leftrightarrow (\forall x \in E, \neg P(x))$$

   $$\neg(\forall x \in E, P(x)) \Leftrightarrow (\exists x \in E, \neg P(x))$$

2. Dans la démonstration, on a le droit d'écrire :

   Le résultat à démontrer est équivalent à *(écrire l'assertion reformulée avec les règles de réécriture précédente)*. On va montrer le résultat sous cette nouvelle forme.

3.1 Preuves par contraposée

Si l'objectif est de la forme...

Objectif : $\boxed{\texttt{P} \to \texttt{Q}}$

---

1. En LEAN, on a le droit d'écrire :

   ```
   contrapose
   ```

   Cela remplace l'objectif par ¬(Q)→ ¬(P).

2. Dans la démonstration, on a le droit d'écrire :

   Raisonnons par contraposée et montrons que $\neg(Q)$ implique $\neg(P)$.