ESISAR 1A P2027 - MA121 2022-2023 - HA PREUVE

Le 12/01/2023 - Salle B141

DUREE : 1h15

L'épreuve se déroule sans calculatrice ni documents. Tout échange de matériel est interdit. Le sujet est recto/verso.

Instructions générales

Après avoir démarré votre poste de travail (Linux), vous vous connecterez avec comme identifiant : `userir` et comme mot de passe : `userir`.

Les exercices ci-après seront réalisés dans le langage L∃∀N , à l'aide de l'outil Web en ligne "Lean Live Web Editor".

- Démarrez le navigateur Firefox (En haut à gauche de l'écran : clic sur "Applications" puis "Web Browser").
- Dans la barre d'adresse, accédez à l'URL : `http://192.168.130.150:8080`
- Effacez l'exemple affiché dans la fenetre d'édition et commencez à travailler
- Sauvegardez régulièrement votre travail
- Le bouton "Save" sauvegarde votre code source dans le fichier `/home/userir/Downloads/_test.lean`

Cinq minutes avant la fin de l'épreuve :

- Renommez ce fichier `_test.lean` en `votre_login.lean` (par ex. pour Jacques Durand : `durandj.lean`). Pour cela, cliquez en haut à gauche de l'écran sur "Applications" puis "File Manager" puis sur le dossier "Downloads" puis clic droit sur `_test.lean` puis clic sur "Rename...".
- Dans le navigateur, accédez à l'outil `http://192.168.130.150` puis cliquez sur "Sélection du fichier", naviguez dans `/home/userir/Downloads`, choisissez votre fichier `votre_login.lean` (ex : `durandj.lean`) et n'oubliez pas de cliquer sur "Envoie".

Attention : comme indiqué ci-dessus, le rendu de l'examen est consitué d'UN SEUL fichier. Pour éviter les conflits de nommage, vous structurerez ce fichier à l'aide de "namespaces", comme ci-contre :

```
import data.nat.basic
import data.set.basic
import data.real.basic

-- ------------------ EXERCICE 1 ------------------
namespace ex1
  --votre reponse a l'exercice 1
  theorem ex_1_1       -- a completer ...
end ex1

-- ------------------ EXERCICE 2 ------------------
namespace ex2
  variables {E F G:Type}
  definition injective  (f: E → F) : Prop  := ∀ (u:E), ∀ (v:E), f u=f v → u=v
  -- votre reponse a l'exercice 2 ...
end ex2

-- ------------------ EXERCICE 3 ------------------
namespace ex3
  -- votre reponse a l'exercice 3 ...
end ex3

-- ------------------ EXERCICE 4 ------------------
namespace ex4
  -- votre reponse a l'exercice 4...
end ex4
```

Afin de rendre un code source sans erreur, il est fortement recommandé de

- Utiliser le mot clé `sorry` pour remplacer les termes de preuve que vous ne parvenez pas à former ;
- Commenter toutes les lignes qui ne sont pas acceptées par L∃∀N (sont considérées comme des commentaires toutes les lignes débutant par `--`).

Rappel : si besoin : les commandes `#check nom_theo` et `#check @nom_theo` permettent d'afficher l'énoncé d'un théorème de la librairie portant le nom `nom_theo`.

Exercice 1

1. Enoncez puis démontrez dans L∃∀N le théorème suivant, que vous nommerez `ex_1_1` :
   "Pour toute proposition P on a : P implique P".
2. Enoncez puis démontrez dans L∃∀N le théorème suivant, que vous nommerez `ex_1_2` :
   "Pour toutes propositions P et Q, si (P et Q) alors (P ou Q)".

Exercice 2

On donne les déclarations suivantes permettant de définir trois types E, F et G et la notion d'injectivité d'une application de type $E \to F$.

```
variables {E F G:Type}
definition injective  (f: E → F) : Prop  := ∀ (u:E), ∀ (v:E), f u =f v → u=v
```

1. Enoncez puis démontrez dans L∃∀N un théorème, nommé `ex_2_1`, affirmant que la composée de deux applications injectives est une application injective.
2. Enoncez puis démontrez dans L∃∀N un théorème, nommé `ex_2_2`, affirmant que pour toutes applications $f : E \to F$ et $g : F \to G$, si $g \circ f$ est injective alors f est injective.

Exercice 3

Pour cet exercice vous importerez la librairie définissant les entiers naturels, en insérant la ligne ci-dessous au début de votre document :

```
import data.nat.basic
```

1. Donnez, dans L∃∀N , la définition d'un entier naturel n divisible par un entier p. Précisément, si `n` et `p` sont de type $\mathbb{N}$, le terme `divisible n p` doit être de type `Prop` et traduire le fait que n est divisible par p.
2. Enoncez, puis démontrez, dans L∃∀N , un théorème nommé `ex_3_2` affirmant que la somme de deux nombres entiers naturels divisibles par un entier naturel p est divisible par p.

   Vous pourrez utiliser le théorème `right_distrib` disponible dans la librairie. Vous pourrez éventuellement avoir besoin de certains des outils suivants : `congr_arg` ; ▶ ; `eq.refl` ; `( ).symm`

Exercice 4

Pour cet exercice vous importerez les librairies définissant les ensembles, et les nombres réels, en insérant les lignes ci-dessous au début de votre document :

```
import data.set.basic
import data.real.basic
```

1. Donnez, dans L∃∀N , la définition d'un majorant d'une partie de $\mathbb{R}$:

   ```
   definition est_majorant (A:set ℝ) (m:ℝ) : Prop :=  -- a compléter
   ```

2. Donnez , dans L∃∀N , la définition du plus grand élément d'une partie de $\mathbb{R}$:

   ```
   definition est_pge  (A:set ℝ) (m:ℝ) : Prop :=  -- a compléter
   ```

3. Enoncez, puis démontrez, dans L∃∀N , un théorème nommé `ex_4_3` affirmant que (s'il existe) le plus grand élément d'une partie de $\mathbb{R}$ est unique.

   Vous pourrez utiliser le théorème `le_antisymm` disponible dans la librairie.

Annexe : saisie des symboles

Symbole	Raccourci
∀	\all
∃	\ex
→	\r
↔	\lr
¬	\not

Symbole	Raccourci
∧	\and
∨	\or
⩾	\ge
∘	\o
λ	\la

Symbole	Raccourci
$\mathbb{N}$	\N
$\mathbb{R}$	\R
⟨	\<
⟩	\>
≤	\le

Symbole	Raccourci
∈	\in
α	\alp
▶	\t
ε	\eps
ℓ	\ell