

TP 2

Programmation fonctionnelle en COQ- GALLINA, encore des preuves en logique propositionnelle

Fichier fourni : lc_tp2.v

Objectifs :

- Écrire des fonctions sur les listes qui seront utilisées dans les TP suivants,
- Encore quelques preuves en logique propositionnelle.

2.1 Retour sur les listes

On définit le type inductif `nlist` des listes de `nat`.

```
Inductive nlist : Set :=  
| nnil : nlist  
| ncons : nat -> nlist -> nlist.
```

On ajoute des notations confortables.

```
Infix ":@" := ncons.  
Notation "[]" := nnil.
```

EXERCICE 1 ▶

Définir la fonction `concat` qui concatène deux `nlist` `l1` et `l2` (par récursion sur `l1`).

```
Fixpoint concat (l1 l2 : nlist) : nlist :=  
[].
```

On note `++` en notation infix pour la concatenation.

```
Infix "++" := concat.
```

EXERCICE 2 ▶

Définir la fonction `length` qui retourne la longueur d'une `nlist`.

EXERCICE 3 ▶

Définir la fonction `appartient` qui retourne vrai si un `nat` appartient à une `nlist`.

2.2 Modélisations et preuves en logique propositionnelle

EXERCICE 4 ▶

Modéliser l'exercice de TD *Georges aime la logique*, prouver que Georges aime la logique.

Le ou - suite

- Introduction du \wedge : `split`

EXERCICE 5 ▶

Modéliser l'exercice de TD *Zoé va à Paris*, prouver que Zoé va à Paris.

Le non

- La notation not : `unfold not`
- La notation not en hypothèse : `unfold not [nom de l'hypothèse avec ~]`

EXERCICE 6 ▶

Montrer $(\sim P \vee \sim Q) \rightarrow \sim(P \wedge Q)$.

- Si on a `toto` et `~toto` dans les hypothèses : le but est alors résolu par `contradiction`

EXERCICE 7 ▶

Montrer $P \rightarrow \sim P \rightarrow Q$.

Le tiers-exclu On introduit la règle de tiers-exclu.

Context (Tiers_exclu: forall X: Prop, X \vee \sim X).

- Pour utiliser le tiers-exclu, c'est-à-dire pour avoir deux sous buts, un avec `toto` en hypothèse, l'autre avec `~toto`: `destruct (Tiers_exclu toto)`.

EXERCICE 8 ▶

Montrer $P \vee \sim P$.

EXERCICE 9 ▶

Montrer $((P \rightarrow Q) \rightarrow P) \rightarrow P$.

EXERCICE 10 ▶

Modéliser l'exercice de TD *Frodon est triste*, prouver que Frodon est triste.

EXERCICE 11 ▶

Modéliser l'exercice de TD *Gérard est déprimé*, prouver que Gérard est déprimé.

EXERCICE 12 ▶

Montrer $(\sim\sim P \rightarrow P) \wedge (P \rightarrow \sim\sim P)$.

Pour l'un des deux sens on aura besoin du tiers-exclu et, en remarquant qu'on peut déduire `False` des hypothèses, de la simplification `exfalso`.

2.3 Encore des preuves en logique propositionnelle

EXERCICE 13 ▶

Montrer $(P \vee Q) \rightarrow (Q \vee P)$.

EXERCICE 14 ▶

Montrer $(P \rightarrow Q \rightarrow R) \leftrightarrow (P \wedge Q \rightarrow R)$.

EXERCICE 15 ▶

Montrer $(P \rightarrow Q) \wedge (P \rightarrow R) \leftrightarrow (P \rightarrow Q \wedge R)$.