

TP2/3 - Fonctions récursives en Isabelle/HOL

On se propose de définir une librairie de manipulation d'ensembles représentés par des listes sans doublons. Définissez les fonctions et vérifiez les propriétés demandées dans les questions suivantes.

TP2 la vérification ne sera effectuée qu'en recherchant des contre-exemples finis (`nitpick`);

TP3 les propriétés seront démontrées en utilisant l'assistant de preuve. A l'issue du TP3, vous pourrez aussi vous exercer à la recherche efficace de contre-exemples sur la théorie `tp3Bis.thy`

1 Construction des ensembles

Exercice 1 Définissez la fonction `member` qui teste l'appartenance d'un élément à une liste.

Exercice 2 Définissez le prédicat (fonction dont le co-domaine est booléen) `isSet` qui teste qu'une liste représente bien un ensemble, i.e. que la liste ne comporte pas de doublons.

Exercice 3 Définissez la fonction `clean` qui supprime les doublons d'une liste.

Exercice 4 Définissez et vérifiez la propriété attendue entre `member` et `clean`.

Exercice 5 Définissez et vérifiez la propriété attendue entre `isSet` et `clean`.

2 Suppression d'un élément

Exercice 6 Définissez la fonction `delete` qui supprime un élément dans un ensemble.

Exercice 7 Définissez et vérifiez les deux propriétés attendues sur `member` et `delete`.

3 Intersection

Exercice 8 Définissez la fonction `intersection` de deux ensembles représentés par des listes.

Exercice 9 Définissez et vérifiez la propriété attendue sur `member` et `intersection`.

Exercice 10 Vérifiez que le résultat de `intersection` satisfait le prédicat `isSet`.

4 Union

Exercice 11 *Définissez la fonction `union` de deux ensembles représentés par des listes.*

Exercice 12 *Définissez et vérifiez la propriété attendue sur `member` et `union`.*

Exercice 13 *Vérifiez que le résultat de `union` satisfait le prédicat `isSet`.*

5 Egalité

Exercice 14 *Définissez la fonction `equal` permettant de tester l'égalité de deux ensembles.*

Exercice 15 *Définissez et vérifiez la propriété attendue sur `member` et `equal`.*

6 Compléments pour le TP3

Sur cette théorie relativement simple, les preuves sont faisables dans un temps limité. Cela ne sera plus le cas pour les prochains TPs et les preuves seront optionnelles. A défaut de faire les preuves, il sera nécessaire de savoir augmenter la puissance de déduction de Nitpick et Quickcheck pour s'assurer que vos lemmes n'admettent pas de contre-exemples finis simples. C'est l'objectif de cet exercice.

1. Copiez et chargez dans Isabelle le fichier `/share/m1info/ACF/TP3/tp3Bis.thy` ;
2. A l'aide de Nitpick et Quickcheck trouvez les contre-exemples pour les 4 lemmes de cette théorie.

7 Bonus (TP2/TP3)... pour les motivés

Exercice 16 *Définissez une fonction `sort` réalisant un tri par insertion sur les listes de naturels de type `nat`. Définissez et démontrez les propriétés attendues sur `sort`.*