

TP1 - Logique en Isabelle/HOL

Fichiers du projet : `tp1.thy` et `tp1bis.thy`

1 Préambule

1.1 Mise en place de l'environnement Isabelle/HOL

- Dans une fenêtre terminal, tapez la commande : `/share/m1info/Isabelle/configTP`
- Dans une fenêtre Terminal, tapez `source ~/.bashrc`

1.2 Première théorie et lancement d'Isabelle/HOL

- Créez un répertoire de travail pour ACF et copiez-y le fichier `/share/m1info/ACF/TP1/tp1.thy`.
- Pour lancer Isabelle/HOL, dans une fenêtre Terminal, tapez `isabelle`.
- Dans Isabelle, chargez votre copie du fichier `tp1.thy`.

1.3 Remarque sur la syntaxe Isabelle/HOL

En Isabelle/HOL, les termes, types, formules et équations de définition de fonctions doivent être entourés de double quotes : `"`.

2 Logique propositionnelle

Avec Isabelle/HOL, complétez le fichier `tp1.thy` avec les propriétés suivantes.

Exercice 1 Redémontrez les lois de De Morgan suivantes :

- $A \wedge (B \vee C) \longleftrightarrow (A \wedge B) \vee (A \wedge C)$
- $\neg(A \wedge B) \longleftrightarrow \neg A \vee \neg B$

Exercice 2 Il existe en Écosse un club très fermé qui obéit aux règles suivantes :

1. tout membre non écossais porte des chaussettes rouges,
2. tout membre portant des chaussettes rouges porte un kilt,
3. les membres mariés ne sortent pas le dimanche,
4. un membre sort le dimanche si et seulement s'il est écossais,
5. tout membre qui porte un kilt est écossais et est marié,
6. tout membre écossais porte un kilt.

Ce club peut-il accepter des membres ? Si l'on retire la première règle que cela change-t-il ?

3 Logique du premier ordre

Dans cette partie, certaines propriétés, même si elle sont vraies, ne pourront être démontré par `apply auto`. Dans ce cas, contentez-vous de vérifier qu'elles n'admettent pas de contreexemple avec `nitpick`.

Exercice 3 *Pour chaque formule, dites si elle est valide, satisfaisable ou contradictoire. Dans le cas où elle est satisfaisable ou contradictoire, proposez une hypothèse à ajouter pour la rendre valide. Dans ces formules, les opérateurs $+$, $*$, $>$ et \leq portent sur les entiers naturels.*

- $\forall x y z. x + y > x + z \longrightarrow x + x > y + z$
- $\forall x y. x + y \leq x * y$
- $\forall x y z. x > y \wedge z > 0 \longrightarrow x * z > y * z$
- $\exists x. P(f(x)) \longrightarrow \forall x. P(f(x))$

Exercice 4 *Redémontrez la commutativité et l'associativité de l'addition pour les entiers naturels. Redémontrez l'existence d'un élément neutre pour l'addition.*

Exercice 5 *Re-vérifiez si l'opérateur `append` de concaténation de listes est commutatif et associatif. Vérifiez s'il admet un élément neutre.*

Exercice 6 *Définissez et vérifiez les propriétés attendues sur les fonctions suivantes :*

- `append` et `length`
- `append` et `map`
- `List.member` et `append`

4 Votre premier “proof contest” sur ProvingForFun

Après les cours 3 et 4 vous aurez accès à deux compétitions spécifiques ACF sur le site ProvingForFun. Ceci est une préparation pour comprendre comment cela fonctionne. Sur la page du cours (google : ACF irisa), dans la section **Related pages** vous trouverez un lien vers **ACF on ProvingForFun**. Créez vous un compte sur ce site et participez à la première épreuve initiatique **First try**. Pour chaque épreuve, vous téléchargez un fichier `Defs.thy`, un fichier `Submission.thy` et un fichier `Check.thy`. Placez les trois dans le même répertoire. Le seul fichier à compléter est `Submission.thy`, avec des preuves et/ou avec des définitions de fonctions de façon à ce que tous les lemmes soient démontrables et terminés par `done`. A défaut les lemmes pourront être terminés par `sorry`. Les lemmes terminés par `sorry` ne rapporteront pas de points. Le fichier `Defs.thy` contient des (éventuelles) définitions supplémentaires. Le fichier `Check.thy` permet de savoir quels sont les lemmes qui vont être effectivement vérifiés et rapporter des points. Quand vous êtes satisfait par votre `Submission.thy` déposez-la sur le serveur. Vous aurez ensuite accès au détail de vos points et à votre classement (leaderboard).

5 Bonus... pour aller plus loin dans les preuves

Pour l'instant nous n'avons vu que la tactique de preuve `apply auto`. Si vous voulez connaître dès maintenant plus d'outils et de techniques pour mener à bien les preuves vous pouvez vous entraîner avec le fichier `/share/m1info/ACF/TP1/tp1bis.thy`.