Travaux dirigés et pratiques
Types énumérés (TD 3)

9 mars 2007

1 Les booléens

L'ensemble des booléens est un exemple classique. Chargez la bibliothéque à l'aide de la commande

```
Require Export Bool.
```

Observez la définition de ce type en tapant

```
Print bool.
```

(attention à ne pas confondre true avec True, false avec False).
Ecrire la fonction `my_xor` qui calcule le ou exclusif de deux booléens. (La fonction prédéfinie s'appelle `xorb`). Démontrer l'associativité de l'opération `my_xor`.

2 Les isométries du triangle équilatéral.

2.1 Les rotations.

On considère l'ensemble des 3 rotations qui conservent le triangle èquilatèral :

```
Inductive Rot : Set :=  R0 : Rot | R1 : Rot | R2 : Rot.
```

On définit la composition de deux rotations :

```
Definition RoR : Rot -> Rot -> Rot :=
fun r1 r2 : Rot =>
match r1 with
  | R0 => r2
  | R1 => match r2 with
          | R0 => R1
          | R1 => R2
          | R2 => R0
          end
  | R2 => match r2 with
```

```
            | R0 => R2
            | R1 => R0
            | R2 => R1
          end
end.
```

Démontrer les lemmes suivants :

```
Lemma RoRNilpotent : forall r : Rot, RoR r (RoR r r) = R0.

Lemma RoRCommut : forall r t : Rot, RoR r t = RoR t r.
```

2.2 Les symétries.

On définit l'ensemble `Sym` des trois symétries indicées par le sommet invariant : `Sa`, `Sb` et `Sc`.
Ecrire cette définition et l'opération de composition de deux symétries `SoS`.
Démontrer les lemmes suivants :

```
Lemma SoSInvolutive : forall s : Sym, SoS s s = R0.

Lemma SoSnonCommut : forall s t : Sym, s <> t -> SoS s t <> SoS t s.
```

On complète avec les définitions des produits d'une symétrie et d'une rotation :

```
Definition RoS : Rot -> Sym -> Sym :=
fun (r1 : Rot) (s2 : Sym)  =>
match r1 with
  | R0 => s2
  | R1 => match s2 with
            | Sa => Sc
            | Sb => Sa
            | Sc => Sb
          end
  | R2 => match s2 with
            | Sa => Sb
            | Sb => Sc
            | Sc => Sa
          end
end.

Definition SoR : Sym -> Rot -> Sym :=
fun (s1 : Sym) (r2 : Rot)  =>
match s1 with
  | Sa => match r2 with
            | R0 => Sa
            | R1 => Sb
            | R2 => Sc
          end
  | Sb => match r2 with
            | R0 => Sb
            | R1 => Sc
            | R2 => Sa
          end
  | Sc => match r2 with
            | R0 => Sc
            | R1 => Sa
            | R2 => Sb
          end
end.
```

2.3 Les isométries.

On définit les isométries du triangle par le type inductif suivant :

```
Inductive Isom : Set :=
  | Cr : Rot -> Isom
  | Cs : Sym -> Isom.
```

Écrivez l'opération de composition sur les isométries :

```
Definition rond : Isom -> Isom -> Isom :=
  fun i1 i2 : Isom => ...
```

Démontrer le lemme suivant :

```
Lemma power_sixt : forall i : Isom,
 rond i (rond i (rond i (rond i (rond i i)))) = Cr R0.
```

2.4 Transformons les sommets du triangle.

Soient `A`, `B` et `C`, les trois sommets du triangle. Déclarez-les dans un type énuméré `Sommet` de type `Set`.
Définir la fonction `app` de type

```
 Isom -> Sommet -> Sommet
```

qui associe à une isométrie et à un sommet, l'image de ce sommet par l'isométrie.
Montrer le lemme suivant :

```
Lemma comp_app : forall (f g : Isom) (x : Sommet),
app f (app g x) = app (rond f g) x.
```

3 Connecteurs logiques de Coq

Nous avons vu la semaine dernière comment définir les connecteurs logiques de façon imprédicative. En Coq, les connecteurs sont fournis dans le langage (en réalité, ils sont définis de façon inductive, mais nous verrons ça plus tard). Le *et* s'écrit avec l'opérateur infixe `/\`, le *ou* avec l'opérateur infixe `\/` et le *non* avec l'opérateur préfixe `~`.

Nous avions défini la semaine dernière les lemmes d'introduction et éventuellement d'élimination pour chaque connecteur. On retrouve ces lemmes sous forme de tactique dans Coq. Les voici résumées :

- les éliminations de tous les connecteurs se font par un elim.
- introduction du et : tactique `split`. L'usage de cette tactique casse le but en deux sous-buts.
- introduction du ou : tactiques `left` et `right`. Servent à choisir quelle membre de l'alternative on va prouver.
- introduction du non : tactique `intro` (`~P` est un raccourci pour `P -> False`).

3.1 Le retour du TD2

En guise d'échauffement, reprenez quelques théorèmes du TD2 suivant votre humeur, récrivez-les en utilisant les connecteurs logique de Coq et démontrez les en utilisant les tactiques de Coq.

3.2 Le club très très très fermé des écossais

Il existe en Écosse un club très fermé qui obéit aux règles suivantes :
- tout membre non écossais porte des chaussettes rouges ;
- tout membre porte un kilt ou ne porte pas de chaussettes rouges ;
- les membres mariés ne sortent pas le dimanche ;
- un membre sort le dimanche si et seulement s'il est écossais ;
- tout membre qui porte un kilt est écossais et marié ;
- tout membre écossais porte un kilt.

Proposez une formalisation de ce club en Coq et démontrez qu'il ne contient personne.