

TP3

Introduction à Coq : Ltac, extraction

Exercice 1

1. Que fait la tactique ? :

```
Require Import Rbase.  
Open Scope R_scope.
```

```
Ltac split_Rmult :=  
  match goal with  
  | |- ((?X1 * ?X2)%R <> 0%R) =>  
    apply Rmult_integral_contrapositive; split; try split_Rmult  
  end.
```

2. Montrer que $\forall x : \mathbf{R}, x \neq 0 \rightarrow 3 * x * 2 \neq 0$

Exercice 2 Ecrire une tactique qui permette de montrer simplement que :

$\forall x : \mathbf{R}, 0 < x \rightarrow 0 < 3 * x * 2$

Exercice 3 Montrer le lemme suivant, l'extraire en Ocaml puis regarder le fichier ml produit.
Lemma double: forall n: nat, {m: nat | m=(plus n n)}.

Exercice 4 [Devoir à rendre le 14/12/15]

Démontrer en Coq :

Lemma sur_2: forall x : nat, {y : nat & {x = 2 * y} + {x = 2 * y + 1}}.

Extraire le programme correspondant en Caml. Rendre la preuve mathématique, la preuve Coq et le programme Ocaml extrait.

Question subsidiaire : tester la fonction `sur_2` du code Ocaml.

Indications :

Commencer par réfléchir à la bonne valeur de y. Suivant les cas (propriété sur x), vous pourrez choisir le cas gauche ou droit. Tout ce dont vous avez besoin pour faire la preuve en Coq se trouve dans les fichiers `Div2.v` et `Even.v` de la bibliothèque standard.