Contrôle continu - Ingénierie de la preuve

Durée : 1h30

Les notes de cours, de travaux dirigés et de travaux pratiques sont autorisées. Le sujet comporte 2 pages et les deux parties sont complètement indépendantes. Les règles logiques utiles pour ce contrôle sont redonnées à la fin du sujet. Le barème est donné à titre indicatif.

On s'attachera à soigner la présentation, en particulier lors des démonstrations par induction et on sera vigilant quant à la syntaxe lors de l'écriture de fragment de code devant être accepté par `Coq`*.*

1 Questions de cours (8 pts)

Question 1 Que signifie le mot-clé `struct` dans une définition par point-fixe ? Expliquer le rôle de ce mot-clé dans la définition de `plus` (que l'on rappellera).

Question 2 On considère la définition en `Coq` de la fonction de soustraction sur les entiers naturels, à savoir :

```
Fixpoint minus (n m:nat) {struct n} : nat :=
  match n with
  | O => O
  | S k => match m with
             | O => S k
             | S l => minus k l
           end
  end.
```

Ecrire les règles de calcul associées à cette définition. On rappelle que les règles demandées sont celles qui s'appliquent lors de l'appel à la tactique `simpl`.

Question 3 On rappelle les types de quelques constantes :

```
or_ind : forall A B P : Prop, (A -> P) -> (B -> P) -> A \/ B -> P
or_intror : forall A B : Prop, B -> A \/ B
or_introl : forall A B : Prop, A -> A \/ B
```

Précisez les types des deux termes de preuve suivants :

```
fun (A B : Prop) (HAB : A -> B) (HnB : ~ B) (HA : A) => HnB (HAB HA)

fun (A B : Prop) (HAB : A \/ B) =>
or_ind (fun HA : A => or_intror B HA) (fun HB : B => or_introl A HB) HAB
```

Question 4 Construisez un terme de preuve pour la formule suivante :

$$\forall A\ B\ C : \texttt{Prop}, (A \to B \to C) \to (A \to B) \to (A \to C)$$

2 Définitions inductives (12 pts)

On veut représenter en `Coq` les entiers rationnels. Pour cela, on choisit d'utiliser des couples d'entiers $(n, m) \in N^2$ et un booléen (qui représentera le signe de la fraction : `true` pour les positifs et `false` pour les négatifs). Afin d'éviter la possibilité de diviser par zéro, on interprète le couple (n, m) comme étant la fraction $\frac{n}{m+1}$.

```
Inductive rationnel : Set := crat : ...
```

Question 5 Précisez le type de l'unique constructeur d'un nombre rationnel en utilisant les types prédéfinis `nat` et `bool`.

Question 6 Programmez par simple filtrage l'opération `opp` de calcul de l'opposé d'un nombre rationnel.

Question 7 Programmez l'opération d'addition de deux nombres rationnels. Déduire de la question précédente l'opération de soustraction de deux nombres rationnels.

Question 8 Programmez l'opération de multiplication de deux nombres rationnels.

Question 9 Programmez une fonction qui teste si les deux nombres rationnels fournis en argument sont égaux. On souhaite bien évidemment que cette fonction renvoie `true` quand on l'appelle avec les arguments $\frac{2}{4}$ et $\frac{1}{2}$.

Question 10 Peut-on démontrer que $\frac{2}{4} = \frac{1}{2}$ où `=` est l'égalité habituelle de `Coq` ?

Question 11 Programmez une fonction de normalisation d'un nombre rationnel. On supposera qu'il existe une fonction `gcd : nat -> nat -> nat` de calcul du pgcd de deux nombres naturels. De quelle autre opération a-t-on besoin ? Comment programmer l'opération de normalisation de manière sûre ?

Question 12 Enoncez une propriété exprimant la commutativité de l'opération d'addition de deux rationnels. Est-elle prouvable dans `Coq` ? Est-il nécessaire pour cela que les rationnels soient normalisés ? Détaillez les tactiques à appliquer pour démontrer ce théorème dans le cas où il est prouvable.