Modélisation et Programmation – Session 1 – 16/12/2019

Bureau d'études – 1h30 – Documents électroniques autorisés (contenu sur moodle et résultats des Travaux Pratiques)

Prénom : Nom : Groupe :

— Télécharger depuis moodle l'archive `source-etudiants.tgz`
— Désarchiver son contenu avec la commande : `tar xzvf source-etudiants.tgz`
— Vous obtenez un répertoire nommé `source-etudiants`
— Renommer ce répertoire sous la forme `source-etudiants_Nom1_Nom2` (en remplaçant Nom1 et Nom2 par le nom des deux membres du binôme par ordre alphabétique). Par exemple, si les membres sont Jacques Requin et Marc Pantel, vous utiliserez la commande : `mv source-etudiants source-etudiants_Pantel_Requin`
— Compiler la bibliothèque avec la commande : `coqc Naturelle.v`
— En fin de séance, vous rendrez sur moodle l'archive contenant le répertoire renommé.

Exercice 1 Soient A, B et C des variables propositionnelles, vous devrez montrer avec l'outil Coq (fichier `coq-exercice-1.v`) **en utilisant la déduction naturelle constructive (c'est-à-dire sans les règles du tiers-exclu (TE dans le résumé de cours) et de l'absurde classique (A dans le résumé de cours))** que la formule suivante est un théorème :

$$(A \to B \to C) \to (A \wedge B \to C)$$

1. Avec les commandes de la bibliothèque `Naturelle` utilisée en travaux pratiques
2. Avec les commandes classiques (sans la bibliothèque `Naturelle` utilisée en travaux pratiques)

Exercice 2 Soient A, B et C des variables propositionnelles, vous devrez montrer, avec l'outil Coq (fichier `coq-exercice-2.v`) **en utilisant la déduction naturelle classique (y compris les règles du tiers-exclu ou de l'absurde classique)**, que la formule suivante est un théorème :

$$\neg A \vee B \to \neg A \vee A \wedge B$$

1. Avec les commandes de la bibliothèque `Naturelle` utilisée en travaux pratiques
2. Avec les commandes classiques (sans la bibliothèque `Naturelle` utilisée en travaux pratiques)

Exercice 3 Nous considérons la spécification des listes munie des équations étudiées en cours et travaux dirigés :

(a) $\forall l \in \texttt{liste}(A).\ \texttt{append}(\texttt{Nil},l) = l$
(b) $\forall t \in A.\forall l, q \in \texttt{liste}(A).\ \texttt{append}(\texttt{Cons}(t,\ q),l) = \texttt{Cons}(t,\ \texttt{append}(q,l))$
(c) $\forall l \in \texttt{liste}(A).\ \texttt{append}(l,\texttt{Nil}) = l$
(d) $\forall l_1, l_2, l_3 \in \texttt{liste}(A).\ \texttt{append}(l_1,\texttt{append}(l_2,l_3)) = \texttt{append}(\texttt{append}(l_1,l_2),l_3)$

Nous complétons cette spécification par celles des fonctions qui assurent respectivement la séparation des éléments d'une liste et l'applatissement d'une liste de listes.
Le comportement de ces fonction peut être modélisé par les équations suivantes :

(e) $\texttt{flatten}(\texttt{Nil}) = \texttt{Nil}$
(f) $\forall t \in \texttt{liste}(A).\forall q \in \texttt{liste}(\texttt{liste}(A)).\ \texttt{flatten}(\texttt{Cons}(t, q)) = \texttt{append}(t, \texttt{flatten}(q))$
(g) $\texttt{split}(\texttt{Nil}) = \texttt{Nil}$
(h) $\forall t \in A.\forall q \in \texttt{liste}(A).\ \texttt{split}(\texttt{Cons}(t, q)) = \texttt{Cons}(\texttt{Cons}(t, \texttt{Nil}), \texttt{split}(q))$

Spécifier ces fonctions `flatten_spec` et `split_spec` dans l'outil Coq (fichier `coq-exercice-3.v`) sous la forme d'axiomes puis montrer que cette fonction satisfait les propriétés suivantes :

(i) $\forall l \in \texttt{liste}(A).\ \texttt{flatten}(\texttt{split}(l)) = l$

Programmer une implantation de la fonction `flatten_impl` puis prouver que cette implantation est correcte vis-à-vis de la spécification `flatten_spec` (théorème `flatten_correctness`).

Exercice 4 Prouver la correction totale du triplet de Hoare suivant (fichier `why3-exercice-4.mlw`) pour un programme calculant le produit de deux entiers positifs A et B. Nous vous suggérons d'exploiter $\texttt{R} = \frac{\texttt{I} \times (\texttt{I}+1)}{2}$ comme invariant et $\texttt{N} - \texttt{I}$ comme variant. Vous compléterez l'invariant si nécessaire pour construire la preuve. Vous indiquerez dans le fichier les commandes Why3 utilisées pour construire la preuve.
$\{\texttt{N} \geq 0\}$

```
R := 0;
I := 0
while (I < N) do
        I := I + 1;
        R := R + I
od;
```

$$\{\mathtt{R} = \frac{\mathtt{N} \times (\mathtt{N} + 1)}{2}\}$$