Modélisation et Architecture – Session 1, 3 et 5 – 20/11/2023

Bureau d'études – 1h30 – Documents autorisés

— Télécharger depuis moodle l'archive `be_2023_S1.tgz`
— Désarchiver son contenu avec la commande : `tar xzvf be_2023_S1.tgz`
— Vous obtenez un répertoire nommé `be_2023_S1`
— Renommer ce répertoire sous la forme `be_2023_S1_Nom1_Prénom1_Nom2_Prénom2` (en remplaçant Nom1, Prénom1, Nom2, Prénom2 par vos noms et prénoms dans l'ordre alphabétique sans caractères accentués). Par exemple, si les étudiants sont Jacques Requin et Pénélope Pieuvre, vous utiliserez la commande : `mv be_2023_S1 be_2023_S1_Requin_Pieuvre`
— Placez vous dans le répertoire `be_2023_S1_Requin_Pieuvre`
— 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 et B 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 :

$$((A \vee B) \wedge (\neg A)) \rightarrow (B \wedge (\neg A))$$

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, c'est-à-dire les commandes `intro`, `elim`, `exact`, `cut`, `split`, `left`, `right`, `destruct`, `absurd` et `apply`)

Exercice 2 Soient A et B 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 :

$$((A \rightarrow B) \rightarrow A) \rightarrow A$$

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, c'est-à-dire les commandes `intro`, `elim`, `exact`, `cut`, `split`, `left`, `right`, `destruct`, `absurd` et `apply` ainsi que l'axiome `classic`)

Exercice 3 Soit une structure de groupe $(E, \mathtt{e}, \mathtt{o})$ composée d'un ensemble E ; d'un opérateur binaire interne `o` associatif à gauche et à droite ; et d'un élément $\mathtt{e} \in E$, neutre à gauche et à droite pour `o`.
Compléter le fichier `coq_exercice_3.v` pour spécifier les propritétés d'associativité de `o` et de neutralité de `e`.
Nous considérons les spécifications des entiers naturels et des listes d'éléments du groupe $(E, \mathtt{e}, \mathtt{o})$ étudiées en cours et travaux dirigés.
Nous complétons ces spécifications par les fonctions de repliement à gauche $\mathtt{foldl}(v, l)$ et à droite $\mathtt{foldr}(v, l)$ des listes l pour la valeur v selon l'opérateur `o`. Le comportement de ces fonctions peut être modélisé par les équations suivantes :

(a) $\forall v \in E, \mathtt{foldl}(v, \mathtt{Nil}) = v$
(b) $\forall v \in E, \forall t \in E, \forall q \in \mathtt{liste}(E),\ \mathtt{foldl}(v, \mathtt{Cons}(t, q)) = \mathtt{foldl}(\mathtt{o}(v, t), q)$
(c) $\forall v \in E, \mathtt{foldr}(v, \mathtt{Nil}) = v$
(d) $\forall v \in E, \forall t \in E, \forall q \in \mathtt{liste}(E),\ \mathtt{foldr}(v, \mathtt{Cons}(t, q)) = \mathtt{o}(v, \mathtt{foldr}(t, q))$

Spécifier la fonction `foldl_spec` dans le fichier `coq_exercice_3.v`.
Montrer que cette fonction satisfait la propriété suivante :

(e) `foldl_append` : $\forall v \in E, \forall l_1, l_2 \in \mathtt{liste}(E).\ \mathtt{foldl}(v, \mathtt{append}(l_1, l_2)) = \mathtt{foldl}(\mathtt{foldl}(v, l_1), l_2)$

Programmer une implantation de la fonction `foldl_impl` puis prouver que cette implantation est correcte vis-à-vis de la spécification `foldl_spec` (théorème `foldl_correctness`).
Spécifier la fonction `foldr_spec` dans le fichier `coq_exercice_3.v`.
Montrer que cette fonction satisfait la propriété suivante :

(f) `foldl_foldr` : $\forall v \in E, \forall l \in \mathtt{liste}(E).\ \mathtt{foldl}(v, l) = \mathtt{foldr}(v, l)$

Exercice 4 Prouver la correction totale du triplet de Hoare suivant (fichier `why3_exercice_4.mlw`) pour un programme calculant les termes de la suite de Fibonacci. Nous vous suggérons d'exploiter $\texttt{C} = S_{\texttt{I}} \wedge \texttt{S} = S_{\texttt{I+1}}$ comme invariant et $\texttt{N} - \texttt{I}$ comme variant. Vous compléterez l'invariant si nécessaire pour construire la preuve.
$\{N \geq 0\}$

```
P := 0;
C := 0;
S := 1;
I := 0;
while (I < N) do
        P := C;
        C := S;
        S := C + P;
        I := I + 1
od
```

$\{\texttt{C} = S_{\texttt{N}}\}$