Université Paris Diderot (Paris 7) - UFR Informatique

Génie Logiciel (L3) : Examen

24 Mai 2013

Durée : 2h30
NB : Seuls les documents manuscrits sont autorisés.

1 Exercice 1

On considère le type des séquences d'entiers $\mathsf{Seq}[\mathbb{Z}]$ dont les constructeurs sont
- la séquence vide $\varepsilon : \mathsf{Seq}[\mathbb{Z}]$,
- l'opération d'ajout en tête (à gauche) d'un élément (entier) à une séquence $\bullet : \mathbb{Z} \times \mathsf{Seq}[\mathbb{Z}] \to \mathsf{Seq}[\mathbb{Z}]$.

Etant donnée une séquence s, on désigne par $|s|$ la longueur de la séquence s. On considère que $|\varepsilon| = 0$. Pour tout entier naturel $i \leq |s|$, on désigne par $s[i]$ l'élément se trouvant à la position i dans s. (On considère que les positions sont indexées de 1 à $|s|$ de gauche à droite.)

Finalement, étant donnée une séquence s, le multi-ensemble des éléments de s est dénoté par $\mathsf{ME}(s)$.

Soit la fonction

[illegible]

2 Exercice 2

On considère un type de séquences $\mathsf{Seq}[\mathbb{E}]$ sur un ensemble $\mathbb{E}$, dont les constructeurs sont la séquence vide (ε) et l'opération d'ajout d'un élément à gauche ($\bullet$), comme dans l'exercice précédent.

On suppose aussi que l'on dispose d'une fonction $\circ : \mathsf{Seq}[\mathbb{E}] \times \mathbb{E} \to \mathsf{Seq}[\mathbb{E}]$ qui ajoute un élément à droite d'une séquence, c'est-à-dire, pour une séquence $s = a_1 \cdots a_n$ et un élément e, produit la séquence $s = a_1 \cdots a_n e$.

L'opération de concaténation de deux séquences est notée $\odot$. On a alors $(a_1 \cdots a_n) \odot (b_1 \cdots b_m) = a_1 \cdots a_n b_1 \cdots b_m$.

On considère alors la fonction $\mathsf{Inverse} : \mathsf{Seq}[\mathbb{E}] \to \mathsf{Seq}[\mathbb{E}]$ qui, étant donnée une séquence d'éléments s, produit une séquence qui est l'image miroir (l'inverse) de s, c'est-à-dire, si $s = a_1 \cdots a_n$, alors $\mathsf{Inverse}(s) = a_n \cdots a_1$. (L'inverse de la séquence vide est elle même.)

Cette fonction est définie récursivement de la manière suivante :

$$\mathsf{Inverse}(\varepsilon) = \varepsilon \quad (1)$$

$$\mathsf{Inverse}(a \bullet s) = \mathsf{Inverse}(s) \circ a \quad (2)$$

On définit l'implémentation itérative suivante de cette fonction :

$\mathsf{InvIter}\ (s) =$
    $t := s$;
    $t' = \varepsilon$;
    $\mathsf{tant\ que}\ \ t \neq \varepsilon\ \ \mathsf{faire}$
        $a := \mathsf{tete}(t)$;
        $t := \mathsf{reste}(t)$;
        $t' := a \bullet t'$;

Question 1 : Montrer en utilisant les règles de la logique de Hoare le fait suivant :

$$\{\mathsf{vrai}\}\ \mathsf{InvIter}(s)\ \{\mathsf{Inverse}(t') = s\}$$

Indication : Trouver un invariant de la boucle qui relie t', t et s à l'aide des fonctions $\mathsf{Inverse}$ et $\odot$.

3 Exercice 3

On considère deux processus (identiques) P_1 et P_2 qui utilisent durant leurs exécutions une ressource partagée R. Cette utilisation doit se faire en exclusion mutuelle, c'est-à-dire, qu'à tout moment, il y

Question 1 : Modéliser par des systèmes de transitions chacun des processus P_1, P_2, et S. Définir une manière de les composer en parallèle (en précisant les actions de synchronisation) qui garantit l'exclusion mutuelle entre P_1 et P_2, c'est-à-dire qu'il n'est jamais possible dans ce système que les deux processus soient simultanément dans leurs états *Utilisation* respectifs. (Donner le système de transition obtenu en composant en parallèle P_1, P_2, et S.)

Question 2 : Est-ce que le système construit à la question précédente satisfait la propriété suivante : "A chaque fois qu'un processus P_i est en attente, il passera inévitablement dans l'état d'utilisation" ?

On considère maintenant un scheduler S' attribue la ressource en alternance aux deux processus : d'abord à P_1 puis à P_2, ensuite à P_1, puis à P_2, et ainsi de suite, dans cet ordre.

Question 3 : Modéliser le nouveau scheduler S' et décrire le système de transition correspondant à la