UFR IMA Master 1 Mars 2006

Option OGL

Durée : devoir à la maison à rendre le 21 mars 2006.
Documents : tous documents autorisés.

EXERCICE (12 points)

On peut définir le prédicat de terminaison et le prédicat avant-après d'une substitution S, notés respectivement $\mathsf{trm}(S)$ et $\mathsf{prd}_x(S)$, de la manière suivante :

$$\begin{array}{lcl} \mathsf{trm}(S) & \hat{=} & [S](x = x) \\ \mathsf{prd}_x(S) & \hat{=} & \neg[S](x' \neq x) \end{array}$$

Intuitivement, S termine (i.e. $\mathsf{trm}(S)$ est vrai) si les pré-conditions sont respectées et qu'il n'y a pas d'erreur à l'exécution. De même, si x désigne les variables du système, alors $\mathsf{prd}_x(S)$ caractérise l'ensemble des couples (x, x') tels que x' désigne les valeurs de l'état après exécution de S et x désigne celles avant exécution de S.

Par application de ces définitions sur les principales substitutions primitives on obtient les résultats suivants :

$\mathsf{trm}(x := e)$	$\hat{=} \mathit{true}$	$\mathsf{prd}_x(x := e)$	$\hat{=} x' = e$
	$\hat{=}$	$\mathsf{prd}_{x,y}(x := e)$	$\hat{=} x' = e \ \wedge \ y' = y$
$\mathsf{trm}(\mathsf{skip})$	$\hat{=} \mathit{true}$	$\mathsf{prd}_x(\mathsf{skip})$	$\hat{=} x' = x$
$\mathsf{trm}(P \mid S)$	$\hat{=} P \ \wedge \ \mathsf{trm}(S)$	$\mathsf{prd}_x(P \mid S)$	$\hat{=} P \Rightarrow \mathsf{prd}_x(S)$
$\mathsf{trm}(S_1 \parallel S_2)$	$\hat{=} \mathsf{trm}(S_1) \ \wedge \ \mathsf{trm}(S_2)$	$\mathsf{prd}_x(S_1 \parallel S_2)$	$\hat{=} \mathsf{prd}_x(S_1) \ \vee \ \mathsf{prd}_x(S_2)$
$\mathsf{trm}(P \Longrightarrow S)$	$\hat{=} P \Rightarrow \mathsf{trm}(S)$	$\mathsf{prd}_x(P \Longrightarrow S)$	$\hat{=} P \ \wedge \ \mathsf{prd}_x(S)$
$\mathsf{trm}(@z \cdot S)$	$\hat{=} \forall z \cdot \mathsf{trm}(S)$	$\mathsf{prd}_x(@z \cdot S)$	$\hat{=} \exists z \cdot \mathsf{prd}_x(S)$

▷ **Question 1** (9 points)

Calculez la terminaison et le prédicat avant-après des substitutions suivantes, et concluez chaque calcul par une phrase en français expliquant ce

que l'on peut en déduire :

- $S_1 \mathrel{\hat{=}} x :\in \mathbb{N}$
- $S_2 \mathrel{\hat{=}}$ **if** $x > 0$ **then** $x := x + 1$ **else** $x := x - 1$ **end**
- $S_3 \mathrel{\hat{=}}$ **pre** $x > 0$ **then** $x := y + 1$ **end**

▷ **Question 2** (3 points) On peut définir la fonction $FN(S)$, qui définit la forme normalisée de toute substitution S de la manière suivante :

$$FN(S) \mathrel{\hat{=}} \mathsf{trm}(S) \mid @x' \cdot (\mathsf{prd}_x(S) \Longrightarrow x := x')$$

Pour toute substitution S et prédicat R, on a la propriété suivante :

$$[S]R \Leftrightarrow [FN(S)]R$$

Vérifiez que cette propriété est vraie pour les substitutions $x := e$, $P \mid S$ et $S_1 \parallel S_2$.

PROBLEME (8 points)

On cherche à modéliser un distributeur de boissons. Les actions possibles de l'utilisateur sont mettre une pièce, choisir une boisson, prendre son gobelet ou annuler la commande. Nous définissons pour cela 3 ensembles : les boissons possibles, les pièces acceptées et les états possibles de la machine.

A cela sont ajoutées des constantes fixant un prix à chaque boisson, une valeur à chaque pièce et le montant maximum que la machine peut accepter. Notons qu'il n'est pas nécessaire de connaître ces valeurs. C'est pourquoi seul le prix de la boisson *rien* est donné.

Pour la partie dynamique de la machine, on s'intéresse à la boisson sélectionnée, au crédit disponible dans la machine, à l'état courant de la machine et à l'affichage indiquant le crédit restant.

▷ **Question 1** (2 points)

Dans la clause **properties**, on voudrait rajouter l'information suivante : la somme maximale acceptée par la machine est au moins supérieure au prix de la boisson la plus chère. Donnez un prédicat logique permettant de formaliser cette propriété.

machine $DistributeurBieres$
sets $BOISSONS = \{rien, Demi, Serieux, Formidable\}$;
$ETATSMACHINE = \{pret, distriEc\}$;
$PIECES$
constants $prix, valeur, SommeMax$
properties
$prix \in BOISSONS \longrightarrow \mathbb{N}$
$\wedge\ valeur \in PIECES \longrightarrow \mathbb{N}$
$\wedge\ prix(rien) = 0$
$\wedge\ SommeMax \in \mathbb{N}$
$\wedge\ \ldots$
variables $boisson, credit, etatc, affichage$
invariant
$boisson \in BOISSONS$
$\wedge\ credit \in 0..SommeMax$
$\wedge\ etatc \in ETATSMACHINE$
$\wedge\ affichage = credit - prix(boisson)$
$\wedge\ (etatc = pret \Rightarrow boisson = rien)$
initialisation $boisson := rien \parallel credit := 0 \parallel affichage := 0 \parallel etatc := pret$
operations
$mettre\_piece(pp) \mathrel{\hat{=}}$ **pre** $pp \in PIECES \wedge credit + valeur(pp) \leq SommeMax \wedge etatc = pret$ **then**
$credit := credit + valeur(pp) \parallel$
$affichage := credit + valeur(pp)$
end ;
$choix\_boisson(bb) \mathrel{\hat{=}}$ **pre** ...**then**
$boisson := bb \parallel$
$affichage := credit - prix(bb) \parallel$
$etatc := distriEc$
end ;
$prendre\_gobelet \mathrel{\hat{=}}$ **pre** $etatc = distriEc$ **then**
$boisson := \ldots \parallel$
$credit := \ldots \parallel$
$affichage := 0 \parallel$
$etatc := pret$
end ;
$annuler \mathrel{\hat{=}}\ \ldots$
end

▷ **Question 2** (3 points)

Précisez la précondition de l'opération $choix\_boisson(bb)$ de telle sorte que l'on ait les garanties suivantes : bb est une boisson qui peut être servie par la machine et la machine est dans l'état $pret$.

La précondition est elle suffisante pour que l'opération vérifie l'invariant ? Si non, complétez la.

▷ **Question 3** (1 point)

Complétez l'opération $prendre\_gobelet$.

▷ **Question 4** (2 points)

Décrivez l'opération $annuler$.