Génie Logiciel-Méthode B
Maîtrise d'Informatique 2001-2002

Examen Final - Feuille n^o 2
Méthode B
3 juin 2002

L'énoncé est composé de 3 pages.
Les notes de cours et de TD manuscrites (les photocopies sont interdites) ainsi que les supports de cours et de TD distribués cette année sont les seuls documents autorisés.
Cette partie devra être rédigée sur une copie séparée sur laquelle le numéro d'anonymat sera reporté

1 Substitutions (2 points)

La substitution CASE E OF EITHER C_1 THEN S_1 OR C_2 THEN S_2 END END dans laquelle E est une expression, C_1 et C_2 sont des listes de constantes et S_1 et S_2 sont des substitutions est définie comme une abréviation pour la substitution

$$((E \in \{C_1\}) \Longrightarrow S_1)[]((E \in \{C_2\}) \Longrightarrow S_2)$$

On suppose défini un type COULEURS=$\{$`pique`,`coeur`,`carreau`,`trèfle`$\}$

1. Soient S_1 la substitution définie par :

```
CASE x OF EITHER pique, trèfle THEN r := TRUE
          OR carreau, coeur THEN r := FALSE
          END
END
```

   et S_2 la substitution définie par :

```
CASE x OF EITHER trèfle THEN r := TRUE
          OR carreau THEN r := FALSE
          END
END
```

   Pour un prédicat P quelconque, expliciter les formules $[S_1]P$ et $[S_2]P$.
   Correction :
   $[S_1]P \equiv x \in \{\texttt{pique}, \texttt{trèfle}\} \Rightarrow [r := \texttt{TRUE}]P \wedge x \in \{\texttt{carreau}, \texttt{coeur}\} \Rightarrow [r := \texttt{FALSE}]P$
   $[S_2]P \equiv x \in \{\texttt{trèfle}\} \Rightarrow [r := \texttt{TRUE}]P \wedge x \in \{\texttt{carreau}\} \Rightarrow [r := \texttt{FALSE}]P$

2. A-t-on $S_1 \sqsubseteq S_2$? $S_2 \sqsubseteq S_1$? justifier.
   Correction :
   On remarque que $x \in \{\texttt{trèfle}\} \Rightarrow x \in \{\texttt{pique}, \texttt{trèfle}\}$ donc si $x \in \{\texttt{pique}, \texttt{trèfle}\} \Rightarrow [r := \texttt{TRUE}]P$ alors a fortiori $x \in \{\texttt{trèfle}\} \Rightarrow [r := \texttt{TRUE}]P$.
   De même si $x \in \{\texttt{carreau}, \texttt{coeur}\} \Rightarrow [r := \texttt{FALSE}]P$ alors $x \in \{\texttt{carreau}\} \Rightarrow [r := \texttt{FALSE}]P$. On en déduit $[S_1]P \Rightarrow [S_2]P$ pour tout P et donc que $S_1 \sqsubseteq S_2$ (S_2 raffine S_1).

Le contraire est faux :
$[S_1]r = r' \equiv x \in \{\mathtt{pique}, \mathtt{trèfle}\} \Rightarrow r' = \mathtt{TRUE} \wedge x \in \{\mathtt{carreau}, \mathtt{coeur}\} \Rightarrow r' = \mathtt{FALSE}$
$[S_2]r = r' \equiv x \in \{\mathtt{trèfle}\} \Rightarrow r' = \mathtt{TRUE} \wedge x \in \{\mathtt{carreau}\} \Rightarrow r' = \mathtt{FALSE}$
Si on prend $x = \mathtt{pique}$ et $r' = \mathtt{FALSE}$, alors $[S_2]r = r'$ est vérifié alors que $[S_1]r = r'$ est faux.

2 Modélisation (4 points)

On propose de modéliser un ensemble de biens qui peuvent être proposés au prêt dans une machine `Prêt`. Pour cela on se donne deux ensembles abstraits : BIEN qui représente les biens et PERSONNE pour représenter les personnes qui peuvent prêter ou bien emprunter un bien. On introduit également un ensemble énuméré TYPE dont les valeurs représentent la nature du bien prêté, ici on se limitera à trois sortes de biens : `voiture`, `livre` et `disque`.
Les variables de la machine `Prêt` sont `biens` qui représente l'ensemble des biens dans la base, `personnes` qui représente l'ensemble des personnes dans la base, `type_de_bien` qui associe à chaque bien de la base son type, `appartient_à` qui relie les biens à leur propriétaire et `a_pour_emprunteur` qui relie un bien à la personne qui l'emprunte. Un bien dans la base a toujours un unique propriétaire mais n'est pas pas forcément emprunté, s'il est emprunté, alors c'est au plus par une seule personne.

1. Écrire l'invariant de la machine `Prêt` en justifiant le type des variables (en particulier pour les relations, on précisera si ce sont des fonctions, si elles sont totales, partielles, injectives, surjectives ...)

   Correction :

   $\mathtt{biens} \subseteq \mathtt{BIEN} \wedge \mathtt{personnes} \subseteq \mathtt{PERSONNES}$
   $\wedge \mathtt{type\_de\_bien} \in \mathtt{biens} \longrightarrow \mathtt{TYPE}$
   $\wedge \mathtt{appartient\_a} \in \mathtt{biens} \longrightarrow \mathtt{personnes}$
   $\wedge \mathtt{emprunte} \in \mathtt{personnes} \nrightarrow \mathtt{biens}$

   `type_de_bien` est une fonction totale car tout bien est classé selon son type, la fonction n'est ni injective (deux biens peuvent avoir le même type), ni surjective (il peut ne pas y avoir de biens d'un certain type dans la base).

   `appartient_a` est une fonction totale car tout bien a un propriétaire, la fonction n'est ni injective (deux biens peuvent avoir le même propriétaire), ni surjective (une personne peut ne rien posséder).

   `emprunte` est une fonction car un bien est emprunté au plus par une personne, partielle car certains biens peuvent ne pas être empruntés, elle n'est pas injective (deux biens peuvent avoir le même emprunteur) ni surjective (une personne peut ne rien emprunter).

2. Donner des expressions du langage B pour représenter :
   (a) l'ensemble `disques` des disques de la base;
   (b) l'ensemble `déjà_loués` des objets qui sont déjà loués;
   (c) la relation `loue_a` entre deux personnes p et p' qui est vraie lorsque la personne loue au moins un bien à la personne p'.

   Correction :

   (a) $\mathtt{disques} = \mathtt{type\_de\_bien}^{-1}[\{\mathtt{disque}\}]$

(b) $\texttt{déjà\_loués} = dom(\texttt{a\_pour\_emprunteur})$

(c) $\texttt{loue\_a} = \texttt{appartient\_a}^{-1}; \texttt{a\_pour\_emprunteur}$

3. Donner les formules du langage B qui expriment les contraintes suivantes (on pourra utiliser les abréviations définies précédemment) :
   (a) une personne ne peut pas emprunter ce qui lui appartient;
   (b) une même personne ne peut emprunter strictement plus de 5 disques.

   Correction :

   (a) $\texttt{loue\_a} \cap \texttt{id} = \emptyset$

   (b) $\forall x.x \in \texttt{personnes} \Rightarrow card(\texttt{a\_pour\_emprunteur}^{-1}[\{x\}] \cap \texttt{disques}) \leq 5$

4. Écrire une opération `location` qui étant donné un bien qui n'est pas encore loué et une personne qui remplit les conditions pour louer ce bien, introduit la nouvelle location dans la base.

   Correction : *Les préconditions à la location sont :*
   – *le bien b est dans la base ainsi que la personne p;*
   – *le bien b n'est pas déjà loué;*
   – *le bien b n'appartient pas à p;*
   – *si b est un disque alors le nombre de disque qu'il emprunte déjà est strictement inférieur à 5.*

   La substitution consiste à mettre à jour la relation $\texttt{a\_pour\_emprunteur}$. *On a donc :*

$$\begin{aligned}\texttt{location}(p,b) \hat{=} \ & PRE\ p \in \texttt{personnes} \wedge b \in \texttt{biens} \\ & \quad \wedge\ b \notin \texttt{déjà\_loués} \wedge \neg \texttt{appartient\_à}(b,p) \\ & \quad \wedge\ (b \in \texttt{disques} \Rightarrow card(\texttt{a\_pour\_emprunteur}^{-1}[\{p\}] \cap \texttt{disques}) < 5) \\ & THEN\ \texttt{a\_pour\_emprunteur}(b) := p \\ & END\end{aligned}$$

5. Énoncer les obligations de preuve à vérifier pour que la condition énoncée à la question 3a soit préservée par l'opération de location.

Correction : *Il faut montrer que la précondition de l'opération* `location` *implique la préservation de la propriété 3a.*
Soit I la propriété :$\forall x.x \in \texttt{personnes} \Rightarrow card(\texttt{a\_pour\_emprunteur}^{-1}[\{x\}] \cap \texttt{disques}) \leq 5$,
la préservation s'écrit :

$$\begin{aligned}& p \in \texttt{personnes} \wedge\ b \in \texttt{biens} \wedge\ b \notin \texttt{déjà\_loués}\ \wedge \neg \texttt{appartient\_à}(b,p) \\ & \wedge\ (b \in \texttt{disques} \Rightarrow card(\texttt{a\_pour\_emprunteur}^{-1}[\{p\}] \cap \texttt{disques}) < 5) \\ & \wedge\ I \\ & \Rightarrow [\texttt{a\_pour\_emprunteur}(b) := p]I\end{aligned}$$

La propriété $[\texttt{a\_pour\_emprunteur}(b) := p]I$ *à montrer se simplifie en :*
$\forall x.x \in \texttt{personnes} \Rightarrow card((\texttt{a\_pour\_emprunteur}\ \lhd\!\!- b \mapsto p)^{-1}[\{x\}] \cap \texttt{disques}) \leq 5$
Pour la montrer il faudrait raisonner par cas sur x $(x = p \vee x \neq p)$ *et sur* b $(b \in \texttt{disques} \vee b \notin \texttt{disques})$