# Université de Sherbrooke, Département d'informatique

IGL501 : Méthodes formelles en génie logiciel, Examen périodique
Professeur : Marc Frappier, mardi 7 octobre 2013, 15h30 à 18h20, local D4-2022

Documentation permise. La correction est, entre autres, basée sur le fait que chacune de vos réponses soit *claire*, c'est-à-dire lisible et compréhensible pour le lecteur; *précise*, c'est-à-dire exacte et sans erreur; *concise*, c'est-à-dire qu'il n'y ait pas d'élément superflu; *complète*, c'est-à-dire que tous les éléments requis sont présents.

Pondération :

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 30 pts | 2 | 10 pts | 3 | 20 pts | 4 | 40 pts | Total | 100 pts |

Prénom: _____ Nom: _____ Matricule : _____

Signature: _____

1. (12 pts) Soit A ={a1,a2,a3} et B ={b1,b2,b3}. Indiquez si les expressions suivantes sont vraies ou fausses, Justifiez votre réponse.

1. $\{a1 \mapsto a2\} \in P(A \times A)$; vrai
2. $\{\{a1 \mapsto a2\}\} \subseteq A \leftrightarrow A$; vrai
3. $\{a1 \mapsto a2,\ a1 \mapsto a3\} \in A \nrightarrow A$; faux; deux images pour a1
4. $\{a1 \mapsto a1,\ a2 \mapsto a3,\ a3 \mapsto a2\} \in A \rightarrowtail A$; vrai
5. $A \nrightarrow B \subseteq A \leftrightarrow A$ ; ~~vrai; une fonction est aussi une relation~~; faux codomaines distincts
6. $A \rightarrowtail B \subseteq A \rightarrow \mathbf{A}$ ; faux; ~~une fonction partielle $f \in A \rightarrowtail B$ avec $dom(f) \in A$ n'appartient pas à $A \rightarrow B$,~~ codomaines distincts

2. (8 pts) Calculez le résultat des expressions suivantes.

1. $dom(\{a1 \mapsto a2, a1 \mapsto a3, a3 \mapsto a2\}) = \{a1,a3\}$
2. $\{a1 \mapsto a2, a1 \mapsto a3, a3 \mapsto a2\};\{a1 \mapsto a2, a1 \mapsto a3, a3 \mapsto a2\} = \{ a1 \mapsto a2 \}$
3. $\{a1\} \lhd \{a1 \mapsto a2, a1 \mapsto a3, a3 \mapsto a2\} \rhd \{a2\} = \{a1 \mapsto a2 \}$
4. $\{a1 \mapsto a2, a1 \mapsto a3, a3 \mapsto a2\} <+ \{ a1 \mapsto a1\} = \{ a1 \mapsto a1, a3 \mapsto a2 \}$

3. (25 pts) Pour chaque opération ainsi que pour l'initialisation de la machine suivante, indiquez si l'invariant est préservé, en faisant le calcul de l'obligation de preuve.

**MACHINE** *Q3*

**CONSTANTS** *A*

**PROPERTIES** $A = 0..1$

**VARIABLES** $x, z$

**INVARIANT** $x \in 1..3$
$\wedge\ z \in A \rightarrowtail\!\!\!\!\!\rightarrow A$

**INITIALISATION**
**ANY** $a$ **WHERE** $a \in A$ **THEN** $z := \mathrm{id}(A) \parallel x := a$ **END**
Réponse: faux, car le cas a = 0 viole l'invariant

**OPERATIONS**

**op1** =
**PRE** $x \in 1..2$
**THEN**
$x := x + 1$
|| **ANY** $a$ **WHERE** $a \in A$ **THEN** $z(a) := a$ **END**
**END;**
Réponse: faux, dans le cas où z(a) /= a avant l'exécution de l'opération

**op2**($y1,y2$) = **PRE** $y1 \in A \wedge y2 \in A$ **THEN** $z(y1) := z(y2) \parallel z(y2) := z(y1)$ **END END**
Réponse: vrai, car cela permute deux valeurs dans la fonction

**op3**($y$) =
**PRE** $y \in -1..1 \wedge y < x$ **THEN**
**SELECT** $y > 0$ **THEN** $x := y+1$
**WHEN** $y > x$ **THEN** $x := y$
**END**
**END;**
Réponse: vrai

**op4** = **CHOICE** $x := 0$ **OR** $x := 1$ **END**
Réponse: faux, car x:= 0 viole l'invariant

**END**

4. (45 pts) Écrivez une spécificartion B pour le système suivant. Le système gère un système de transactions d'une firme de courtage la bourse. Chaque courtier est soumis à une limite quotidienne pour le total de ses transactions. Cette limite quotidienne varie en fonction du courtier, mais elle ne doit pas dépasser une certaine limite globale fixée par la firme de courtage. Une transaction qui dépasse une certaine limite doit être approuvée par le superviseur du courtier; cette limite pour une transaction varie en fonction du courtier. Chaque courtier a un superviseur. On distingue les transactions complétées (ie, approuvées ou qui n'ont pas besoin d'approbation à cause de leur montant), des transactions en attente d'approbation. Le respect des limites quotidiennes d'un courtier doit prendre en compte autant les transactions en attente que les transactions complétées. Pour simplifier, votre modèle n'a pas a prendre en compte l'évolution du temps; il ne doit gérer que les transactions d'une journée; il n'a pas a prendre en compte plusieurs journées à la fois; les limites fixées ne varient pas durant une journée. Le système doit garder une trace des approbations et des refus, aux fins de vérification externe par des vérificateurs de l'Autorité des marchés financiers. Vous aurez besoin de l'opérateur **somme**(*S*), définit ci-dessous, qui retourne la somme des éléments de *S*; il retourne 0 si *S* est vide. Complétez la spécification suivante.

**MACHINE** *Courtier*
**SETS**
*COURTIER* = {*c1*,*c2*};
*SUPERVISEUR* = {*s1*,*s2*};
*TRANSACTION* = {*t1*,*t2*,*t3*,*t4*,*t5*}
**DEFINITIONS**
$somme(S) == \sum(z).(z \in S \mid z)$
**VARIABLES**
*courtier*, *superviseur*, *supervisePar*, *limiteQuotidienne*, *limiteParTransaction*, *limiteGlobale*, *transaction*, *montant*, *auteur*, *transactionEnAttente*, *approbation*, *refus*

**INVARIANT**

| | |
|---|---|
| $courtier \subseteq COURTIER$ | /* ensemble des courtiers */ |
| $\wedge$ $superviseur \subseteq SUPERVISEUR$ | /* ensemble des superviseurs */ |
| $\wedge$ $superviserPar \in courtier \rightarrow superviseur$ | /* indique le superviseur du courtier */ |
| $\wedge$ $limiteQuotidienne \in courtier \rightarrow \mathbf{NAT}$ | /* indique la limite quotidienne du courtier */ |
| $\wedge$ $limiteGlobale \in \mathbf{NAT}$ | /* indique la limite quotidienne du système */ |
| $\wedge$ $limiteParTransaction \in courtier \rightarrow \mathbf{NAT}$ | /* indique la limite d'une transaction d'un courtier */ |
| $\wedge$ $transaction \subseteq TRANSACTION$ | /* ensemble des transactions */ |
| $\wedge$ $montant \in transaction \rightarrow \mathbf{NAT}$ | /* montant d'une transaction*/ |
| $\wedge$ $auteur \in transaction \rightarrow courtier$ | /* indique le courtier qui a effectué la transaction */ |
| $\wedge$ $transactionEnAttente \subseteq transaction$ | /* transactions qui doivent être approuvées */ |
| $\wedge$ $approbation \in transaction \twoheadrightarrow superviseur$ | /* indique le superviseur qui a approuvé une transaction */ |
| $\wedge$ $refus \in transaction \twoheadrightarrow superviseur$ | /* indique le superviseur qui a refusé une transaction */ |

/* La somme des transactions ne dépasse pas la limite du courtier */
$\wedge \forall(c).(c \in courtier \Rightarrow somme(montant[auteur^{-1}[\{c\}]-\mathbf{dom}(refus)]) \leq limiteQuotidienne(c))$

/* La limite quotidienne d'un courtier ne dépasse pas la limite de la firme */
$\wedge \forall(c).(c \in courtier \Rightarrow limiteQuotidienne(c) \leq limiteGlobale)$

/* Les transactions en attente ont besoin d'être approuvées */

$\wedge\ \forall(t).\ (\ t \in transactionEnAttente$
$\Rightarrow$
$montant(t) > limiteParTransaction(auteur(t))$
$)$

/* Les transaction complétées sont approuvées si nécessaire */
$\wedge\ \forall(t).\ (\quad t \in transaction$
$\wedge\ t \notin transactionEnAttente$
$\wedge\ montant(t) > limiteParTransaction(auteur(t))$
$\Rightarrow$
$t \mapsto superviseePar(auteur(t)) : (approbation \cup refus)$
$)$
/* Une transaction est soit en attente, soit approuvée ou refusée */
$\wedge\ (\mathbf{dom}(approbation) \cup \mathbf{dom}(refus)) \cap transactionEnAttente = \varnothing$
$\wedge\ \mathbf{dom}(approbation) \cap \mathbf{dom}(refus) = \varnothing$

**INITIALISATION**
*… rien à compléter ici …*

**OPERATIONS**

/* Cette opération effectue une transaction par le courtier $c$ pour le montant $m$. Le no de la transaction est choisi de manière aléatoire par le système. */
**transiger**($c$,$m$) =
**PRE**
$c \in courtier$
$\wedge\ m \in \mathbf{NAT1}$
$\wedge\ somme(montant[auteur^{-1}[\{c\}]]) + m \leq limiteQuotidienne(c)$
$\wedge\ TRANSACTION - transaction \neq \varnothing$
**THEN**
**ANY** $t$
**WHERE**
$t \in TRANSACTION - transaction$
**THEN**
$transaction := transaction \cup \{t\}$
|| **IF**
$m > limiteParTransaction(c)$
**THEN**
$transactionEnAttente := transactionEnAttente \cup \{t\}$
**END**
|| **auteur**($t$) := $c$
|| **montant**($t$) := $m$
**END**
**END**;

/* Cette opération permet à un superviseur $s$ d'approuver la transaction $t$. */
**approuver**($s$,$t$) =

**PRE**
$s \in superviseur$
$\wedge\ s = supervisePar(auteur(t))$
$\wedge\ t \in transactionEnAttente$
**THEN**
$transactionEnAttente := transactionEnAttente - \{t\}$
$\|$ **approbation**$(t) := s$
**END**;

**refuser**$(s,t)$ =
**PRE**
$s \in superviseur$
$\wedge\ s = supervisePar(auteur(t))$
$\wedge\ t \in transactionEnAttente$
**THEN**
$transactionEnAttente := transactionEnAttente - \{t\}$
$\|$ **refus**$(t) := s$
**END**
**END**

**5. (10 pts)** Pour chaque question suivante, indiquez si l'opération abstraite Op1 est raffinée par l'opération concrète Op2, avec l'invariant de collage x = x' (l'invariant de collage est l'invariant *J* du REFINEMENT, c'est-à-dire la machine concrète). L'invariant *I* de la machine abstraite est x ∈ -1 .. 1. Notez qu'il n'y a pas de variable de sortie ici et pas de précondition, donc l'obligation de preuve devient simplement

$$I \wedge J \Rightarrow [op2]\ \neg\ [op1]\ \neg J$$

a)

Op1 = ANY z WHERE z ∈ {−1,1} THEN x := z END
Op2 = CHOICE x' := 1 OR x' := -1 END

Solution:

$[CHOICE\ x' := 1\ OR\ x' := -1\ END]\neg[\ ANY\ z\ WHERE\ z \in \{-1,1\}\ THEN\ x := z\ END\ ]\neg(\ x = x')$
$\Leftrightarrow\ [CHOICE\ x' := 1\ OR\ x' := -1\ END]\neg(\ \forall\ z\ .\ z \in \{-1,1\} \Rightarrow [x := z]\ \neg(\ x = x'))$
$\Leftrightarrow\ \ [x' := 1]\neg(\ \forall\ z\ .\ z \in \{-1,1\} \Rightarrow [x := z]\ \neg(\ x = x'))$
$\wedge\ [x' := -1]\neg(\ \forall\ z\ .\ z \in \{-1,1\} \Rightarrow [x := z]\ \neg(\ x = x'))$
$\Leftrightarrow\ \ \neg(\ \forall\ z\ .\ z \in \{-1,1\} \Rightarrow \neg(\ z = 1))$
$\wedge\ \neg(\ \forall\ z\ .\ z \in \{-1,1\} \Rightarrow \neg(\ z = 1))$
$\Leftrightarrow\ \ \exists z\ .\ z \in \{-1,1\} \wedge\ z = 1$
$\wedge\ \exists z\ .\ z \in \{-1,1\} \wedge z = -1$
$\Leftrightarrow\ vrai$

b)

Op1 = CHOICE x' := 1 OR x' := -1 END
Op2 = ANY z WHERE z ∈ −1..1 THEN x := z END

Solution:

[ ANY z WHERE z ∈ –1..1 THEN x' := z END ]¬[CHOICE x := 1 OR x := -1 END] ¬( x = x')
⇔ ∀ z . z ∈ –1..1 => ¬ ([x:=1] ¬( x = z) ∧ [x:=-1] ¬( x = z))
⇔ ∀ z . z ∈ –1..1 => ¬ (¬( 1 = z) ∧ ¬( -1 = z))
⇔ ∀ z . z ∈ –1..1 => (1 = z ∨ -1 = z)
⇔ *faux* pour z = 0