INP ENSEEIHT

3A TLS-SEC et Mastère "Sécurité Informatique" – 3ème Année. 2019-2020

Développement de logiciels sécurisé

18 Décembre 2020.

Durée 1h30

Documents autorisés

Partie 1. Questions de cours

La méthode B propose de décrire des spécifications de systèmes comprenant des variables, des invariants, une initialisation des variables ainsi que des évènements.

A. Une obligation de preuve déchargée (prouvée), peut-elle être considérée comme un théorème ?

B. Expliquer pourquoi les obligations de preuve peuvent-elles générées automatiquement ?

C. Quel mécanisme de preuve est associé à la preuve d'un invariant ?

D. Une machine décrit des invariants et des théorèmes. Tout deux décrivent des propriéts de la machines. Pourquoi théorèmes et invariants ont-ils été différenciés ?

E. Lorsqu'un variant est introduit, il doit décroitre grâce aux évènements dit "convergents". Quelles propriétés peut-on garantir grâce à la présence d'un variant décroissant ?

F. Le raffinement établit une relation de simulation entre deux machines. Expliquer le rôle de la relation de simulation ?

Il est demandé de répondre en quelques lignes seulement.

Pour les parties 2, 3 et 4, si vous souhaitez introduire des précisions, veuillez les décrire dans votre copie.

Partie 2. Une machine Event-B

On veut spécifier un système de gestion de comptes bancaires. Les comptes, pris dans un ensemble A possèdent un solde compris entre 0 et une constante C entier naturel positif.

CONTEXT
$COMPTES$
SETS
C // pour définir l'ensemble des comptes
CONSTANTS
S // Pour définir le solde maximal pour un compte
AXIOMS
$S > 0$
END

La machine Event-B *GestCompte* ci-dessous décrit ce modèle.

Un ensemble *comptes* de comptes bancaires est défini dans l'état (clause *variables*) de ce système. Il est possible d'obtenir le solde de chaque compte par la fonction totale *solde* également définie dans la même clause.

À l'initialisation, les ensembles *comptes* et *solde* sont vides.

MACHINE
$GestCompte$
SEES
CTX

VARIABLES
$comptes, solde$

INVARIANT
INV1 : $comptes \subseteq C$
INV2 : $solde \in comptes \longrightarrow \mathbb{N}$

INITIALISATION
$comptes := \emptyset$
$solde := \emptyset$

EVENTS
$Ouvrir = \ldots$

$Fermer = \ldots$

$Créditer = \ldots$

$Débiter = \ldots$

Les évènements suivants sont introduits.

— Ouvrir et Fermer qui permettent d'ouvrir ou de fermer un compte bancaire

— Créditer et débiter qui permettent de créditer ou de débiter une somme donnée sur un compte donné.

Questions

1.1. Compléter la machine abstraite GestCompte en écrivant un invariant inv3 qui indique que le solde de tous les comptes est compris entre 0 et S

1.2. Compléter la machine abstraite GestCompte en exprimant la spécification formelle des 4 évènements Ouvrir, Fermer, Créditer et Débiter.
Vous veillerez à respecter l'invariant dans cette spécification.

1.3. Ecrire les obligations de preuve d'invariants associées aux évènements Fermer et Débiter.

1.4. Justifier, en quelques lignes ou par une démonstration, la correction des évènements Fermer et Débiter par rapport à l'invariant proposé en question 1.1.

Partie 3. Un raffinement Event-B

On souhaite prendre en compte le virement entre deux comptes bancaires. Le squelette de la machine GestCompte_Ref_1 ci-dessous introduit l'évènement Virer.

Cet évènement considère deux comptes sour et dest ainsi qu'une somme som et procède au transfert de cette somme depuis le compte sour vers le compte dest en retirant som de sold(sour) pour l'ajouter à solde(dest).

La machine ci-dessous décrit le squelette du raffinement demandé

MACHINE
GestCompte_Ref_1
REFINES
GestCompte
VARIABLES
...

INVARIANT
...

INITIALISATION
...

EVENTS
Ouvrir = ...

Fermer = ...

Créditer = ...

Débiter = ...

Virer = ...

Questions

2.1. Quel évènement est raffiné par l'évènement Virer ?

2.2. Compléter le raffinement ci-dessous en décrivant l'évènement Virer.
Il est inutile de recopier les variables, invariant, initialisation, événements qui ne changent pas par rapport à la machine GestCompte

2.3. Le raffinement obtenu est-il correct ? Justifier votre réponse.

Partie 4. Encore du raffinement

L'objectif de ce raffinement est d'introduire les clients titulaires de comptes et une caractérisation plus fine des comptes bancaires.

Ainsi, en plus des ouverture, fermeture, retrait et virement nous souhaitons introduire :

— les clients CLIENTS , personnes qui peuvent être titulaires de comptes bancaires

— le type de compte CPTYPE , avec des comptes courant et des comptes livret.

Pour cela, on introduit le contexte ci-dessous qui définit ces nouveaux ensembles

CONTEXT
$COMPTES\_EXT$

EXTENDS
$COMPTES$

SETS
$CLIENTS$ Ensemble de clients
$CPTYPE$ Ensemble de types de comptes bancaires

CONSTANTS
$COURANT$
$LIVRET$

AXIOMS
$PARTITION(CPTYPE, COURANT, LIVRET)$
– les comptes sont soit des comptes courant ou livret

END

La machine obtenue par le raffinement précédent GestCompte_Ref_1 devra à son tour être raffinée pour prendre en compte les exigences nouvelles suivantes :

[Req1] Chaque compte est associé à un et un seul titulaire de compte

[Req2] Chaque compte est soit un compte courant, soit un compte livret

[Req3] Créditer un compte ne peut être effectué par le titulaire du compte (un client) seulement et uniquement sur un compte courant

[Req4] Débiter un compte ne peut être effectué par le titulaire du compte (un client) seulement et uniquement sur un compte courant

[Req5] Virer ne peut être réalisé qu'entre compte courant et livret d'un même titulaire de compte.

La machine ci-dessous décrit le squelette du raffinement demandé. Elle introduit deux variables propr et type pour associer, à un compte, un propriétaire (titulaire) et un type (courant ou livret) respectivement.

MACHINE
$GestCompte\_Ref\_2$
REFINES
$GestCompte\_Ref\_1$

SEES
$COMPTES\_EXT$

VARIABLES
$propr, type$

INVARIANT
INV4 : $propr \quad \in comptes \longrightarrow CLIENTS$
INV5 : $Type \quad \in comptes \longrightarrow CPTYPE$

INITIALISATION
...

EVENTS
Ouvrir Refines ... = ...

Fermer Refines ...= ...

Créditer Refines ...= ...

Débiter Refines ... = ...

Virer Refines ... = ...

Ici les invariants inv4 et inv5 formalisent les exigences Req1 et Req2 respectivement.

Questions

3.1. Compléter le raffinement GestCompte_Ref_2 pour prendre en compte les nouvelles exigences. Vous décrirez les évènements raffinés ainsi que les nouveaux évènements qui pourraient apparaître.

3.2. Le raffinement obtenu est-il correct ? Justifier votre réponse.