Travaux Dirigés No2 – Composants

Frédéric Peschanski

1er février 2017

Exercice : Spécifications bancaires

Dans cet exercice, on utilise le langage de spécifications semi-formelles vu en cours pour modéliser un système bancaire simplifié, décrit dans l'énoncé.

Question 1 : Compte bancaire

Spécifier un service Compte représentant un compte bancaire et offrant les fonctionnalités suivantes :

- Observateurs pour le nom, le numéro de compte, le solde, la limite autorisée de découvert, le montant éventuel de ce dernier (uniquement si le compte est à découvert),
- Observateurs pour tester si le compte est découvert ou provisionné
- Observateurs pour tester si le retrait d'une somme donnée est possible
- Un constructeur à partir d'un nom, d'un numéro de compte strictement positif et d'un montant de découvert autorisé
- Un constructeur à partir d'un autre compte
- Une opération de dépôt d'une somme positive sur le compte
- Une opération de retrait d'une somme positive depuis le compte, si ce retrait est autorisé

Question 2 : Propriétés

Le service Compte est-il selon-vous cohérent ? complet ? Expliquez ce qu'il faudrait faire pour démontrer cela formellement. Le service est-il activable ? Justifiez cette fois-ci de façon formelle.

Finalement, discutez de la convergence de chaque opérateur. Pour chaque opérateur convergent, modifiez la spécification en conséquence et justifiez formellement.

Question 3 : Agence

On donne ci-dessous la spécification partielle d'un service Agence qui centralise la gestion d'un ensemble de comptes.

```
service : Agence
observators :
    const nom : [Agence] → String
    numeros : [Agence] → Set<int>
    nbComptes : [Agence] → int
```

```
    compteExiste : [Agence] × int → bool
    getCompte : [Agence] × int → Compte
        pre getCompte(A,num) require compteExiste(A,num)
Constructors :
    init : String → [Agence]
        pre init(nom) require nom ≠ ""
Operators :
    // cf. question
Observations :
[invariants]
    nbComptes(A) = card(numeros(A)) // card(E) est le cardinal de l'ensemble fini E
    compteExiste(A,num) = num ∈ numeros(A)
[init]
    nom(init(nm)) = nm
    numeros(init(nm)) = ∅
```

On souhaite compléter le service avec deux opérations :

- operation de création de compte, à partir d'un nom, d'un numéro non utilisé et d'une autorisation de découvert
- operation de fermeture (définitive) de compte

Question 4 : Virements

On souhaite dans cette question ajouter une opération de virement de compte à compte à l'intérieur d'une même agence. Spécifier l'opération avec sa signature, sa précondition éventuelle ainsi que les observations associées.