Travaux Dirigés No4 – Raffinement

Frédéric Peschanski

9 février 2016

Exercice 1 : Système de contrôle de circulation sur un pont

Dans cet exercice, nous construisons un système simplifié de contrôle de circulation sur un pont reliant le continent (*mainland*) à une île (*island*). L'objectif est d'exploiter au mieux le raffinement et/ou l'inclusion de service.

Question 1 : Section de route et route limitée

Soit les services suivants :

service : RoadSection
types : int
observators :
    // nombre de voitures sur la section de route
    nbCars : [RoadSection] → int
Constructors :
    init : → [RoadSection]
Operators :
    enter : [RoadSection] → [RoadSection]
    leave : [RoadSection] → [RoadSection]
        pre leave(R) **require** nbCars(R) > 0
Observations :
[invariants]
    nbCars(R) ≥ 0
[init]
    nbCars(init()) $= 0$
[enter]
    nbCars(enter(R)) = nbCars(R) + 1
[leave]
    nbCars(leave(R)) = nbCars(R) - 1

service : LimitedRoad
refine : RoadSection
types : boolean, int
observators :
    // nombre max. de voitures sur la route
    const limit : [LimitedRoad] → int
    full : [LimitedRoad] → boolean
Constructors :
    init : int → [RoadSection]
        pre init(lim) **require** lim > 0
Operators :
    enter : [RoadSection] → [RoadSection]
        pre enter(R) **require** ¬full(R)
Observations :
[invariants]
    full(R) $\stackrel{\text{min}}{=}$ nbCars(R) = limit(R)
    nbCars(R) $\leq$ limit(R)
[init]
    nbCars(init(lim)) $= 0$
    limit(init(lim)) = lim

Question Quelles conditions faut-il vérifier pour que le raffinement proposé soit correct ? Ces conditions sont-elles vérifiées ? Si non donner une observation qui contredit le raffinement et proposer des solutions.

Question 2 : Modèle du pont

On souhaite définir un service Bridge représentant un service concret pour le modèle de pont à voie unique. Ce service est conçu comme un raffinement de LimitedRoad.

La spécification partielle de ce service est la suivante :

service : Bridge
refine : LimitedRoad
types : boolean,int
observators :
    nbIn : [Bridge] → int
    nbOut : [Bridge] → int
Constructors :
    init : int → [Bridge]
        pre init(lim) **require** lim > 0
Operators :
    enterIn : [Bridge] → [Bridge] *// entrée depuis le continent*
        pre enterIn(B) **require** ¬full(B)
    leaveIn : [Bridge] → [Bridge]
        pre leaveIn(B) **require** nbIn(B) > 0
    enterOut : [Bridge] → [Bridge] *// entrée depuis l'île*
        pre enterOut(B) **require** ¬full(B)
    leaveOut : [Bridge] → [Bridge]
        pre leaveOut(B) **require** nbIn(B) > 0
Observations :
[invariants]
    // COMPLETER LES INVARIANTS
[init]
    nbIn(init(lim)) = 0
    nbOut(init(lim)) = 0
    limit(init(lim)) = lim
[enterIn]
    nbIn(enterIn(B)) = nbIn(B) + 1
    nbOut(enterIn(B)) = nbOut(B)
[leaveIn]
    nbIn(leaveIn(B)) = nbIn(B) - 1
    nbOut(leaveIn(B)) = nbOut(B)
[enterOut]
    nbIn(enterOut(B)) = nbIn(B)
    nbOut(enterOut(B)) = nbOut(B) + 1
[leaveOut]
    nbIn(leaveOut(B)) = nbIn(B)
    nbOut(leaveOut(B)) = nbOut(B) - 1

L'objectif est de compléter ce service pour notamment assurer :
— la correction du raffinement de LimitedRoad
— la complétude du modèle Bridge

Exercice 2 : Héritage et services requis/fournis

On suppose un système de flot de données (*dataflow*) avec les services suivants :
— ServInt : service de production d'objets entiers relatifs de type Int
— ServReal : service de production d'objets réel de type Real
— ServComplex : service de production d'objets complexes de type Complex
— Dummy : un service annexe de nature indéterminée
On suppose que ServInt raffine ServReal qui raffine ServComplex [1].

Un composant de *dataflow* possède des entrées (services requis) et sorties (services fournis) de nombres. Le type d'un composant qui requiert les services $R_1, \ldots, R_n$ et fournis les services $P_1, \ldots, P_m$ sera notée $\langle R_1, \ldots, R_n \mid P_1, \ldots, P_m \rangle$.

Question 1 Les raffinements suivants sont ils corrects ? Si oui montrer un cas possible de substitution, si non donner un contre-exemple :

1. Le raffinement entre types de nombres selon leur inclusion ensembliste ne produit pas de résultats très probants en informatique, du fait de leur représentation en générale très différente, et surtout des types des opérations que l'on souhaite effectuer sur des nombres.

1. $\langle \mathsf{ServReal} \mid \mathsf{ServReal} \rangle$ raffine $\langle \mathsf{ServReal} \mid \mathsf{ServReal} \rangle$
2. $\langle \mathsf{ServNat} \mid \mathsf{ServReal} \rangle$ raffine $\langle \mathsf{ServReal} \mid \mathsf{ServReal} \rangle$
3. $\langle \mathsf{ServComplex} \mid \mathsf{ServReal} \rangle$ raffine $\langle \mathsf{ServReal} \mid \mathsf{ServReal} \rangle$
4. $\langle \mathsf{ServReal} \mid \mathsf{ServComplex} \rangle$ raffine $\langle \mathsf{ServReal} \mid \mathsf{ServReal} \rangle$
5. $\langle \mathsf{ServReal} \mid \mathsf{ServNat} \rangle$ raffine $\langle \mathsf{ServReal} \mid \mathsf{ServReal} \rangle$
6. $\langle \mathsf{ServComplex}, \mathsf{ServNat} \mid \mathsf{ServReal} \rangle$ raffine $\langle \mathsf{ServReal} \mid \mathsf{ServReal} \rangle$
7. $\langle \mathsf{ServReal} \mid \mathsf{ServNat}, \mathsf{ServComplex} \rangle$ raffine $\langle \mathsf{ServReal} \mid \mathsf{ServReal} \rangle$
8. $\langle \mathsf{ServReal} \mid \mathsf{ServReal} \rangle$ raffine $\langle \mathsf{ServReal}, \mathsf{Dummy} \mid \mathsf{ServReal} \rangle$
9. $\langle \mathsf{ServReal} \mid \mathsf{ServReal} \rangle$ raffine $\langle \mathsf{ServReal} \mid \mathsf{ServReal}, \mathsf{Dummy} \rangle$
10. $\langle \mathsf{ServReal}, \mathsf{Dummy} \mid \mathsf{ServReal} \rangle$ raffine $\langle \mathsf{ServReal} \mid \mathsf{ServReal} \rangle$
11. $\langle \mathsf{ServReal} \mid \mathsf{ServReal}, \mathsf{Dummy} \rangle$ raffine $\langle \mathsf{ServReal} \mid \mathsf{ServReal} \rangle$

Question 2 En déduire les conditions générales pour vérifier :

$$\langle R'_1, \ldots, R'_{n'} \mid P'_1, \ldots, P'_{m'} \rangle \text{ raffine } \langle R_1, \ldots, R_n \mid P_1, \ldots, P_m \rangle$$