TD 23. Premières Spécifications - Solutions

Jean-Louis BOULANGER

Décembre 2000

Résumé

Le but de ce premier TD est de manipuler les substitutions généralisées.

Mots Clés : Spécification, Substitution Généralisée.

Table des matières

1 Introduction

Les exercices de ce TD consistent à traduire un texte en français constitué d'affirmations sous la forme de machine abstraite.

2 exercice 1

2.1 Gestion d'un ensemble d'avion

On cherche a gérer la gestion des avions d'une compagnie aérienne. A chaque avion est associé un *IDENTIFICATEUR*.

Mettre en place une première spécifiation qui tient compte de cette première exigence et qui comporte les opérations suivantes :

- création d'un avion
- destruction d'un avion

```
MACHINE
   AVIONS

SETS
   AVION
;  IDENTIFICATEUR

VARIABLES
   avion
,  vol_id

INVARIANT
   avion <: AVION
&  vol_id : avion --> IDENTIFICATEUR

INITIALISATION
   avion, vol_id := {}, {}

OPERATIONS

   aa <-- creer_avion(vid) =
      PRE
         vid   :  IDENTIFICATEUR
      &  avion /= AVION
      THEN
         ANY
               oo
         WHERE
               oo : AVION - avion
         THEN
               aa         := oo
         ||    avion      := avion \/ {oo}
         ||    vol_id(oo) := vid
```

```
            END
        END;

    detruire_avion(aa) =
        PRE
            aa : AVION
        &   aa : avion
        THEN
                avion       := avion - {aa}
        END
END
```

Commentaires. Les symboles introduits ont la signification suivante :

- `<:` représente $\subseteq$,
- `A -> B` représente une fonction totale qui est définie sur tout son domaine de `A` vers `B` ($A \to B$),
- `x : A` représente $x \in A$,
- `{}` représente l'ensemble vide ($\emptyset$), on le voit ici l'ensemble vide est une valeur de fonction car en mathématiques une fonction est représentée par son graphe (un ensemble de couples),
- `A /= B` représente $A \neq B$,
- `A - B` représente $A \setminus B$,
- `A / B` représente $A \cup B$,
- `||` correspond à une indépendance dans l'éxecution des Substitutions Généralisées,
- `f(x)` représente la valeur de f en x.

Explication de la première opération : De manière informelle, l'opération attend en entrée quelque chose qui est de type `IDENTIFICATEUR` et renvoie un élément d'un ensemble noté `AVION - avion` qui correspond aux avions auxquels on n'a pas encore affecté de nom (par exemple un numéro de vol : AF240). Le choix de l'avion n'a pas d'importance à ce niveau de la spécification, on souhaite simplement que l'opérations nous en renvoie un.

C'est une solution, mais elle a un petit problème :

```
+------------+----+-----+-----+-----+-----+-----+-----+
| COMPONENT  | TC | POG | Obv | nPO | nUn | %Pr | BOC |
+------------+----+-----+-----+-----+-----+-----+-----+
| AVIONS     | OK | OK  |   4 |   8 |   2 |  75 | OK  |
+------------+----+-----+-----+-----+-----+-----+-----+
```

2.2 Gestion des pilotes

Voici une soution a notre problème.

```
MACHINE
    BDD
```

```
INCLUDES
   AVIONS
,  PERSONNES

VARIABLES
   bdd

INVARIANT
   bdd : les_personnes <-> avion

INITIALISATION
   bdd := {}

OPERATIONS

      pp <-- create_pilote =
      BEGIN
          pp <--Personne_Creer
      END;

      delete_pilote (pp) =
      PRE
        pp : LES_PERSONNES
      & pp : les_personnes
      THEN
           bdd := {pp} <<| bdd
        || Personne_Supprimer (pp)
      END;

      add_authorise(pp,aa) =
      PRE
         pp   : LES_PERSONNES
      &  pp   : les_personnes
      &  aa   : AVION
      &  aa   : avion
      THEN
         bdd := bdd \/ { pp |-> aa}
      END;

      delete_authorisation (pp,aa) =
      PRE
         pp   : LES_PERSONNES
      &  pp   : les_personnes
      &  aa   : AVION
      &  aa   : avion
      THEN
         bdd := bdd - {pp |-> aa}
      END;
```

```
      print_pilote (pp) =
      PRE
         pp   : LES_PERSONNES
      THEN
         skip
      END
END
```

```
+--------------------+----+-----+-----+-----+-----+-----+-----+
| COMPONENT          | TC | POG | Obv | nPO | nUn | %Pr | BOC |
+--------------------+----+-----+-----+-----+-----+-----+-----+
| AVIONS             | OK | OK  |   4 |   8 |   2 |  75 | OK  |
| BDD                | OK | OK  |   6 |   5 |   0 | 100 | OK  |
| PERSONNES          | OK | OK  |   6 |   2 |   0 | 100 | OK  |
+--------------------+----+-----+-----+-----+-----+-----+-----+
```

Références

[Sommerville] Ian Sommerville, *Le génie logiciel*,
Addison-Wesley,
1992