

# Travaux Pratiques de Génie Logiciel - Contraintes Z

Sébastien Laborie et Yves Ledru

## 1 Objectifs

Les objectifs de ce TP consistent, entre autres, à:

- écrire en Z des contraintes dans l'environnement RoZ. La syntaxe Z utilisée est disponible dans le placard électronique.
- valider (**et invalider**) ces contraintes en spécifiant différents jeux de tests avec Jaza.

## 2 Contraintes sur les attributs de classe

Dans l'environnement RoZ, il est possible d'écrire à l'aide du langage Z diverses contraintes sur chaque élément du diagramme UML. Pour cela, il suffit de sélectionner un élément du diagramme et cliquer sur la fonction *Browse*  $\rightarrow$  *Specification*. Le champ *Documentation* d'une spécification permet de spécifier ces contraintes (celles-ci sont séparées par `\\`). En ce qui concerne la spécification de contraintes sur les attributs il suffit de sélectionner un attribut, d'accéder à sa spécification et de remplir son champ *Description* (cette contrainte peut faire référence à d'autres attributs de cette classe).

Téléchargez le fichier *Vol\_Ligne\_Instance\_Avion.mdl* ainsi que le fichier *roztypes.zed* disponibles dans le placard électronique. Attention, ces fichiers ont été adaptés par rapport aux séances de TD. A chaque execution de Jaza, tapez en première commande : `set setsize 20000`.

**Question 1** *Comme nous l'avons vu en TD, spécifiez en Z qu'un vol ne peut durer plus de 2 heures et que son heure de départ sera toujours avant celle de son heure d'arrivée. Puis, vérifiez avec Jaza que ces contraintes sont contrôlées.*

Lorsqu'une contrainte échoue, utilisez la commande `why` pour identifier la contrainte non-satisfaite.

## 3 Contraintes sur les extensions de classe

Dans RoZ, les contraintes sur l'extension d'une classe s'écrivent aussi dans le champ *Documentation* de cette classe. Cependant, nous faisons bien évidemment la distinction entre intention et extension. Par exemple, AVION, qui est le nom de la classe, représente l'intention de la classe alors que Avion représentera son extension. La syntaxe Z dans RoZ sera relativement similaire à celle employée en TD, cependant  $\forall$ ,  $\exists$ ,  $\wedge$ ,  $\vee$ ,  $\neq$ ,  $\bullet$  seront traduits respectivement par `\forall`, `\exists`, `\land`, `\lor`, `\neq`, `@`.

**Question 2** *Une première contrainte possible sur l'extension d'une classe est la contrainte de clé. Spécifiez en Z la contrainte de clé de la classe AVION. Puis, vérifiez avec Jaza que cette contrainte est contrôlée.*

**Question 3** *Spécifiez en Z la contrainte de clé de la classe INSTANCE. Puis, vérifiez avec Jaza que cette contrainte est contrôlée.*

Bien évidemment, il est aussi possible d'écrire d'autres contraintes sur l'extention d'une classe.

**Question 4** *Spécifiez que tous les avions de même type ont un rayon d'action identique. Puis, vérifiez avec Jaza que cette contrainte est contrôlée.*

## 4 Contraintes sur les relations entre deux classes

En ce qui concerne les contraintes sur les relations, par exemple celle entre VOL et LIGNE, il n'est pas nécessaire, comme nous avons pu le voir en TD, de spécifier VolLigneRel1 et VolLigneRel2. En effet, ces deux schémas sont générés automatiquement à l'aide des arités de la relation définis dans le diagramme UML. Par contre, il est possible de spécifier des contraintes supplémentaires sur les relations. On utilisera  $\backslash dom$  et  $\backslash ran$  pour dénoter le domaine ainsi que le co-domaine d'une fonction.

**Question 5** *Spécifiez qu'un vol pour une ligne donnée ne peut partir deux fois à la même heure. Puis, vérifiez avec Jaza que cette contrainte est contrôlée.*

**Question 6** *Spécifiez que l'avion prévu pour une instance permette de transporter la capacité ou le tonnage réservé. Puis, vérifiez avec Jaza que cette contrainte est contrôlée.*

**Question 7** *Spécifiez la contrainte de clé importée entre VOL et INSTANCE. Puis, vérifiez avec Jaza que cette contrainte est contrôlée.*

## 5 Contraintes sur les vues

Pour spécifier une contrainte entre deux classes non liées par une relation directe, par exemple entre les classes AVION et VOL, RoZ utilise la notion de vue. Pour cela, il suffit de sélectionner le champ *LogicalView* dans le cadre supérieur gauche de RoZ et de compléter son champ *Documentation*.

**Question 8** *Spécifiez que, pour une instance donnée, le rayon d'action de l'avion est supérieur à la distance de la ligne. Puis, vérifiez avec Jaza que cette contrainte est contrôlée en exécutant la commande *CheckGlobalInvariant*.*

**Question 9** *Spécifiez qu'un même vol ne peut assurer deux instances simultanément (dont les horaires se chevauchent). Puis, vérifiez avec Jaza que cette contrainte est contrôlée en exécutant la commande *CheckGlobalInvariant*.*

## 6 S'il vous reste du temps

**Question 10** *Spécifiez d'autres contraintes vues en TD. Puis, vérifiez avec Jaza que ces contraintes sont contrôlées.*