

## Exercice : opérations sur un compte bancaire

Soient des comptes bancaires de deux catégories différentes. Chaque compte a un numéro unique et un solde unique ; de plus un compte est soit de la catégorie C1 soit de la catégorie C2.

Un compte bancaire de la catégorie C1 est un compte pour lequel la banque n'autorise aucun découvert (le solde doit toujours être positif ou nul) et il y a une limite maximale à la somme totale qu'on peut déposer sur ce compte.

Un compte de la catégorie C2 est un compte pour lequel un découvert de seuilBasC2 est autorisé ; il n'y a pas de limite maximale dans ce cas.

Analyser les données et les propriétés ; modélisez les puis écrivez une machine abstraite avec les opérations nécessaires pour effectuer les dépôts et les retraits sur des comptes de catégorie C1 ou C2. Pour une opération de retrait ou d'approvisionnement, le propriétaire du compte donne le numéro du compte et le montant associé à l'opération.

## Etude de cas : modélisation d'un éditeur graphique

On veut spécifier en B un (logiciel) éditeur graphique qui permet de manipuler des **formes** géométriques : triangle, carré, rectangle, etc.

Pour ce faire, on élabore une machine abstraite B où on manipule des **objets** qui ont les diverses formes géométriques. L'éditeur dispose d'une zone d'affichage des objets délimitée en abscisse par xMin, xMax et en ordonnée par yMin, yMax. Ce sont des constantes.

Dans un premier temps, on caractérise chaque objet par une des formes géométriques et une position. Une position est décrite ici par deux valeurs entières (x, y).

L'éditeur graphique dispose à tout moment d'un ensemble éventuellement vide d'objets créés (objets courants), d'une **corbeille** qui peut contenir des objets (copiés ou coupés comme on le verra ci-après). L'éditeur dispose aussi d'un objet courant, c'est celui qui est sélectionné.

**Spécification en B d'une machine abstraite EdiGraphA avec les opérations suivantes :**

- **CreerObjet.** Cette opération attend comme paramètre la forme de l'objet à créer. Un objet est créé, la forme donnée en paramètre lui est attribuée ; l'objet est ajouté à l'ensemble des objets courants de l'éditeur. Une position par défaut est donnée à l'objet.
- **SelectionnerObj.** Cette opération attend deux paramètres : x et y qui doivent être des valeurs entières. L'opération sélectionne l'objet dont les valeurs de position sont identiques à x, y. Il y a au plus un objet sélectionné à tout moment. Une sélection remplace la sélection précédente.
- **ModifierPosition.** Cette opération attend deux paramètres  $n_x$ ,  $n_y$  qui doivent être des entiers. La position de l'objet courant est modifiée pour devenir  $n_x$ ,  $n_y$ .
- **CouperObjet.** Cette opération permet de supprimer l'objet courant. L'objet ainsi coupé tombe dans la corbeille de l'éditeur.
- **CopierObjet.** Cette opération copie l'objet actuellement sélectionné, la copie va dans la corbeille.

**Q#1** Ecrivez la machine abstraite EdiGraphA avec les opérations présentées.

**Q#2** Exprimez les obligations de preuve de cohérence de deux de vos opérations.

**Q#3** Prouvez une des obligations de preuve de cohérence exprimées précédemment.

### **Extension de la machine abstraite EdiGraphA**

On veut donner plus de détails sur les objets manipulés. On considère un ensemble de couleurs prédéfinies (rouge, vert, bleu, noir, jaune, etc ). Tout objet a maintenant une couleur. À la création d'un objet, la couleur noire lui est attribuée par défaut.

**Q#4** Ecrivez une machine EdiGraphC qui étend la précédente en prenant en compte les couleurs. Vous modifierez au besoin l'invariant et les opérations.

**Q#5** On veut écrire l'opération suivante : **ModifierCouleur**. Cette opération attend comme paramètre d'entrée une couleur. L'opération modifie la couleur de l'objet courant avec la couleur donnée en paramètre.

## **Etude de cas : modélisation d'un système simple de communication**

Utilisation de : **injection, surjection, union généralisée/quantifiée, intersection généralisation/quantifiée ; spécifications de propriétés additionnelles.**

Le système simple d'échange téléphonique présenté ici concerne des souscripteurs (les usagers qui ont souscrit au système) qui s'appellent pour communiquer/échanger.

Le système d'échange téléphonique permet de satisfaire efficacement des demandes d'appels à connexion entre n'importe quel nombre de souscripteurs. Pour une demande de connexion effectuée par un souscripteur au système, les souscripteurs appelés sont soit immédiatement connectables, auquel cas ils deviennent liés à une connexion en cours, soit ils ne sont pas immédiatement connectables et on met la demande en attente, on la diffère ainsi pour une connexion tardive.

### **Analyse du cas**

Après une analyse préliminaire, nous vous proposons un début de la modélisation formelle.

On a un ensemble de souscripteurs *Sousc*, et un ensemble de connexions *Conx*. On va devoir exprimer des relations entre *Sousc* et *Conx* mais il faut avant tout bien expliciter les contraintes liées à ces relations ; pour ce faire on interviewe le client !

### **Explicitation du système et des contraintes**

- Les connexions : une connexion implique un appelant et un ou plusieurs appelés.
- Les échanges : à tout moment, un échange reflète, un nombre de connexions demandées et un nombre de connexions en cours (actuelles) tels que deux connexions en cours données n'ont aucun souscripteur en commun.

Les connexions en cours sont toutes des connexions demandées. On garde toutes les connexions demandées (comme historique) ; on ne les efface pas. Il n'y a pas de connexions demandées qui ne soit pas en cours, et qui n'a pas de souscripteurs communs avec des connexions en cours.

- Les connexions demandées : toutes les connexions demandées pour un échange forment un ensemble de connexions.

- Les connexions en cours : toutes les connexions en cours pour un échange donné, forment un sous-ensemble des connexions demandées de cet échange de telle sorte que deux connexions en cours données ne partagent pas de souscripteurs.

## Analyse du cahier de charge et élicitation d'hypothèses de travail

Faite ressortir du cahier de charges, les principales informations.  
Complétez au besoin ce cahier de charges par des hypothèses de travail.  
Imaginez des scénarios d'utilisation du système de communication ainsi décrit.

## Début de modélisation

On utilisera des ensembles abstraits CONNEXION et SOUSCRIPTEUR, et les propriétés invariantes suivantes :

**P1** Les connexions en cours (où  $conx \subseteq CONNEXION$ ) :  $cEnCours \subseteq Conx$

**P2** Les connexions demandées :  $cDemandees \subseteq Conx$

**P3** L'appelant d'une connexion et les appelés d'une connexion :

$appelantC : Conx \rightarrow Sousc$

$appelesC : Conx \rightarrow P(Sousc)$  (ou bien sous forme de relation  $appelesC : Conx \leftrightarrow Sousc$ )

**Q#6** Ecrivez une machine B ; définissez les ensembles de base, les variables d'état puis l'invariant de la machine en tenant compte des propositions ci-dessus.

Vous allez maintenant compléter progressivement la machine.

## Expression des propriétés/contraintes : complétez l'invariant

Soit la contrainte suivante :

**P4** les connexions en cours sont des connexions demandées  $cEnCours \subseteq cDemandees$

Cette spécification est-elle juste ?

Spécifiez les propriétés/contraintes suivantes et ajoutez les à l'invariant avec le commentaire approprié :

**P5** deux connexions en cours données n'ont ni le même appelant ni des appelés communs

**P6** Il n'y a pas d'appels en attente qui n'ait pas de souscripteur dans les en cours.

## Spécification d'opérations

**Q#7** Spécifiez une opération qui donne les souscripteurs à une connexion  $c$  : c'est l'appelant et les appelés

**Q#8** Spécifiez une opération qui donne les souscripteurs des appels en cours