UNIVERSITÉ DE NANTES

Faculté des Sciences et des Techniques
MASTER2 ALMA - 2006/2007
Méthodes Formelles - B (C. Attiogbé)

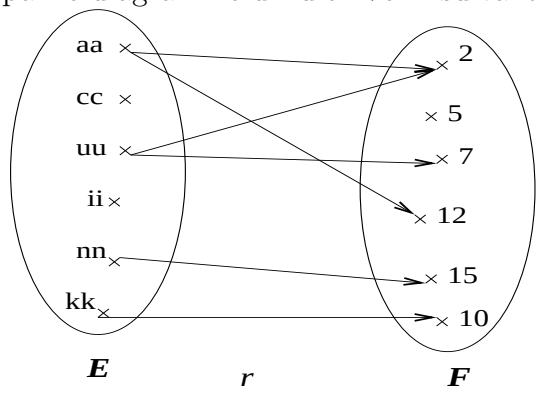
Examen - Méthodes formelles - durée 1h30

Mars 2007 - documents du cours autorisés

Pour vous évaluer, nous allons privilégier la qualité de vos réponses à leur quantité.

Exercice 1 - (20 mn)

Soit la relation r donnée par le diagramme d'Euler-Venn suivant :



1. Ecrivez une machine B qui contient la relation r
2. Ecrivez une opération qui transforme r en incrémentant de 1 l'image (les images) de chaque élément.
3. Ecrivez une opération qui extrait la sous-relation de r dont le co-domaine ne contient que des nombres impairs.
4. Ecrivez une opération qui extrait de r tous les couples dont le premier membre est un élément d'un ensemble prm donné.
5. Proposez un raffinement pour la machine ainsi étudiée. Indiquez vos hypothèses de travail quant à l'utilisation éventuelle de machines de librairie.

Exercice 2 - (20 mn)

La conception et programmation orientée objet est une technique bien connue aujourdhui. On y utilise des *classes*, des *objets* et divers mécanismes de structuration.
Nous nous intéressons, dans le cadre d'une étude, à la spécification en B d'un *éditeur spécifique de classes*. L'idée générale est de construire des classes satisfaisant certaines propriétés de correction. Pour ce faire nous considérons l'une des deux principales caractéristiques d'une classe : la *signature de la classe*. L'autre caractéristique est la sémantique de la classe ; on n'en tient pas compte ici.

Ainsi, pour la partie signature de classe,
– une *classe* a un *ensemble de variables* et un *ensemble de méthodes* (ou opérations).

– chaque variable a un *type*,
– chaque méthode a une *interface* (types des paramètres d'entrée et sortie),
– une interface est une suite de types avec un ordre précis (qui est celui d'appariton des paramètres).
– une méthode ne peut pas porter le même nom qu'une classe et vice versa.

Sur ces bases, spécifiez une machine abstraite B qui permet de manipuler des classes ainsi décrites (complétez au besoin l'analyse). Les commentaires de la spécification sont obligatoires.

Etude de cas - (50mn)

Etude formelle d'un logiciel de contrôle pour un système industriel (montage, convoiement, ...). On veut s'assurer de la correction du logiciel[1] en entreprenant un développement en B.

En situation de fonctionnement, le logiciel devra offrir des opérations activables par un opérateur humain. Le logiciel à son tour donne des ordres au système industriel. Nous nous occupons ici du développement du logiciel. Pour simplifier, on ne prendra pas en compte les ordres donnés au système industriel.

Le système industriel est équipé d'un dispositif de sécurité qui peut être activé ou désactivé. Lorsque le dispositif de sécurité est désactivé le système ne doit pas fonctionner ; c'est à dire que des ordres sont sans effet (par exemple il ne répond pas aux ordres de mise en route).

Avec la méthode B, le travail revient à spécifier (et prouver la correction) des machines abstraites, de façon à être sûre que le système fonctionne bien dans les bonnes conditions.
Le système peut fonctionner en deux modes :
– le mode *normal* ; c'est le mode de fonctionnement normal, il faut ici que le dispositif de sécurité soit activé.
– le mode *maintenance* dans lequel, on peut désactiver le dispositif de sécurité. Après une réparation ou une maintenance, il faut activer le système de sécurité avant de pouvoir remettre le système en fonctionnement.

Chaque opération nécessite des conditions d'appel particulières pour assurer un bon fonctionnement du système.

Les opérations offertes par le logiciel et activables par l'opérateur sont :
– `activer_fonctionnement` : permet d'activer le fonctionnement du système (mise en route),
– `désactiver_fonctionnement` : permet d'arrêter le fonctionnement du système,
– `activer_securité` : permet d'activer le dispositif de sécurité,
– `désactiver_sécurité` : permet de désactiver le dispositif de sécurité,
– `mode_maintenance` : permet de mettre le système en mode de maintenance,
– `mode_fonctionnement` : permet de mettre le système en mode de fonctionnement normal.

1. Après avoir analysé et complété au besoin la description informelle précédente, donnez-en une spécification formelle sous forme d'une machine abstraite B avec toutes les clauses nécessaires. Les commentaires de la spécification sont obligatoires.
2. Enoncez (en français) les obligations de preuve de cohérence pour la machine.
3. Pour chacune des opérations, donnez les obligations de preuve. Faites ou esquissez la preuve.

[1] En effet, pour éviter les risques pour son environnement, le système ne doit pas fonctionner dans n'importe quelle condition.