La Méthode B

Objectifs

Une méthode rigoureuse de développement logiciel :
- Spécification formelle
- Conception explicite par raffinement
- Implémentation simplifiée en couches
- Vérification et validation

Prérequis :
- Cahier des charges
- Description des besoins

Contexte et motivations

- Modélisation :
  - On démarre de spécifications générales informelles
  - Petit-à-petit le modèle est raffiné
  - On arrive à un modèle mathématique
- Spécification formelle :
  - Écriture de ce qui est attendu dans un langage formel
  - Permet de raisonner rigoureusement sur les propriétés attendues

- Modèle d'un programme :
  - Modèle usuel : une fonction qui transforme la mémoire de la machine.
  - Permet de prouver qu'un programme exécute bien les tâches demandées.
- Raffinement : décomposition d'un problème en sous-problèmes plus simples.
- Réutilisation : résolution d'un problème en utilisant des bibliothèques génériques bien établies

Correction d'un programme

- Certification
- Test

Méthode B

- Jean-Raymond Abrial (années 80)
- Méthode qui intervient tôt dans le développement du logiciel
- Notation de machine abstraite, raffinement, implémentation
- Théorie des ensembles
- Logique du premier ordre
- Langage de substitutions

Applications

Cette technologie est utilisée au milieu industriel pour la modélisation de logiciels critiques avec besoins du zéro-défaut.

- Coût du développement (Ariane)
- Attaques malveillantes (commerce, santé)
- Vies humaines (transport, médecine)
  - Système freinage RER A
  - (1998) Ligne de métro 14. Le logiciel critique embarqué a été modélisé, prouvé et généré à partir de spécifications B.
  - (2005) Ligne de métro 1.
  - Depuis : métros et navettes aéroports dans le monde entier.

Outils

- `B-tool` (Anglaterre), voir `http://www.b-core.com/btool.html`
- Atelier B, voir `http://www.atelierb.eu/`
- J.R. Abrial : The B-Book. Cambridge University Press, 1996. Atelier B : http ://www.atelierb.societe.com
- Actualités de B, voir `http://vasco.imag.fr/B/`
- Projet LL supportant la méthode B `https://gna.org/projects/brillant/`

Le formalisme

- Expressions arithmétiques
- Ensembles de base
- Formules
- Relations
- Fonctions
- Substitutions
- Séquences

Les expressions arithmétiques

Opérateurs $+, -, \times, /, \texttt{mod}$. Exemple :

$$8 \times (n + 11)$$

Les ensembles de base

$\{\}$ ou $\emptyset$	L'ens. vide
NAT	L'ens. des entiers naturels
INT	L'ens. des entiers relatifs
$STRING$	L'ens. des chaînes de caractères
$BOOL$	L'ens. des booléens
$\{e\}$	L'ens. réduit à un élément e
$\{e_1, ..., e_n\}$	L'ens. composé des éléments e1,...,en

Exemple : $\{1, 2, 3\}$ et $\{\{\}, \{1, 2\}\}$.

Opérations sur les ensembles

$E_1 \cup E_2$	union
$E_1 \cap E_2$	intersection
$E_1 - E_2$	différence

Les ensembles par compréhension

Appartenance à un ensemble : $z \in S$.

L'ens. de tous les objets qui vérifient la formule P est $\{z|P\}$ ou $\{z \in S \mid P\}$.

L'ens. des entiers entre n1 et n2 (inclus) avec n1 et n2 des expressions représentants des entiers positifs : $(n_1..n_2)$.

`card(E)` est le cardinal (nombre d'éléments) d'un ensemble `E` fini.

$\mathbb{P}(E)$ est l'ens. de sous-ens. de E.

Formules de base

$$
\begin{array}{lcl}
e_1 & = & e_2 \\
n_1 & > & n_2 \\
n_1 & < & n_2 \\
e_1 & \neq & e_2 \\
n_1 & \geq & n_2 \\
n_1 & \leq & n_2
\end{array}
$$

Formules ensemblistes

$e \in E$	L'expression e est un objet de l'ensemble E.
$e \notin E$	L'expression e n'est pas un objet de l'ensemble E.
$E_1 \subseteq E_2$	E1 est un sous-ensemble de E2.
$E_1 \not\subseteq E_2$	E1 n'est pas un sous-ensemble de E2.

Formules composées

$not(F)$	négation
$F_1 \wedge F_2$	conjonction
$F_1 \vee F_2$	disjonction
$F_1 \Rightarrow F_2$	implication
$F_1 \Leftrightarrow F_2$	équivalence
$\exists\ var.F$	quantification existentielle
$\forall\ var.F$	quantification universelle

Quantification multiple : $\forall\ (x, y).(x = y \Rightarrow y = x)$

Les formules bien typées

$\forall (a,b).(a,b) \in \mathbb{P}(E) \times \mathbb{P}(E) \Rightarrow \{x \mid x \in E \wedge x \in a \wedge x \in b\} \subseteq E$

- a de type $\mathbb{P}(E)$
- b de type $\mathbb{P}(E)$
- x de type E
- $x \in a$ bien typé
- $x \in b$ bien typé
- $\{x \mid x \in E \wedge x \in a \wedge x \in b\}$ de type $\mathbb{P}(E)$
- $\{x \mid x \in E \wedge x \in a \wedge x \in b\} \subseteq E$ bien typé

Substitution élémentaire

$$[var := E]P$$

Exemple : $[n := 3]n > m$ est égal à $3 > m$.

Variables liées

$$\exists\, x.F$$

$$\forall\, x.F$$

$$\{x \mid P\}$$

Une variable peut etre libre et liée en même temps :

$$x > 0 \ \wedge \ \forall\, x.x = 0$$

Substitution multiple

C'est une substitution simultanée

$$[x_1, \ldots, x_n := e_1, \ldots, e_n]P$$

ou

$$x_1 := e_1 \ || \ \ldots || \ x_n := e_n \ ; P$$

Exercice : Donner un exemple pour montrer que la substitution séquentielle et la substitution simultanée ne sont pas les mêmes.

Exercice : Exprimer la substitution simultanée en utilisant la substitution séquentielle.

Relations

$E_1 \leftrightarrow E_2$	Relations entre éléments de E_1 et de E_2
$R_1; R_2$	Composition des relations R_1 et R_2
$\mathtt{id}(E)$	la relation identité sur l'ensemble E
R^{-1}	relation inverse de R
$R^\star$	clôture réflexive-transitive de R
$E \lhd R$	la restriction de la relation R au domaine E
$R \rhd E$	la restriction de la relation R à l'image E

$\mathtt{dom}(R)$: domaine de R

$\mathtt{ran}(R)$: image de R

Fonctions

Les fonctions sont des relations fonctionnelles : si $(x, y) \in R$ et $(x, z) \in R$, alors $y = z$.

Les fonctions partielles de A en B : $A \nrightarrow B$

Les fonctions totales de A en B : $A \rightarrow B$

Les fonctions injectives de A en B : $A \rightarrowtail B$

Séquences

Les séquences sont des suites ordonnées d'objets d'un ensemble E. Une manière de les modéliser mathématiquement est de les voir comme des fonctions de N dans E dont le domaine est soit l'ensemble vide (dans le cas de la séquence vide) soit l'ensemble 1..n pour n un entier strictement positif.

Exemple

Modélisation de l'ensemble des entiers naturels de Peano.

1. $0 \in N$.

   N n'est pas vide.
2. $\forall\, n.(n \in N \Rightarrow succ(n) \in N)$.
3. $\forall\, n.(0 \neq succ(n))$.

   N possède un premier élément.
4. $\forall\, (n, m).(succ(n) = succ(m) \Rightarrow n = m)$.
5. $[E \in \mathbb{P}(N) \wedge 0 \in E \wedge \forall\, n.(n \in E \Rightarrow succ(n) \in E)] \Rightarrow E = N$.

   N vérifie le principe de récurrence.

Exercice

Étant donné un ensemble personnes, on cherche à modéliser :

- Les ensembles femmes, hommes
- Les relations epoux, epouse, mere, pere, parent, enfant, grand-parent, ancetre, frere, soeur, frere-soeur, enfant

1. Chaque personne est un homme ou une femme.

   $\texttt{hommes} \subseteq \texttt{personnes}$

   $\texttt{femmes} \subseteq \texttt{personnes}$

   $\texttt{personnes} \subseteq \texttt{hommes} \cup \texttt{femmes}$

2. Nulle personne n'est à la fois homme et femme.

   $\texttt{hommes} \cap \texttt{femmes} = \emptyset$

3. Seules les femmes ont des époux qui sont des hommes.

   $\texttt{epoux} \in \texttt{femmes} \leftrightarrow \texttt{hommes}$

4. Chaque femme a au plus un époux.

   $\texttt{epoux} \in \texttt{femmes} \nrightarrow \texttt{hommes}$

   (i.e. $(f, h_1) \in \text{epoux} \wedge (f, h_2) \in \text{epoux} \Rightarrow h_1 = h_2$).

5. Les hommes ne peuvent être mariés qu'à au plus une femme.

   $\texttt{epoux} \in \texttt{femmes} \rightarrowtail \texttt{hommes}$

   (i.e. $(f_1, h) \in \text{epoux} \wedge (f_2, h) \in \text{epoux} \Rightarrow f_1 = f_2$).

   ou

   $\texttt{epouse} = \texttt{epoux}^{-1}$

   $\texttt{epouse} \in \texttt{hommes} \nrightarrow \texttt{femmes}$

6. Une mère est une femme mariée.

   $\texttt{mere} \in \texttt{personnes} \nrightarrow \texttt{dom(epoux)}$

7. Le père est l'époux de la mère.

   $\texttt{pere} = \texttt{mere}\,;\,\texttt{epoux}$

8. Modéliser la notion de parent.

   $\texttt{parent} = \texttt{mere} \cup \texttt{pere}$

9. Modéliser la notion d' enfant.

   $\texttt{enfant} = \texttt{parent}^{-1}$

10. Modéliser la notion de grand-parent.

    $\texttt{grand-parent} = \texttt{parent}\,;\,\texttt{parent}$

11. Modéliser la notion d'ancetre.

    $\texttt{ancetre} = \texttt{parent}\,;\,\texttt{parent}^{*}$

12. Modéliser la notion de frere.

    $\texttt{frere} = (\text{mere}\,;\,\text{mere}^{-1}) \rhd \texttt{hommes} - \texttt{id(hommes)}$

13. Modéliser la notion de soeur.

    $\texttt{soeur} = (\text{mere}\,;\,\text{mere}^{-1}) \rhd \texttt{femmes} - \texttt{id(femmes)}$

14. Modéliser la notion de frere-soeur.

    $\texttt{frere-soeur} = \texttt{frere} \cup \texttt{soeur}$

15. Démontrer $\texttt{mere} = \texttt{pere}\,;\,\texttt{epoux}^{-1}$.

$\texttt{pere}\,;\,\texttt{epoux}^{-1}$

$(\texttt{mere}\,;\,\texttt{epoux}\,)\,;\,\texttt{epoux}^{-1}=$

$\texttt{mere}\,;\,(\texttt{epoux}\,;\,\texttt{epoux}^{-1})=$

$\texttt{mere}\,;\,\texttt{id}(\texttt{dom}(\texttt{epoux}))\,=$

$\texttt{mere}$