- Intro
- Données
- SubsG
- Machine

La méthode B

Cours donné à l'Ensimag
Partie 2

Marie-Laure Potet

LSR-IMAG, Grenoble, France

- Intro
- Données
- SubsG
- Machine

Plan du cours

1. Raffinement et implémentation
2. Simulation et raffinement

Raffinement : principe

- Le raffinement est le fait de transformer une spécification abstraite en un texte plus proche de la programmation, pour finalement obtenir un programme
- L'effet des appels d'opérations de la machine abstraite doit être préservé, vu de l'utilisateur
- Le raffinement de machine se fait opération par opération
- Il y a (éventuellement) raffinement de l'état
- Pour chaque opération :
  - reformulation en fonction du changement d'état
  - affaiblissement des préconditions
  - réduction du non-déterminisme

LSR-IMAG
- Intro
- Données
- SubsG
- Machine

Exemple : une machine

MACHINE $RAFF\_EX1$

VARIABLES yy

INVARIANT $yy \subseteq \mathsf{NAT}_1$

INITIALISATION $yy := \varnothing$

OPERATIONS

$ajouter(nn)$ = PRE $nn \in \mathsf{NAT}_1$

THEN $yy := yy \cup \{nn\}$

END;

$vv \longleftarrow choix$ = PRE $yy \neq \varnothing$

THEN $vv :\in yy$

END

END

Une machine qui "fait presque la même chose"

MACHINE $RAFF\_EX2$

VARIABLES zz

INVARIANT $zz \in$ NAT

INITIALISATION $zz := 0$

OPERATIONS

$ajouter(nn)$ = PRE $nn \in$ NAT$_1$

THEN $zz := nn$

END;

$vv \longleftarrow choix$ = $vv := zz$

END

Une machine qui "fait presque la même chose"

LSR-IMAG
- Intro
- Données
- SubsG
- Machine

MACHINE $RAFF\_EX2$

VARIABLES zz

INVARIANT $zz \in$ NAT

INITIALISATION $zz := 0$

OPERATIONS

$ajouter(nn)$ = PRE $nn \in$ NAT$_1$

THEN $zz := nn$

END;

$vv \longleftarrow choix$ = $vv := zz$

END

Relation de simulation : $zz \in yy \ \vee \ yy = \emptyset$

Composant raffinement en B

REFINEMENT $RAFF\_EX1\_R$ REFINES $RAFF\_EX1$

VARIABLES zz

INVARIANT $zz \in \mathsf{NAT}$

$\wedge \ (zz \in yy \ \vee \ yy = \varnothing)$

INITIALISATION $zz := 0$

OPERATIONS

$ajouter(nn)$ = PRE $nn \in \mathsf{NAT}_1$

THEN $zz := nn$

END;

$vv \longleftarrow choix$ = $vv := zz$

END

Machine initiale

Syntaxe du raffinement

Raffinement d'une substitution dans le même état

Définition du raffinement de S par T :

$S \sqsubseteq T$	$\forall R \cdot ([S]\, R \ \Rightarrow \ [T]\, R)$

- Toute propriété vérifiée par S est aussi vérifiée par T
- définition non opérationnelle

Forme normalisée d'une substitution

- Intro
- Données
- SubsG
- Machine

Forme normalisée d'une substitution S :

$$\text{trm}(S) \mid (@\, v' \,.\, \text{prd}_v(S) \implies v := v')$$

avec :

$\text{trm}(S)$	=	$[S]true$	condition de terminaison
$\text{prd}_v(S)$	=	$\neg[S](v \neq v')$	relation avant/après

Exemple :

$$\text{prd}(if\ v = 0\ then\ v := 0\ else\ v := 1\ end)$$
$$=$$
$$(v = 0 \Rightarrow v' = 0) \wedge (v \neq 0 \Rightarrow v' = 1)$$

Définition opérationnelle du raffinement sans changement d'état

- Intro
- Données
- SubsG
- Machine

$S \sqsubseteq T$	$\text{trm}(S) \Rightarrow \text{trm}(T)$ $\text{trm}(S) \wedge \text{prd}(T) \Rightarrow \text{prd}(S)$

Exemple

$$v > 0 \mid v :\in \{1, 2\}$$
$$\sqsubseteq$$
$$if\ v = 0\ then\ v := 0\ else\ v := 1\ end$$

Obligation de preuve relative à la terminaison :

$v > 0 \Rightarrow\ true$

$true$

Obligation de preuve relative aux relations avant-après :

$v > 0 \wedge\ (v = 0 \Rightarrow v' = 0) \wedge\ (v \neq 0 \Rightarrow v' = 1) \Rightarrow\ v' \in \{1, 2\}$

$v > 0 \wedge\ v' = 1 \Rightarrow\ v' \in \{1, 2\}$

$true$

Raffinement avec changement de représentation : formules

$S \sqsubseteq_L T$	$L\ \wedge\ \mathsf{trm}(S)\ \Rightarrow\ \mathsf{trm}(T)$ $L\ \wedge\ \mathsf{prd}_y(T)\ \Rightarrow\ \exists x' \cdot (\mathsf{prd}_x(S)\ \wedge\ [x, y := x', y']\, L)$

Raffinement avec changement de représentation : formules

- Intro
- Données
- SubsG
- Machine

$S \sqsubseteq_L T$	$L \wedge \mathsf{trm}(S) \Rightarrow \mathsf{trm}(T)$ $L \wedge \mathsf{prd}_y(T) \Rightarrow \exists x' \cdot (\mathsf{prd}_x(S) \wedge [x, y := x', y']\, L)$

Autre formulation :

$S \sqsubseteq_L T$	$L \wedge \mathsf{trm}(S) \Rightarrow [T]\,\neg[S]\,\neg L$

Application à l'exemple : $ajouter$ (1)

- Intro
- Données
- SubsG
- Machine

Raffinement de $ajouter$ avec changement de représentation :

L	$yy \subseteq \mathsf{NAT}_1 \wedge zz \in \mathsf{NAT} \wedge (zz \in yy \vee yy = \varnothing))$
$ajouterA$	$nn \in \mathsf{NAT}_1 \mid yy := yy \cup \{nn\}$
$ajouterC$	$nn \in \mathsf{NAT}_1 \mid zz := nn$

Application à l'exemple : $ajouter$ (1)

Raffinement de $ajouter$ avec changement de représentation :

$$L \quad yy \subseteq \mathsf{NAT}_1 \ \wedge\ zz \in \mathsf{NAT} \ \wedge\ (zz \in yy \ \vee\ yy = \varnothing))$$

$$ajouterA \quad nn \in \mathsf{NAT}_1 \mid yy := yy \cup \{nn\}$$

$$ajouterC \quad nn \in \mathsf{NAT}_1 \mid zz := nn$$

$ajouterA \sqsubseteq_L ajouterC$:

$yy \subseteq \mathsf{NAT}_1 \ \wedge\ zz \in \mathsf{NAT} \ \wedge$

$(zz \in yy \ \vee\ yy = \varnothing) \ \wedge$ /* L */

$nn \in \mathsf{NAT}_1$ /* $\mathsf{trm}(ajouterA)$ */

$\Rightarrow$

$[ajouterC] \neg [ajouterA] \neg L$

Raffinement de $ajouter$ (2)

- Intro
- Données
- SubsG
- Machine

$yy \subseteq \mathsf{NAT}_1 \ \wedge\ zz \in \mathsf{NAT} \ \wedge$

$(zz \in yy \ \vee\ yy = \varnothing) \ \wedge$ /* L */

$nn \in \mathsf{NAT}_1$ /* $\mathsf{trm}(ajouterA)$ */

$\Rightarrow$

$nn \in \mathsf{NAT}_1 \ \wedge$ /* $\mathsf{trm}(ajouterC)$ */

$yy \cup \{nn\} \subseteq \mathsf{NAT}_1 \ \wedge\ nn \in \mathsf{NAT} \ \wedge$

$nn \in yy \cup \{nn\}$

Raffinement de machines : Syntaxe

MACHINE M	REFINEMENT N REFINES M
VARIABLES x	VARIABLES y
INVARIANT I	INVARIANT J
INITIALISATION U	INITIALISATION V
OPERATIONS	OPERATIONS
$r \longleftarrow nom\_op(w)$ =	$r \longleftarrow nom\_op(w)$ =
PRE P THEN K END	PRE Q THEN L END
END	END

Obligations de preuves

Obligation de preuve des raffinements

Initialisation :

$$[V]\ \neg[U]\ \neg J$$

Obligation de preuve pour chaque opération :

$$I \ \wedge\ J \ \wedge\ P \ \Rightarrow\ Q$$

$$I \ \wedge\ J \ \wedge\ P \ \Rightarrow\ [L]\ \neg[K]\ \neg J$$

$$I \ \wedge\ J \ \wedge\ P \ \Rightarrow\ [[r := r']L]\ \neg[K]\ \neg(J \ \wedge\ r = r')$$

Raffinement et Simulation

Soit M un composant raffiné par R. Le principe de correction du raffinement est le suivant :

- Toute substitution externe admise par M est admise par R :
  - interface(M) = Interface(R)
  - trm$(U_M) \Rightarrow$ trm(U_R)
- Les valeurs observables du niveau raffiné sont des valeurs observables du niveau abstrait :

$$\begin{array}{l} \quad \text{VAR } v_M \;\; \text{IN } \; init_M \;\; ; \; U_M \text{ END} \\ \sqsubseteq_{id} \text{VAR } v_R \;\; \text{IN } \; init_R \;\; ; \; U_R \text{ END} \end{array}$$

Principe

$\Rightarrow$ Un “ grand ” nombre d'utilisations externes

Méthode effective :

1. une relation d'abstraction α qui lie les valeurs abstraites et les valeurs concrètes
2. Une notion de commutativité du raffinement

$$\begin{tikzcd} X \arrow[r, "A"] \arrow[d, shift left, "\alpha^{-1}"] & X' \arrow[d, shift left, "\alpha^{-1}"] \\ Y \arrow[u, shift left, "\alpha"] \arrow[r, "C"'] & Y' \arrow[u, shift left, "\alpha"] \end{tikzcd}$$

LSR-IMAG

- Intro
- Données
- SubsG
- Machine

Différentes simulations

- L-simulation (forward ou downward simulation) :

$$\alpha^{-1} \,;\, C \;\subseteq\; A \,;\, \alpha^{-1}$$

i.e : $\forall a, c' \; (\exists c \; (\alpha \wedge C) \Rightarrow \exists a' (A \wedge \alpha'))$

- L^{-1}-simulation (backward ou upward simulation) :

$$C \,;\, \alpha \;\subseteq\; \alpha \,;\, A$$

i.e : $\forall c, a' \; (\exists c' \; (C \wedge \alpha') \Rightarrow \exists a \; (\alpha \wedge A))$

- U-simulation : $\alpha^{-1} \,;\, C \,;\, \alpha \;\subseteq\; A$
- U^{-1}-simulation : $C \;\subseteq\; \alpha^{-1}; A; \alpha$

LSR-IMAG

- Intro
- Données
- SubsG
- Machine

Quelques résultats

$\Rightarrow$ Correction ? complétude ?

- Méthode B (et en général) : L-simulation
- incomplétude de ces différentes notions
- Complétude : L-simulation et L^{-1}-simulation
- Si α est une fonction totale alors équivalence de ces différentes notions

Exemple

MACHINE $CASINO1$
VARIABLES ii
INVARIANT $ii \in 0..36$
INITIALISATION $ii :\in 0..36$
OPERATIONS
$rr \longleftarrow spin \hat{=} rr := ii \mid\mid ii :\in 0..36$
END

MACHINE $CASINO2$
OPERATIONS
$rr \longleftarrow spin \hat{=} rr :\in 0..36$
END

casino2 $\sqsubseteq$ casino1 mais casino1 $\not\sqsubseteq$ casino2
S. Dunne, ZB 2003