INSA RENNES – DEPARTEMENT INFORMATIQUE

Verification

Annales

Quentin Decré

2010-2011

TABLE DES MATIERES

1. DS de Mai 2009

1.1. Questions de cours

1.1.1. La surcharge

Notation : q <+ r

Définition formelle : $(\mathrm{dom}(r) \trianglelefteq q) \cup r$

r et q sont des relations entre les mêmes domaines.

La surcharge permet de mettre à jour une relation en écrasant d'anciens couples de q avec des nouveaux provenant de r.

1.1.2. Utilisation de la surcharge et de l'union

Si $\mathrm{dom}(r) \cap \mathrm{dom}(q) = \emptyset$, alors on peut utiliser la surcharge ou l'union indifféremment. En effet, cela implique que $(\mathrm{dom}(r) \trianglelefteq q) = q$, et donc cela revient à faire $q \cup r$.

1.1.3. Definition formelle de la substitution generalisee $x :\in P$

$$[x :\in P]R$$

$$[ANY\ y\ WHERE\ y \in P\ THEN\ x \coloneqq y\ END]R$$

$$[@y.(y \in P \Longrightarrow x \coloneqq y)]R$$

1.1.4. Developper

$$[y \coloneqq 4]\neg[ANY\ z\ WHERE\ z > 3\ THEN\ CHOICE\ x \coloneqq z\ OR\ x \coloneqq 6\ END]\neg(x = y)$$

On remplace le any par sa définition formelle (facilités syntaxiques)

$$[y \coloneqq 4]\neg[@z.(z > 3 \Longrightarrow\ CHOICE\ x \coloneqq z\ OR\ x \coloneqq 6\ END]\neg(x = y)$$

On remplace le choice par sa définition formelle (facilités syntaxiques)

$$[y \coloneqq 4]\neg[@z.(z > 3 \Longrightarrow\ x \coloneqq z\ [\,]\ x \coloneqq 6)]\neg(x = y)$$

On utilise l'axiome de la substitution généralisée ANY

$$[y \coloneqq 4]\neg\forall z.[z > 3 \Longrightarrow\ x \coloneqq z\ [\,]\ x \coloneqq 6]\neg(x = y)$$

On utilise l'axiome de la substitution généralisée GARDE

$$[y \coloneqq 4]\neg\forall z.\big(z > 3 \Rightarrow [\ x \coloneqq z\ [\,]\ x \coloneqq 6]\neg(x = y)\big)$$

On utilise l'axiome de la substitution généralisée CHOICE

$$[y := 4]\neg\forall z.\Big(z > 3 \Rightarrow \big([x := z]\neg(x = y) \wedge [x := 6]\neg(x = y)\big)\Big)$$

On applique Morgan

$$[y := 4]\exists z.\Big(z > 3 \wedge \neg\big([x := z]\neg(x = y) \wedge [x := 6]\neg(x = y)\big)\Big)$$

$$[y := 4]\exists z.\Big(z > 3 \wedge \big(\neg[x := z]\neg(x = y) \vee \neg[x := 6]\neg(x = y)\big)\Big)$$

On peut passer les négations après les substitutions simples (S5)

$$[y := 4]\exists z.\Big(z > 3 \wedge \big([x := z](x = y) \vee [x := 6](x = y)\big)\Big)$$

Application des substitutions (S1/S2)

$$[y := 4]\exists z.\Big(z > 3 \wedge \big((z = y) \vee (6 = y)\big)\Big)$$

D'après S5 et S7 (on suppose $y\backslash z$)

$$\exists z.[y := 4]\Big(z > 3 \wedge \big((z = y) \vee (6 = y)\big)\Big)$$

On distribue la substitution avec S3

$$\exists z.\Big([y := 4](z > 3) \wedge [y := 4]\big((z = y) \vee (6 = y)\big)\Big)$$

Avec S5 et S4 on peut distribuer sur le $\vee$, et S12 pour distribuer sur le $=$

$$\exists z.\Big(([y := 4]z > [y := 4]3) \wedge \big(([y := 4]z = [y := 4]y) \vee ([y := 4]6 = [y := 4]y)\big)\Big)$$

On applique S1 S2

$$\exists z.\Big((z > 3) \wedge \big((z = 4) \vee (6 = 4)\big)\Big)$$

En prenant $z = 4$, on peut démontrer que c'est vrai.

1.1.5. PRODUCTION DE LA FORMULE PRECEDENTE

$$[y := 4]\neg[ANY\ z\ WHERE\ z > 3\ THEN\ CHOICE\ x := z\ OR\ x := 6\ END]\neg(x = y)$$

On peut avoir la formule précédente lors de la preuve d'une opération (ou initialisation) d'un raffinement.

x est une variable de la machine abstraite et y une variable de la machine raffinée. Les deux sont liées par l'invariant de liaison.

Ici, on a réduit le non déterminisme.

Point Bonus : (x=y) ce n'est pas un raffinement de données.

1.1.6. Relation / Ensemble image

Avec la relation, on gagne le lien entre les époux et les épouses. On sait donc qui est marié avec qui, en plus de pouvoir récupérer les ensembles des épouses et époux. Cela permet la composition et évite de maintenir de multiples variables. On évite la redondance et les mises à jours multiples.

On peut récupérer ces ensembles avec le domaine et le codomaine (ou image).

1.2. Demonstration dans le systeme derive

$$\forall(x,y).(R \Rightarrow \neg P \lor Q) \Rightarrow \forall(x,y).(P \land R \Rightarrow R \land Q) \text{ sous la condition } x \setminus y$$

Note 1 : les démonstrations de non liberté seront indiquées en rouge et réalisées à la fin.

Note 2 : bien faire attention à la portée des quantificateurs, qui nous empêche d'utiliser GEN au début. On ne peut qu'utiliser DED.

Note 3 : indiquer lorsqu'on utilise des règles de réécriture.

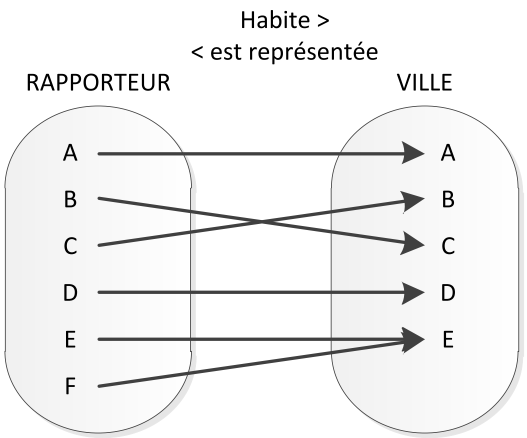
$$DED\ \dfrac{GEN\ \dfrac{D8\ \dfrac{DED*2\ \dfrac{CNJ\ \dfrac{BS2\ \dfrac{}{H,P,R \vdash R} \qquad D11\ \dfrac{D7\ \dfrac{DB1\ \dfrac{}{H,P,R \vdash \neg R \Rightarrow Q} \qquad D7\ \dfrac{DB1\ \dfrac{}{H,P,R \vdash \neg P \Rightarrow Q} \qquad DED\ \dfrac{BS2\ \dfrac{}{H,P,R \vdash Q}}{H,P,R \vdash Q \Rightarrow Q}}{H,P,R \vdash (P \Rightarrow Q) \Rightarrow Q}}{H,P,R \vdash \overbrace{\cancel{[x,y := E,F]}}^{\substack{\text{on choisi x pour E} \\ \text{et y pour F, ce qui élimine} \\ \text{cette substitution}}} \big(R \Rightarrow (P \Rightarrow Q)\big) \Rightarrow Q}}{H,P,R \vdash Q}}{H,P,R \vdash R \land Q}}{H \vdash P \Rightarrow \big(R \Rightarrow (R \land Q)\big)}}{H \vdash (P \land R \Rightarrow R \land Q)}}{\forall(x,y).(R \Rightarrow (P \Rightarrow Q)) \vdash \forall(x,y).(P \land R \Rightarrow R \land Q)}\ (x,y) \text{"\textbackslash"}\ H}{\vdash \forall(x,y).\left(R \Rightarrow \underbrace{(P \Rightarrow Q)}_{\substack{\text{écriture du}\lor \\ \text{en}\Rightarrow}}\right) \Rightarrow \forall(x,y).(P \land R \Rightarrow R \land Q)}$$

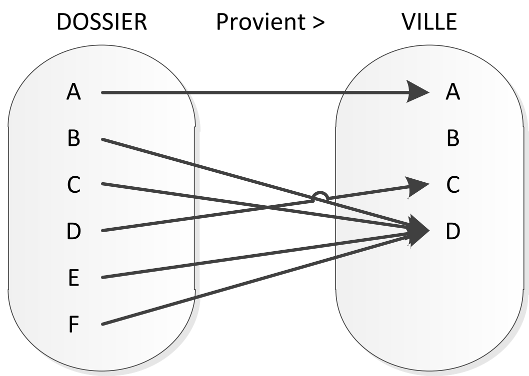
Tests de non liberté :

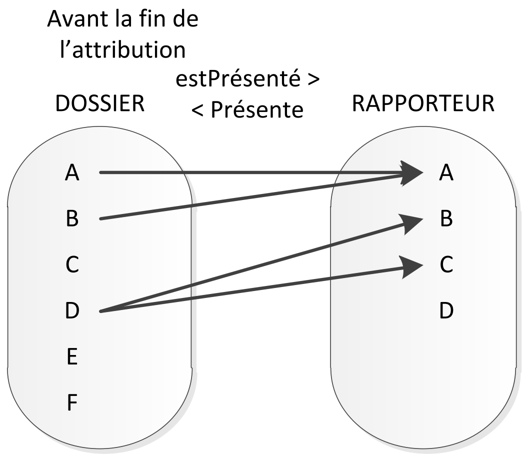
(x,y) "\" $\forall(x,y).\big(R \Rightarrow (P \Rightarrow Q)\big)$ est vrai d'après NF5

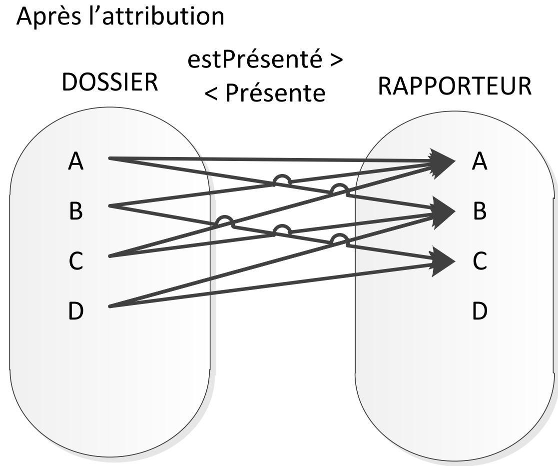
1.3. MACHINES ABSTRAITES (CORRECTION NON VERIFIEE)

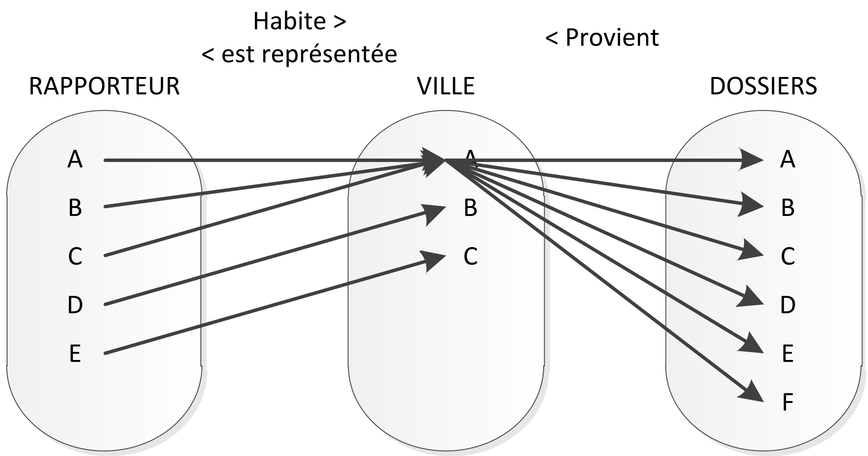
1.3.1. QUESTION 3.1 – PATATES











1.3.2. QUESTION 3.2 – INCOMPATIBILITE FORTE

Exemple du cas suivant :

D et E devront s'occuper de tous les dossiers, soit 6 dossiers, alors qu'ils devraient être sous les $\frac{6}{5} + 2$ dossiers.

$$\forall v. (v \in Ville \Longrightarrow card\big(dom(Provient[v])\big) < card\big(dom(habite \trianglelefteq v)\big)$$

Il faut que le nombre de de rapports venant d'une ville soit inférieur au nombre de rapporteurs venant des autres villes.

1.3.3. QUESTION 3.3

Dans l'état final, il doit y avoir exactement 2 rapporteurs par dossiers. Et un rapporteur doit avoir :

$$\frac{card(dossiers)}{card(rapporteurs)} \pm 2$$

Dossiers à prendre en charge.

Il est donc nécessaire d'avoir un invariant qui assure que cela sera vrai dans l'état final.

$\forall x. (x \in dossier \Rightarrow card\big(estprésenté(x)\big) \leq 2)$ `//au plus deux rapporteurs`
$\forall x. (x \in rapporteurs \Rightarrow card\big(présente(x)\big) \leq \frac{card(dossiers)}{card(rapporteurs)} + 2$
`//on empêche de dépasser le nombre max de dossiers par rapporteur`

Il faudra donc vérifier que l'on respectera l'invariant en initialisant la relation présente.

Dans la première phase, on peut prendre n'importe quel ensemble de Rapporteurs <--> Dossiers vérifiant l'invariant.

Pour ajouter_un_lien, dans les préconditions, il faut vérifier les éléments suivants :

Statut = 2ePhase $\wedge$
D $\in$ dossiers $\wedge$
R $\in$ rapporteurs $\wedge$
$(r, d) \notin$ présente $\wedge$
$\forall x. (x \in dossier \Rightarrow card\big(estprésenté(x)\big) < 2$ $\wedge$
$\forall x. (x \in rapporteurs \Rightarrow card\big(présente(x)\big) < Nbmr + 2 \wedge$
Habite(r) != Provient^{-1}(d)

1.3.4. MODELE DE L'APPLICATION

```
MACHINE
      Jury

CONSTRAINTS


SETS
      RAPPORTEUR,
      DOSSIER,
      VILLE,
      STATUT = {init, 2ePhase, final}

CONSTANTS
      Rapporteurs,
      Dossiers,
      Villes,
      Habite,
      Provient

PROPERTIES                 Q
      Rapporteurs ∈ RAPPORTEUR ∧
      Dossiers ∈DOSSIER ∧
      Villes ∈VILLE ∧
      Habite ∈ RAPPORTEUR →> VILLE ∧
      Provient∈ DOSSIER → VILLE ∧
      ∀v.(v ∈Ville⇒card(dom(Provient[v]))<card(dom(habite⊴v))
      // CF question 2


DEFINITIONS
      Nbmr = card(Dossiers)/card(Rapporteurs) ∧
      Estprésenté = présente-1

VARIABLES
      Présente, statut

INVARIANT                  I
      Présente ∈ Rapporteurs <-> Dossiers ∧
      Statut ∈ STATUT ∧
      (Statut = final) ⇒ ∀x.(x ∈ dossier ⇒ card(estprésenté(x)) = 2) ∧
      ¬(Statut = final) ⇒ ∀x.(x ∈ dossier ⇒ card(estprésenté(x)) ≤ 2) ∧
      ∀x.(x ∈ rapporteurs ⇒ card(présente(x)) ≤ Nbmr + 2 ∧
      (Statut = final) ⇒ ∀x.(x ∈ rapporteurs ⇒ card(présente(x)) ≥ Nbmr − 2 ∧
      (Habite ; Provient-1) ∩ Présente = ∅


INITIALISATION
      Statut := init

OPERATIONS
      Premiere_phase =
      PRE
            Statut = init
      THEN
            ANY sel
            WHERE
                  Sel ∈ Dossiers → Rapporteurs ∧
                  (Habite ; Provient-1) ∩ Sel-1 = ∅) ∧
                  ∀x.(x ∈ rapporteurs ⇒ card(Sel-1 (x)) ≤ Nbmr + 2
            THEN
                  Présente := Sel-1 ||
                  Statut := 2ePhase
            END
      END

      Ajouter_un_lien(d,r) =
      PRE
            Statut = 2ePhase ∧               P
            d ∈dossiers ∧
            r ∈ rapporteurs ∧
```

```
        (r,d) ∉ présente ∧
        ∀x.(x ∈ dossier ⇒ card(estprésenté(x)) < 2 ∧
        ∀x.(x ∈ rapporteurs ⇒ card(présente(x)) < Nbmr + 2 ∧
        Habite(r) != Provient-1(d)

THEN
        Présente := Présente ∪{(r,d)}        S
END


Effacer_un_lien(d,r) =
PRE
        Statut = 2ePhase ∧
        d ∈dossiers ∧
        r ∈ rapporteurs ∧
        (r,d) ∈ présente
THEN
        Présente := Présente − {(r,d)}

END


Valider =
PRE
        Statut = 2ePhase ∧
        ∀x.(x ∈ dossier ⇒ card(estprésenté(x)) = 2)  ∧
        ∀x.(x ∈ rapporteurs ⇒ card(présente(x)) ≥ Nbmr − 2
THEN
        Statut := final
END


DD ← dossiers_non_completement_attribués =
        ANY sol
        WHERE
                Sol ⊂ dossiers ∧
                ∀x.(x ∈ (dossiers − sol) ⇒ card(estprésenté(x)) = 2)
        THEN
                DD := sol
        END
END


RR ← rapporteurs_sous_affectés =
        ANY sol
        WHERE
                Sol ⊂ rapporteurs ∧
                ∀x.(x ∈ rapporteurs − sol ⇒ card(présente(x)) ≥ Nbmr − 2
        THEN
                DD := sol
        END
END
```

1.3.5. QUESTION 3.5 – OBLIGATION DE PREUVE AJOUTER LIEN

$$\underbrace{A}_{\text{paramètres}} \wedge \underbrace{C}_{\text{contraintes}} \wedge \overbrace{B}^{\text{ensembles}} \wedge \overbrace{Q}^{\text{propriétés}} \wedge \underbrace{I}_{\text{invariants}} \wedge \overbrace{P}^{\text{préconditions}} \Rightarrow \underbrace{[S]}_{\text{substitution}} \overbrace{I}^{\text{invariants}}$$

Si on retire les ensemble vides dans ce cas :

$$\overbrace{B}^{\text{ensembles}} \wedge \overbrace{Q}^{\text{propriétés}} \wedge \underbrace{I}_{\text{invariants}} \wedge \overbrace{P}^{\text{préconditions}} \Rightarrow \underbrace{[S]}_{\text{substitution}} \overbrace{I}^{\text{invariants}}$$

On note les ensembles sur la machine abstraite, à l'exception de ceux détaillés ci-dessous :

RAPPORTEUR $\in F_1(INT)$ ∧
DOSSIER $\in F_1(INT)$ ∧
VILLE $\in F_1(INT)$ ∧
STATUT $\in F_1(INT)$ ∧
Init != 2ePhase ∧
Init != final ∧
2ePhase != final

Invariant après substitution :

Présente $\cup \{(r,d)\} \in \text{Rapporteurs} <-> Dossiers$ ∧
Statut ∈ STATUT ∧
(Statut = final) ⇒ $\forall x.(x \in dossier \Rightarrow card\big((\text{Présente } \cup \{(r,d)\})^{-1}(x)\big) = 2)$ ∧
¬(Statut = final) ⇒ $\forall x.(x \in dossier \Rightarrow card\big((\text{Présente } \cup \{(r,d)\})^{-1}(x)\big) \leq 2)$ ∧
$\forall x.(x \in rapporteurs \Rightarrow card\big((\text{Présente } \cup \{(r,d)\})(x)\big) \leq Nbmr + 2$ ∧
(Statut = final) ⇒ $\forall x.(x \in rapporteurs \Rightarrow card\big((\text{Présente } \cup \{(r,d)\})(x)\big) \geq Nbmr - 2$
∧ (Habite ; Provient^{-1}) ∩ Présente $\cup \{(r,d)\} = \emptyset$

1.3.6. QUESTION 3.6 – PREUVES INFORMELLES

Après avoir retiré les invariants identiques à ceux avant l'implication (car $A \Rightarrow A$ est une tautologie), on obtient les buts suivants :

Présente $\cup \{(r,d)\} \in \text{Rapporteurs} <-> Dossiers$ ∧
// vrai car D ∈ dossiers ∧ R ∈ rapporteurs

(Statut = final) ⇒ $\forall x.(x \in dossier \Rightarrow card\big((\text{Présente } \cup \{(r,d)\})^{-1}(x)\big) = 2)$ ∧
// vrai car statut = 2ePhase et final != 2ePhase

¬(Statut = final) ⇒ $\forall x.(x \in dossier \Rightarrow card\big((\text{Présente } \cup \{(r,d)\})^{-1}(x)\big) \leq 2)$ ∧
// vrai car $\forall x.(x \in dossier \Rightarrow card\big(estprésenté(x)\big) < 2$

$\forall x.(x \in rapporteurs \Rightarrow card\big((\text{Présente } \cup \{(r,d)\})(x)\big) \leq Nbmr + 2$ ∧
// vrai car $\forall x.(x \in rapporteurs \Rightarrow card\big(présente(x)\big) < Nbmr + 2$

(Statut = final) ⇒ $\forall x.(x \in rapporteurs \Rightarrow card\big((\text{Présente } \cup \{(r,d)\})(x)\big) \geq Nbmr - 2$
// vrai car statut = 2ePhase et final != 2ePhase

∧ (Habite ; Provient^{-1}) ∩ Présente $\cup \{(r,d)\} = \emptyset$
// vrai car Habite(r) != Provient^{-1}(d)

1.3.7. Technologies pour implementer l'operation premiere_phase

Le cours de contrainte est idéal pour résoudre ce problème !