TD - Modélisation des données

Victor Lezaud

30 janvier 2018

Sommaire

Séance de Travaux Dirigés I

Langages relationnels

1 Bornes des cardinaux

Soient R un symbole de relation , r et s deux relations R, $A \in schema(R)$, $c \in dom(A)$ et F une formule sur R. Donner des bornes aux cardinaux des expressions suivantes en fonction de $|r|$ et $|s|$:

1.1 Sélection

$0 \leq |\sigma_F(r)| \leq |r|$

1.2 Union

$max(|r|, |s|) \leq |r \cup s| \leq |r| + |s|$

1.3 Soustraction

$0 \leq |r \setminus s| \leq |r|$

1.4 Projection

$1 \leq |\pi_{\{A\}}(r)| \leq |r|$ si $|r| \geq 1$

1.5 Division

$0 \leq |r./s| \leq |r|/|s|$

2 Calcul relationnel autorisé

Soit $Q = \{< x : A, y : B >| F(x, y)\}$ une requête du calcul relationnel sur R avec $schema(R) = \{A, B\}$. On considère les formules suivantes :

2.1 $F(x, y) = R(x, 2) \vee R(3, y)$

2.1.1 Q est-elle bien formée ?

F est une combinaison de formule atomique lié par l'opérateur OU, Q est donc bien formée.

2.1.2 Q est-elle autorisée ?

Les variables x et y sont des variables libres, or y est négatif dans $R(x, 2)$ et x est négatif dans $R(3, y)$, donc par l'opérateur OU x et y sont négatifs donc Q n'est pas autorisé.

2.1.3 Soit r_0 une relation sur R. $Ans(Q, \{r_0\})$ est-il toujours fini ?

N'importe quel $y \in B$ vérifie la condition $R(x, 2)$ si $x \in r_0[A]$ et inversement n'importe quel $x \in A$ vérifie la condition $R(3, y)$ si $y \in r_0[B]$. Ainsi si r_0 n'est pas vide $|Ans(Q, \{r_0\})| \geq |A| + |B|$, donc $Ans(Q, \{r_0\})$ n'est donc pas toujours fini.

2.1.4 Existe-t-il une requête équivalente en algèbre relationnelle ? en SQL ?

Contrairement au calcul relationnel, l'algèbre relationnelle et le SQL travaille directement sur les relations et ne peuvent donc pas avoir de résultats qui n'appartiennent à aucune relation.

2.2 $F(x, y) = (R(x, 2) \vee R(3, y)) \wedge R(x, y)$

2.2.1 Q est-elle bien formée ?

F est une combinaison de formule atomique lié par les opérateurs OU et ET, Q est donc bien formée.

2.2.2 Q est-elle autorisée ?

Les variables x et y sont des variables libres, or x et y sont positifs dans $R(x, y)$, donc par l'opérateur ET x et y sont positifs donc Q est autorisé.

2.2.3 Soit r_0 une relation sur R. $Ans(Q, \{r_0\})$ est-il toujours fini ?

Q est une formule de calcul autorisée donc $Ans(Q, \{r_0\})$ est toujours finie.

2.2.4 Existe-t-il une requête équivalente en algèbre relationnelle ? en SQL ?

- Algèbre relationnelle : $\sigma_{B=2 \ OU \ A=3}(r_0)$
- SQL : `SELECT * FROM r0 WHERE B=2 OR A=3`

2.3 $F(x, y) = \exists x' : A(R(x, y) \wedge R(x', y) \wedge \neg(x = x'))$

2.3.1 Q est-elle bien formée ?

F est une combinaison de formules atomiques liées par les opérateurs NOT et ET, Q est donc bien formée.

2.3.2 Q est-elle autorisée ?

Les variables x et y sont des variables libres et x' est déclaré par $\exists x' : A$, or x, y et x' sont positifs dans $R(x, y) \wedge R(x', y)$, donc par l'opérateur ET x, x' et y sont positifs donc Q est autorisé.

2.3.3 Soit r_0 une relation sur R. $Ans(Q, \{r_0\})$ est-il toujours fini ?

Q est une formule de calcul autorisée donc $Ans(Q, \{r_0\})$ est toujours finie.

2.3.4 Existe-t-il une requête équivalente en algèbre relationnelle ? en SQL ?

— Algèbre relationnelle : $\pi_{\{R\}}(\sigma_{A \neq A'}(r_0 \bowtie \rho_{A \rightarrow A'}(r_0))$
— SQL : `SELECT A,B FROM r0, r0 as r1 WHERE r0.B=r1.B AND r0.A<>r1.B`

2.4 $F(x, y) = \forall z : A(\forall t : B(\neg R(z, t) \vee R(t, z)))$

2.4.1 Q est-elle bien formée ?

F est une combinaison de formules atomiques liées par les opérateurs NOT et OU, Q est donc bien formée.

2.4.2 Q est-elle autorisée ?

Les variables x et y sont des variables libres, or x et y n'apparaissent pas dans F, Q n'est donc pas autorisée.

2.4.3 Soit r_0 une relation sur R. $Ans(Q, \{r_0\})$ est-il toujours fini ?

x et y n'apparaissent dans F donc soit $Ans(Q, r_\{0\}) = \emptyset$ soit $Ans(Q, \{r_0\}) = A \times B$. Ainsi $Ans(Q, \{r_0\})$ n'est pas toujours fini.

2.4.4 Existe-t-il une requête équivalente en algèbre relationnelle ? en SQL ?

NON

3 Équivalence des langages de requête

Soient R, S deux symboles de relation avec $schema(R) = \{A, B, C, D\}$ et $schema(S) = \{C, E\}$ et soient r_1, r_2 deux relations sur R et s une relation sur S. Pour chacune des requêtes ci-dessous, donnez les requêtes équivalentes dans les langages vus en cours.

3.1 $\pi_{\{A,B\}}(r_1) \cup \pi_{\{A,B\}}(r_2)$

3.1.1 Calcul relationnel

$Ans(Q, \{r_1\}) \cup Ans(Q, \{r_2\})$ avec $Q = \{< x : A, y : B >| \exists z : C(\exists t : D(R(x, y, z, t))\}$

3.1.2 SQL

```
SELECT A,B
FROM r1
UNION
SELECT A,B
FROM r2
```

3.2 $\pi_{\{A,B\}}(r_1 \cup r_2)$

3.2.1 Calcul relationnel

$Ans(Q, \{r_1 \cup r_2\})$ avec $Q = \{< x : A, y : B >| \exists z : C(\exists t : D(R(x, y, z, t))\}$

3.2.2 SQL

```
SELECT A,B
FROM r1 UNION r2
```

3.3 $Ans(Q, \{r_1, s\})$ avec $Q = \{< x : A, y : B >| \exists z : C(\exists t : D(R(x, y, z, t) \wedge \exists u : E(S(z, u) \wedge u = t)))\}$

3.3.1 Algèbre relationnelle

$\pi_{\{A,B\}}(r_1 \bowtie \rho_{E \to D}(s))$

3.3.2 SQL

```
SELECT A,B
FROM r1,s
WHERE r1.C = s.C AND D = E
```

4 Équivalence de requête

Soient R et U des symboles de relation avec $schema(R) = \{A, B, C, D\}$ et $schema(U) = \{C, D\}$. On considère une base de données contenant les relations r, s et v sur R et u sur U.

4.1 Donner une preuve des équivalences suivantes :

4.1.1 $\pi_{\{A,B\}}(r \cup s) = \pi_{\{A,B\}}(r) \cup \pi_{\{A,B\}}(r)$

Soit t un tuple sur $\{A, B\}$.
$t \in \pi_{\{A,B\}}(r \cup s)$
$\Leftrightarrow \exists t' \in r \cup s \ avec \ t'[A, B] = t$
$\Leftrightarrow \exists t' \in r \ avec \ t'[A, B] = t \ ou \ \exists t'' \in s \ avec \ t''[A, B] = t$
$\Leftrightarrow t \in \pi_{\{A,B\}}(r) \ ou \ t \in \pi_{\{A,B\}}(s)$
$\Leftrightarrow t \in \pi_{\{A,B\}}(r) \cup \pi_{\{A,B\}}(s)$

4.1.2 $r \cap (s \setminus v) = (r \cap s) \setminus (r \cap v)$

Soit t un tuple sur R
$t \in r \cap (s \setminus v)$
$\Leftrightarrow t \in r \ et \ (t \in s \ et \ t \notin v)$
$\Leftrightarrow (t \in rett \in s) \ et \ (t \notin v \ et \ t \in r)$
$\Leftrightarrow t \in r \cap s \ et \ (t \in r \ et \ t \notin r \cap t)$
$\Leftrightarrow t \in (r \cap s) \setminus (r \cap v)$

4.1.3 $(r \div u) \cap (s \div u) = (r \cap s) \div u$

Soit t un tuple sur $\{A, B\}$
$t \in (r \div u) \cap (s \div u)$
$\Leftrightarrow \forall x : ADOM(u), \exists t' \in r \; avec \; t'[A, B] = t \; et \; t'[U] = x \; et \; \forall y :$
$ADOM(u), \exists t'' \in s \; avec \; t''[A, B] = t \; et \; t''[U] = y$
$\Leftrightarrow \forall x : ADOM(u), \exists t' \in r, t'' \in s \; avec \; t'[A, B] = t = t''[A, B] \; et \; t'[U] = x =$
$t''[U]$
$\Leftrightarrow \forall x : ADOM(u), \exists t' \in r \cap s \; avec \; t'[A, B] = t \; et \; t'[U] = x$
$\Leftrightarrow t \in (r \cap s) \div u$

4.2 Donner un contre-exemple aussi concis que possible pour les équivalences suivantes :

Pour les prochaines questions nius considèrerons la base de données suivantes :

r	A	B	C	D
	0	0	0	0

u	A	B	C	D
	0	0	1	1

s	C	D
	0	0
	1	1

4.2.1 $\pi_{\{A,B\}}(r \setminus s) = \pi_{\{A,B\}}(r) \setminus \pi_{\{A,B\}}(s)$

$\pi_{\{A,B\}}(r \setminus s) = \{(0, 0)\}$
$\pi_{\{A,B\}}(r) \setminus \pi_{\{A,B\}}(s) = \emptyset$

4.2.2 $\pi_{\{A,B\}}(r \cap s) = \pi_{\{A,B\}}(r) \cap \pi_{\{A,B\}}(s)$

$\pi_{\{A,B\}}(r \cap s) = \emptyset$
$\pi_{\{A,B\}}(r) \cap \pi_{\{A,B\}}(s) = \{(0, 0)\}$

4.2.3 $(r \div u) \cup (s \cup u) = (r \cup s) \div u$

$(r \div u) \cup (s \cup u) = \emptyset$
$(r \cup s) \div u = \{(0, 0)\}$

Séance de Travaux Dirigés II

Algèbre, Calcul relationnel et SQL

Soit $U = \{num, cnom, pnom, qte, ville, fnom, prix, status\}$ un univers. On définit R un symbole de base de données sur U avec $schema(R) = \{Commande, Fournisseur, Produit\}$ et $schema(Commande) = \{num, cnom, pnom, qte\}$, $schema(Fournisseur) = \{fnom, status, ville\}$, $schema(Produit) = \{pnom, fnom, prix\}$. On considère la base de données d sur R avec $d = \{commandes, fournisseurs, produits\}$ rempli tel que ci dessous :

commandes	num	cnom	pnom	qte
	1535	Jean	Cornas	6
	1854	Jean	Bordeaux	20
	1254	Paul	Chablis	20
	1259	Paul	Chablis	25
	1596	Paul	Cornas	12

fournisseurs	fnom	status	ville
	Vini	SARL	Dijon
	BonVin	SA	Dijon
	Chapoutier	SA	Valence
	SaV	Association	Antraôgues

produits	pnom	fnom	prix
	Cornas	BonVin	20
	Cornas	Chapoutier	18
	Bordeaux	Vini	8.2
	Boudes	Vini	4.3
	Bordeaux	Chapoutier	18.5
	Chapoutier	Chapoutier	5.1
	Chablis	Chapoutier	5

5 Requête dans les trois langages

Pour chaque question ci-dessous, donner une expression en algèbre relationnelle, calcul relationnel autorisé et SQL.

5.1 Donner les informations sur toutes les commandes

5.1.1 Algèbre relationnelle

$commandes$

5.1.2 Calcul relationnel autorisé

$ans(Q, \{commandes\})$ avec $Q = \{< x : num, y : cnom, z : pnom, t : qte >| Commande(x, y, z, t)\}$

5.1.3 SQL

```
SELECT *
FROM commandes
```

5.2 Donner le nom de tous les produits commandés

5.2.1 Algèbre relationnelle

$\pi_{\{pnom\}}(commandes)$

5.2.2 Calcul relationnel autorisé

$ans(Q, \{commandes\})$ avec $Q = \{< x : pnom >| \exists a : num(\exists b : conm(\exists c : qte(Commande(a, b, x, c))))\}$

5.2.3 SQL

```
SELECT pnom
FROM commandes
```

5.3 Donner le nom des produits commandés par Jean

5.3.1 Algèbre relationnelle

$\pi_{\{pnom\}}(\sigma_{cnom='Jean'}(commandes))$

5.3.2 Calcul relationnel autorisé

$ans(Q, \{commandes\})$ avec $Q = \{< x : pnom >| \exists a : num(\exists b : qte(Commande(a,' Jean', x, b)))\}$

5.3.3 SQL

```
SELECT pnom
FROM commandes
WHERE cnom = 'Jean'
```

5.4 Donner le nom des fournisseurs de Bordeaux ou de Cornas à un prix égal à 10 euros

5.4.1 Algèbre relationnelle

$\pi_{\{fnom\}}(\sigma_{pnom='Bordeaux' oupnom='Cornas}(\sigma_{prix=10}(produits)))$

5.4.2 Calcul relationnel autorisé

$ans(Q, \{produits\})$ avec $Q = \{< x : fnom >| (Produit('Cornas', x, 10) \vee Produit('Bordeaux', x, 10)))\}$

5.4.3 SQL

```
SELECT fnom
FROM produits
WHERE qte=10 AND (pnom='Cornas' OR pnom='Bordeaux')
```

5.5 Quels sont les produits dont le nom est le même que le nom des fournisseurs

5.5.1 Algèbre relationnelle

$\pi_{\{pnom\}}(\sigma_{pnom=fnom}(produits))$

5.5.2 Calcul relationnel autorisé

$ans(Q, \{produits\})$ avec $Q = \{< x : pnom >| \exists a : prix(Produit(x, x, a))\}$

5.5.3 SQL

```
SELECT pnom
FROM produits
WHERE pnom = fnom
```

5.6 Donner le nom, le prix et le fournisseur des produits commandés par Jean

5.6.1 Algèbre relationnelle

$\pi_{\{pnom,fnom,prix\}}(\sigma_{cnom='Jean'}(commandes) \bowtie produits)$

5.6.2 Calcul relationnel autorisé

$ans(Q, \{produits, commandes\})$ avec $Q = \{< x : pnom, y : fnom, z : prix >| \exists a : num(\exists b : qte(Produit(x, y, z) \wedge Commande(a,' Jean', x, b)\}$

5.6.3 SQL

```
SELECT pnom, fnom, prix
FROM produits NATURAL JOIN commandes
WHERE cnom = 'Jean'
```

5.7 Quels sont les fournisseurs qui habitent dans la même ville (2 à 2)

5.7.1 Algèbre relationnelle

$\pi_{\{fnom,fnom'\}}(\sigma_{fnom>fnom'}(\rho_{fnom\rightarrow fnom'}(\pi_{\{fnom,ville\}}(fournisseurs)) \bowtie \pi_{\{fnom,ville\}}(fournisseurs)))$

5.7.2 Calcul relationnel autorisé

$ans(Q, \{fournisseurs\})$ avec $Q = \{< x : fnom, y : fnom >| \exists a : ville(\exists b : status(Fournisseur(x, b, a)) \wedge \exists c : status(Fournisseur(y, c, a)) \wedge x > y)\}$

5.7.3 SQL

```
SELECT f1.fnom, f2.fnom
FROM fournisseurs as f1, fournisseurs as f2
WHERE f1.ville = f2.ville AND f1.fnom > f2.fnom
```

5.8 Quels sont les produits qui coûtent 15 euros ou qui sont commandés par Jean ?

5.8.1 Algèbre relationnelle

$$\pi_{\{pnom\}}(\sigma_{cnom='Jean'}(commandes)) \cup \pi_{\{pnom\}}(\sigma_{prix=15}(produits))$$

5.8.2 Calcul relationnel autorisé

$ans(Q, \{commandes, produits\})$ avec $Q = \{< x : pnom >| \exists a : num(\exists b : qte(Commande(a, 'Jean', x, b))) \vee \exists c : fnom(Produit(c, x, 15))\}$

5.8.3 SQL

```
SELECT pnom
FROM commandes
WHERE cnom = 'Jean'
UNION
SELECT pnom
FROM produits
WHERE prix = 15
```

5.9 Quels sont les produits qui n'ont pas été commandés ?

5.9.1 Algèbre relationnelle

$$\pi_{\{pnom\}}(produits) - \pi_{\{pnom\}}(commandes)$$

5.9.2 Calcul relationnel autorisé

$ans(Q, \{commandes, produits\})$ avec $Q = \{< x : pnom >| \forall a : num(\forall b : cnom(\forall c : qte(\exists d : fnom(\exists e : prix(\neg Commande(a, b, x, c) \wedge Produit(x, d, e))))))\}$

5.9.3 SQL

```
SELECT pnom
FROM produits
MINUS
SELECT pnom
FROM commandes
```

5.10 Quels sont les produits commandés en quantité égale à 10 et dont le prix est égal à 15 ?

5.10.1 Algèbre relationnelle

$$\pi_{\{pnom\}}(\sigma_{qte=10}(commandes) \bowtie \sigma_{prix=15}(produits))$$

5.10.2 Calcul relationnel autorisé

$ans(Q, \{commandes, produits\})$ avec $Q = \{< x : pnom >| \exists a : num(\exists b : cnom(\exists c : fnom(Commande(a, b, x, 10) \wedge Produit(x, c, 15))))\}$

5.10.3 SQL

```
SELECT pnom
FROM produits NATURAL JOIN commandes
WHERE qte=10 AND prix=15
```

5.11 Quels sont les produits qui sont fournis par tous les fournisseurs ?

5.11.1 Algèbre relationnelle

$pi_{\{pnom,fnom\}}(produits) \div pi_{\{fnom\}}(fournisseurs)$
$\pi_{\{pnom\}}(produits) - \pi_{\{pnom\}}((\pi_{\{pnom\}}(produits) \times \pi_{\{fnom\}}(fournisseurs) - \pi_{\{pnom,fnom\}}(produits)))$

5.11.2 Calcul relationnel autorisé

$ans(Q, \{fournisseurs, produits\})$ avec $Q = \{< x : pnom >| \exists b : fnom(\exists c : prix(Produit(x, b, c))) \wedge \forall d : fnom(\forall e : status(\forall f : ville(\exists g : prix(Produit(x, d, g)) \vee \neg Fournisseur(d, e, f))))\}$

5.11.3 SQL

```
SELECT pnom
FROM produits
MINUS
SELECT pnom
FROM(
----SELECT pnom,fnom
----FROM
--------(SELECT pnom FROM produits)
----CROSS JOIN
--------(SELECT fnom FROM fournisseurs)
----MINUS
----SELECT pnom,fnom
----FROM produits
)
```

5.12 Quels sont les clients qui ont commandés tous les produits en quantité de 5 ?

5.12.1 Algèbre relationnelle

$\pi_{\{cnom,pnom\}}(\sigma_{qte=15}(commandes)) \div \pi_{\{pnom\}}(produits)$
$\pi_{\{cnom\}}(\sigma_{qte=15}(commandes)) - \pi_{\{cnom\}}(\pi_{\{cnom\}}(\sigma_{qte=15}(commandes)) \times \pi_{\{pnom\}}(produits) - \pi_{\{cnom,pnom\}}(\sigma_{qte=15}(commandes)))$

5.12.2 Calcul relationnel autorisé

$ans(Q, \{commandes, produits\})$ avec $Q = \{< x : cnom >| \exists a : num(\exists b : pnom(Commande(x, a, b, 5))) \wedge \forall c : pnom(\forall d : fnom(\forall e : prix(\exists f : num(Commande(f, x, c, 15)) \vee \neg Produit(c, d, e)))))\}$

5.12.3 SQL

```
SELECT cnom
FROM commandes
WHERE qte=15
MINUS
SELECT cnom
FROM(
----SELECT cnom,pnom
----FROM
--------(SELECT cnom FROM commandes WHERE qte=15)
----CROSS JOIN
--------(SELECT pnom FORM produits)
----MINUS
----SELECT cnom,pnom
----FROM commandes
----WHERE qte = 15
)
```

Séance de Travaux Dirigés III

Contraintes

6 Dépendances Fonctionnelles et d'Inclusion

6.1 Définition

Soient R un symbole de relation, r une relation sur R et $X, Y \subseteq schema(R)$ avec $X \cap Y = \emptyset$. Rappeler les définitions suivantes :

6.1.1 Définir $r \not\models X \to Y$

$r \not\models X \to Y \Leftrightarrow \exists t_1, t_2 \in r \; avec \; t_1[X] = t_2[X] \; et \; t_1[Y] \neq t_2[Y]$

6.1.2 Définir $\{r\} \not\models R[X] \subseteq R[Y]$

$\{r\} \not\models R[X] \subseteq R[Y] \Leftrightarrow \exists t \in r, \; t[X] \notin ADOM(Y)$

6.2 Satisfaction de dépendances

Soit r_0 la relation sur R suivante :

r_0	A	B	C
t_1	1	1	1
t_2	1	1	2
t_3	1	2	3
t_4	1	3	4

6.2.1 Les assertions suivantes sont-elles vraies ? Si non, donner un contre-exemple.

1. $r_0 \models C \to AB$ VRAI
2. $r_0 \models A \to B$ FAUX : $t_2[A] = t_3[A]$ et $t_2[B] \neq t_3[B]$
3. $r_0 \models B \to C$ FAUX : $t_2[B] = t_3[B]$ et $t_2[C] \neq t_3[C]$
4. $\{r_0\} \models R[C] \subseteq R[B]$ FAUX : $4 \in R[C]$ et $4 \notin R[B]$
5. $\{r_0\} \models R[B] \subseteq R[C]$ VRAI

6.2.2 Requête de contre-exemple de $A \to B$

Considérons la dépendance conditionnelle $A \to B$. Écrire une requête qui affiche tous les contre-exemples de $A \to B$ dans r_0, en affichant les 2 tuples du contre-exemple sur une même ligne.

En algèbre : $\sigma_{B>B'}(\rho_{B\to B'}(\pi_{\{A,B\}}(r_0)) \bowtie \pi_{\{A,B\}}(r_0))$

En calcul : $ans(Q, \{r_0\})$ avec $Q = \{< x : A, y : B, z : B >| \exists c : C(\exists c' : C(R(x, y, c) \wedge R(x, z, c') \wedge y > z))\}$

SQL :
```
SELECT r0.A,r0.B,r1.B
FROM r0, r0 as r1
WHERE r0.A = r1.A AND r0.B>r1.B
```

6.2.3 De même avec $R[C] \subseteq R[A]$ dans r_0

En algèbre : $\pi_{\{C\}}(r_0) \setminus \pi_{\{A\}}(r_0)$

En calcul : $ans(Q, \{r_0\})$ avec $Q = \{< x : A >| \exists a : B(\exists b : C(R(x, a, b))) \wedge \forall c : A(\forall d : B(\neg R(c, d, x)))\}$

SQL :
```
SELECT A
FROM r0
MINUS
SELECT C
FROM r0
```

6.3 Caractérisation de la "satisfaction" des dépendances

6.3.1 Trouver une expression algébrique équivalente à $\{r, s\} \models R[X] \subseteq S[Y]$

$\pi_{\{X\}}(r) - \rho_{Y \to X}(\pi_{\{Y\}}(s)) = \emptyset$

6.3.2 Démontrer là

$\pi_{\{X\}}(r) - \rho_{Y \to X}(\pi_{\{Y\}}(s)) = \emptyset$
$\Leftrightarrow \nexists t \in r[X]$ *et* $t \notin s[Y]$
$\Leftrightarrow r[X] \subseteq s[Y]$
$\Leftrightarrow \{r, s\} \models R[X] \subseteq S[Y]$

6.3.3 Trouver une expression algébrique équivalente à $r \models X \to Y$

$|\sigma_{\{Y \neq Y'\}}(\rho_{Y \to Y'}(\pi_{\{XY\}}(r)) \bowtie \pi_{\{XY\}}(r)))| = 0$

6.3.4 Démontrer là

$|\sigma_{\{Y \neq Y'\}}(\rho_{Y \to Y'}(\pi_{\{XY\}}(r)) \bowtie \pi_{\{XY\}}(r)))| = 0$
$\Leftrightarrow \nexists t \in \sigma_{\{Y \neq Y'\}}(\rho_{Y \to Y'}(\pi_{\{XY\}}(r)) \bowtie \pi_{\{XY\}}(r)))$
$\Leftrightarrow \nexists t \in \rho_{Y \to Y'}(\pi_{\{XY\}}(r)) \bowtie \pi_{\{XY\}}(r))$ *tel que* $t[Y] \neq t[Y']$
$\Leftrightarrow \nexists t', t'' \in r,$ *tel que* $t'[X] = t''[X]$ *et* $t[Y] \neq t[Y']$
$\Leftrightarrow r \models X \to Y$

6.4 Preuve d'implication

Soient les notations suivantes :
— r,s deux relations sur R
— t une relation sur T avec $schema(T) \cap schema(R) = \emptyset$
— u une relation sur U
— $X, Y \subseteq schema(R)$

En supposant que $r \models X \to Y$, indiquer si les expressions ci-dessous sont vraies ou fausses. Si elles sont vraies, donner une démonstration succincte. Si elles sont fausses, donner un contre-exemple.

6.4.1 $\sigma_F(r) \models X \to Y$

$\sigma_F(r) \subseteq r$ *et* $r \models X \to Y$
$\Rightarrow \sigma_F(r) \models X \to Y$

6.4.2 $r \cup s \models X \to Y$

Dans les relations ci-dessous on a bien $r \models X \to Y$ mais $r \cup s \not\models X \to Y$

r	X	Y
	0	0

s	X	Y
	1	0

$r \cup s$	X	Y
	0	0
	1	0

6.4.3 $r \setminus s \models X \to Y$

$r \setminus s \subseteq r$ *et* $r \models X \to Y$
$\Rightarrow r \setminus s \models X \to Y$

6.4.4 Soit $X \cup Y \subseteq W$. $\pi_W(r) \models X \to Y$

$\pi_W(r) \subseteq r$ *et* $r \models X \to Y$
$\Rightarrow \pi_W(r) \models X \to Y$

6.4.5 $r \bowtie s \models X \to Y$

$r \bowtie s = r \cap s \subseteq r$ *et* $r \models X \to Y$
$\Rightarrow r \bowtie s \models X \to Y$

6.4.6 $r \bowtie t \models X \to Y$

$r \bowtie t = r \times t$ or $(r \times t)[R] = r$ et $r \models X \to Y$
$\Rightarrow r \bowtie t \models X \to Y$

6.4.7 $r \bowtie u \models X \to Y$

$(r \bowtie u)[XY] \subseteq r[XY]$ et $r \models X \to Y$
$\Rightarrow r \bowtie u \models X \to Y$

7 Raisonnement sur les DF

7.1 Théorie de la preuve

En utilisant les règles d'inférence d'Armstrong, donner une dérivation (ou une preuve) de :

7.1.1 $\{A \to B, A \to C\} \vdash A \to BC$

$\{A \to B, A \to C\}$
$\Rightarrow \{A \to AB, AB \to BC\}$ par Augmentation
$\Rightarrow A \to BC$ par Transitivité

7.1.2 $\{A \to BC\} \vdash A \to B$

$\{A \to BC\}$ et par réflexivité $B \subseteq BC \Rightarrow BC \to B$
$\Rightarrow A \to B$ par transitivité

7.2 Problème de l'implication

Soit R un schéma de relation avec $schema(R) = \{A, B, C, D, E, F\}$ et $F = \{ABC \to E, BD \to A, CF \to B\}$ un ensemble de DF sur R.

7.2.1 Montrer que $F \vdash CDF \to E$

$F \vdash CF \to B$
$\Rightarrow F \vdash CDF \to BD$ par augmentation
$\Rightarrow F \vdash CDF \to ABD$ par transitivité
$\Rightarrow F \vdash CDF \to ABCD$ par réflexivité
$\Rightarrow F \vdash CDF \to ABCDE$ par transitivité
$\Rightarrow F \vdash CDF \to E$ par décomposition

7.2.2 Montrer que $F \models CDF \to E$

Soit r une relation sur R qui satisfait F et t_1 et t_2 deux tuples de r avec $t_1[CDF] = t_2[CDF] \Rightarrow t_1[BD] = t_2[BD] \Rightarrow t_1[ABC] = t_2[ABC] \Rightarrow t_1[E] = t_2[E] \Rightarrow F \models CDF \to E$

7.2.3 Calculer CDF_F^+

$CDF_F^+ = BCDF_F^+ = ABCDF_F^+ = ABCDEF$ Ainsi CDF est une superclé de R.

Séance de Travaux Dirigés IV

Conception de schémas normalisés

8 Projection et couverture d'un ensemble de DF

Soit R un symbole de relation avec $schema(R) = \{A, B, C, D\}$. Soit F un ensemble de DF sur R avec $F = \{A \to CD, AC \to B, BC \to AD\}$.

8.1 Calculer la projection de F sur AB, AD et ABC

Par transitivité $F \vdash A \to CD$ et $F \vdash AC \to B$ implique $F \vdash A \to B$ donc :
$F[AB] = \{A \to B\}$
$F[AD] = \{A \to D\}$
$F[ABC] = \{A \to C, AC \to B, BC \to A\}$

8.2 Calculer une couverture non redondante de F.

Pour trouver une couverture non-redondante de F on essaie d'enlever une des DF de F et de la retrouver avec les autres.

8.2.1 Si on enlève $A \to CD$

Soit r une relation sur R vérifiant $F \setminus \{A \to CD\}$.

r	A	B	C	D
t_0	0	0	0	0
t_1	0	1	1	1

On a bien $r \models AC \to B$ et $r \models BC \to AD$ mais $r \not\models A \to CD$ car $t_0[A] = t_1[A]$ et $t_0[CD] \neq t_1[CD]$. On ne peut donc pas enlever $A \to CD$ si l'on veut avoir une couverture de F.

8.2.2 Si on enlève $AC \to B$

Soit r une relation sur R vérifiant $F \setminus \{AC \to B\}$.

r	A	B	C	D
t_0	0	0	0	0
t_1	0	1	0	0

On a bien $r \models A \to CD$ et $r \models BC \to AD$ mais $r \not\models AC \to B$ car $t_0[AC] = t_1[AC]$ et $t_0[B] \neq t_1[B]$. On ne peut donc pas enlever $AC \to B$ si l'on veut avoir une couverture de F.

8.2.3 Si on enlève $BC \to AD$

Soit r une relation sur R vérifiant $F \setminus \{BC \to AD\}$.

r	A	B	C	D
t_0	0	0	0	0
t_1	1	0	0	0

On a bien $r \models A \to CD$ et $r \models AC \to B$ mais $r \not\models BC \to AD$ car $t_0[BC] = t_1[BC]$ et $t_0[AD] \neq t_1[AD]$. On ne peut donc pas enlever $BC \to AD$ si l'on veut avoir une couverture de F.

8.2.4 Conclusion

F est déjà une couverture non-redondante de lui-même.
Elle n'est pas unique car dans la question suivante on va trouver une couverture minimale (donc non-redondante) de F différente.

8.3 Calculer une couverture minimale de F

$F \vdash A \rightarrow CD$ et $F \vdash AC \rightarrow B$ par transitivité
$\Rightarrow F \vdash A \rightarrow BCD$
$F \vdash A \rightarrow BCD$ par décomposition et Augmentation
$F \vdash A \rightarrow CD$ et $F \vdash AC \rightarrow B$

Ainsi $H = \{A \rightarrow BC, BC \rightarrow AD\}$ est la couverture minimale de F

9 Fermés et relation d'Armstrong

Soit R un symbole de relation avec $schema(R) = \{A, B, C, D\}$. Soit F un ensemble de DF sur R avec $F = \{A \rightarrow CD, AC \rightarrow B, BC \rightarrow AD\}$.

9.1 Calculer $CL(F)$ de R par rapport à F

A_F^+	$ABCD$	AB_F^+	$ABCD$	ABC_F^+	$ABCD$
B_F^+	B	AC_F^+	$ABCD$	ABD_F^+	$ABCD$
C_F^+	C	AD_F^+	$ABCD$	ACD_F^+	$ABCD$
D_F^+	D	BC_F^+	$ABCD$	BCD_F^+	$ABCD$
		BD_F^+	BD		
		CD_F^+	CD		

$CL(F) = \{\emptyset, B, C, D, BD, CD, ABCD\}$

9.2 En déduire l'ensemble des éléments irréductibles $IRR(F)$

$\emptyset = B \cap C \Rightarrow IRR(F) = \{B, C, BD, CD\}$

9.3 Générer une relation d'Armstrong r_0 pour F

r_0	A	B	C	D
t_0	0	0	0	0
t_1	1	0	1	1
t_2	2	2	0	2
t_3	3	3	3	0
t_4	4	0	4	0
t_4	5	5	0	0

10 Algorithmes de décomposition

10.1 Premier exemple

Soit R un symbole de relation avec $schema(R) = \{A, B, C\}$. Soit F un ensemble de F sur R avec $F = \{A \rightarrow B, B \rightarrow C\}$.

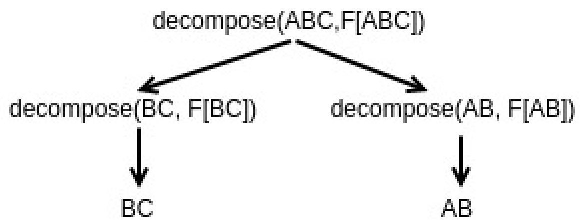
10.1.1 Donner une décomposition en 3FN de R par rapport à F

Pour obtenir une décomposition en 3FN on utilise l'algorithme de synthèse. On crée donc un symbole de relation R_i pour chaque DF de F, cela donne :

- R_0 avec $schema(R_0) = \{A, B\}$
- R_1 avec $schema(R_1) = \{B, C\}$

10.1.2 Donner une décomposition en FNBC de R par rapport à F

Pour obtenir une décomposition en FNBC on utilise l'algorithme récursif de décomposition avec $F = \{A \to AB, B \to C\}$



Résultat :

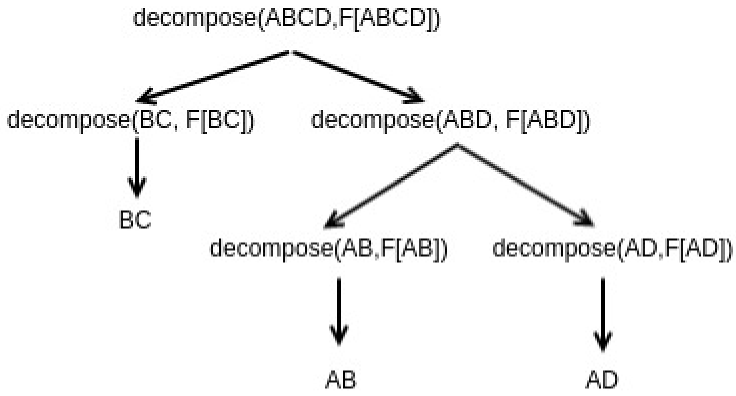
- R_0 avec $schema(R_0) = \{A, B\}$
- R_1 avec $schema(R_1) = \{B, C\}$

10.2 Même questions pour $schema(R) = \{A, B, C, D\}$

10.2.1 Décomposition 3FN

- R_0 avec $schema(R_0) = \{A, B\}$
- R_1 avec $schema(R_1) = \{B, C\}$
- R_2 avec $schema(R_2) = \{A, B, C, D\}$

10.2.2 Décomposition FNBC



Résultat :
- R_0 avec $schema(R_0) = \{A, B\}$
- R_1 avec $schema(R_1) = \{B, C\}$
- R_2 avec $schema(R_2) = \{A, D\}$

11 Relations d'Armstrong

Soit R un symbole de relation avec $schema(R) = \{A, B, C\}$. Soit F un ensemble de F sur R avec $F = \{A \to B, B \to C\}$.

11.1 Donner une relation d'Armstrong r_0 pour F

A_F^+	ABC	AB_F^+	ABC
B_F^+	BC	AC_F^+	ABC
C_F^+	C	BC_F^+	BC

Ainsi $CL(F) = \{\emptyset, C, BC, ABC\}$ donc $IRR(F) = \{\emptyset, C, BC\}$

r_0	A	B	C
t_0	0	0	0
t_1	1	1	1
t_2	2	2	0
t_3	3	0	0

11.1.1 De même pour $F = \{AB \to C, BC \to A\}$

A_F^+	A	AB_F^+	ABC
B_F^+	BC	AC_F^+	AC
C_F^+	C	BC_F^+	ABC

Ainsi $CL(F) = \{\emptyset, A, B, C, AC, ABC\}$ donc $IRR(F) = \{A, B, C, AC\}$

r_0	A	B	C
t_0	0	0	0
t_1	0	1	1
t_2	2	0	2
t_3	3	3	0
t_4	0	4	0

12 Conception à partir d'un texte

Une conférence est organisée en différentes sessions, chaque session portant un nom et ayant en moyenne trois papiers. Un papier est caractérisé par un titre, une liste ordonnée d'auteurs et un nombre de pages et est présentée à une seule conférence. La session est dirigée par un président de session issu des membres du comité de programme. Chaque papier est évalué par plusieurs membres du comité de programme (CP). Une évaluation comporte une note et un commentaire.
Modéliser cette application en utilisant l'approche de conception dite de la relation universelle.

12.1 Identifier les attributs décrivant l'univers du discours

$U = \{conference, session, nomSession, papier, titrePapier, auteur, rangAuteur$
$nomAuteur, nPages, presSession, membreCP, nomMembre, evaluation, note, commentaire\}$

12.2 Coder les contraintes avec des Dépendances Fonctionnelles

Soit F un ensemble de DF sur U,F contient :
- $session \rightarrow conference$
- $session \rightarrow nomSession$
- $session \rightarrow presSession$
- $papier \rightarrow titrePapier$
- $papier \rightarrow nPages$
- $papier \rightarrow session$
- $papier, auteur \rightarrow rangAuteur$
- $papier, rangAuteur \rightarrow auteur$
- $auteur \rightarrow nomAuteur$
- $membreCP \rightarrow nomMembre$
- $membreCP, papier \rightarrow evaluation$
- $evaluation \rightarrow note$
- $evaluation \rightarrow commentaire$

La couverture minimale de F est :
- $session \rightarrow conference, nomSession, presSession$
- $papier \rightarrow titrePapier, nPages, session$
- $papier, auteur \rightarrow rangAuteur$
- $papier, rangAuteur \rightarrow auteur$
- $auteur \rightarrow nomAuteur$
- $membreCP \rightarrow nomMembre$
- $membreCP, papier \rightarrow evaluation$

— $evaluation \rightarrow note, commentaire$

12.3 Proposer un schéma de base de données en 3FN

Soit R un symbole de base de données sur U,on trouve $schema(R)$ en 3FN avec l'algorithme de synthèse sur la couverture minimale de F ci-dessus.
Résultat :

— R_0 avec $schema(R_0) = \{session, conference, nomSession, presSession\}$
— R_1 avec $schema(R_1) = \{papier, titrePapier, nPages, session\}$
— R_2 avec $schema(R_2) = \{auteur, nomAuteur\}$
— R_3 avec $schema(R_1) = \{papier, auteur, rangAuteur\}$
— R_4 avec $schema(R_4) = \{membreCP, nomMembre\}$
— R_5 avec $schema(R_5) = \{membreCP, papier, evaluation\}$
— R_6 avec $schema(R_6) = \{evaluation, note, commentaire\}$