Corrigé de l'interrogation de bases de données
Licence d'Informatique - IUP Miage
Université de Paris XI

V. Benzaken, D. Faure, S. Raynaud, S. Rey, V. Ventos

18 avril 2000

Exercice 1 (S. Raynaud)

Soit le schéma R(A,B,C,D,E) et l'ensemble F={A→BC, B→D, D→E, C→AD} de dépendances fonctionnelles.

Pour trouver les clés de ce schéma, il faut calculer la fermeture de chaque ensemble d'attributs de R.

A

$[A]^+$ = A avec A→BC on a $[A]^+$ = ABC

$[A]^+$ = ABC avec B→D on a $[A]^+$ = ABCD

$[A]^+$ = ABCD avec D→E on a $[A]^+$ = ABCDE

Finalement : $[A]^+$ = ABCDE donc A est une clé

C

$[C]^+$ = C avec C→AD on a $[C]^+$ = ACD

$[C]^+$ = ACD avec A→BC on a $[C]^+$ = ABCD

$[C]^+$ = ABCD avec D→E on a $[C]^+$ = ABCDE

Finalement : $[C]^+$ = ABCDE par conséquent C est une clé

A et C sont des clefs. En effet, A et C sont des singletons. Or, B, D, E, BD, BE, DE et BDE ne sont pas des clés puisque leur fermeture ne ramène pas tous les éléments de R.

$[B]^+$ = BDE, $[D]^+$= $[DE]^+$ = DE, $[E]^+$ = E et il en est de même pour $[BD]^+$ = $[BE]^+$ = $[BDE]^+$ = BDE.

Les autres ensembles d'attributs (ex : AD, BCE, etc) sont en fait des surclés qui seraient composées de A et/ou de C et donc pas des clés. Par conséquent, les seules clés du schéma sont A et C.

Rappel :

une clé est une surclé. Une surclé n'est pas forcément une clé.

Exercice 2 (S. Rey)

a)

Trouvez les dépendances fonctionelles pour l'instance cine de schéma CINEMA.

- salle $\rightarrow$ film
- salle, nom $\rightarrow$ film
- salle, snack $\rightarrow$ film
- film $\rightarrow$ salle
- film, nom $\rightarrow$ salle
- film,snack $\rightarrow$ salle
- nom, salle, snack $\rightarrow$ film
- nom, film, snack $\rightarrow$ salle

b)

Soit Γ = {AB→C, A→D, CD→EF}. Montrer que l'on peut déduire AB→F avec les seules trois règles d'Armstrong.

A→D $\vdash_{augmentation}$ AC→CD
AC→CD, CD→EF $\vdash_{transitivite}$ AC→EF
AB→C $\vdash_{augmentation}$ AB→AC
AB→AC, AC→EF $\vdash_{transitivite}$ AB→EF
De plus on a F $\subset$ EF $\vdash_{reflexivite}$ EF→F.
AB→EF et EF→F $\vdash_{transitivite}$ AB→F

Exercice 3 (D. Faure)

F_1={P → M; P,H → C; H,S → P; H,C → S } et F_2= {P → M; H,S → P; H,C → S }

a)

On remarque que $F_2 \subset F_1$. Pour que F_1 soit équivalent F_2, il faut que la dépendance supplémentaire de F_1 puisse tre déduite partir de F_2.

Calculons $[P, H]^+$ par rapport F_2 pour essayer de déduire la dépendance manquante.

$[P, H]^+$ = P,H,M

On ne peut dont pas déduire la DF P,H → C donc nous en déduisons que F_2 n'est pas équivalent F_1.

b)

Décomposition BCNF selon F_2

F_2= {P → M; H,S → P; H,C → S }

Calcul de la clef pour F_2. $[H,C]^+$ = {H,C,S,P,M} donc {H,C} est une surclef. Or, $[H]^+$ = {H} et $[C]^+$ = {C}. Donc H,C est une clef.

La DF P → M viole BCNF.

Calcul de $[X]^+$: S_1 = **{P;M}**, DF_1 = {P → M} avec P comme clef. S_1 est en **BCNF**.

Calcul de $[(R-X^+)\bigcup X\}$: S_2 = {H;S;C;P}, DF_2 = {H,S → P; H,C → S} avec HC comme clef.

La DF H,S → P viole BCNF.

Calcul de $[X]^+$: S_3 = **{H;S;P}**, DF_3 = { H,S → P } avec HS comme clef. S_3 est en **BCNF.**

Calcul de $[(R-X^+)\bigcup X\}$: S_4 = **{C;H,S}**, DF_4 = {H,C → S} avec HC comme clef. S_4 est en **BCNF.**

Décomposition selon F_1

F_1={P → M; P,H → C; H,S → P; H,C → S }

HC est une clef car $[H,C]^+$ = {HCSPM}, et $[H]^+$ = H et $[C]^+$ = C.

P → M viole BCNF.

Calcul de $[X]^+$: $[P]^+$ = PM

S_1 = **{P;M}**, DF_1 = {P → M} avec P comme clef. S_1 est en **BCNF.**

Calcul de $[(R-X^+)\bigcup X\}$: S_2 = {H;S;C;P}, DF_2 = {P,H → C; H,S → P; H,C → S }

PH est une clef car $[P;H]^+$ = PHSC et $[P]^+$ = {P} et $[H]^+$ = {H}.

HS est une clef car $[H;S]^+$ = HSPC et $[S]^+$ = {S}.

HC est une clef car $[H;C]^+$ = HCPS et $[C]^+$ = {C}.

aucune DF ne viole BCNF donc S_2 est en BCNF.

Exercice 4 (V. Ventos)

a)

r1	vol	avion	r2	vol	pilote
	v1	a1		v1	p1
	v2	a1		v2	p1
	v3	a2		v3	p2
	v4	a3		v4	p2

b)

r1	vol	avion
	v1	a1
	v2	a1
	v3	a2
	v4	a1

Remarque preliminaire :

La decomposition ne correspond pas a l'application de l'algorithme conduisant a une décomposition en forme BCNF

Si l'on decompose la relation R en r1 et r2 comme dans le a) et que l'on remplace dans r1 le n-uplet (v4,a3) par (v4,a1), nous aurons les problemes suivants :

1. l'avion a3 n'apparait plus dans la base (mais c'est aussi le cas si l'on remplace le n-uplet (v4,a3,p2) par le n-uplet (v4,a1,p2) ou (v4,a1,P1) en respectant la DF avion $\rightarrow$ pilote.
2. on perd l'information a3 pilote par p2
3. il faut mettre a jour r2 en remplacant le n-uplet (v4,p2) par (v4,p1) si l'on veut que la DF (perdue dans notre decomposition) avion $\rightarrow$ pilote soit verifiee et que la base ne soit pas incoherente.

Exercice 5 (`V. Benzaken`)

Soit F un ensemble de dépendances fonctionnelles sur le schéma R(U). Soit X $\rightarrow$ Y une dépendance de F. Soit A un attribut de U. On dit que A est étranger à X $\rightarrow$ Y relativement à F si :

- (F - { X $\rightarrow$ Y }) $\cup$ { X $\rightarrow$ (Y - A) } $\vdash$ X $\rightarrow$ Y ou si
- (F - { X $\rightarrow$ Y }) $\cup$ { (X - A) $\rightarrow$ Y } $\vdash$ X $\rightarrow$ Y

Développez un algorithme qui prenne en entrée un ensemble F de dépendances fonctionnelles et produise comme résultat un ensemble F' de dépendances qui lui soit équivalent mais ne possédant aucun attributs étrangers.

Préambule

Je rappelle qu's'agissait d'un exercice de reflexion.

Que nous dit le définition ?

La définition contient deux cas : étranger en partie gauche de dépendances et étranger en partie droite. Dans les deux cas, un attribut A est étranger à une dépendance si l'on peut l'enlever de la DF (obtenant ainsi un nouvel ensemble de dépendances F') sans que ceci n'altère l'ensemble des conséquences logiques de F. C'est à dire que F et F' sont équivalents ($F^+ = F'^+$).

Comment ca marche ?

En premier lieu, à quoi peut bien servir (très intuitivement) une telle notion? Il faut s'en convaincre à partir d'exemples.

Soit F = { A $\rightarrow$ BC, B $\rightarrow$ A, AD $\rightarrow$ E, BD $\rightarrow$ C }

- B est t'il étranger à gauche pour BD $\rightarrow$ C ? Pour le vérifier il faut calculer $[BD]^+$ relativement à { A $\rightarrow$ BC, B $\rightarrow$ A, AD $\rightarrow$ E, D $\rightarrow$ C} pour savoir si on retrouve BD $\rightarrow$ C

  $[BD]^+$ = BDCAE donc oui.
- A est t'il étranger à gauche pour AD $\rightarrow$ E

  calculons $[AD]^+$ relativement à { A $\rightarrow$ BC, B $\rightarrow$ A, D $\rightarrow$ E, D $\rightarrow$ C }.

  $[AD]^+$ = ADCEB donc oui.

- C est t'il étranger à droite pour $A \rightarrow BC$

  Calculons $[A]^+$ pour { $A \rightarrow B$, $B \rightarrow A$, $D \rightarrow E$, $D \rightarrow C$ }.

  $[A]^+ = AB$ donc non.

- B est t'il étranger à droite pour $A \rightarrow BC$

  De même calculons $[A]^+$ pour { $A \rightarrow C$, $B \rightarrow A$, $D \rightarrow E$, $D \rightarrow C$ }

  $[A]^+ = AC$ donc non.

Ce que nous obtenons est un ensemble F' = { $A \rightarrow BC$, $B \rightarrow A$, $D \rightarrow CE$} qui est équivalent à F et contenant moins de dépendances et moins de symboles d'attributs que F.

La définition vous a troublé ? Adaptez la !

Pour la première règle (étranger à droite) si Y se réduit à un singleton (un unique attribut B) alors tout attribut A distinct de B est étranger. ((B-A) = B) ce qui n'est pas très interessant relativement au problème qui était posé.

De même, pour la seconde règle (étranger à gauche) si X se réduit au singleton B, tout attribut A différent de B est étranger. En fait, pour cette règle, tout attribut est étranger!

La définition n'est véritablement utile que si l'on considére qu'elle ne s'applique qu'aux dépendances dont la partie gauche et/ou droite n'est pas réduite à un singleton.

L'algorithme peut alors s'exprimer par :

```
Reduire(F)
  reduire-Gauche(F);
  compacter(F);
  reduire-Droit(F);
fin

reduire-Gauche(G)
 pour chaque X --> Y dans G telle que
          X = A1,...An (n>1)
 faire
          si G-etranger(Ai, G) alors
              G := G - {X-->Y} U {(X-Ai) --> Y}
          fsi
 finpour
fin

compacter(G)
/* transforme toutes les DF de meme partie gauche
    en une seule */

reduire-Droit(G)
 pour chaque X --> Y dans G telle que
           Y = B1,...,Bk (k>1)
 faire
           si D-etranger(Bi, G) alors
```

```
                G := G -{X--> Y} U {X -->(Y-Bi)}
            fsi
 finpour
fin

G-etranger(Ai, G)
/*  calcul de [X]+ relativement a G - {X-->Y} U {(X-Ai) --> Y}
  si Y est dans [X]+ alors return(true) sinon false.
*/

similaire pour D-etranger(Bi,G)
```

Il y avait d'autres formulations possibles. Cet algorithme est correct, son résultat est bien un ensemble de df équivalent à l'ensemble initial puisque à chaque pas où l'on modifie F, on ne le fait qu'en vérifiant que l'ensemble obtenu lui est équivalent.

Reprenons notre exemple

F = { $A \to BC$, $B \to A$, $AD \to E$, $BD \to C$ }

Réduire à gauche :

- A pour $AD \to E$?
  { $A \to BC$, $B \to A$, $D \to E$, $BD \to C$ } $[AD]^+ = ADEBC$ oui.
- B pour $BD \to C$
  { $A \to BC$, $B \to A$, $D \to E$, $D \to C$ } $[BD]^+ = ABCDE$ donc oui.

Compacter
{ $A \to BC$, $B \to A$, $D \to EC$ }
Reduire à gauche F reste inchangé.

Remarque

Suivant le choix des attributs candidats on trouve des résultats différents.

Réduire à gauche :

- D pour $AD \to E$?
  { $A \to BC$, $B \to A$, $A \to E$, $BD \to C$ } $[AD]^+ = ADEBC$ oui.
- D pour $BD \to C$
  { $A \to BC$, $B \to A$, $A \to E$, $B \to C$ } $[BD]^+ = ABCDE$ donc oui.

Compacter
{ $A \to BCE$, $B \to AC$ }
Reduire à gauche

- C pour $A \to BCE$
  { $A \to BE$, $B \to AC$ } $[A]^+ = ABCE$ donc oui.

Résultat

$$\{ A \to BE, B \to AC \}$$

En fait, Dave Maier a démontré dans un article paru dans le Journal of the ACM en 1980 que cette dernière solution qui possède un nombre minimum de dépendances et qui met en jeu un nombre minimum de symboles d'attributs peut être obtenue par un algorithme dont la complexité est de l'ordre de $O(n^2)$.

Conclusion

En fait, à partir d'un ensemble initial d'attributs U et un ensemble de dépendances F sur U nous avons simplifié F. C'est à dire que nous avons obtenu F' équivalent à F mais contenant moins de DF et moins de symboles d'attributs. Cela **ne veut pas dire** que nous avons éliminé les attributs étrangers de notre schéma. U reste inchangé, seules les dépendances ont été simplifiées.

Il est alors possible d'opérer la décomposition BCNF de U par rapport à F'. Les attributs étrangers qui n'aparaissent plus dans les DF apparaitront dans les clefs !