Bases de données

L2 sciences et technologies, mention informatique

les requêtes conjonctives

ou : comment extraire de l'information de ceci ?

films	titre	réalisateur	année
	starwars	lucas	1977
	nikita	besson	1990
	locataires	ki-duk	2005
	dune	lynch	1984

réalisateurs	nom	nationalité
	lucas	américaine
	lynch	américaine
	besson	française
	ki-duk	coréenne

patrick.marcel@univ-tours.fr
http://celene.univ-tours.fr/course/view.php?id=3131

requête

exemple de requêtes :

1. qui est le réalisateur de “dune” ?
2. en quelle année est sorti “nikita” ?
3. quelle est la nationalité du réalisateur de “locataires” ?
4. lister les films réalisés par des américains

la requête 4

"lister les films réalisés par des américains"

avec des variables portant sur les tuples :

si il y a des tuples t_1, t_2 dans *films* et *réalisateurs*
tels que *nationalité* de t_2 est "américaine"
et *réalisateur* de t_1 = *nom* de t_2
alors le résultat contient *titre* de t_1

la requête 4

"lister les films réalisés par des américains"

avec des variables portant sur les constantes du domaine :

si il y a des tuples $(r,$"américaine"$)$,(t,r,a) dans *réalisateurs* et *films*
alors le tuple (t) fait partie du résultat

la requête 4

sous forme de règle :

$\text{film\_américain}(t) \leftarrow \text{réalisateurs}(r, \text{"américaine"}), \text{films}(t, r, a).$

si

- il existe une valeur de r associée à "américaine" dans l'instance de réalisateurs, et
- on retrouve cette valeur dans l'instance de films

alors la valeur de t associée à la valeur de r dans l'instance de films fait partie du résultat

requête conjonctive exprimée sous forme de règle

langage à base de règles

une *requête conjonctive* sur un schéma de base de données D est une expression de la forme :

$$ans(u) \leftarrow R_1(u_1), \ldots, R_n(u_n)$$

- $ans(u)$ est appelé la *tête* de la règle
- $R_1(u_1), \ldots, R_n(u_n)$ est appelé le *corps* de la règle
- les $R(u_i)$ sont appelés des *atomes*

dans cette règle

R_i est un nom de relation de D

$ans \notin D$ est un nom de relation

u_i est une expression de la forme $e_1, \ldots, e_{m_i}$

les e_j sont des variables de **var** ou constantes de **dom**

les variables de cette règle

condition de *champs restreint* :

chaque variable apparaissant dans u doit apparaitre au moins une fois dans $u_1, \ldots, u_n$

on note $var(q)$ l'ensemble des variables de la requête q

exemple

"qui est le réalisateur de dune ?"

$$ans(r) \leftarrow films(\text{“}dune\text{”},r,a).$$

"en quelle année est sorti nikita ?"

$$ans(a) \leftarrow films(\text{“}nikita\text{”},r,a).$$

valuation

soit $V \subset \mathbf{var}$

une *valuation* v sur V est une fonction de V dans **dom**

une valuation v associe une valeur à chaque variable

tuple libre

soit U un ensemble d'attributs, dans l'approche nommée

un *tuple libre* sur U est une fonction de U dans $\mathbf{var} \cup \mathbf{dom}$

soit t un tuple libre et v une valuation

$v(t)$ est le tuple t où les variables sont remplacées par leur valuation

exemple

soit $V = \{t,r,a\} \subset \mathbf{var}$

v_1, v_2, v_3 sont trois valuations :

- $v_1(t) =$ starwars, $v_1(r) =$ lucas, $v_1(a) = 1977$
- $v_2(t) =$ dune, $v_2(r) =$ lynch, $v_2(a) = 1984$
- $v_3(t) = 1977$, $v_3(r) = 1984$, $v_3(a) = 1977$

$v_1(\mathit{films}(t,r,a)) =$ films(starwars, lucas, 1977)
$v_2(\mathit{films}(t,r,a)) =$ films(dune, lynch, 1984)
$v_3(\mathit{films}(t,r,a)) =$ films(1977, 1984, 1977)

l'image d'une requête q

q une requête $ans(u) \leftarrow R_1(u_1), \ldots, R_n(u_n)$

I une instance de base de données de schéma D

l'*image* de (la réponse à) q sur I est :

$$q(I) = \{v(u) | v \text{ est une valuation sur } var(q) \text{ et } v(u_i) \in I(R_i) \text{ pour tout } i \in [1,n] \}$$

exemple

la requête 4 : film_américain(t) ←réalisateurs(r,"américaine"),films(t,r,a).

I : l'instance de base de données suivante :

I={films(starwars,lucas,1977),films(nikita,besson,1990),
films(locataires,ki-duk,2005),films(dune,lynch,1984)
réalisateurs(lucas,américaine),réalisateurs(lynch,américaine),
réalisateurs(ki-duk,coréenne),réalisateurs(besson,française)}

exemple

les valuations v_1 et v_2 telles que :

- $v_1(\mathrm{t}) = \mathrm{starwars},\ v_1(\mathrm{r}) = \mathrm{lucas},\ v_1(\mathrm{a}) = 1977$
- $v_2(\mathrm{t}) = \mathrm{dune},\ v_2(\mathrm{r}) = \mathrm{lynch},\ v_2(\mathrm{a}) = 1984$

$q(I) = \{\mathrm{film\_américain}(\text{“starwars”}), \mathrm{film\_américain}(\text{“dune”})\}$

constitue la réponse à la requête

exemple

$ans(w) \leftarrow films(x,y,z)$

n'est pas à champs restreint

la réponse à cette requête est infinie...

domaine actif

pour I une instance de base de données, q une requête

on note :

$adom(I)$	l'ensemble des constantes apparaissant dans I *domaine actif de l'instance*
$adom(q)$	l'ensemble des constantes apparaissant dans q *domaine actif de la requête*

exemple

dans l'exemple précédent :

$adom(I)$={starwars, lucas, américaine, 1984, dune, ...}

$adom(q)$={américaine}

qu'est-ce que $q(I)$?

on note $adom(q,I) = adom(q) \cup adom(I)$

q est une requête à champs restreint sur I

donc $adom(q(I)) \subseteq adom(q,I)$

donc $q(I)$ est un ensemble fini

donc c'est une instance

extension et intention

$$ans(u) \leftarrow R_1(u_1), \ldots, R_n(u_n)$$

si les relations R_i sont stockées

on dit qu'elles sont définies en *extension*

si *ans* n'est pas stockée

c'est une relation définie en *intention*

requête booléenne

exemple : connaît-on un film sorti en 1990 ?

$$ans() \leftarrow film(t,r,1990)$$

réponse

$\{()\}$ si oui

$\emptyset$ sinon

le calcul conjonctif

calcul conjonctif

$ans(e_1,\ldots,e_m) \leftarrow R_1(u_1),\ldots,R_n(u_n)$

variante syntaxique :

$\{e_1,\ldots,e_m | \exists x_1,\ldots,x_k(R_1(u_1) \wedge \ldots \wedge R_n(u_n))\}$

- $x_1,\ldots,x_k$ sont les variables apparaissant dans le corps et pas dans la tête
- $\wedge$ est la conjonction (“et”)
- $\exists$ est la quantification existentielle (“il existe”)

exemple

la requête 4 exprimée dans le calcul conjonctif :

$$\{t | \exists r, a, \text{ réalisateurs}(r, \text{"américaine"}) \wedge \text{films}(t, r, a)\}$$

la syntaxe du calcul conjonctif

soit D un schéma de base de données

une formule sur D est une expression de la forme :

1. un atome $R(e_1, \ldots, e_n)$ sur D
2. $(\varphi \wedge \psi)$ où φ et ψ sont des formules sur D, ou
3. $\exists x \varphi$ où x est une liste de variables et φ une formule sur D

exemple

des formules du calcul :

$\text{films}(\text{"starwars"}, r, \text{"1977"})$

$\text{réalisateur}(\text{"lucas"}, n) \wedge \text{réalisateur}(\text{"lynch"}, n)$

$\exists y\ \text{réalisateur}(x, \text{"française"}) \wedge \text{films}(\text{"starwars"}, y, x)$

variable libre/liée

une (occurrence de) variable x dans une formule φ est *libre* si

1. φ est un atome, ou
2. $\varphi = (\psi \wedge \xi)$ où x est libre dans ψ ou ξ
3. $\varphi = \exists y \psi$ où y est distincte de x et x est libre dans ψ

une variable qui n'est pas libre est *liée*

$\mathit{free}(\varphi)$: ensemble des variables libres de φ

requête du calcul conjonctif

un requête est une expression de la forme

$$\{e_1, \ldots, e_n | \varphi\}$$

où φ est une formule du calcul, et les variables de $(e_1, \ldots, e_n)$ sont exactement $free(\varphi)$

exemple

dans

$$\{t | \exists r, a \ (\text{réalisateurs}(r, \text{"américaine"}) \wedge \text{films}(t, r, a))\}$$

t est libre

r et a sont liées

valuation

définie comme précédemment, pouvant s'écrire $\{x_1/a_1, \ldots, x_n/a_n\}$

on note $v|_V$ la restriction de v à l'ensemble V

v une valuation sur V, $x \notin V$, $c \in \mathbf{dom}$, $v \cup \{x/c\}$ est une valuation sur $V \cup \{x\}$

- identique à v sur V
- associant x avec c

satisfaction d'une formule

I une instance de base de données *satisfait* une formule φ sous une valuation v (noté $I \models \varphi[v]$) si

1. $\varphi = R(u)$ est un atome et $v(u) \in I(R)$, ou
2. $\varphi = (\psi \wedge \xi)$ et $I \models \psi[v|_{free(\psi)}]$ et $I \models \xi[v|_{free(\xi)}]$, ou
3. $\varphi = \exists x \psi$ et il existe $c \in dom$, $I \models \psi[v \cup \{x/c\}]$

exemple

soit l'instance de base de donnée I de la diapo 1

soit la formule $\varphi = \exists r,a\ (\text{réalisateurs}(r,\text{"américaine"}) \wedge \text{films}(t,r,a))$

I satisfait φ sous v si v est telle que $v(\text{t}) = \text{starwars}$

I ne satisfait pas φ sous v' telle que $v'(\text{t}) = \text{nikita}$

image

soit $q = \{e_1, \ldots, e_m | \varphi\}$ une requête conjonctive sur D et I une instance de D

l'image de I par q, notée $q(I)$, est :

$$q(I) = \{v((e_1, \ldots, e_m)) | I \models \varphi[v] \text{ et } v \text{ est une valuation sur } free(\varphi)\}$$

exemple

soit la requête $q = \{t | \exists r,a \text{ (réalisateurs}(r,\text{"américaine"}) \wedge \text{ films}(t,r,a))\}$

soit l'instance I de la diapo 1

$q(I) = \{(\text{"starwars"}),(\text{"dune"})\}$

propriétés des requêtes conjonctives

pourquoi étudie-t-on les requêtes conjonctives ?

- elles sont simples
- elles représentent une part importante des requêtes usuelles
- elles ont de bonnes propriétés

monotonie

une requête q sur D est *monotone* si pour toute instance I, J de D :

$I \subseteq J$ implique $q(I) \subseteq q(J)$

exemple

la requête q=film_américain(t) ←réalisateurs(r,"américaine"),films(t,r,a).

I et J : deux instances de base de données avec

I={films(starwars,lucas,1977),films(nikita,besson,1990),
films(locataires,ki-duk,2005),films(dune,lynch,1984),
réalisateurs(lucas,américaine),réalisateurs(lynch,américaine),
réalisateurs(ki-duk,coréenne),réalisateurs(besson,française)}

exemple

J={films(nikita,besson,1990),films(locataires,ki-duk,2005),
films(dune,lynch,1984), réalisateurs(lynch,américaine),
réalisateurs(ki-duk,coréenne),réalisateurs(besson,française)}

$J \subset I$

exemple

$q(I) = \{\text{film\_américain}(\text{“starwars”}), \text{film\_américain}(\text{“dune”})\}$

$q(J) = \{\text{film\_américain}(\text{“dune”})\}$

$q(J) \subset q(I)$

requête non monotone

exemple de requête non monotone :

soit la relation acteur de schéma acteur[nom,a_tourné_avec]

quels sont les acteurs qui n'ont tourné que avec lucas ?

$I(\text{acteur}) = \{(\text{ford},\text{lucas}),(\text{ford},\text{spielberg})\}$, $q(I) = \emptyset$

$J(\text{acteur}) = \{(\text{ford},\text{lucas})\}$, $q(J) = \{\text{ford}\}$

satisfiabilité

une requête q est *satisfiable* si il existe une instance I telle que $q(I)$ est non vide

exemple de requête non satisfiable :

est-ce qu'il existe un film qui s'appelle "starwars" et "dune" ?

propriétés des requêtes conjonctives

théorème :

les requêtes conjonctives sont monotones et satisfiables

à démontrer en TD...

propriétés des requêtes conjonctives

toute requête conjonctive q peut être écrite sous la forme

$$\{e_1, \ldots, e_m | \exists x_1, \ldots, x_p (R_1(u_1) \wedge \ldots \wedge R_n(u_n))\}$$

évaluer q sur une instance I demande juste à regarder dans $adom(q, I)$

propriétés (suite)

soient $q = \{u|\varphi\}$ et $q' = \{w|\psi\}$ une autre requête conjonctive

avec $free(q) = free(q')$

q est *équivalente* à $q'(q \equiv q')$ si

quelles que soient I et v, $I \models \varphi[v] \iff I \models \psi[v]$

exemple

$\{x|\exists y,z$ films$(y,x,z) \wedge$ réalisateur$(x,$coréenne$)\}$

et

$\{a|\exists b,c$ réalisateur$(a,$coréenne$) \wedge$ films$(b,a,c)\}$

sont 2 requêtes équivalentes

propriétés (suite)

le langage à base de règles pour décrire des requêtes conjonctives Q_1 et le calcul conjonctif Q_2 sont équivalents

ils permettent d'exprimer *exactement* les même requêtes

formellement :

$$\forall q_1 \in Q_1, \exists q_2 \in Q_2, q_1 \equiv q_2$$

$$\forall q_1 \in Q_2, \exists q_2 \in Q_1, q_1 \equiv q_2$$

ajout de l'égalité

l'égalité entre variables ou entre variables et constantes peut être utilisée

exemples :

$\text{film\_américain}(t) \leftarrow \text{réalisateurs}(r_1, n), n = \text{"américaine"}, \text{films}(t, r_2, a), r_1 = r_2.$

ajout de l'égalité

problème : quelle est la réponse à

$ans(x,y) \leftarrow R(x), y = z$ où $x,y,z \in$ **var**

on considère seulement les règles à *champs restreint* :

toute variable du corps doit être égale à

- une constante, ou
- une variable apparaissant dans un atome

ajout de l'égalité

problème : quelle est la réponse à

$ans(x) \leftarrow R(x), a = b$ où $x \in \mathbf{var}, a, b \in \mathbf{dom}$

on considère seulement les règles satisfiables

toute requête satisfiable avec égalité peut s'écrire sous une forme sans égalité

composition de requêtes

un *programme conjonctif* P sur une base de données D est une séquence de requêtes conjonctives

$S_1(u_1) \leftarrow body_1$
$S_2(u_2) \leftarrow body_2$
...
$S_n(u_n) \leftarrow body_n$

tous les S_i sont distincts, n'appartiennent pas à D

composition de requêtes

les relations pouvant apparaitre dans $body_i$ sont

- les relations de D et
- $S_1, \ldots, S_{i-1}$

tout programme conjonctif peut être réécrit sous la forme d'une seule règle

exemple

le programme

$S(x,y) \leftarrow R(x,y),Q(y).$

$T(y) \leftarrow Q(x),S(x,y).$

$U(x,y) \leftarrow T(x),R(x,y).$

peut être réécrit en

$$U(x,y) \leftarrow R(x,y),Q(z),R(z,x),Q(x).$$

clôture par composition

théorème :

la composition de requêtes conjonctives est une requête conjonctive