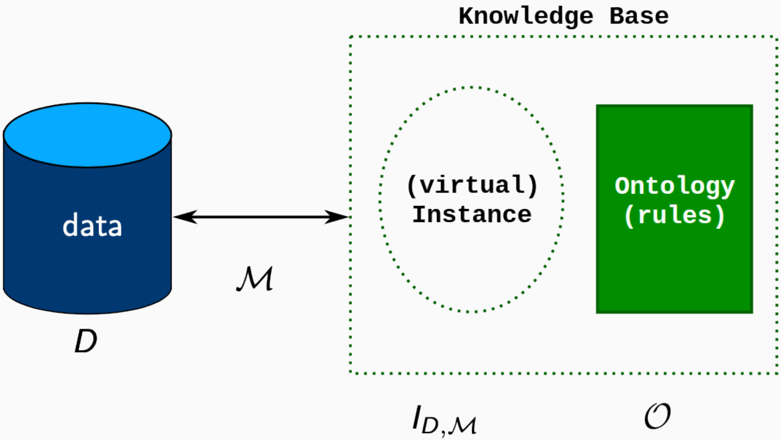
Ré-écriture d'unions de requêtes conjonctives avec des règles existentielles disjonctives

Guillaume PÉRUTION-KIHLI

21 octobre 2022

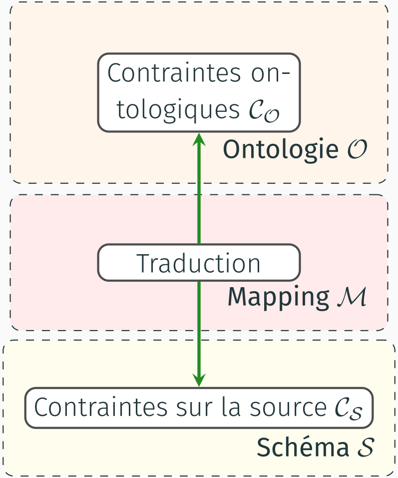
Contexte

Cadre OBDA


Knowledge Base
data
D
$\mathcal{M}$
(virtual)
Instance
Ontology
(rules)
$I_{D,\mathcal{M}}$
$\mathcal{O}$

Qualité des données : les contraintes d'intégrité

- Peuvent être exprimées sur une source ou au niveau ontologique
- Comment traduire une contrainte sur la source au niveau ontologique ? Et inversement ?
- Intérêts :
  - Corriger les données ou l'ontologie
  - Permettre à l'utilisateur de mieux comprendre les sources de données
  - Permettre d'ajouter des contraintes ontologiques aux sources de données



Classes de contraintes

Il en existe beaucoup, et notamment :

- Contraintes négatives : spécifie une contradiction
  Exemple : rien ne peut être à la fois un chat et un chien

$$\forall X.\ chat(X) \wedge chien(X) \rightarrow \bot$$

  On peut aussi le voir comme "il n'existe rien qui est à la fois un chien et un chat" :

$$\neg(\exists X.\ chat(X) \wedge chien(X))$$

- Contraintes positives : spécifie une information obligatoire
  Exemple : tout animal de compagnie doit avoir un nom

$$\forall X.\ animalDeCompagnie(X) \rightarrow \exists Y.aPourNom(X, Y)$$

Pour le moment, on s'intéresse au cas des contraintes négatives

Traduction de contraintes : préserver la satisfaction et la violation

Soient $I_{(D,\mathcal{M})}$ la matérialisation de $\mathcal{M}$ sur D, $C_{\mathcal{O}}$ une contrainte sur l'ontologie et $C_{\mathcal{S}}$ une contrainte sur la source

- **Préservation de la satisfaction :** $\mathcal{C}_{\mathcal{O}}$ préserve la satisfaction de $\mathcal{C}_{\mathcal{S}}$ si pour toute base de données D, $D \models \mathcal{C}_{\mathcal{S}}$ implique $I_{(D,\mathcal{M})} \models \mathcal{C}_{\mathcal{O}}$
- **Préservation de la violation :** $\mathcal{C}_{\mathcal{O}}$ préserve la violation de $\mathcal{C}_{\mathcal{S}}$ si pour toute base de données D, $I_{(D,\mathcal{M})} \models \mathcal{C}_{\mathcal{O}}$ implique $D \models \mathcal{C}_{\mathcal{S}}$
- Une traduction est **parfaite** si elle préserve les deux

De même pour $\mathcal{C}_{\mathcal{S}}$ par rapport à $\mathcal{C}_{\mathcal{O}}$

Des contraintes négatives aux CQ

- $Q \rightarrow \bot$: Q est une conjonction d'atomes pouvant être vue comme une requête conjonctive booléenne (contrainte satisfaite ssi pas de réponse à Q)
- **Traduire une contrainte négative se réduit donc à traduire une CQ**
- **Préservation de la satisfaction** = existence d'une traduction Q' **adéquate** (*sound*) de Q (càd $ans(Q', target) \subseteq ans(Q, source)$)
- **Préservation de la violation** = existence d'une traduction Q' **complète** de Q (càd $ans(Q, source) \subseteq ans(Q', target)$)
- **Traduction parfaite** = traduction **adéquate et complète**

Traduction de requête : ontologie vers données

Dans le sens ontologie vers données, on utilise les techniques bien connues de ré-écriture de requête, et on peut toujours avoir une traduction parfaite

Traduction de requête : ontologie vers données

Dans le sens ontologie vers données, on utilise les techniques bien connues de ré-écriture de requête, et on peut toujours avoir une traduction parfaite

$$Q_O = Chien(X) \wedge Chat(X)$$

$$animal(X, chat) \rightarrow Chat(X)$$
$$animal(X, chien) \rightarrow Chien(X)$$
$$animal(X, chaton) \rightarrow Chat(X)$$

$$\mathcal{Q}_S = (animal(X, chat) \wedge animal(X, chien)) \vee (animal(X, chaton) \wedge animal(X, chien))$$

Traduction de requête : données vers ontologie

En sens contraire, la traduction est plus difficile

Traduction de requête : données vers ontologie

En sens contraire, la traduction est plus difficile

$Q_S = animal(X, chat) \wedge animal(X, chien)$
$animal(X, chat) \rightarrow Chat(X)$
$animal(X, chien) \rightarrow Chien(X)$
$animal(X, chaton) \rightarrow Chat(X)$

animal	
Félix	chat
Leo	chien
Leo	chaton

- Il n'existe pas de traduction parfaite ici
- La requête $Q_O^C = Chien(X) \wedge Chat(X)$ a plus de réponses que Q_S, la traduction est complète mais pas adéquate
- La seule traduction adéquate est $Q_O^A = \perp$
- Il n'existe pas toujours de traduction maximalement adéquate finie

Traduction s-to-o maximalement adéquate

- Prolongement des travaux de [Cima & al.]
  - Requêtes CQJFE et mappings Pure GAV : il existe toujours une traduction maximalement adéquate finie + algorithme force brute
- Extension aux UCQ (unions de requêtes conjonctives) / mappings GLAV
  - Nouvelle technique de traduction inverse utilisant une notion de mapping inverse :
    - calculer $\mathcal{M}'$ l'inverse de $\mathcal{M}$ permettant de récupérer le maximum d'information de la base de données de départ
    - Puis calcul d'une o-to-s traduction parfaite de $\mathcal{Q}_\mathcal{S}$ avec $\mathcal{M}'$

$$I_{(D,\mathcal{M})} \models Q_G \Rightarrow D \models Q_\varphi$$

$\mathrm{rew}(Q_\varphi, \mathcal{M}')$

Mapping de récupération maximum

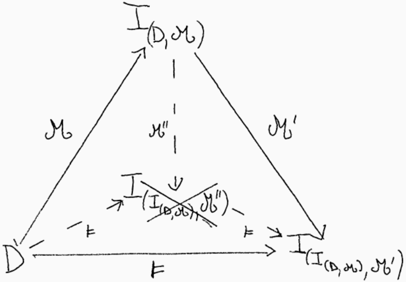
- Mapping défini dans les travaux de thèse de [Pérez & al.]

Mapping de récupération maximum

- Mapping défini dans les travaux de thèse de [Pérez & al.]
- Un mapping $\mathcal{M}'$ est un mapping de récupération de $\mathcal{M}$ si, toute requête $Q_{\mathcal{S}}$ et pour toute base de données D, $I_{(D,\mathcal{M})}, \mathcal{M}' \vDash \mathcal{Q}_{\mathcal{S}} \Rightarrow D \vDash \mathcal{Q}_{\mathcal{S}}$

Mapping de récupération maximum

- Mapping défini dans les travaux de thèse de [Pérez & al.]
- Un mapping $\mathcal{M}'$ est un mapping de récupération de $\mathcal{M}$ si, toute requête $Q_{\mathcal{S}}$ et pour toute base de données D, $I_{(D,\mathcal{M})}, \mathcal{M}' \vDash \mathcal{Q}_{\mathcal{S}} \Rightarrow D \vDash \mathcal{Q}_{\mathcal{S}}$
- $\mathcal{M}'$ est dit maximum s'il n'existe pas d'autre mapping de récupération conservant plus de réponses aux requêtes



Mapping de récupération maximum

- Mapping défini dans les travaux de thèse de [Pérez & al.]
- Un mapping $\mathcal{M}'$ est un mapping de récupération de $\mathcal{M}$ si, toute requête $Q_{\mathcal{S}}$ et pour toute base de données D, $I_{(D,\mathcal{M})}, \mathcal{M}' \models \mathcal{Q}_{\mathcal{S}} \Rightarrow D \models \mathcal{Q}_{\mathcal{S}}$
- $\mathcal{M}'$ est dit maximum s'il n'existe pas d'autre mapping de récupération conservant plus de réponses aux requêtes
- **Problème** : $\mathcal{M}'$ est un mapping pouvant contenir des règles disjonctives !

Mapping de récupération maximum

- Mapping défini dans les travaux de thèse de [Pérez & al.]
- Un mapping $\mathcal{M}'$ est un mapping de récupération de $\mathcal{M}$ si, toute requête $Q_{\mathcal{S}}$ et pour toute base de données D, $I_{(D,\mathcal{M})}, \mathcal{M}' \vDash \mathcal{Q}_{\mathcal{S}} \Rightarrow D \vDash \mathcal{Q}_{\mathcal{S}}$
- $\mathcal{M}'$ est dit maximum s'il n'existe pas d'autre mapping de récupération conservant plus de réponses aux requêtes
- **Problème** : $\mathcal{M}'$ est un mapping pouvant contenir des règles disjonctives !

$\mathcal{M}$	$\mathcal{M}'$
$animal(X, lezard) \rightarrow Chien(X)$ $animal(X, chat) \rightarrow Chat(X)$ $animal(X, chaton) \rightarrow Chat(X)$	$Chien(X) \rightarrow animal(X, chien)$ $Chat(X) \rightarrow animal(X, chat) \vee animal(X, chaton)$

Retour à la s-to-o traduction maximalement adéquate

Résultat en cours :

Soit $\mathcal{M}'$ le mapping de récupération maximum de $\mathcal{M}$, soit une requête $\mathcal{Q}_\mathcal{S}$, soit $\mathcal{Q}_\mathcal{O}$, une disjonction de CQ (éventuellement infinie). Alors :

$\mathcal{Q}_\mathcal{O}$ est une s-to-o traduction maximalement adéquate de $\mathcal{Q}_\mathcal{S}$ à travers $\mathcal{M}$ i.e. :

- pour toute base de données D, si $D, \mathcal{M} \models \mathcal{Q}_\mathcal{O}$ alors $D \models \mathcal{Q}_\mathcal{S}$
- et quelle que soit $\mathcal{Q}'_\mathcal{O}$ adéquate, $\mathcal{Q}'_\mathcal{O} \sqsubseteq_\mathcal{M} \mathcal{Q}_\mathcal{O}$

ssi

$\mathcal{Q}_\mathcal{O}$ est une o-to-s traduction parfaite de $\mathcal{Q}_\mathcal{S}$ avec $\mathcal{M}'$, i.e. :

- pour toute instance I, $I, \mathcal{M}' \models \mathcal{Q}_\mathcal{S}$ ssi $I \models \mathcal{Q}_\mathcal{O}$

Rappels sur l'unification

Rappels : unification

$$\overbrace{q(x,y,z)}^{B} \rightarrow \overbrace{p(x,y,z)}^{H} \qquad \overbrace{\underbrace{p(u,u,a)}_{Q'} \wedge t(a)}^{Q}$$

$$\begin{tikzcd} p(x,y,z) \arrow[dr, "{\sigma_1 = \{y \mapsto x,\ z \mapsto a\}}"'] & & p(u,u,a) \arrow[dl, "{\pi_Q = \{u \mapsto x\}}"] \\ & p(x,x,a) & \end{tikzcd}$$

$$u = \{y \mapsto x, u \mapsto x, z \mapsto a\}$$

$$u(p(x,y,z)) = u(p(u,u,a))$$

$$Q'' = u(B) \wedge u(Q \setminus Q') = u(q(x,y,z)) \wedge u(t(a))$$
$$= q(x,x,a) \wedge t(a)$$

Rappels : unification par pièce

$$F, \mathcal{R} \models Q \Leftrightarrow chase(F, \mathcal{R}) \models Q \Leftrightarrow F \models rewriting(Q, \mathcal{R})$$

Unification par atome incorrecte avec les règles existentielles

Exemple erroné :

$R = t(X) \rightarrow \exists Y. p(X, Y)$

$Q = p(U, V) \wedge p(W, V) \wedge r(U, W)$

Rappels : unification par pièce

$$F, \mathcal{R} \models Q \Leftrightarrow chase(F, \mathcal{R}) \models Q \Leftrightarrow F \models rewriting(Q, \mathcal{R})$$

Unification par atome incorrecte avec les règles existentielles

Exemple erroné :

$R = t(X) \rightarrow \exists Y.p(X, Y)$

$Q = p(U, V) \wedge p(W, V) \wedge r(U, W)$

$Q' = t(U) \wedge p(W, V) \wedge r(U, W)$

Rappels : unification par pièce

$$F, \mathcal{R} \models Q \Leftrightarrow chase(F, \mathcal{R}) \models Q \Leftrightarrow F \models rewriting(Q, \mathcal{R})$$

Unification par atome incorrecte avec les règles existentielles

Exemple erroné :

$R = t(X) \rightarrow \exists Y.p(X, Y)$

$Q = p(U, V) \wedge p(W, V) \wedge r(U, W)$

$Q' = t(U) \wedge p(W, V) \wedge r(U, W)$

$F = t(a) \wedge p(c, d) \wedge r(b, c)$

Rappels : unification par pièce

$$F, \mathcal{R} \models Q \Leftrightarrow chase(F, \mathcal{R}) \models Q \Leftrightarrow F \models rewriting(Q, \mathcal{R})$$

Unification par atome incorrecte avec les règles existentielles

Exemple erroné :

$R = t(X) \rightarrow \exists Y.p(X, Y)$

$Q = p(U, V) \wedge p(W, V) \wedge r(U, W)$

$Q' = t(U) \wedge p(W, V) \wedge r(U, W)$

$F = t(a) \wedge p(c, d) \wedge r(b, c)$

$chase(F, \{R\}) = t(a) \wedge p(c, d) \wedge r(b, c) \wedge p(a, Y_0)$

Rappels : unification par pièce

$$F, \mathcal{R} \models Q \Leftrightarrow chase(F, \mathcal{R}) \models Q \Leftrightarrow F \models rewriting(Q, \mathcal{R})$$

Unification par atome incorrecte avec les règles existentielles

Exemple erroné :

$R = t(X) \rightarrow \exists Y.p(X, Y)$

$Q = p(U, V) \wedge p(W, V) \wedge r(U, W)$

$Q' = t(U) \wedge p(W, V) \wedge r(U, W)$

$F = t(a) \wedge p(c, d) \wedge r(b, c)$

$chase(F, \{R\}) = t(a) \wedge p(c, d) \wedge r(b, c) \wedge p(a, Y_0)$

$F \models Q'$ mais $chase(F, \{R\}) \not\models Q$

Rappels : unification par pièce

$$\forall F, F, \mathcal{R} \models Q \Leftrightarrow chase(F, \mathcal{R}) \models Q \Leftrightarrow F \models rewriting(Q, \mathcal{R})$$

Solution : unification par pièce

Exemple correct :

$R = t(X) \rightarrow \exists Y.p(X, Y)$

$Q = p(U, V) \wedge p(W, V) \wedge r(U, W)$

Rappels : unification par pièce

$$\forall F, F, \mathcal{R} \vDash Q \Leftrightarrow chase(F, \mathcal{R}) \vDash Q \Leftrightarrow F \vDash rewriting(Q, \mathcal{R})$$

Solution : unification par pièce

Exemple correct :

$R = t(X) \rightarrow \exists Y.p(X, Y)$

$Q = p(U, V) \wedge p(W, V) \wedge r(U, W)$

$u = \{U \mapsto X, V \mapsto Y\}$: V variable séparatrice unifiée avec une variable existentielle : non !

Une variable est séparatrice si elle à la fois présente dans la partie unifiée et non-unifiée de la requête

Rappels : unification par pièce

$$\forall F, F, \mathcal{R} \models Q \Leftrightarrow chase(F, \mathcal{R}) \models Q \Leftrightarrow F \models rewriting(Q, \mathcal{R})$$

Solution : unification par pièce

Exemple correct :

$R = t(X) \rightarrow \exists Y.p(X, Y)$

$Q = p(U, V) \wedge p(W, V) \wedge r(U, W)$

$u = \{U \mapsto X, W \mapsto X, V \mapsto Y\}$: V n'est plus séparatrice

Rappels : unification par pièce

$$\forall F, F, \mathcal{R} \models Q \Leftrightarrow chase(F, \mathcal{R}) \models Q \Leftrightarrow F \models rewriting(Q, \mathcal{R})$$

Solution : unification par pièce

Exemple correct :

$R = t(X) \rightarrow \exists Y.p(X, Y)$

$Q = p(U, V) \wedge p(W, V) \wedge r(U, W)$

$u = \{U \mapsto X, W \mapsto X, V \mapsto Y\}$: V n'est plus séparatrice

$Q' = t(X) \wedge r(X, X)$

Rappels : unification par pièce

$$\forall F, F, \mathcal{R} \vDash Q \Leftrightarrow chase(F, \mathcal{R}) \vDash Q \Leftrightarrow F \vDash rewriting(Q, \mathcal{R})$$

Solution : unification par pièce

Exemple correct :

$R = t(X) \rightarrow \exists Y.p(X, Y)$

$Q = p(U, V) \wedge p(W, V) \wedge r(U, W)$

$Q' = t(X) \wedge r(X, X)$

$F = t(a), r(a, a)$

Rappels : unification par pièce

$$\forall F, F, \mathcal{R} \models Q \Leftrightarrow chase(F, \mathcal{R}) \models Q \Leftrightarrow F \models rewriting(Q, \mathcal{R})$$

Solution : unification par pièce

Exemple correct :

$R = t(X) \rightarrow \exists Y.p(X, Y)$

$Q = p(U, V) \wedge p(W, V) \wedge r(U, W)$

$Q' = t(X) \wedge r(X, X)$

$F = t(a), r(a, a)$

$chase(F, \{R\}) = t(a), r(a, a), p(a, Y_0)$

Rappels : unification par pièce

$$\forall F, F, \mathcal{R} \vDash Q \Leftrightarrow chase(F, \mathcal{R}) \vDash Q \Leftrightarrow F \vDash rewriting(Q, \mathcal{R})$$

Solution : unification par pièce

Exemple correct :

$R = t(X) \rightarrow \exists Y.p(X, Y)$

$Q = p(U, V) \wedge p(W, V) \wedge r(U, W)$

$Q' = t(X) \wedge r(X, X)$

$F = t(a), r(a, a)$

$chase(F, \{R\}) = t(a), r(a, a), p(a, Y_0)$

$F \vDash Q'$ et $chase(F, \{R\}) \vDash Q$

Rappels sur le chase disjonctif

Rappels : chase disjonctif

$$R = \forall \bar{x}, \bar{y}.(B[\bar{x}, \bar{y}] \rightarrow \bigvee_{i=1}^{n} \exists \bar{z}_i.H_i[\bar{x}, \bar{z}_i])$$

Déclencheur sur $F \in \mathcal{F}$: (R, π) tq $\pi(B[\bar{x}, \bar{y}]) \subseteq F$

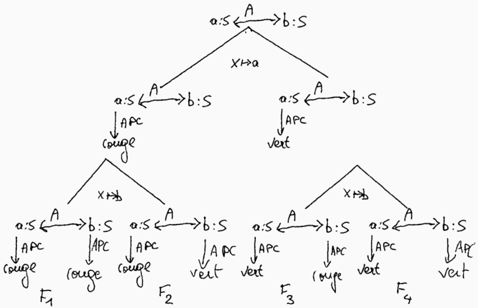
Application d'un déclencheur : $t = (R, \pi)$ sur F : $\alpha_\vee(F, R, \pi) = \{F \cup \pi^{safe_i}(H_i) \mid 1 \leqslant i \leqslant n\}$

Arbre de dérivation : arbre enraciné étiqueté (V, E, λ) avec sommets V, arêtes E et fonction d'étiquetage λ associant à un sommet une base de faits, et tel que :

- la racine v est tq $\lambda(v) = F$;
- l'ensemble des enfants d'un nœud interne labelisé F' contient exactement les nœuds étiquetés avec les faits $\alpha_\vee(F', R, \pi)$ avec (R, π) un déclencheur sur F'.

Exemple de chase disjonctif

$F = sommet(a) \wedge arc(a, b) \wedge arc(b, a) \wedge sommet(b)$
$sommet(X) \rightarrow aPourCouleur(X, rouge) \vee aPourCouleur(X, vert)$



Résultat : $\mathcal{F} = F_1 \vee \cdots \vee F_4$

Réponses aux requêtes

- $\mathcal{F} = F_1 \vee \cdots \vee F_k \models Q_1 \vee \cdots \vee Q_n$
  $\Leftrightarrow F_1 \models Q_1 \vee \cdots \vee Q_n$ et ... et $F_k \models Q_1 \vee \cdots \vee Q_n$
  $\Leftrightarrow$ ($F_1 \models Q_1$ ou ... ou $F_1 \models Q_n$) et ... et ($F_k \models Q_1$ ou ... ou $F_k \models Q_n$)

Réponses aux requêtes

- $\mathcal{F} = F_1 \vee \cdots \vee F_k \vDash Q_1 \vee \cdots \vee Q_n$
  $\Leftrightarrow F_1 \vDash Q_1 \vee \cdots \vee Q_n$ **et** $\ldots$ **et** $F_k \vDash Q_1 \vee \cdots \vee Q_n$
  $\Leftrightarrow$ **(**$F_1 \vDash Q_1$ **ou** $\ldots$ **ou** $F_1 \vDash Q_n$**) et** $\ldots$ **et (**$F_k \vDash Q_1$ **ou** $\ldots$ **ou** $F_k \vDash Q_n$**)**
- $F \vDash Q \Leftrightarrow Q \geqslant F$, où $Q \geqslant F$ signifie que Q s'envoie dans F par homomorphisme

Réponses aux requêtes

- $\mathcal{F} = F_1 \vee \cdots \vee F_k \vDash Q_1 \vee \cdots \vee Q_n$
  $\Leftrightarrow F_1 \vDash Q_1 \vee \cdots \vee Q_n$ **et** $\ldots$ **et** $F_k \vDash Q_1 \vee \cdots \vee Q_n$
  $\Leftrightarrow (F_1 \vDash Q_1$ **ou** $\ldots$ **ou** $F_1 \vDash Q_n)$ **et** $\ldots$ **et** $(F_k \vDash Q_1$ **ou** $\ldots$ **ou** $F_k \vDash Q_n)$
- $F \vDash Q \Leftrightarrow Q \geqslant F$, où $Q \geqslant F$ signifie que Q s'envoie dans F par homomorphisme
- $F_1 \vee \cdots \vee F_k \vDash Q_1 \vee \cdots \vee Q_n$
  $\Leftrightarrow (Q_1 \geqslant F_1$ **ou** $\ldots$ **ou** $Q_n \geqslant F_1)$ **et** $\ldots$ **et** $(Q_1 \geqslant F_k$ **ou** $\ldots$ **ou** $Q_n \geqslant F_k)$

Réponses aux requêtes

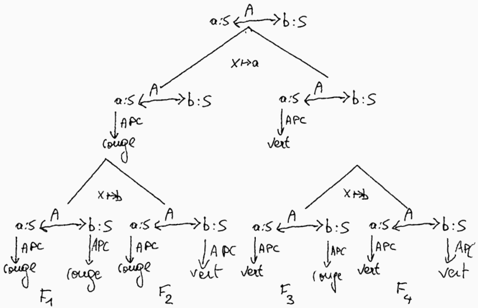
- $\mathcal{F} = F_1 \vee \cdots \vee F_k \vDash Q_1 \vee \cdots \vee Q_n$
  $\Leftrightarrow F_1 \vDash Q_1 \vee \cdots \vee Q_n$ **et** $\ldots$ **et** $F_k \vDash Q_1 \vee \cdots \vee Q_n$
  $\Leftrightarrow (F_1 \vDash Q_1$ **ou** $\ldots$ **ou** $F_1 \vDash Q_n)$ **et** $\ldots$ **et** $(F_k \vDash Q_1$ **ou** $\ldots$ **ou** $F_k \vDash Q_n)$
- $F \vDash Q \Leftrightarrow Q \geqslant F$, où $Q \geqslant F$ signifie que Q s'envoie dans F par homomorphisme
- $F_1 \vee \cdots \vee F_k \vDash Q_1 \vee \cdots \vee Q_n$
  $\Leftrightarrow (Q_1 \geqslant F_1$ **ou** $\ldots$ **ou** $Q_n \geqslant F_1)$ **et** $\ldots$ **et** $(Q_1 \geqslant F_k$ **ou** $\ldots$ **ou** $Q_n \geqslant F_k)$

Autrement dit : une UCQ $\mathcal{Q}$ est satisfaite sur un ensemble de mondes $\mathcal{F}$ ssi pour tous les mondes $F_i \in \mathcal{F}$, il existe une CQ $Q_j \in \mathcal{Q}$ tq $Q_j \geqslant F_i$.

On le note $\mathcal{Q} \geqslant_{\vee} \mathcal{F}$.

Exemple de chase disjonctif

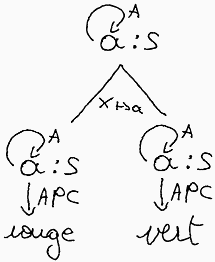
$F = sommet(a) \wedge arc(a, b) \wedge arc(b, a) \wedge sommet(b)$
$sommet(X) \rightarrow aPourCouleur(X, rouge) \vee aPourCouleur(X, vert)$



$Q = arc(U, V) \wedge aPourCouleur(U, T) \wedge aPourCouleur(V, T)$: est-ce que le graphe n'est pas 2-colorable ?

Exemple de chase disjonctif

$G = sommet(a) \wedge arc(a, a)$
$sommet(X) \rightarrow aPourCouleur(X, rouge) \vee aPourCouleur(X, vert)$



$Q = arc(U, V) \wedge aPourCouleur(U, T) \wedge aPourCouleur(V, T)$: est-ce que le graphe n'est pas 2-colorable ?

Ré-écrire avec des règles existentielles disjonctives

Comment ré-écrire en présence de règles existentielles disjonctives ?

$R = p(X) \rightarrow q(X) \vee r(X)$
$\mathcal{Q} = q(a) \vee r(a)$

Comment fait-on pour ré-écrire $\mathcal{Q}$ avec R ?
Combien il y a-t-il de ré-écritures possibles ?

Comment ré-écrire en présence de règles existentielles disjonctives ?

$R = p(X) \rightarrow q(X) \vee r(X)$
$\mathcal{Q} = q(a) \vee r(a)$

Correction :

$Q_1 = q(a), Q_2 = r(a)$

Comment ré-écrire en présence de règles existentielles disjonctives ?

$$R = p(X) \rightarrow q(X) \vee r(X)$$

$\mathcal{Q} = q(a) \vee r(a)$

Correction :

$Q_1 = q(a), Q_2 = r(a)$

$u_1 = \{X \mapsto a\}, u_2 = \{X \mapsto a\}$

Comment ré-écrire en présence de règles existentielles disjonctives ?

$$R = p(X) \rightarrow q(X) \vee r(X)$$

$\mathcal{Q} = q(a) \vee r(a)$

Correction :

$Q_1 = q(a), Q_2 = r(a)$

$u_1 = \{X \mapsto a\}, u_2 = \{X \mapsto a\}$

On les agrège : $u = \{X \mapsto a\}$

Comment ré-écrire en présence de règles existentielles disjonctives ?

$$R = p(X) \rightarrow q(X) \vee r(X)$$

$\mathcal{Q} = q(a) \vee r(a)$

Correction :

$Q_1 = q(a), Q_2 = r(a)$

$u_1 = \{X \mapsto a\}, u_2 = \{X \mapsto a\}$

On les agrège : $u = \{X \mapsto a\}$

$Q' = u(p(X)) = p(a)$

Comment ré-écrire en présence de règles existentielles disjonctives ?

$R = p(X) \rightarrow q(X) \vee r(X)$
$\mathcal{Q} = q(a) \vee r(b)$

Comment fait-on pour ré-écrire Q avec R ?
Combien il y a-t-il de ré-écritures possibles ?

Comment ré-écrire en présence de règles existentielles disjonctives ?

$R = p(X) \rightarrow q(X) \vee r(X)$
$\mathcal{Q} = q(a) \vee r(b)$

Correction :

$Q_1 = q(a), Q_2 = r(b)$

Comment ré-écrire en présence de règles existentielles disjonctives ?

$$R = p(X) \rightarrow q(X) \vee r(X)$$

$$\mathcal{Q} = q(a) \vee r(b)$$

Correction :

$Q_1 = q(a), Q_2 = r(b)$

$u_1 = \{X \mapsto a\}, u_2 = \{X \mapsto b\}$

Comment ré-écrire en présence de règles existentielles disjonctives ?

$$R = p(X) \rightarrow q(X) \vee r(X)$$

$$\mathcal{Q} = q(a) \vee r(b)$$

Correction :

$Q_1 = q(a), Q_2 = r(b)$

$u_1 = \{X \mapsto a\}, u_2 = \{X \mapsto b\}$

u_1 et u_2 sont incompatibles entre eux (X est unifié avec deux constantes différentes)

Comment ré-écrire en présence de règles existentielles disjonctives ?

$$R = p(X) \rightarrow q(X) \vee r(X)$$

$\mathcal{Q} = q(a) \vee r(b)$

Correction :

$Q_1 = q(a), Q_2 = r(b)$

$u_1 = \{X \mapsto a\}, u_2 = \{X \mapsto b\}$

u_1 et u_2 sont incompatibles entre eux (X est unifié avec deux constantes différentes)

Pas de ré-écriture possible ici

Comment ré-écrire en présence de règles existentielles disjonctives ?

$R = p(X) \rightarrow q(X) \vee r(X)$
$\mathcal{Q} = q(a) \vee (r(U) \wedge t(U))$

Comment fait-on pour ré-écrire Q avec R ?
Combien il y a-t-il de ré-écritures possibles ?

Comment ré-écrire en présence de règles existentielles disjonctives ?

$R = p(X) \rightarrow q(X) \vee r(X)$
$\mathcal{Q} = q(a) \vee (r(U) \wedge t(U))$

Correction :

$Q_1 = q(a), Q_2 = r(U) \wedge t(U)$

Comment ré-écrire en présence de règles existentielles disjonctives ?

$$R = p(X) \rightarrow q(X) \vee r(X)$$

$\mathcal{Q} = q(a) \vee (r(U) \wedge t(U))$

Correction :

$Q_1 = q(a), Q_2 = r(U) \wedge t(U)$

$u_1 = \{X \mapsto a\}, u_2 = \{X \mapsto U\}$

Comment ré-écrire en présence de règles existentielles disjonctives ?

$$R = p(X) \rightarrow q(X) \vee r(X)$$

$\mathcal{Q} = q(a) \vee (r(U) \wedge t(U))$

Correction :

$Q_1 = q(a), Q_2 = r(U) \wedge t(U)$

$u_1 = \{X \mapsto a\}, u_2 = \{X \mapsto U\}$

On les agrège : $u = \{X \mapsto a, U \mapsto a\}$

Comment ré-écrire en présence de règles existentielles disjonctives ?

$$R = p(X) \rightarrow q(X) \vee r(X)$$

$\mathcal{Q} = q(a) \vee (r(U) \wedge t(U))$

Correction :

$Q_1 = q(a), Q_2 = r(U) \wedge t(U)$

$u_1 = \{X \mapsto a\}, u_2 = \{X \mapsto U\}$

On les agrège : $u = \{X \mapsto a, U \mapsto a\}$

$Q' = u(p(X)) \wedge u(t(U)) = p(a) \wedge t(a)$

Comment ré-écrire en présence de règles disjonctives ?

$R = p(X, Y) \rightarrow t(X) \vee q(Y)$
$\mathcal{Q} = t(U) \wedge q(U)$

Comment fait-on pour ré-écrire Q avec R ?
Combien il y a-t-il de ré-écritures possibles ?

Comment ré-écrire en présence de règles disjonctives ?

$R = p(X, Y) \rightarrow t(X) \vee q(Y)$
$\mathcal{Q} = t(U) \wedge q(U)$

Correction :

Soient $Q_1 = t(U_1) \wedge q(U_1)$ et $Q_2 = t(U_2) \wedge q(U_2)$ des copies sûres

Comment ré-écrire en présence de règles disjonctives ?

$$R = p(X, Y) \rightarrow t(X) \vee q(Y)$$

$$\mathcal{Q} = t(U) \wedge q(U)$$

Correction :

Soient $Q_1 = t(U_1) \wedge q(U_1)$ et $Q_2 = t(U_2) \wedge q(U_2)$ des copies sûres de Q

$u_1 = \{X \mapsto U_1\}, u_2 = \{Y \mapsto U_2\}$

Comment ré-écrire en présence de règles disjonctives ?

$$R = p(X, Y) \rightarrow t(X) \vee q(Y)$$

$$\mathcal{Q} = t(U) \wedge q(U)$$

Correction :

Soient $Q_1 = t(U_1) \wedge q(U_1)$ et $Q_2 = t(U_2) \wedge q(U_2)$ des copies sûres de Q

$u_1 = \{X \mapsto U_1\}, u_2 = \{Y \mapsto U_2\}$

On les agrège : $u = \{X \mapsto U_1, Y \mapsto U_2\}$

Comment ré-écrire en présence de règles disjonctives ?

$$R = p(X, Y) \rightarrow t(X) \vee q(Y)$$

$\mathcal{Q} = t(U) \wedge q(U)$

Correction :

Soient $Q_1 = t(U_1) \wedge q(U_1)$ et $Q_2 = t(U_2) \wedge q(U_2)$ des copies sûres de Q

$u_1 = \{X \mapsto U_1\}, u_2 = \{Y \mapsto U_2\}$

On les agrège : $u = \{X \mapsto U_1, Y \mapsto U_2\}$

$Q' = u(q(U_1) \wedge p(X, Y) \wedge t(U_2)) = q(U_1) \wedge p(U_1, U_2) \wedge t(U_2)$

Comment ré-écrire en présence de règles disjonctives ?

$$R = p(X,Y) \to t(X) \vee q(Y)$$

$$\mathcal{Q} = t(U) \wedge q(U)$$

Correction :

Soient $Q_1 = t(U_1) \wedge q(U_1)$ et $Q_2 = t(U_2) \wedge q(U_2)$ des copies sûres de Q

$u_1 = \{X \mapsto U_1\}, u_2 = \{Y \mapsto U_2\}$

On les agrège : $u = \{X \mapsto U_1, Y \mapsto U_2\}$

$Q' = u(q(U_1) \wedge p(X,Y) \wedge t(U_2)) = q(U_1) \wedge p(U_1,U_2) \wedge t(U_2)$

Mais ce n'est pas fini !

Comment ré-écrire en présence de règles disjonctives ?

$R = p(X, Y) \rightarrow t(X) \vee q(Y)$
$\mathcal{Q} = (t(U) \wedge q(U)) \vee (q(U_1) \wedge p(U_1, U_2) \wedge t(U_2))$

Soient $Q_3 = t(U_3) \wedge q(U_3)$ et $Q_4 = q(U_4) \wedge p(U_4, U_5) \wedge t(U_5)$ des copies sûres

Comment ré-écrire en présence de règles disjonctives ?

$R = p(X, Y) \rightarrow t(X) \vee q(Y)$
$\mathcal{Q} = (t(U) \wedge q(U)) \vee (q(U_1) \wedge p(U_1, U_2) \wedge t(U_2))$

Soient $Q_3 = t(U_3) \wedge q(U_3)$ et $Q_4 = q(U_4) \wedge p(U_4, U_5) \wedge t(U_5)$ des copies sûres
$u_3 = \{X \mapsto U_3\}, u_4 = \{Y \mapsto U_4\}$

Comment ré-écrire en présence de règles disjonctives ?

$R = p(X, Y) \rightarrow t(X) \vee q(Y)$
$\mathcal{Q} = (t(U) \wedge q(U)) \vee (q(U_1) \wedge p(U_1, U_2) \wedge t(U_2))$

Soient $Q_3 = t(U_3) \wedge q(U_3)$ et $Q_4 = q(U_4) \wedge p(U_4, U_5) \wedge t(U_5)$ des copies sûres
$u_3 = \{X \mapsto U_3\}, u_4 = \{Y \mapsto U_4\}$
On les agrège : $u = \{X \mapsto U_3, Y \mapsto U_4\}$

Comment ré-écrire en présence de règles disjonctives ?

$R = p(X, Y) \rightarrow t(X) \vee q(Y)$
$\mathcal{Q} = (t(U) \wedge q(U)) \vee (q(U_1) \wedge p(U_1, U_2) \wedge t(U_2))$

Soient $Q_3 = t(U_3) \wedge q(U_3)$ et $Q_4 = q(U_4) \wedge p(U_4, U_5) \wedge t(U_5)$ des copies sûres
$u_3 = \{X \mapsto U_3\}, u_4 = \{Y \mapsto U_4\}$
On les agrège : $u = \{X \mapsto U_3, Y \mapsto U_4\}$
$Q_2 = u(q(U_3) \wedge p(X, Y) \wedge p(U_4, U_5) \wedge t(U_5)) = q(U_3) \wedge p(U_3, U_4) \wedge p(U_4, U_5) \wedge t(U_5)$

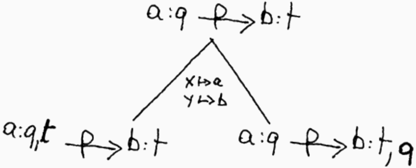
Comment ré-écrire en présence de règles disjonctives ?

$R = p(X, Y) \to t(X) \vee q(Y)$
$\mathcal{Q} = (t(U) \wedge q(U)) \vee (q(U_1) \wedge p(U_1, U_2) \wedge t(U_2))$

Soient $Q_3 = t(U_3) \wedge q(U_3)$ et $Q_4 = q(U_4) \wedge p(U_4, U_5) \wedge t(U_5)$ des copies sûres
$u_3 = \{X \mapsto U_3\}, u_4 = \{Y \mapsto U_4\}$
On les agrège : $u = \{X \mapsto U_3, Y \mapsto U_4\}$
$Q_2 = u(q(U_3) \wedge p(X, Y) \wedge p(U_4, U_5) \wedge t(U_5)) = q(U_3) \wedge p(U_3, U_4) \wedge p(U_4, U_5) \wedge t(U_5)$
Il y a un nombre infini de ré-écritures incomparables entre elles !

Les ré-écritures étaient-elles correctes ?

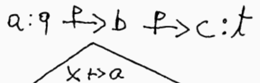
$$R = p(X, Y) \rightarrow t(X) \vee q(Y)$$
$$Q = t(U) \wedge q(U)$$
$$Q' = q(U_1) \wedge p(U_1, U_2) \wedge t(U_2)$$
$$F = q(a) \wedge p(a, b) \wedge t(b)$$



Les ré-écritures étaient-elles correctes ?

$$R = p(X, Y) \to t(X) \vee q(Y)$$
$$Q = t(U) \wedge q(U)$$

$$Q'' = q(U_1) \wedge p(U_1, U_2) \wedge p(U_2, U_3) \wedge t(U_3)$$
$$F = q(a) \wedge p(a, b) \wedge p(b, c) \wedge t(c)$$



Généralisons un peu

$$R = \forall \bar{x}, \bar{y}.(B[\bar{x}, \bar{y}] \rightarrow \bigvee_{i=1}^{n} \exists \bar{z}_i.H_i[\bar{x}, \bar{z}_i])$$

$$\mathcal{Q} = \bigvee_{j=1}^{k} Q_j$$

1. On prend n copies sûres de CQ dans $\mathcal{Q}$: $Q_1^s, \ldots, Q_n^s$
2. On calcule n unificateurs u_n entre $H_i[\bar{x}, \bar{z}_i]$ et Q_i^s pour $0 < i \leqslant n$
3. S'ils sont compatibles : on crée une substitution agrégée u
4. Pour chaque Q_i^s, soit $Q_i^{s'}$ la partie unifiée avec $H_i'[\bar{x}, \bar{z}_i] \subseteq H_i[\bar{x}, \bar{z}_i]$

   Le résultat est $u(B[\bar{x}, \bar{y}]) \wedge \left(\bigwedge_{i=1}^{n} u(Q_i^s \setminus Q_i^{s'})\right)$

Adéquation et complétude

Comment démontrer l'adéquation et la complétude ?

Idée : on a

$$F, \mathcal{R} \vDash Q \Leftrightarrow chase_{\vee}(F, \mathcal{R}) \vDash Q$$

et on va montrer

$$\forall F, chase_{\vee}(F, \mathcal{R}) \vDash Q \Leftrightarrow F \vDash rewriting_{\vee}(Q, \mathcal{R})$$

On va montrer ça pour une règle et pour une étape de chase et de ré-écriture

La propriété se montre ensuite par récurrence

Algorithme

Algorithme

Algorithme : ALGORITHME DE RÉ-ÉCRITURE GÉNÉRIQUE

Données : Un ensemble de règles disjonctives $\mathcal{R}$, une UCQ $\mathcal{Q}$

Accès : Un opérateur de ré-écriture disjonctives $\mathtt{rew}_{\vee}$, une fonction de couverture $\mathtt{cover}$

Résultat : une couverture de l'ensemble de toutes les ré-écritures de $\mathcal{Q}$

$\mathcal{Q}_F \leftarrow \mathcal{Q}$; // *Résultat*

$\mathcal{Q}_E \leftarrow \mathcal{Q}$; // *requêtes à explorer*

tant que $\mathcal{Q}_E \neq \emptyset$ **faire**

  $\mathcal{Q}_C \leftarrow \mathtt{cover}(\mathcal{Q}_F \cup \mathtt{rew}_{\vee}(\mathcal{Q}_E, \mathcal{R}))$; // *Mise à jour de la couverture*

  $\mathcal{Q}_E \leftarrow \mathcal{Q}_C \backslash \mathcal{Q}_F$; // *Sélection des requêtes non-explorées*

  $\mathcal{Q}_F \leftarrow \mathcal{Q}_C$;

retourner $\mathcal{Q}_F$

L'algorithme s'arrête ssi il existe une ré-écriture adéquate et complète finie de $\mathcal{Q}$ avec $\mathcal{R}$ (qu'il produit).

Mappings disjonctifs

Cas particulier des mappings

- Même avec une seule règle non-récursive, la ré-écriture peut ne pas terminer
- La ré-écriture peut être infinie à travers un mapping disjonctif alors qu'on a un nombre fini de requêtes incomparables entre elles sur la base de données

Exemple :

$$P(X, Y) \to s_1(Y) \vee s_2(X)$$
$$A1(X) \to s_1(X) \wedge s_2(X)$$

$$\mathcal{Q} = s_1(U) \wedge s_2(U)$$

$$\mathcal{Q}' = A_1(U) \vee \bigvee_{n \in \mathbb{N}} \left(s_1(U_0) \wedge \left(\bigwedge_{i=1}^{n} p(U_{i-1}, U_i) \right) \wedge s_2(U_n) \right)$$

$$\mathcal{Q}'' = A_1(U)$$

Conclusion

Merci pour votre attention !