

Raisonnement distribué en pair à pair pour le Web sémantique



Marie-Christine Rousset

LIG, Université de Grenoble



travail commun sur le projet **SomeWhere** avec
P.Adjiman, P.Chatalic, F. Goasdoué, G-H Nguyen, L. Simon

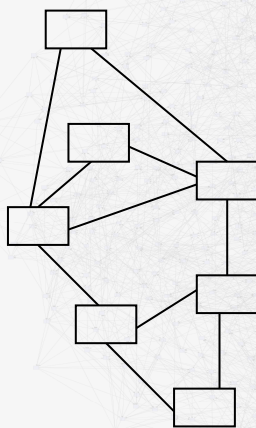


IASI/Gemo LRI/INRIA Univ.Paris Sud



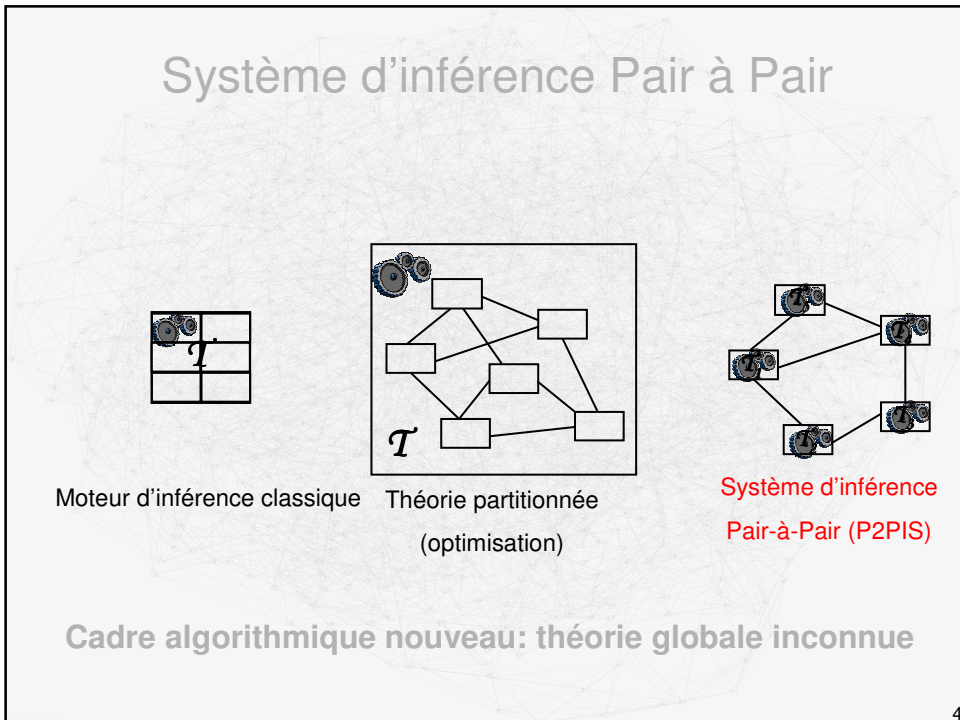
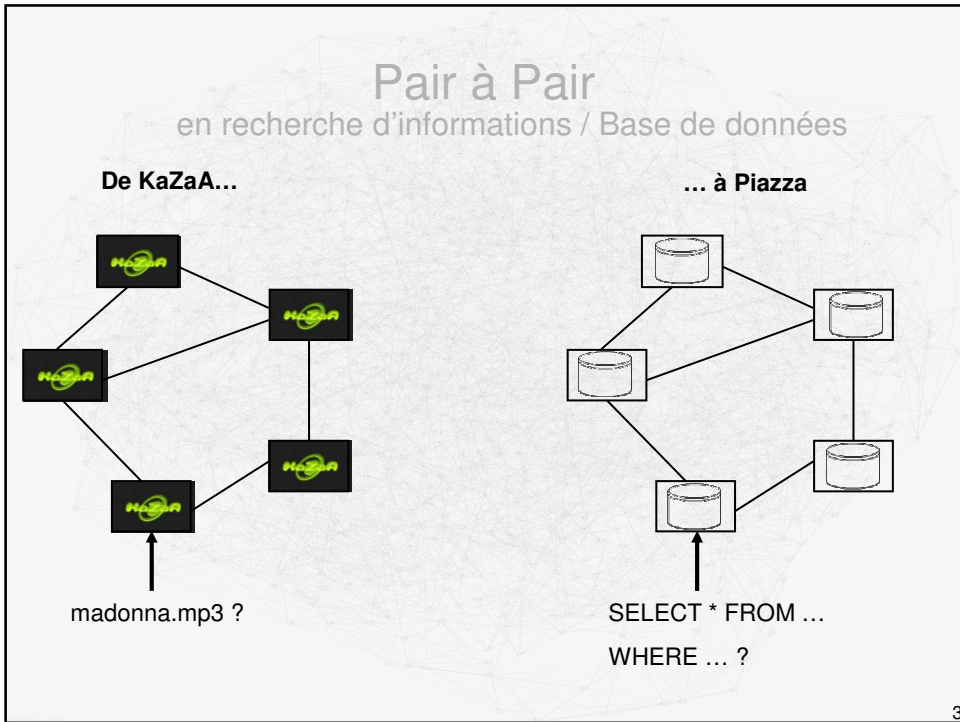
1

Systeme Pair à Pair



- Coalition de serveurs d'information indépendants (pairs)
- Chaque pair peut être soit client soit serveur
- Architecture dynamique

2



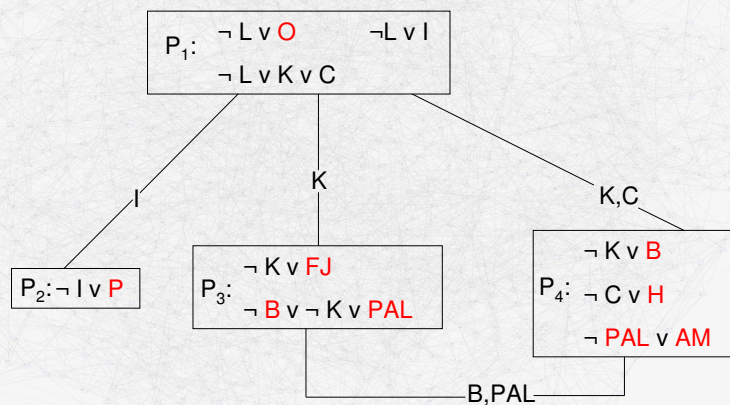
Plan de la présentation

1. Raisonnement dans les P2PIS
 - B. L'algorithme DeCA
 - C. Raisonnement en présence d'inconsistences
2. Construire des PDMS pour le web sémantique à partir de DeCA
 - A. Contexte
 - B. Le PDMS SomeOWL
 - C. Le PDMS SomeRDFS
3. Passage à l'échelle
4. Conclusion

5

Contexte et définition du problème

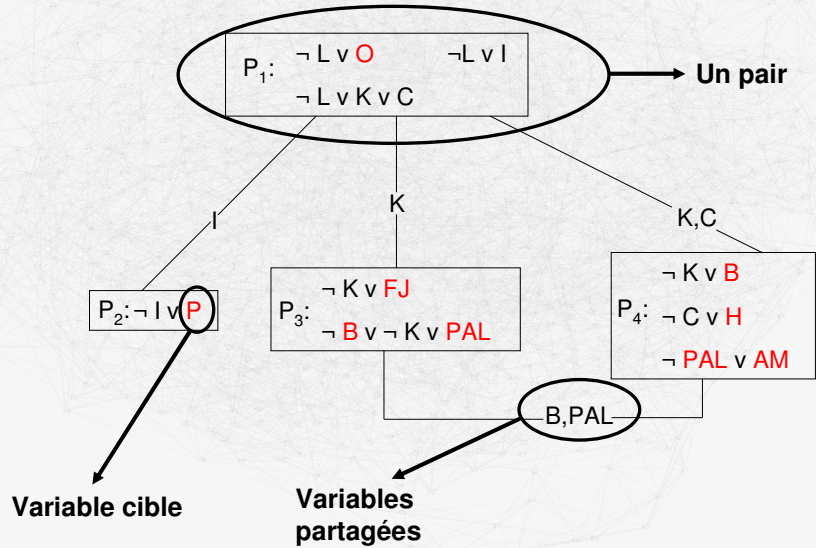
Système d'inférence pair-à-pair (P2PIS)



Théories propositionnelles *distribuées*

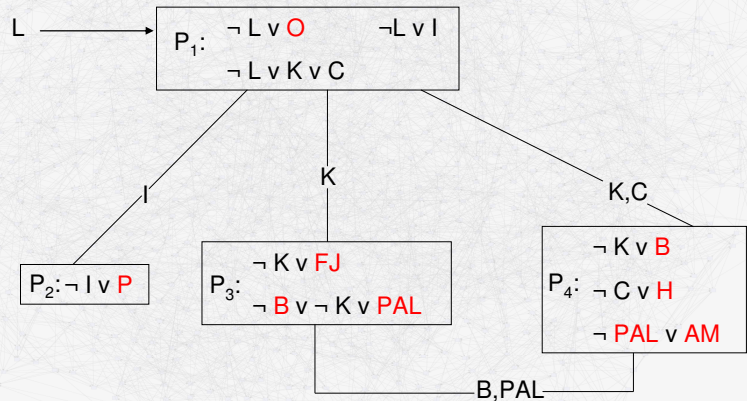
6

Contexte et définition du problème



7

Contexte et définition du problème



Trouver tous les *impliqués premiers propres* de L relativement à $\bigcup_{i \in \{1,4\}} P_i$ et contenant uniquement des **variables cibles**

8

Contexte et définition du problème

Impliqué premier propre (ppi) :

Soit P une théorie clausale et q une clause. Une clause m est dite:

- *impliqué* de q rel. à P ssi $P \cup \{q\} \models m$
- impliqué *premier* de q rel. à P ssi m est un impliqué de q rel. à P , et pour toute clause m' impliqué de q rel. à P , si $m' \models m$ alors $m' \equiv m$
- impliqué premier *propre* de q rel. à P ssi elle est impliqué premier de q rel. à P mais $P \not\models m$

Problème de calcul de conséquence :

Entrée: un pair P , une clause q (utilisant le vocabulaire de P)

Sortie: tout les ppi de q rel. à $\bigcup_{i \in 1..n} P_i$ et ne contenant que des **variables cibles**

Principale difficulté algorithmique :

Pas de connaissance globale de $\bigcup_{i \in 1..n} P_i$

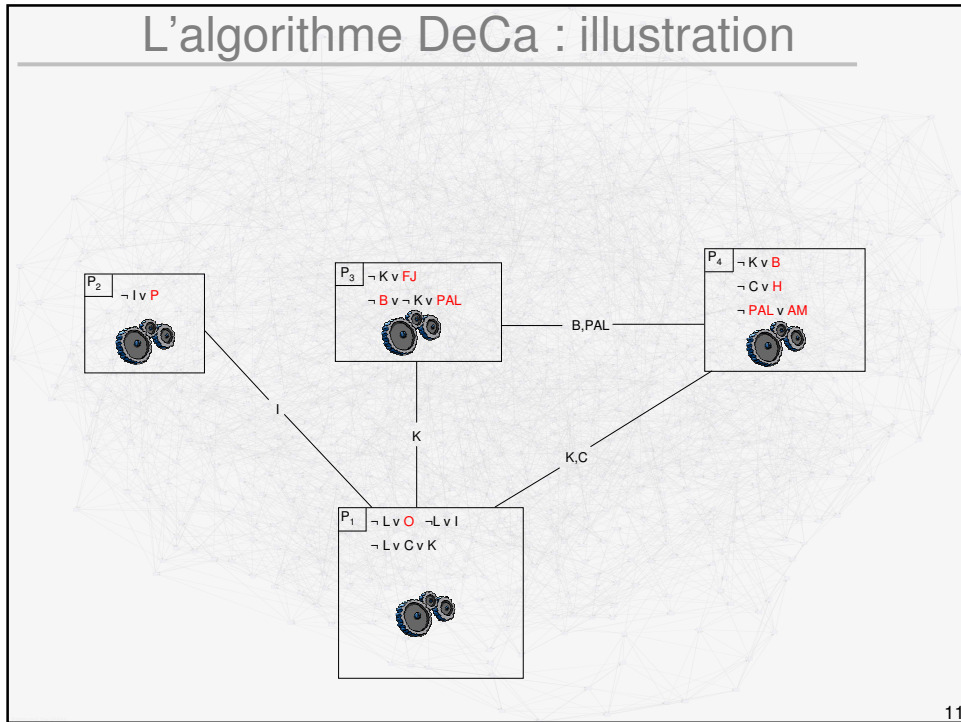
9

Plan de la présentation

1. Raisonement dans les P2PIS
 - B. L'algorithme DeCA
2. Construire des PDMS pour le web sémantique à partir de DeCA
 - A. Contexte
 - B. Le PDMS SomeOWL
 - C. Le PDMS SomeRDFS
3. Passage à l'échelle
4. Conclusion

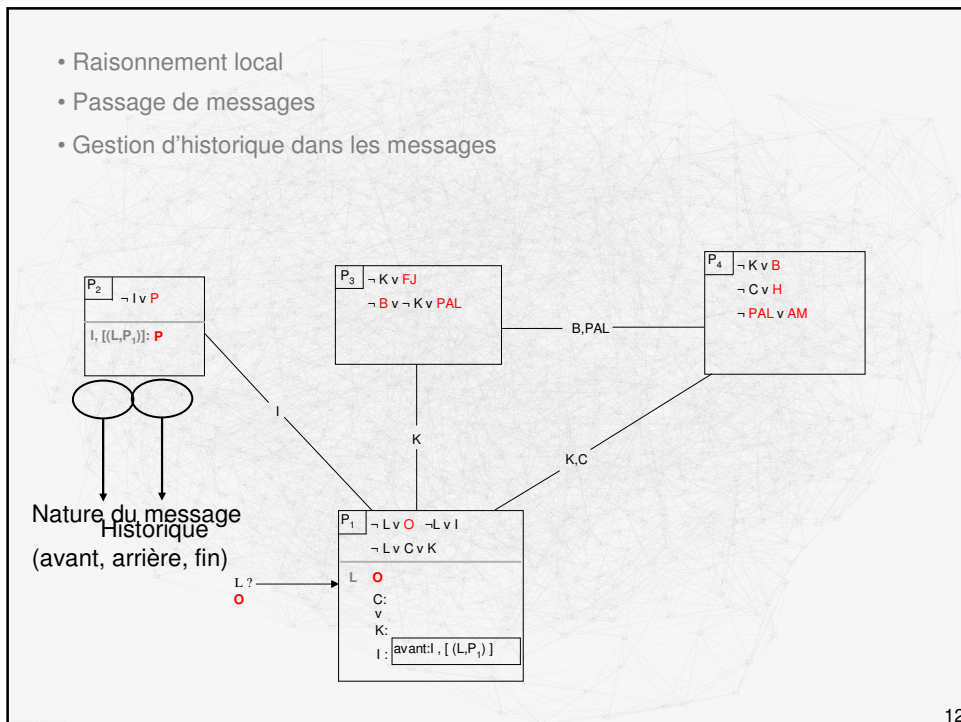
10

L'algorithme DeCa : illustration



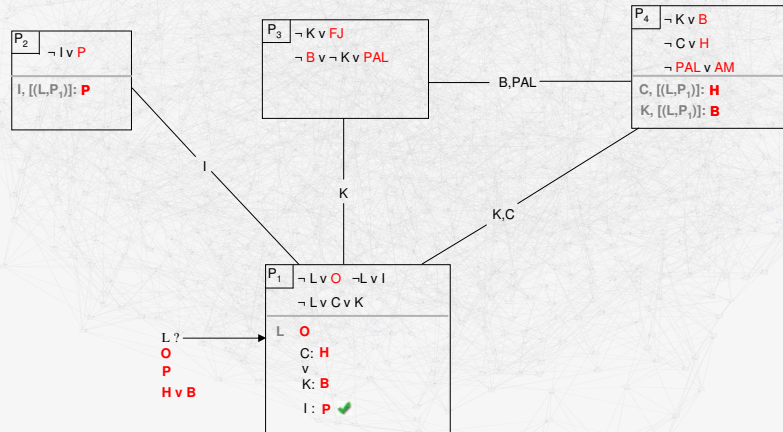
11

- Raisonnement local
- Passage de messages
- Gestion d'historique dans les messages



12

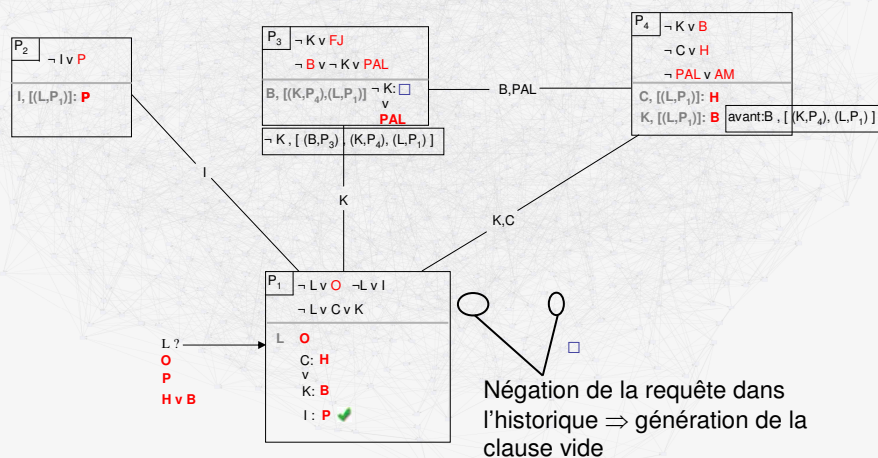
- Raisonnement local
- Passage de messages
- Gestion d'historique dans les messages
- Découpage / recombinaison



15

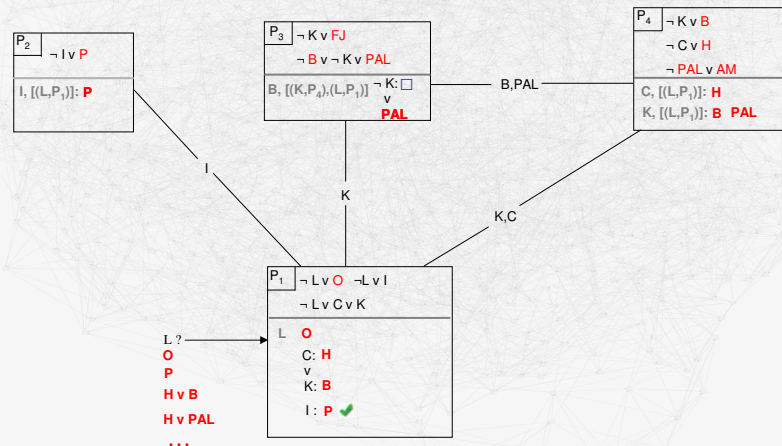
- Raisonnement local
- Passage de messages
- Gestion d'historique dans les messages
- Découpage / recombinaison

Génération de la clause vide
 Détection de cycles



16

- Raisonnement local
- Passage de messages
- Gestion d'historique dans les messages } Génération de la clause vide
- Découpage / recombinaison } Détection de cycles



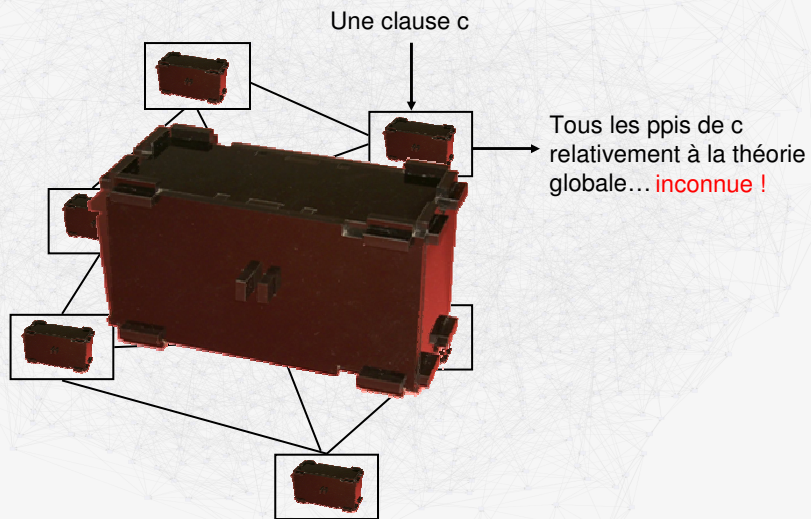
17

L'algorithme DeCa : propriétés

- **Anytime**
- **Termine et notifie sa terminaison**
- **Correct**
- **Condition suffisante de complétude:**
 tout couple de pairs ayant une variable **A** en commun est relié par un chemin dont toutes les arêtes sont étiquetées par **A**

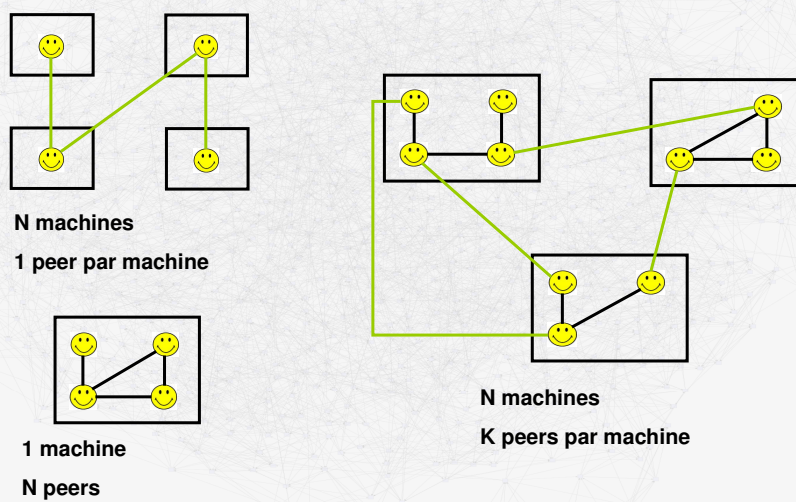
18

L'algorithme DeCa : résumé



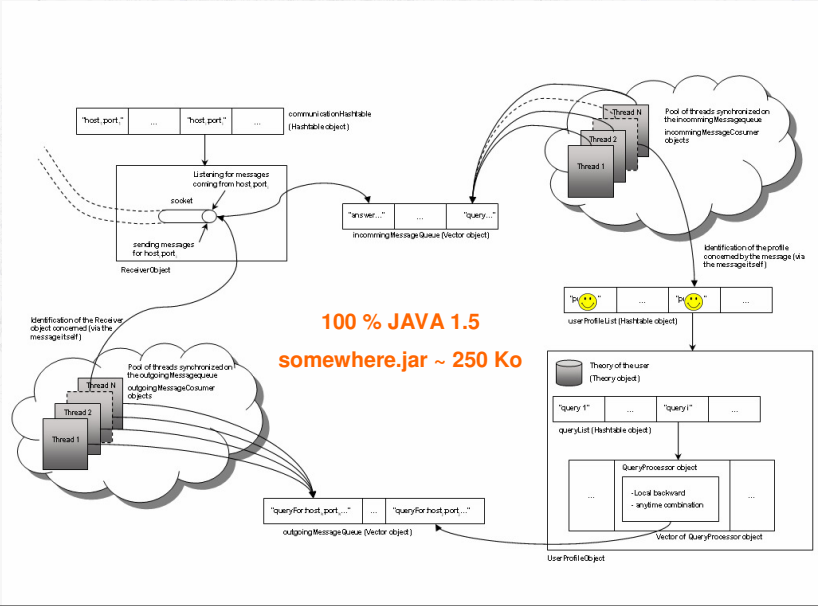
19

SomeWhere infrastructure de déploiement de DeCA



20

Zoom sur une machine



Plan de la présentation

1. Raisonement dans les P2PIS
 - A. Contexte et définition du problème
 - C. Raisonement en présence d'inconsistances
2. Construire des PDMS pour le web sémantique à partir de DeCA
 - A. Contexte
 - B. Le PDMS SomeOWL
 - C. Le PDMS SomeRDFS
3. Passage à l'échelle
4. Conclusion

1

Le problème de l'inconsistance en P2P

- la théorie globale peut être inconsistante même si chaque théorie locale est consistante
 - à cause des mappings entre pairs
 - un mapping: une formule contenant des variables partagées
- Pas d'autorité centrale et pairs autonomes
 - on « fait avec » l'inconsistance :
on raisonne de manière « bien-fondée » en présence d'inconsistances

2

Notre approche

- On calcule toutes les causes d'inconsistances et on les caractérise comme des **nogoods** stockés de manière distribuée
- On adopte une sémantique simple pour définir la notion de conséquence **bien fondée** :
 - celles qui peuvent être dérivées à partir d'un sous-ensemble consistant de la théorie globale
 - calculé au cours du raisonnement

Problème : concevoir des algorithmes décentralisés

- de calcul et stockage des nogoods
- de raisonnement bien-fondé
 - fondé sur l'exploitation des nogoods

3

Nogoods revisités en P2P

Notre hypothèse : $O = \bigcup O_i$ est consistante

- Pour chaque pair, soit $T_i = O_i \cup M_i$ sa théorie
- Les inconsistances sont causées par des mappings

Definition:

un *nogood* ng est un sous-ensemble de $M = \bigcup M_i$
tel que $ng \cup O$ est inconsistant

4

Calcul des nogoods

Propriété: si ng est un nogood, il existe un mapping m dans ng dont on peut dériver la clause vide

Approche **incrémentale** :

A chaque ajout d'un nouveau mapping m par un pair P

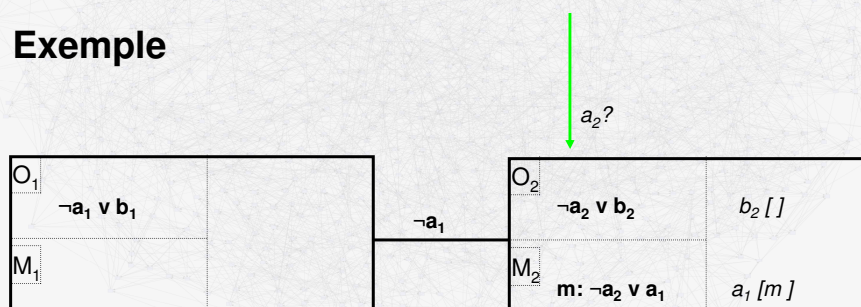
1. Calculer toutes les conséquences de m (et construire toutes leurs preuves)
2. Pour chaque preuve (de la clause vide), garder le **mapping support** $ms : \{m\} \cup ms$ est un nogood
3. Stocker les nogoods ainsi trouvés dans P

5

Mapping supports

Définition: le mapping support d'une preuve est l'ensemble des mappings participant à la preuve

Exemple



6

L'algorithme P2P-NG

Une adaptation de DeCA, appliqué à chaque ajout d'un nouveau mapping par P

P2P-NG(m,P):

- calcule les mapping supports de toutes les preuves de la clause vide à partir de m.
- Les conditions de terminaison sont différentes
- Les branches de raisonnement qui ne peuvent aboutir à la clause vide sont arrêtées
→ évite de nombreux calculs faits par DeCA

7

Propriétés de P2P-NG

- Correct
- Complet
Tous les nogoods minimaux sont trouvés
(indépendamment de l'ordre dans lequel les mappings sont ajoutés)
- Stockage distribué des nogoods
un nogood trouvé par P2P-NG(m,P) est stocké dans P (et contient m) **seulement s'il** est minimal par rapport à ceux déjà stockés dans P

8

Raisonnement bien fondé

- Exploitation des nogoods pour ne renvoyer que les impliqués premiers propres bien fondés
Un impliqué est bien fondé si un de ses mapping supports ne contient aucun nogood
- **L'algorithme WF-DeCA**, une adaptation de DeCA :
 - conserve et combine les mapping supports pour chaque preuve
 - récolte et combine les nogoods stockés dans chaque pair visité lors de la construction de la preuve
 - élimine les preuves en construction dont le mapping support contient un des nogoods récoltés au cours de la preuve

9

Propriétés de WF-DeCA

- Terminaison
- Correction
s'appuie sur la complétude de P2P-NG
- Malgré le stockage distribué des nogoods, **tous les nogoods potentiellement pertinents pour la preuve en cours de construction sont collectés au cours de la preuve.**

10

Plan de la présentation

1. Raisonnement dans les P2PIS
 - A. Contexte et définition du problème
 - B. L'algorithme DeCA
 - C. Raisonnement en présence d'inconsistances
2. Construire des PDMS pour le web sémantique à partir de DeCA
 - B. Le PDMS SomeOWL
 - C. Le PDMS SomeRDFS
3. Passage à l'échelle
4. Conclusion

11

Contexte : le web sémantique

- La "prochaine génération du Web": une infrastructure permettant d'exprimer l'information de façon précise, interprétable par les humains et les machines.
- HTML insuffisant: aucune distinction entre sémantique et présentation.

Titre de l'oeuvre

Nom de l'artiste

```
<A HREF=" ../dali.html"><B>DALI, Salvador</B></A><BR>
The Persistence of Memory<br>1931<br>
Oil on canvas<br>9 1/2 x 13 in (24.1 x 33 cm)<br>
The Museum of Modern Art, New York<br>
<IMG SRC="
http://www.artchive.com/artchive/D/dali/persistence.jpg.html" >
```

Musée

Référence de l'image



12

Contexte : le web sémantique

- Nécessité d'un support permettant l'exploitation sémantique des données: les ontologies.

"In computer science, an ontology is the product of an attempt to formulate an exhaustive and rigorous conceptual schema about a domain" (Wikipedia)

Exemple:

Arts Movies, Television, Music...	Business Jobs, Real Estate, Investing...	Computers Internet, Software, Hardware...
Games Video Games, RPGs, Gambling...	Health Fitness, Medicines, Alternative...	Home Family, Consumer, Cooking...
Kids and Teens Arts, School, Toys, Teen Life...	News Media, Newspapers, Weather...	Recreation Travel, Food, Outdoors, Hobbies...
Reference Maps, Education, Libraries...	Regional US, Canada, UK, Europe...	Science Biology, Psychology, Physics...
Shopping Autos, Clothing, Gifts...	Society Foods, Religion, Issues...	Sports Baseball, Soccer, Basketball...
World Deutsch, Español, Français, Italiano, Japanese, Nederlands, Polski, Dansk, Svenska...		

- Standards de description d'ontologies recommandés par le W3C: OWL et RDFS

13

Notre vision du Web sémantique

- Un gigantesque système de gestion de données pair-à-pair (Peer-to-peer Data Management System ou PDMS)
- Chaque pair annote ses données relativement à une ontologie *simple* représentant son point de vue personnel
- Chaque pair peut établir des mappings avec ses voisins

Challenge: répondre à des requêtes en P2P

Les PDMSs SomeOWL et SomeRDFS

14

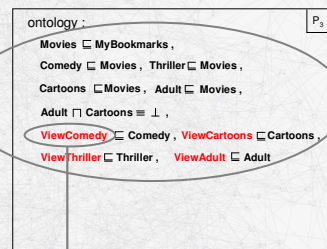
Plan de la présentation

1. Raisonnement dans les P2PIS
 - A. Contexte et définition du problème
 - B. L'algorithme DeCA
 - C. C. Raisonnement en présence d'inconsistances
2. Construire des PDMS pour le web sémantique à partir de DeCA
 - B. Le PDMS SomeOWL
 - C. Le PDMS SomeRDFS
3. Passage à l'échelle
4. Conclusion

15

Le PDMS SomeOWL : un exemple

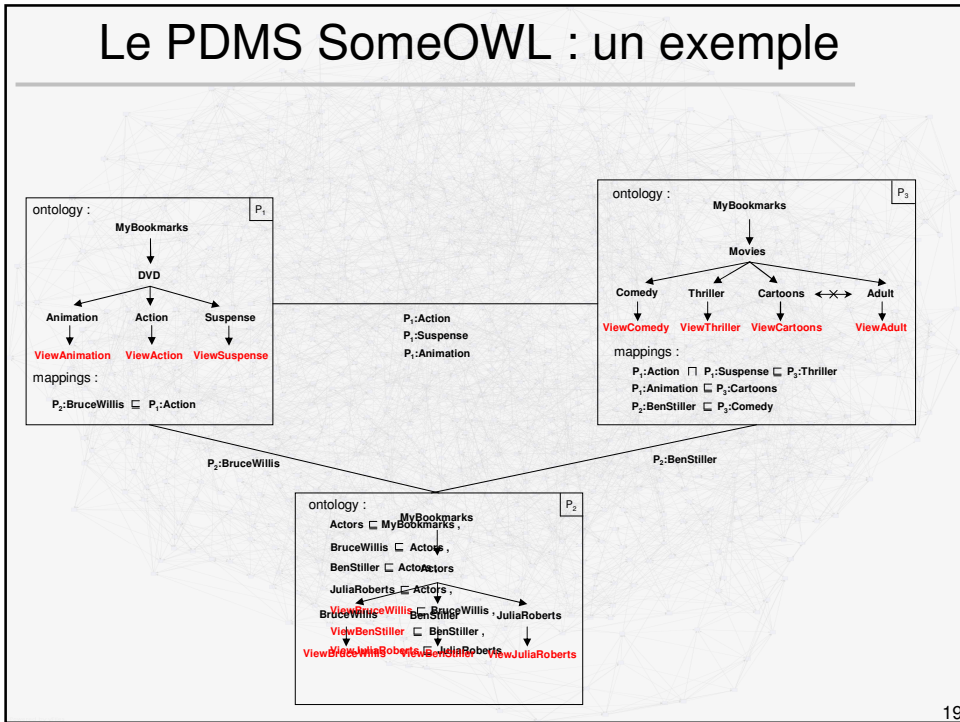
Schéma d'un pair:
Décrit en OWL-PL:
langage à base de
classes utilisant les
constructeurs \neg, \sqcup, \sqcap
(logique de description
CLU)



Classe extensionnelle

16

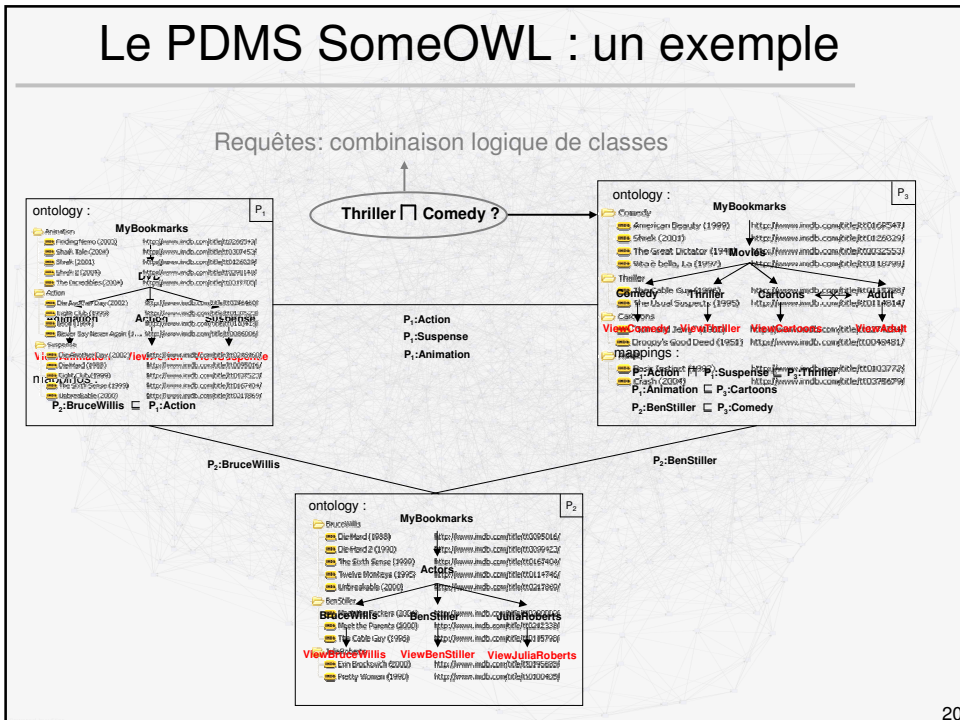
Le PDMS SomeOWL : un exemple



19

Le PDMS SomeOWL : un exemple

Requêtes: combinaison logique de classes



20

Le PDMS SomeOWL : calcul des réponses

- un problème de raisonnement

Soit \mathcal{S} un PDMS someOWL et \mathcal{C} l'ensemble de ses données

$$Q(\mathcal{S}) = \{\bar{t} \in \mathcal{C}^n \mid \mathcal{S} \models Q(\bar{t})\} \quad (\text{réponse certaine})$$

Comment trouver toutes les réponses certaines ?

- Par réécriture de requêtes :

- **Étape 1** : calcul de *réécritures* = reformulation de la requête en des combinaisons de *classes extensionnelles*

- **Étape 2** : Évaluation des réécritures (calcul des extensions) donnant des réponses

21

Répondre par réécriture: intérêt

- Permet de raisonner en pair-à-pair uniquement au niveau du schéma (pas de transport de données)
- Les réécritures indiquent les pairs à interroger indirectement
 - Une dizaine parmi des milliers potentiels

Propriété

S'il existe un nombre fini de réécritures *conjonctives maximales* de la requête initiale relativement au PDMS, l'évaluation de ces réécritures donne *toutes* les réponses certaines en temps polynomial dans la taille des données [Goasdoué & Rousset 2004]

22

SomeOWL : calcul des réécritures

Par encodage propositionnel direct

- $Prop(\top) = true, Prop(\perp) = false$
- $Prop(A) = A$, if A is an atomic class
- $Prop(D_1 \sqcap D_2) = Prop(D_1) \wedge Prop(D_2)$
- $Prop(D_1 \sqcup D_2) = Prop(D_1) \vee Prop(D_2)$
- $Prop(\neg D) = \neg(Prop(D))$
- $Prop(C \sqsubseteq D) = Prop(C) \Rightarrow Prop(D)$
- $Prop(C \equiv D) = Prop(C) \Leftrightarrow Prop(D)$
- $Prop(C \sqcap D \equiv \perp) = \neg Prop(C) \vee \neg Prop(D)$

Propriété de transfert

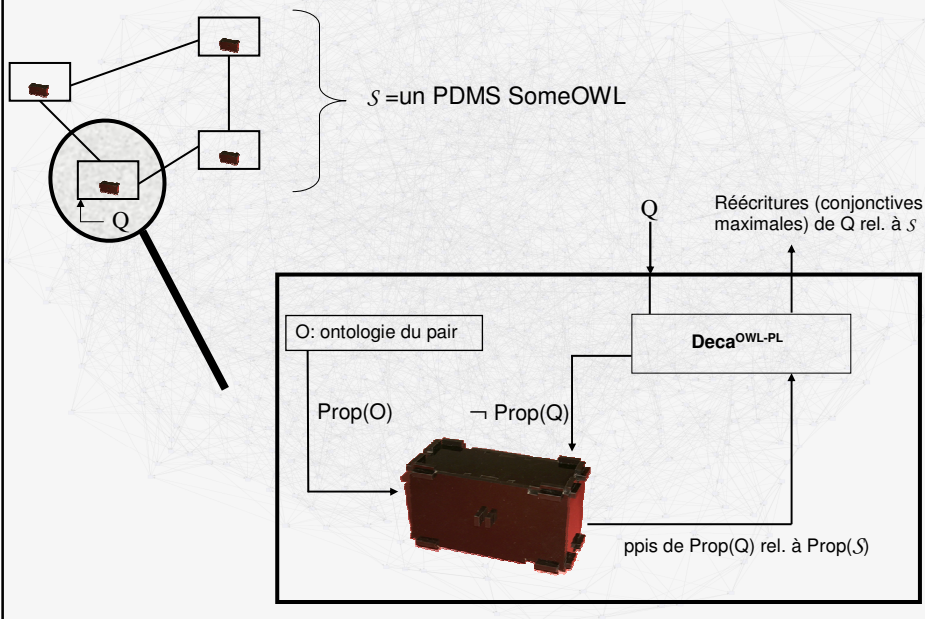
Soit P un PDMS et $Prop(P)$ son encodage propositionnel. Q_e est *réécriture maximale* d'une requête Q ssi $\neg Prop(Q_e)$ est un impliqué premier propre de $\neg Prop(Q)$ rel. à $Prop(P)$

Conséquence directe

La tâche de réécriture peut être effectuée par DeCA !

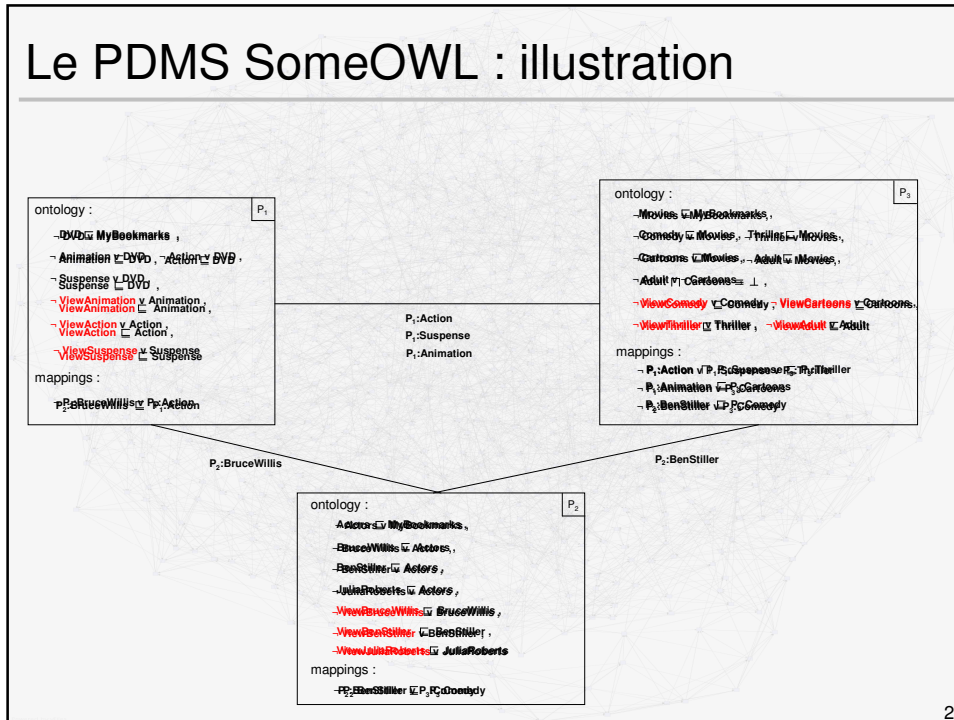
23

Le PDMS SomeOWL : réécrire une requête en utilisant DeCA

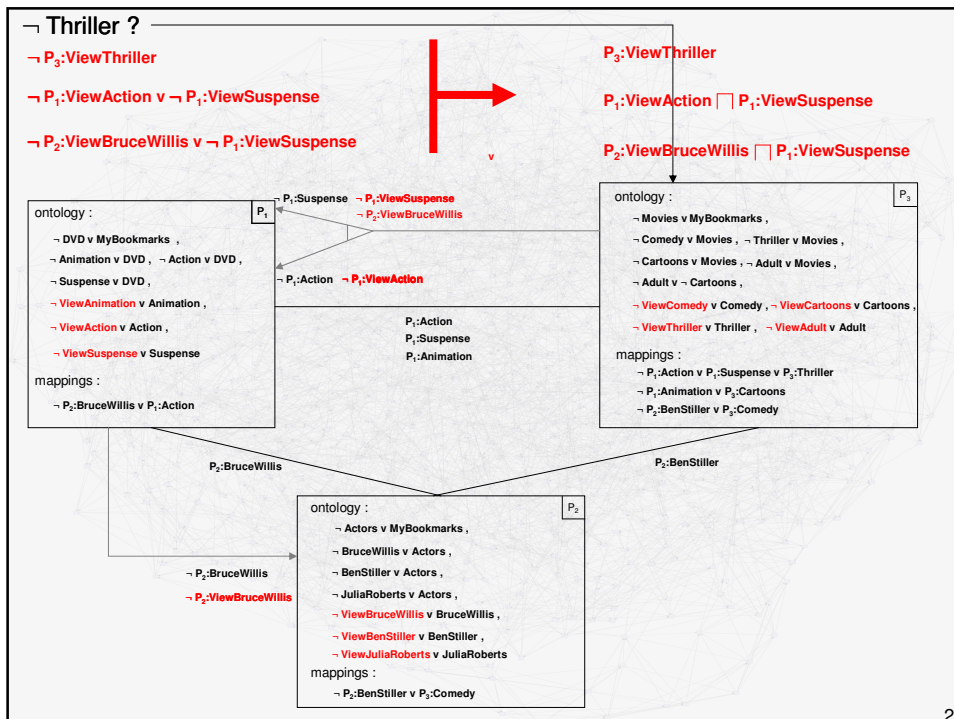


24

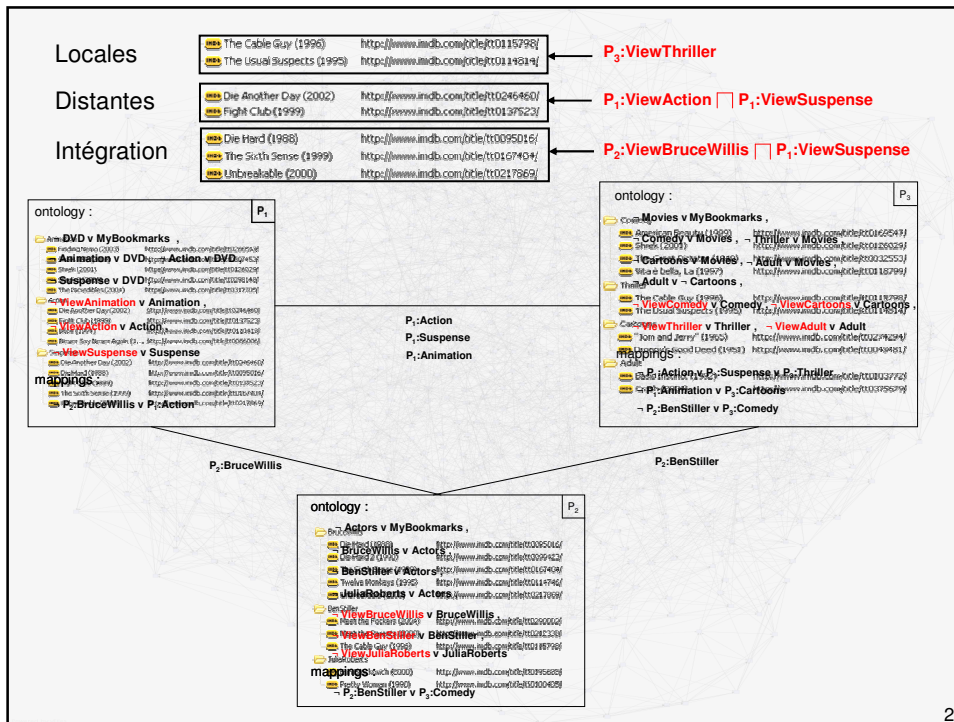
Le PDMS SomeOWL : illustration



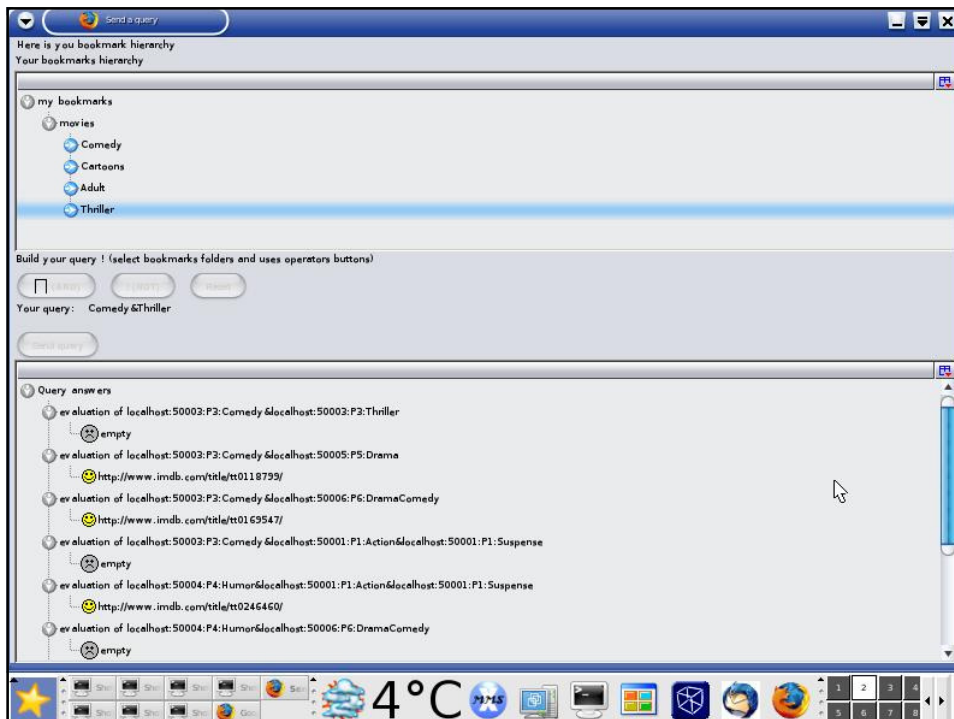
25



26



27



Plan de la présentation

1. Raisonnement dans les P2PIS
 - A. Contexte et définition du problème
 - B. L'algorithme DeCA
2. Construire des PDMS pour le web sémantique à partir de DeCA
 - A. Contexte
 - B. [REDACTED]
 - C. Le PDMS SomeRDFS
3. Passage à l'échelle
4. Conclusion

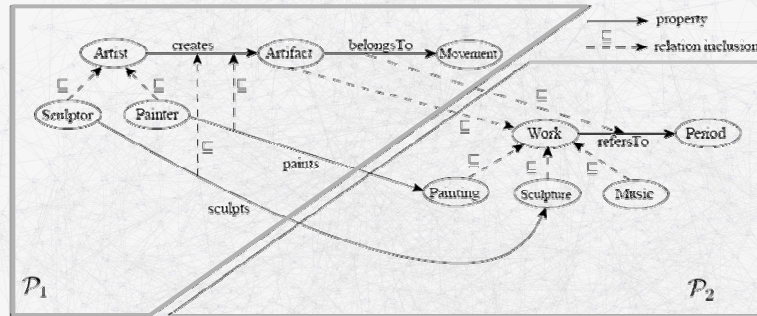
1

SomeRDFS

- **Extension du modèle de données à RDF(S)**
- **RDF(S)**
 - Recommandation du W3C pour décrire les ressources du web
 - Relation unaire (classes) et binaire (propriétés) entre objets
 - Chaque objet est identifié de manière unique (URI)

2

SomeRDFS : illustration

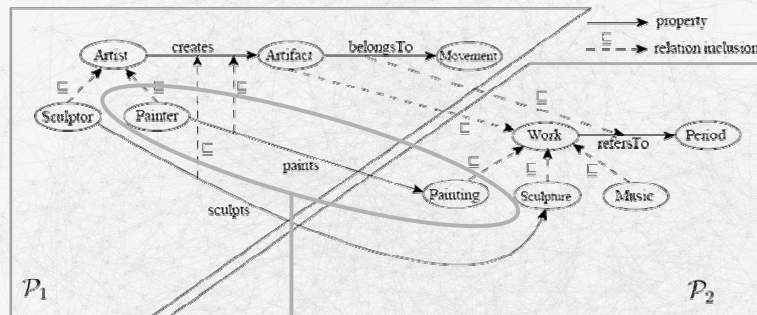


Constructeur	Axiomatisation FOL
Inclusion de classe	$\forall X (C_1(X) \Rightarrow C_2(X))$
Inclusion de propriété	$\forall X \forall Y (P_1(X, Y) \Rightarrow P_2(X, Y))$
Typage du domaine d'une propriété	$\forall X \forall Y (P(X, Y) \Rightarrow C(X))$
Typage du co-domaine d'une propriété	$\forall X \forall Y (P(X, Y) \Rightarrow C(Y))$

Schéma: coreRDFS
-sémantique de la FOL

3

SomeRDFS : illustration

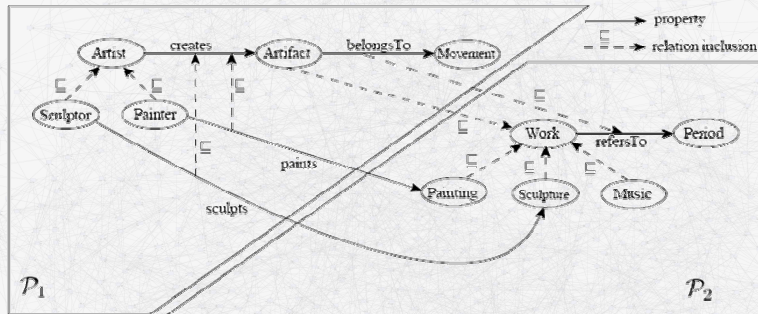


Mappings:

Constructeur	Axiomatisation FOL
Inclusion de classe	$\forall X (C_1(X) \Rightarrow C_2(X))$
Inclusion de propriété	$\forall X \forall Y (P_1(X, Y) \Rightarrow P_2(X, Y))$
Typage du domaine d'une propriété	$\forall X \forall Y (P(X, Y) \Rightarrow C(X))$
Typage du co-domaine d'une propriété	$\forall X \forall Y (P(X, Y) \Rightarrow C(Y))$

4

SomeRDFS : illustration



\mathcal{P}_1 Data

\mathcal{P}_1 : paints(Picasso, Les-demoiselles-d-Avignon)

\mathcal{P}_1 : sculpts(Picasso, La-femme-au-chapeau)

\mathcal{P}_1 : belongsTo(Les-demoiselles-d-Avignon, Picasso-pink)

\mathcal{P}_1 : belongsTo(La-femme-au-chapeau, Modern-art)

\mathcal{P}_2 Data

\mathcal{P}_2 : Painting(Les-dejeuner-des-carottiers)

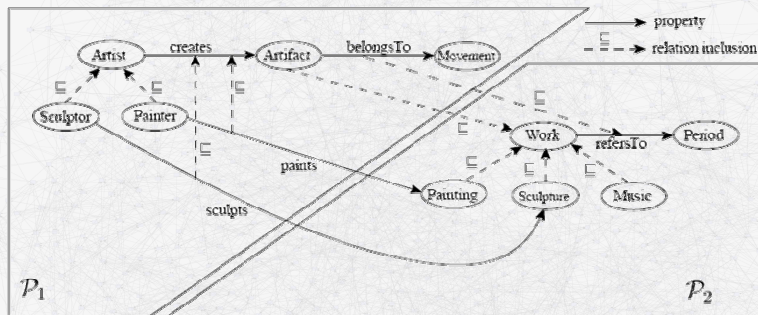
\mathcal{P}_2 : refersTo(Les-demoiselles-d-Avignon, Cubism)

\mathcal{P}_2 : Sculpture(The-statue-of-David)

\mathcal{P}_2 : Music(Nutcracker)

5

SomeRDFS : illustration



$$Q_2(X, Y) \equiv \mathcal{P}_2:\text{Painting}(X) \wedge \mathcal{P}_2:\text{refersTo}(X, Y)$$

Requêtes conjonctives

6

SomeRDFS: réécriture par encodage propositionnel

Fonction de propositionnalisation

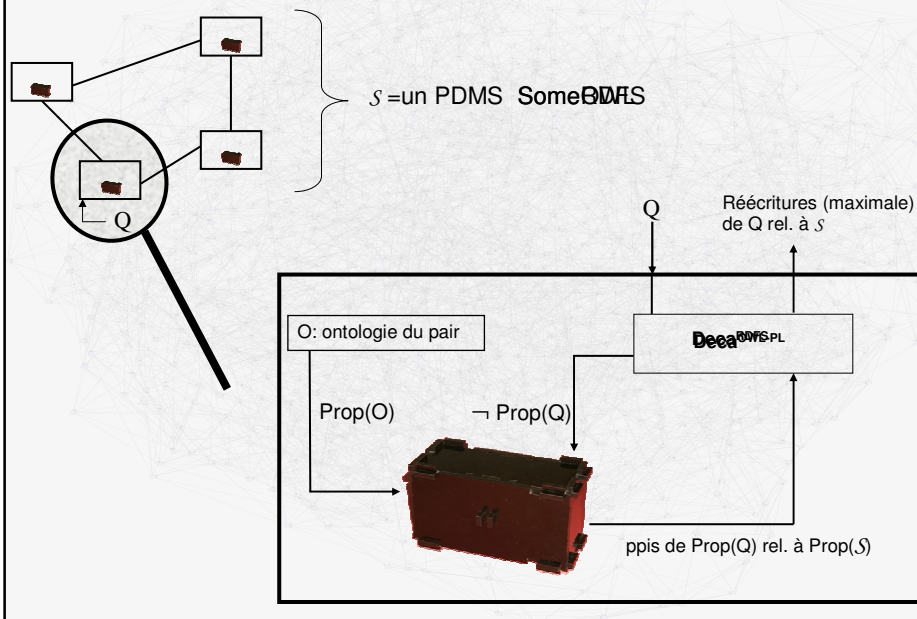
- si $\forall X(C_1(X) \Rightarrow C_2(X))$ est dans \mathcal{P} , $\neg C_1^{dom} \vee C_2^{dom}$ et $\neg C_1^{range} \vee C_2^{range}$ sont dans $Prop(\mathcal{P})$.
- si $\forall X, Y(P(X, Y) \Rightarrow C(X))$ est dans \mathcal{P} , $\neg P^{prop} \vee C^{dom}$ est dans $Prop(\mathcal{P})$.
- si $\forall X, Y(P(X, Y) \Rightarrow C(Y))$ est dans \mathcal{P} , $\neg P^{prop} \vee C^{range}$ est dans $Prop(\mathcal{P})$.
- si $\forall X, Y(P_1(X, Y) \Rightarrow P_2(X, Y))$ est dans \mathcal{P} , $\neg P_1^{prop} \vee P_2^{prop}$ est dans $Prop(\mathcal{P})$.

Propriété de transfert (réécriture conjonctive maximale \Rightarrow impliqué premier propre)

La tâche de réécriture peut être effectuée par DeCA !

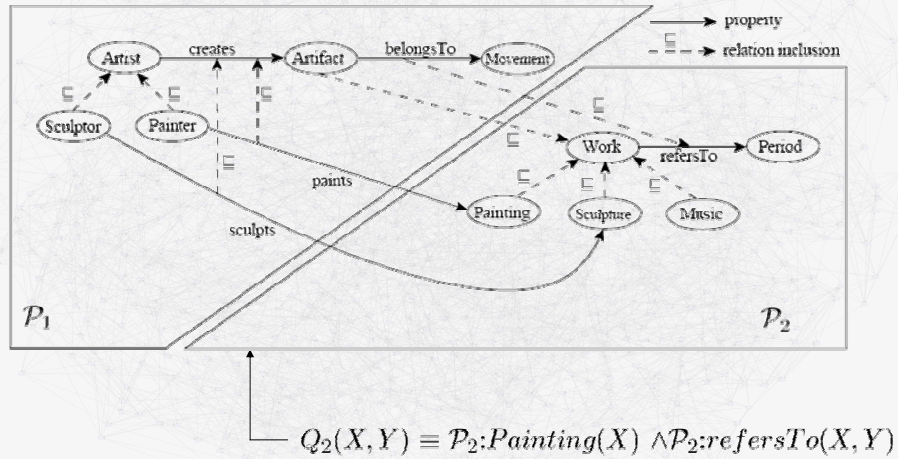
7

SomeRDFS: répondre à une requête avec DeCA



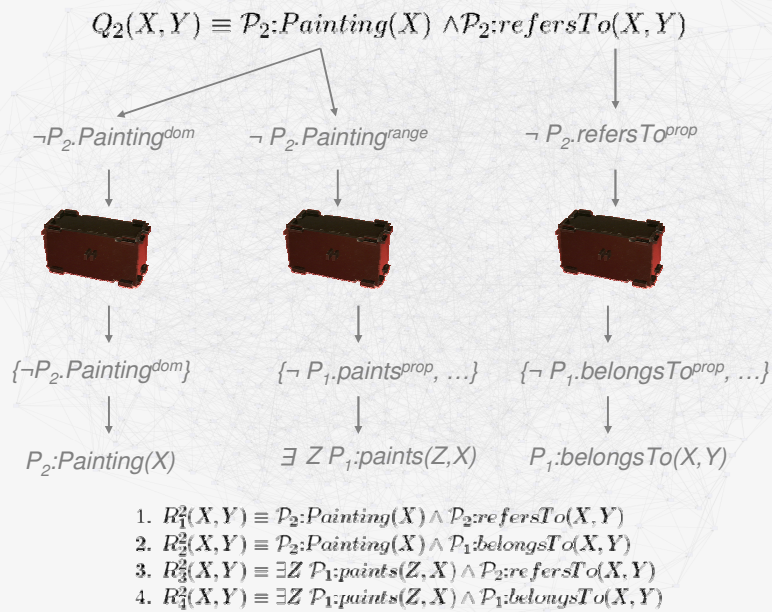
8

SomeRDFS: répondre à une requête avec DeCA



9

SomeRDFS: répondre à une requête avec DeCA



10

SomeRDFS: répondre à une requête avec DeCA

$$Q_2(X, Y) \equiv \mathcal{P}_2:Painting(X) \wedge \mathcal{P}_2:refersTo(X, Y)$$

Évaluation

\mathcal{P}_1 Data	\mathcal{P}_2 Data
$\mathcal{P}_1:paints(Picasso, Les-demoiselles-d-Avignon)$	$\mathcal{P}_2:Painting(Les-jeuneux-des-carotiers)$
$\mathcal{P}_1:sculpts(Picasso, La-femme-au-chapeau)$	$\mathcal{P}_2:refersTo(Les-demoiselles-d-Avignon, Cubism)$
$\mathcal{P}_1:belongsTo(Les-demoiselles-d-Avignon, Picasso-pink)$	$\mathcal{P}_2:Sculpture(The-statue-of-David)$
$\mathcal{P}_1:belongsTo(La-femme-au-chapeau, Modern-art)$	$\mathcal{P}_2:Music(Nutcracker)$

$$Q_2(\mathcal{S}) = \underbrace{\emptyset}_{R_1^2(\mathcal{S})} \cup \underbrace{\emptyset}_{R_2^2(\mathcal{S})} \cup \underbrace{\{(Les-demoiselles-d-Avignon, Cubism)\}}_{R_3^2(\mathcal{S})} \\ \cup \underbrace{\{(Les-demoiselles-d-Avignon, Picasso-pink)\}}_{R_4^2(\mathcal{S})}$$

1. $R_1^2(X, Y) \equiv \mathcal{P}_2:Painting(X) \wedge \mathcal{P}_2:refersTo(X, Y)$
2. $R_2^2(X, Y) \equiv \mathcal{P}_2:Painting(X) \wedge \mathcal{P}_1:belongsTo(X, Y)$
3. $R_3^2(X, Y) \equiv \exists Z \mathcal{P}_1:paints(Z, X) \wedge \mathcal{P}_2:refersTo(X, Y)$
4. $R_4^2(X, Y) \equiv \exists Z \mathcal{P}_1:paints(Z, X) \wedge \mathcal{P}_1:belongsTo(X, Y)$

11

Plan de la présentation

1. Raisonnement dans les P2PIS
 - A. Contexte et définition du problème
 - B. L'algorithme DeCA
2. Construire des PDMS pour le web sémantique à partir de DeCA
 - A. Contexte
 - B. Le PDMS SomeOWL
 - C. Le PDMS SomeRDFS

3. Passage à l'échelle

4. Conclusion

12

Expérimentations

- Génération de P2PIS artificiels

- Ce qui a été mesuré

- la profondeur du raisonnement dans le réseau
- le temps de calcul pour produire les conséquences

- Sur des réseaux « petit monde »

1000 peers Ring,
10 neighbors/peer

p: probabilité de
rediriger une arête

P=0.1

Small world

- Taille des théories locales

- 70 clauses de taille 2 à partir de 70

les clauses 40 sont des variables c.

clauses de taille 2 ou 3

parages $\Rightarrow 2+p$ théories clauseales

Modèle de Watts and Strogatz

parmi
toirement

ies

Graphe aléatoire

13

Expérimentations

- Plateforme d'expérimentation

- cluster de 75 ordinateurs, processeur Athlon avec 1 GB de RAM: 26 à 1.4 GHZ, 9 à 1.8 GHZ et 40 à 2 GHZ. Réseau haut débit

- Chaque ordinateur simulait environ 14 pairs affectés aléatoirement $\cong 1050$ pairs

- Paramètres que l'on a fait varier

- **q**: le nombre de variables partagées

- **%3cnf**: pourcentage de clauses de taille 3

- timeout de 30 secondes

14

Résultats (résumé)

- Faible profondeur du traitement de la requête
 - moins de 7, même dans les cas difficiles
- Temps de calcul
 - dans 90% des cas, la première réponse est produite en moins de 2s
 - Cas facile (mappings simple) :
 - Peu de réponses par requêtes (5 en moyenne)
 - Très rapide (moins de 0.1s) de les calculer toutes sans timeouts
 - Cas difficiles (mappings plus nombreux et plus complexes)
 - Autour de 1000 réponses par requête (> 30% avec timeouts)
 - Assez rapide de les obtenir toutes (moins de 20s)

15

Résultats (quelques détails)

Temps et nombre de réponses (valeurs moyennes pour 300 requêtes)

Réseau	Très Facile	Facile	Moyen	Difficile	Très Difficile
	$q = 2$ % <i>conf</i> = 0	$q = 3$ % <i>conf</i> = 20	$q = 3$ % <i>conf</i> = 100	$q = 5$ % <i>conf</i> = 100	$q = 10$ % <i>conf</i> = 100
1 ^{ère} rép.	0.04s (100%)	1.26s (99.6%)	1.58s (95.6%)	1.39s (89.3%)	2.66s (49.7%)
10 ^{ème} rép.	0.06s (14.3%)	1.37s (25.6%)	0.99s (33.3%)	1.13s (12.0%)	5.38s (29.9%)
100 ^{ème} rép.	-	2.11s (12.7%)	0.84s (27.0%)	4.09s (10.7%)	11.0s (9.0%)
1000 ^{ème} rép.	-	4.17s (6.80%)	4.59s (21.2%)	11.35s (7.15%)	16.6s (1.80%)
tout	0.07s	5.56s	14.6s	21.23s	27.74s
% timeout	0%	13.9%	37.5%	66.9%	86.9%
#réponses	5.17	364	1006	1004	65
%unsat	4.4%	3.64%	3.76%	1.84%	1.81%

- PDMS correspondant :
 - SomeRDFS \cong Très Facile
 - SomeOWL \cong de Facile à Très Difficile

16

Conclusion

- SomeWhere: une infrastructure générique fondée sur la logique
 - pour la médiation sémantique entre ressources distribuées et hétérogènes
 - données, services, « devices »
 - thématique transversale à différents domaines
 - Intelligence artificielle
 - Bases de données
 - Informatique distribuée
 - Web sémantique

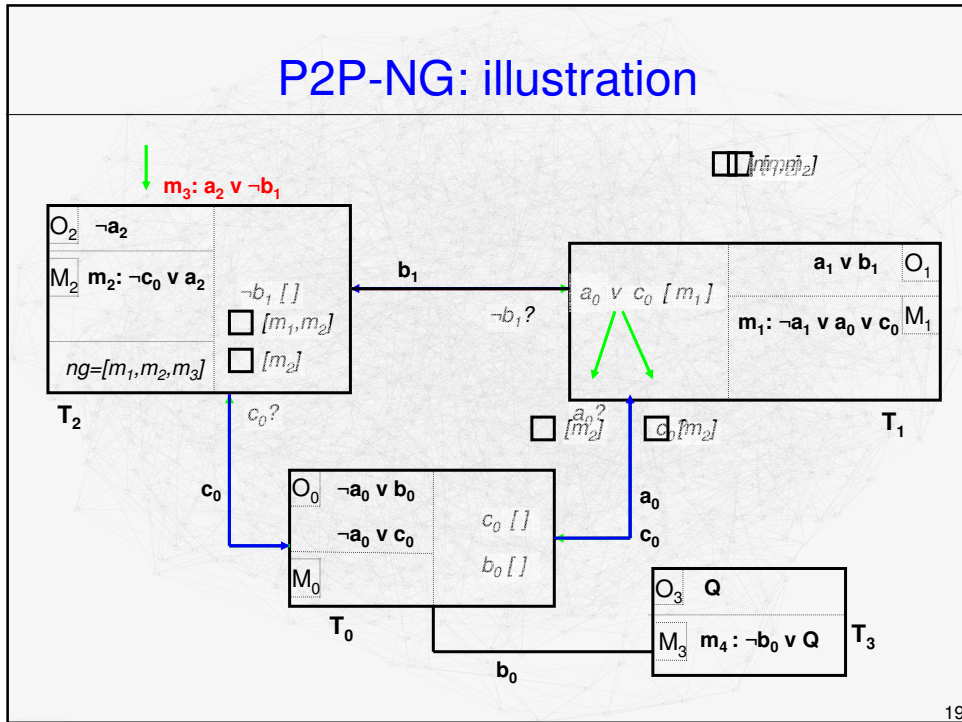
17

Travaux en cours

- Sémantique épistémique pour les mappings
 - Mappings orientés
 - Impact sur le raisonnement
 - moins de conséquences/réponses
 - Travail de F. Goasdoué et N Abdallah [RFIA 08]
 - Modélisation et estimation de la confiance dans des PDMS
 - Découverte et composition de « smart devices »
 - en se fondant sur la description sémantique des fonctionnalités, des entrées/sorties des devices
 - Découverte automatique de mappings
 - Par fouille de méta-données
- Exposé de Rémi Tournaire cet après-midi

18

P2P-NG: illustration



WF-DeCA: illustration

