

# Aligner des ontologies lourdes : une méthode basée sur les axiomes

Frédéric Fürst, Francky Trichet



# Alignement des ontologies

## Principe de l'alignement des ontologies

- Contexte : 2 ontologies couvrant des domaines connexes
- But : établir des liens conceptuels entre primitives conceptuelles (concepts et relations) des 2 ontologies
- Les liens conceptuels peuvent être des identités, des subsomptions ou d'autres liens conceptuels (disjonctions, inverses, ...)

## Les approches existantes

- Terminologiques : analyse des labels des primitives pour déterminer des identités. Raffinement par utilisation de lexiques.
- Basées sur les données : analyse des occurrences des instances des primitives dans les corpus pour déterminer des analogies entre primitives
- Basées sur la sémantique : analyse de la sémantique attachée aux primitives pour déterminer des analogies entre primitives. *Limitées essentiellement aux subsomptions*

# Contexte du travail

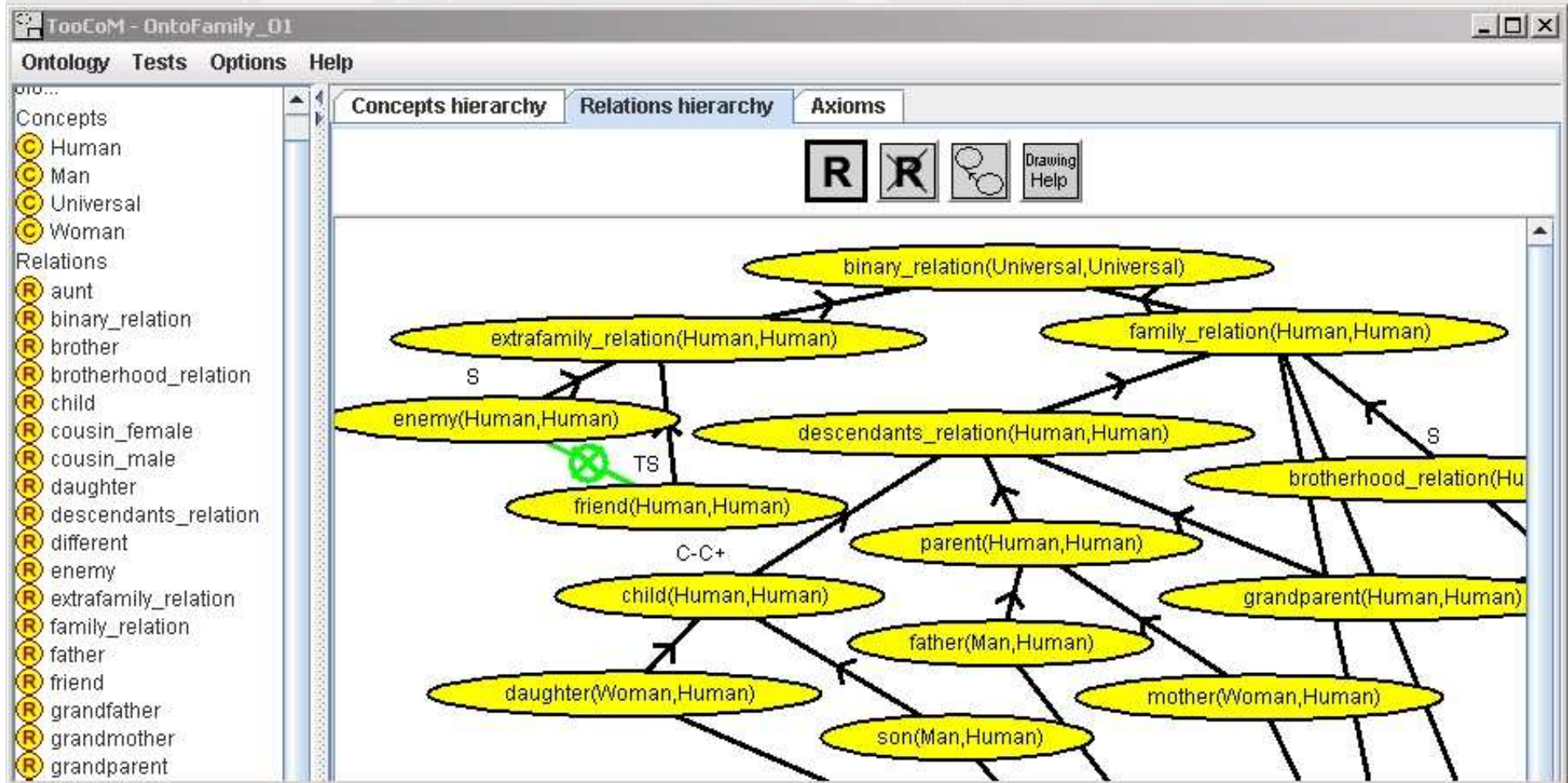
- Alignement d'ontologies « lourdes », intégrant toute la sémantique d'un domaine, au travers de :
  - **propriétés classiques** (schémas d'axiomes) : subsomptions, abstraction des concepts, propriétés algébriques des relations, etc.
  - **axiomes de domaine**, exprimant des propriétés ne relevant pas des schémas classiques
- Les ontologies sont exprimées en **OCGL** (Ontology Conceptual Graphs Language), un langage graphique de représentation d'ontologies issu du modèle des **Graphes Conceptuels** et ses extensions :
  - **hiérarchie de concepts** décorée par des schémas d'axiome
  - **hiérarchie de relations** décorée par des schémas d'axiome
  - **axiomes de domaine** correspondant à une implication logique
- OCGL est implémenté dans l'outil **TooCoM** (a Tool to Operationalize an Ontology in the Conceptual Graphs model), disponible sur <http://sourceforge.net/projects/toocom/>

# OGL dans TooCoM (1/3)

The screenshot displays the TooCoM software interface for an ontology named 'OntoFamily\_O1'. The main window is divided into several sections:

- Left Panel (Ontology Tree):** Shows a hierarchical structure under 'Ontology'. The 'Concept Types' folder is expanded, showing 'Human', 'Man', 'Universal', and 'Woman'. 'Human' is selected, and its children 'Woman' and 'Man' are highlighted with yellow circles.
- Top Panel (Navigation):** Contains tabs for 'Concept types hierarchy', 'Relation types hierarchy', and 'Axioms'. Below the tabs are icons for 'C' (Concept), 'G' (Graph), 'D' (Drawing), and 'Drawing Help'.
- Main Area (Diagram):** Displays a graph of the ontology. 'Universal' is connected to 'Human' by a horizontal arrow. 'Human' is connected to 'Woman' and 'Man' by diagonal arrows. 'Woman' and 'Man' are connected to each other by a vertical line with a green circle containing a cross at the intersection, indicating a disjoint relationship.
- Bottom Right (Properties Dialog):** A dialog box titled 'Properties of Woman' is open. It shows the following fields:
  - Term: Woman (English)
  - Parents: Human
  - Children: (empty)
  - Abstract:
  - Disjoint with: Man
  - (use Ctrl key)
  - OK button

# OGL dans TooCoM (2/3)



# OGL dans TooCoM (3/3)

TooCoM - OntoFamily\_O1

Ontology Tests Options Help

Ontology : OntoFamily\_O1

- Concept Types
- Relation Types
- Instances
- Axioms Schemata
- Axioms
  - Aunt\_Definition
  - Brotherhood\_Definition
  - BrotherViaBrotherhood\_Definition
  - CousinFemale
  - CousinMale
  - Enemy\_Enemy
  - Enemy\_Friend
  - Friend\_Enemy
  - GrandFather\_Definition
  - GrandMother\_Definition
  - HusbandWife
  - NephewViaAunt\_Definition
  - NephewViaUncle\_Definition
  - NieceViaAunt\_Definition
  - NieceViaUncle\_Definition
  - SisterViaBrotherhood\_Definition
  - Uncle\_Definition
  - WifeHusband

Concept types hierarchy Relation types hierarchy Axioms

Remove Axiom New Axiom Logical Form Notes C R X Drawing Help

**C Aunt\_Definition logical form**

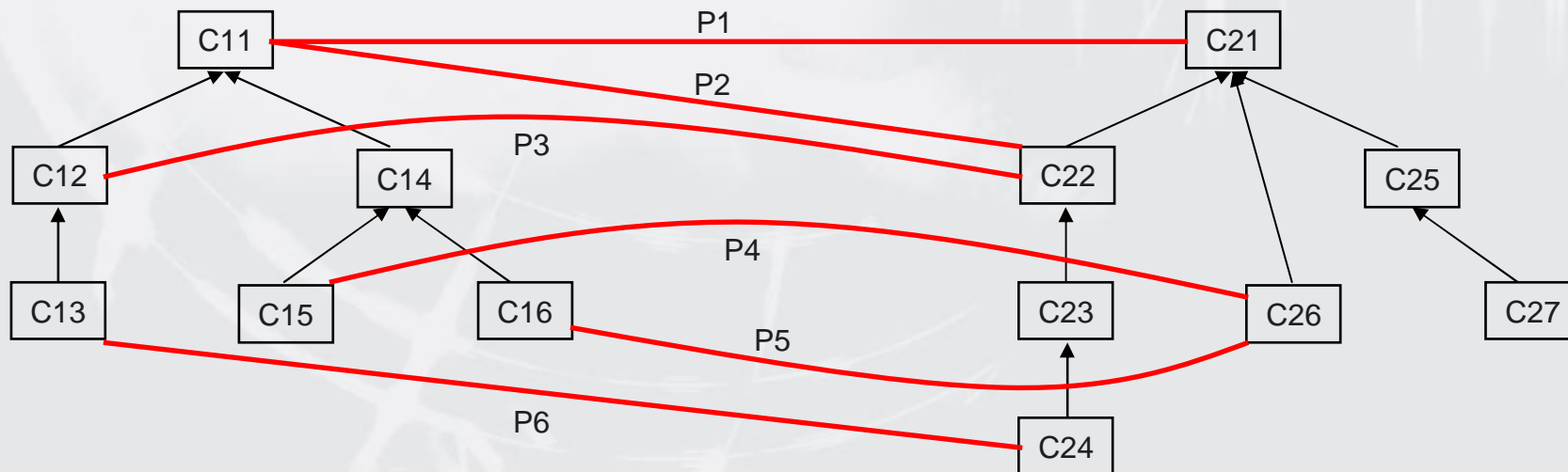
FOR ALL x1,x2,x3 SUCH AS Woman(x1) AND Human(x2) AND Human(x3) AND sister(x1,x2) AND parent(x2,x3) , THEN aunt(x1,x3)

**C Aunt\_Definition note**

[From Wikipedia, the free encyclopedia - <http://www.wikipedia.com>]  
One's aunt (or aunty) is either a female sibling of one of one's parents or the wife of an uncle who is the male sibling of a parent.

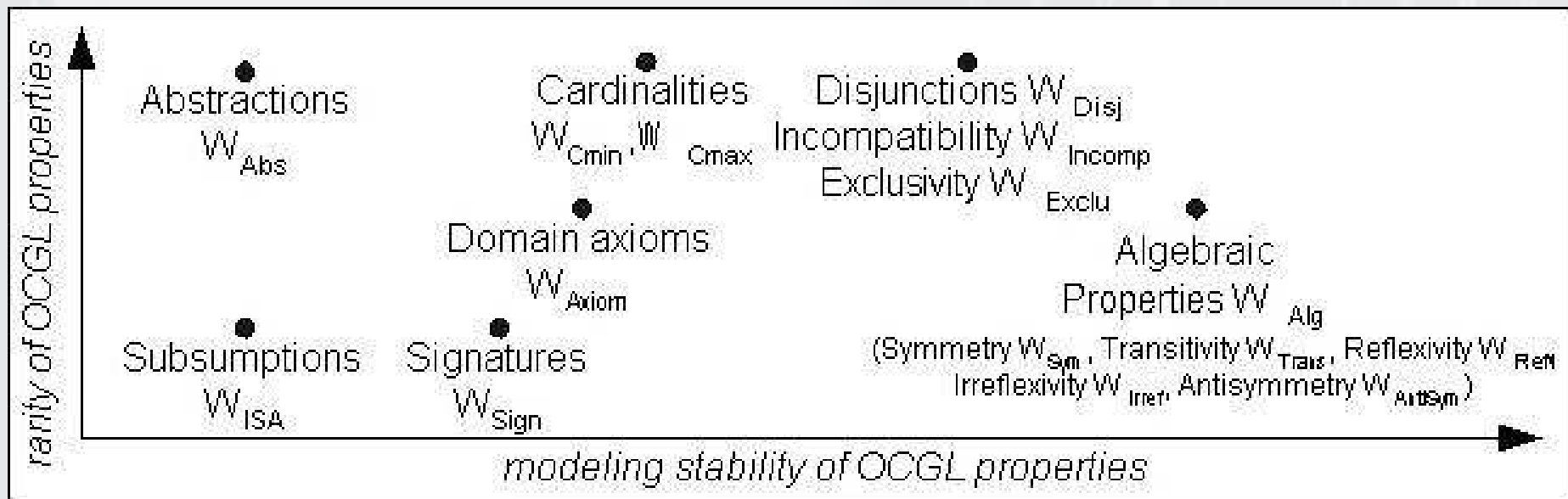
# Principes de l'algorithme (1/5)

- Objectif : étant donné 2 ontologies exprimées en OCL, mettre en évidence des **identités** (appariements) entre concepts et entre relations des 2 ontologies, identités pondérées par des **coefficients de vraisemblance**
- Moyen : baser la découverte d'appariements et leur pondération sur la **comparaison de la sémantique attachée à chaque primitive** à travers schémas d'axiome et axiomes de domaine



# Principes de l'algorithme (2/5)

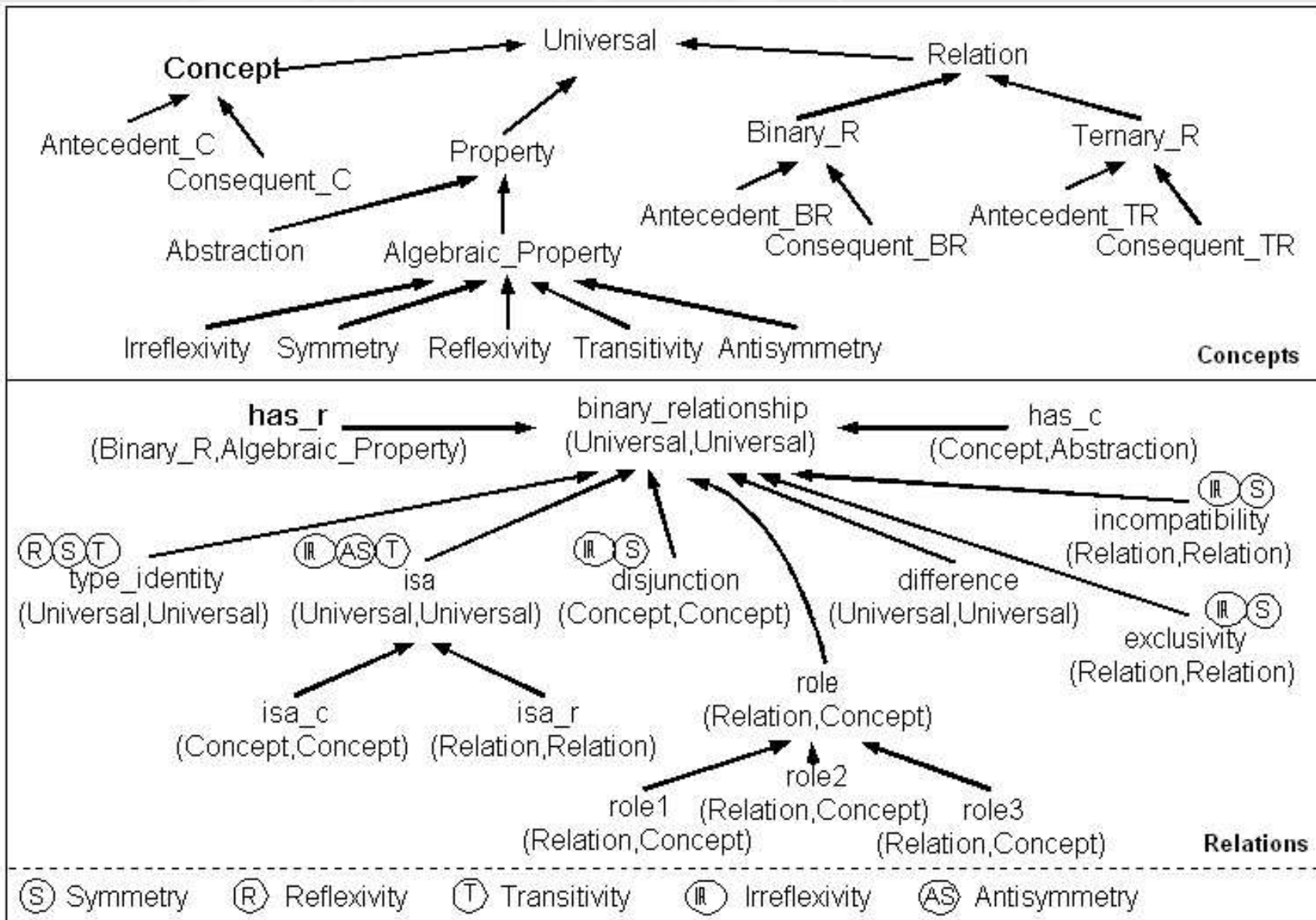
- Poids relatif de chaque propriété dans l'algorithme fixé a priori par :
  - La **stabilité de modélisation** de la propriété, constance de la représentation d'une propriété d'une ontologie à une autre : *plus une propriété est stable, plus elle est pertinente dans la mise en évidence d'appariements*
  - La **rareté** d'une propriété à l'intérieur d'une ontologie : *plus une propriété est rare, plus elle est pertinente dans la mise en évidence d'appariements*



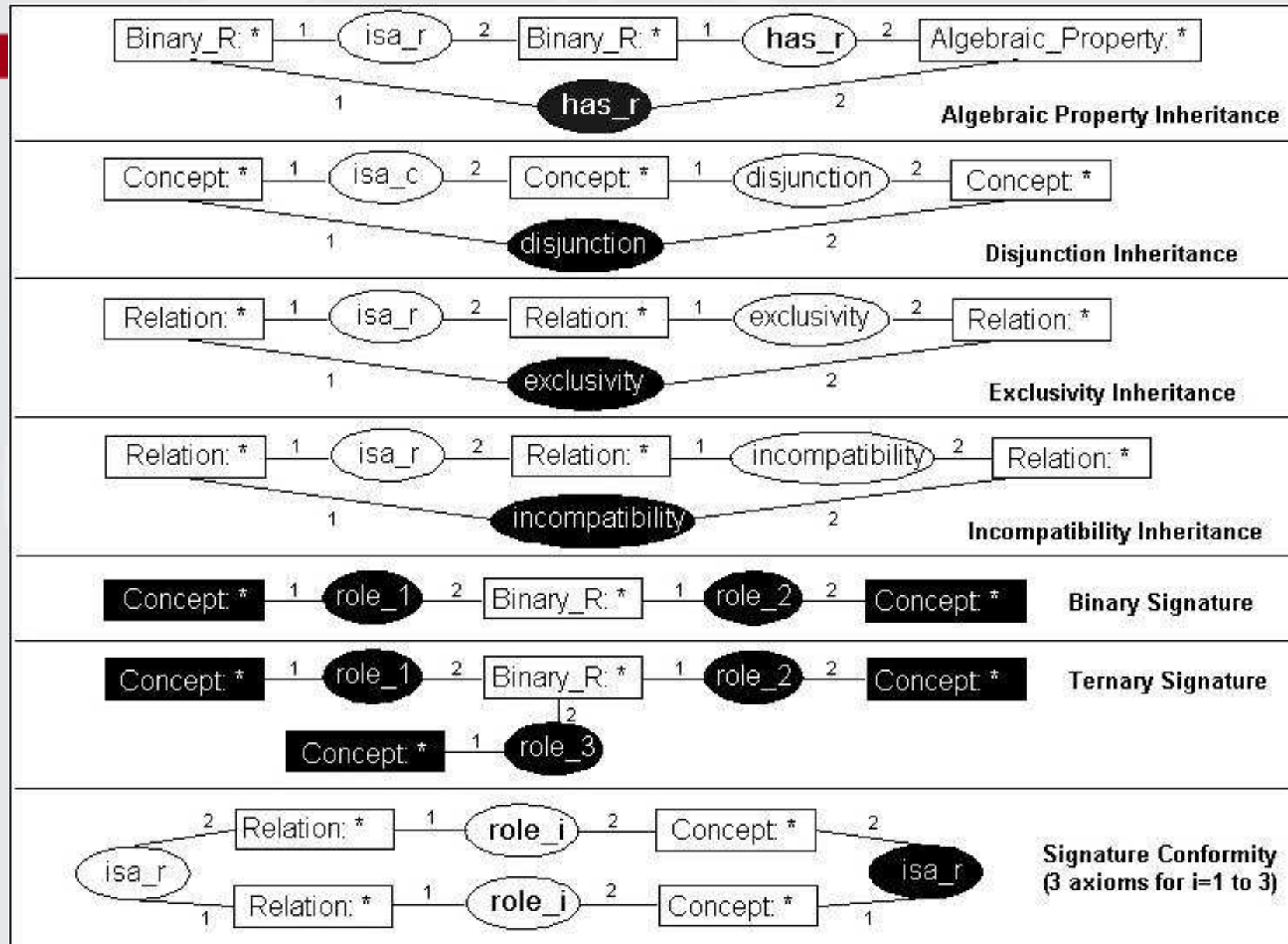
# Principes de l'algorithme (3/5)

- Utiliser les schémas d'axiome pour découvrir et pondérer des appariements de primitives :
  - Pour toute propriété P, si les primitives P1 et P2 de O1 et O2 portent toutes deux P, le poids de l'appariement (P1,P2) est augmenté de celui de P
  - Si l'appariement n'existe pas, il est créé avec le poids correspondant
  - Le poids de l'appariement est diminué du poids de la propriété si une seule des primitives porte la propriété
  - L'absence d'une propriété chez les deux primitives n'entraîne pas de modification (l'absence d'information n'est pas considérée comme de l'information)
- Utiliser les axiomes de domaine pour découvrir et pondérer des appariements de primitives :
  - Comparer topologiquement les axiomes de domaine pour s'abstraire du niveau terminologique : utilisation d'une *ontologie de représentation* pour représenter les axiomes de domaines au niveau « meta »

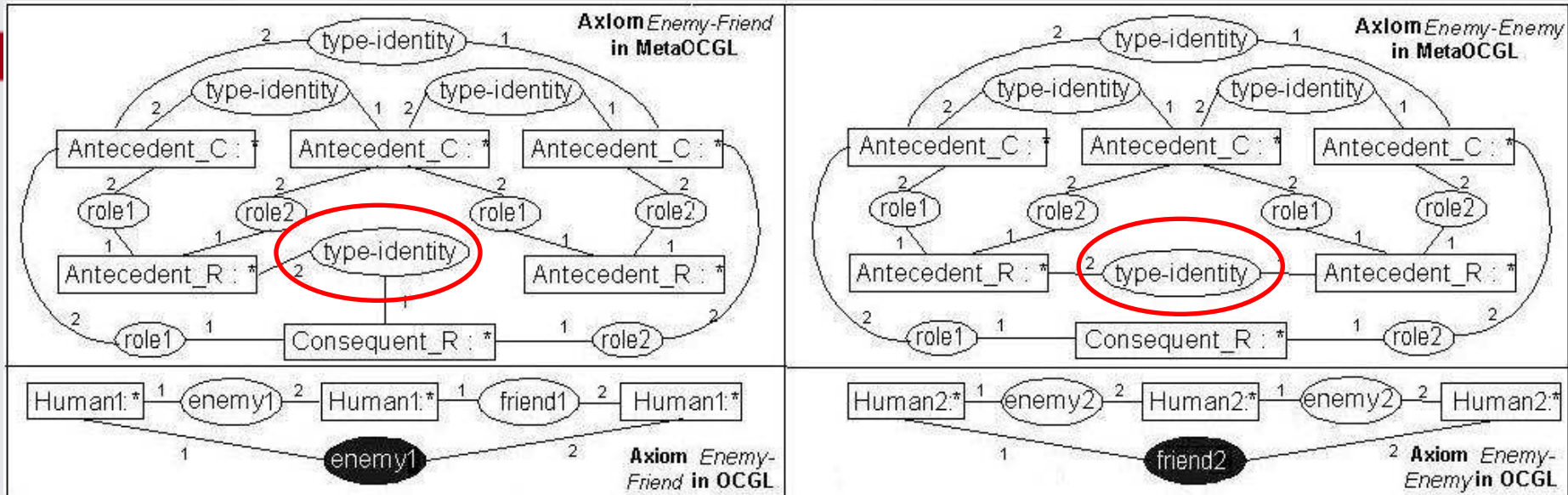
# MetaOCGL (1/3)



# MetaOCGL (2/3)



# Principes de l'algorithme (4/5)



- Comparaison topologique des axiomes via l'opération de **projection** :
  - **Equivalence simple** : double projection entre les meta-représentations des 2 axiomes, sans tenir compte des identités de type
  - **Equivalence typée** : double projection entre les meta-représentations des 2 axiomes incluant les identités de type

Appariements créés ou modifiés : (Human1, Human2), 3\*Wsimple  
 (enemy1, enemy2), Wsimple  
 (friend1, enemy2), Wsimple  
 (enemy1, friend2), Wsimple

# Principes de l'algorithme (5/5)

- Utilisation des **signatures** des relations :
  - Pour modifier les poids des appariements de concepts en fonction des appariements de relations *ou l'inverse*. Exemple : friend(Human,Human)
  - Le choix est basé sur le **taux d'appariements découverts** par rapport au nombre de primitives : *si le taux d'appariements de concepts est supérieur à celui des appariements de relations, les signatures servent à modifier les poids des appariements de relations à partir des appariements de concepts, une fois les appariements de concepts résolus, et vice-versa*
- Résolution des appariements :
  - Une des 2 ontologies est privilégiée arbitrairement
  - Filtrage des appariements les plus significatifs : seuil arbitraire (mais paramétrable)
  - Les appariements sont classés par poids décroissants, l'appariement de plus fort poids, pour chaque primitive de l'ontologie privilégiée, est retenu, les autres rejetés
  - L'intervention de l'utilisateur peut être nécessaire pour trancher entre des conflits liés aux valeurs des poids des appariements (poids égaux par exemple)

# Expérience réalisée

- 2 ontologies du domaine des relations familiales et amicales construites par 2 personnes différentes :
- Onto\_family O1 : 3 concepts, 31 relations, 11 schémas d'axiome et 18 axiomes de domaine
- Onto\_family O2 : 3 concepts, 23 relations, 10 schémas d'axiome et 27 axiomes de domaine

## Résultats expérimentaux

- 9 appariements de concepts (sur 9 possibles) et 192 appariements de relations (sur 713 possibles) : après filtrage, 9 appariements de concepts et 69 appariements de relations.
- Les taux d'appariements sont de 3 pour les concepts et 2.55 pour les relations => utilisation des signatures pour modifier les poids des appariements de relations
- L'algorithme met en évidence les identités entre concepts (Human,Human), (Woman,Female) et (Man, Male)
- L'algorithme met en évidence 9 identités de relations parmi 17 « vraies » identités

# Conclusion

- Nous avons proposé une **première approche de l'utilisation des axiomes**, et surtout des axiomes de domaine pour l'alignement d'ontologies lourdes :
  - Permettant d'exploiter toute la richesse sémantique des ontologies
  - **Complémentaire des approches plus classiques** (terminologiques ou basées sur des corpus)
  - Utilisant une représentation des axiomes au niveau «meta» pour s'abstraire des différences terminologiques
- Il s'agit d'un **travail préliminaire** qui nécessite
  - Des tests sur des ontologies plus larges (mais « lourdes »)
  - La prise en compte des **subsomptions**
  - La généralisation du passage au niveau meta par l'utilisation d'une ontologie de l'alignement d'ontologie et d'ontologie de représentation pour automatiser le processus
  - Une combinaison avec les approches plus classiques

# Liens

---

- Un rapport de recherche détaillant la méthode et l'expérience
  - <http://193.52.99.12/site-statique/fr/research/reports/rr2005/>
- Le site de l'outil TooCoM implémentant l'algorithme
  - <http://sourceforge.net/projects/toocom/>