Peer-to-Peer Inference Systems and (Non-)Conservative Extension of a KB

AI Communications: The European Journal on AI, 2009

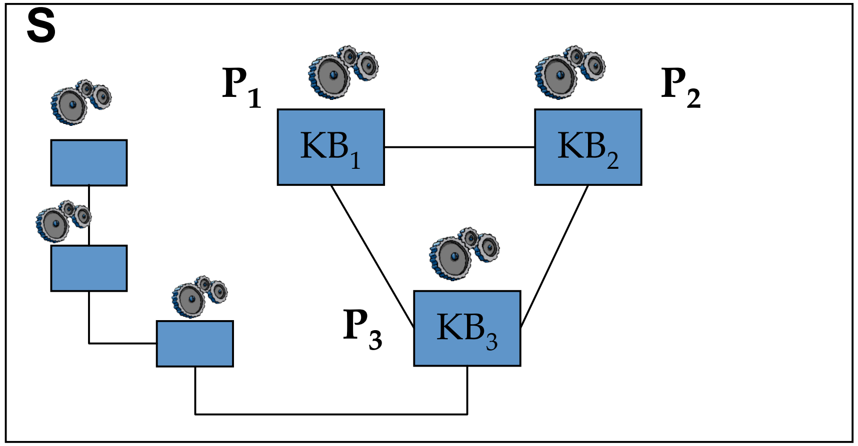
Nada Abdallah **François Goasdoué**

Université Paris-Sud & CNRS (LRI/IASI) – INRIA (Saclay/GEMO)

Peer-to-peer inference system

Each peer has:

- a KB using its *own* symbols
- a set of mappings with other peers
- a copy of a decentralized reasoning algorithm

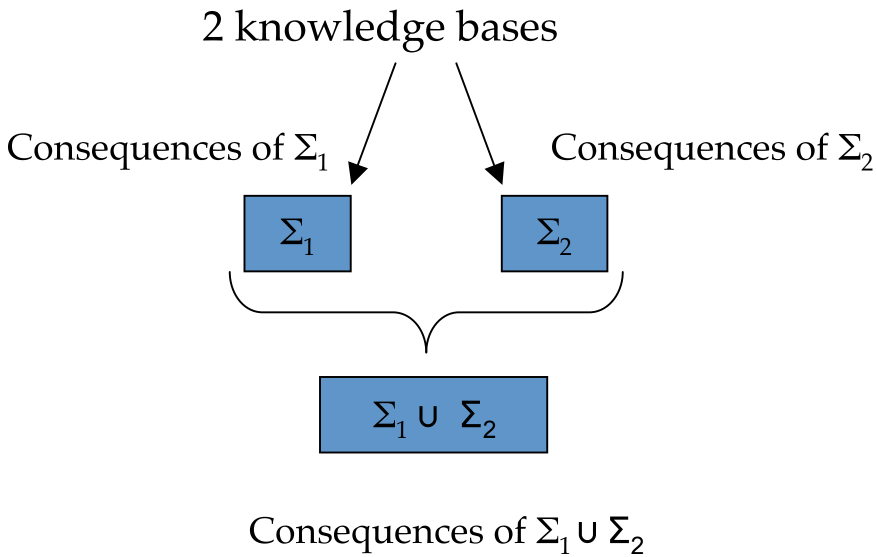


Is S a conservative extension of P_1?

No, in the general case.

S may provide extra knowledge about the own application domain/arera of expertise of P_1, whether P_1 *agrees with or not*!

Conservative extension of a KB



$\Sigma_1 \cup \Sigma_2$ is a conservative extension of Σ_1 iff

$\forall$ f using the symbols of Σ_1 *only*

If $f \in$ Consequences of $\Sigma_1 \cup \Sigma_2$

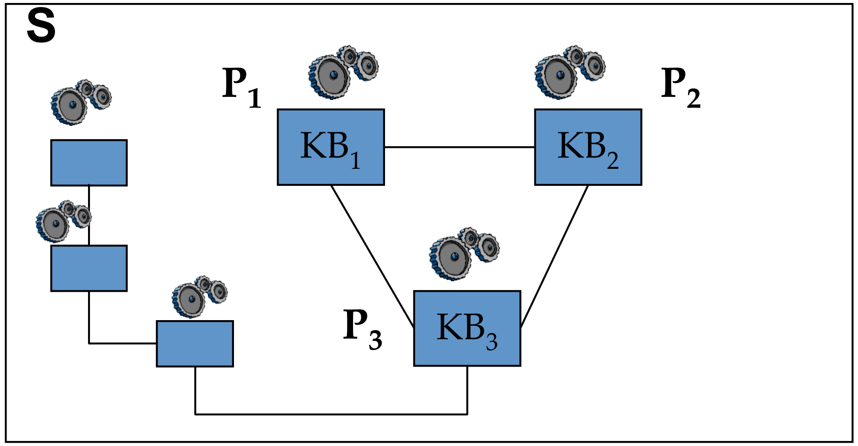
Then $f \in$ Consequences of Σ_1

Σ1 ∪ Σ2 does not have more knowledge in terms of the *own* symbols of Σ1 than Σ1 itself.

Peer-to-peer inference system

Each peer has:

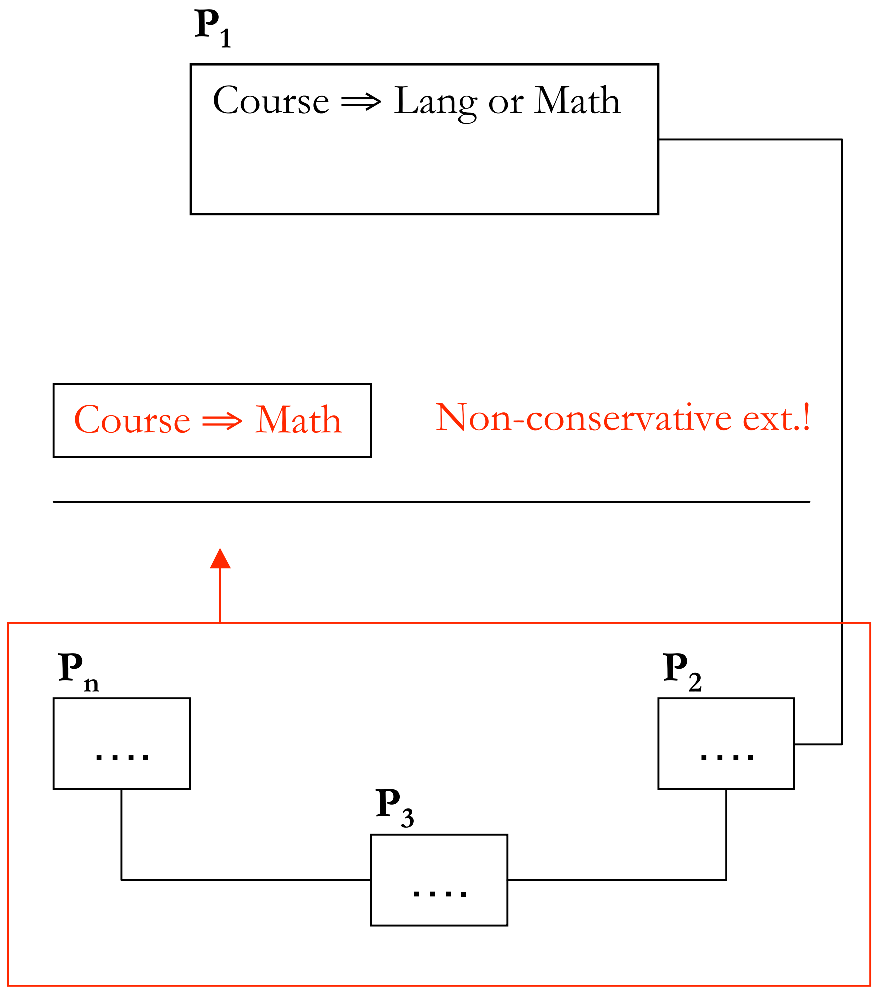
- a KB using its *own* symbols
- a set of mappings with other peers
- a copy of a decentralized reasoning algorithm



Is S a conservative extension of P_1?

No, in the general case.

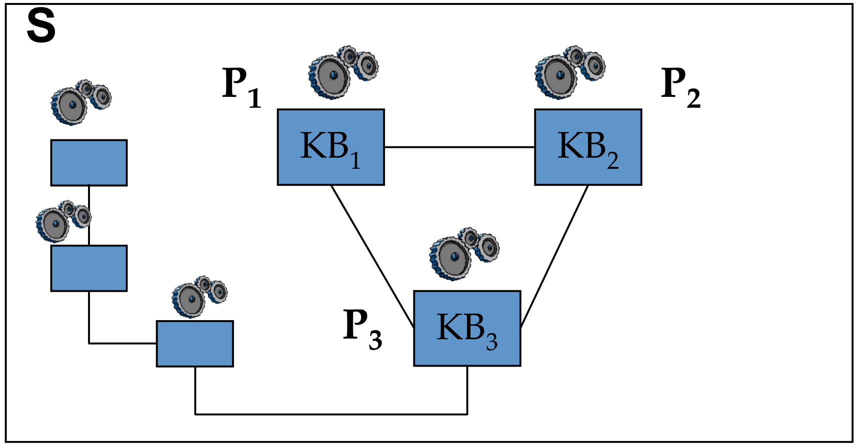
S may provide extra knowledge about the own application domain/arera of expertise of P_1, whether P_1 *agrees with or not*!



Peer-to-peer inference system

Each peer has:

- **a KB using its *own* symbols**
- **a set of mappings with other peers**
- **a copy of a decentralized reasoning algorithm**



Is S a conservative extension of P_1?

No, in the general case.

S may provide extra knowledge about the own application domain/arera of expertise of P_1, whether P_1 ***agrees with or not!***

Does it matter?

→ Non-conservative extension of P_1 amounts to

1. → Knowledge corruption

P_1 has a complete description of its application domain : it is an expert.

Claim: it should be able to forbid the extra knowledge.

2. → Possible knowledge corruption

P_1 has an incomplete description of its application domain but it is an expert.

Claim: it should be able to forbid the inaccurate extra knowledge.

3. → Knowledge to learn

P_1 has an incomplete description of its application domain and it is not an expert.

Claim: it should be able to get the extra knowledge.

P2PISs and Conservative Extension Checking

Contributions

We study two problems from a theoretical and a decentralized algorithmic perspectives:

1.Deciding whether a P2PIS is a conservative extension of a given peer

2.Computing the witnesses to non-conservativeness of a peer, together with their causes

Motivations

To allow a peer to forbid or to learn the extra knowledge that a P2PIS has wrt its application domain/area of expertise

Setting

Propositional P2PISs

→ A good tradeoff between expressivity and scalability.

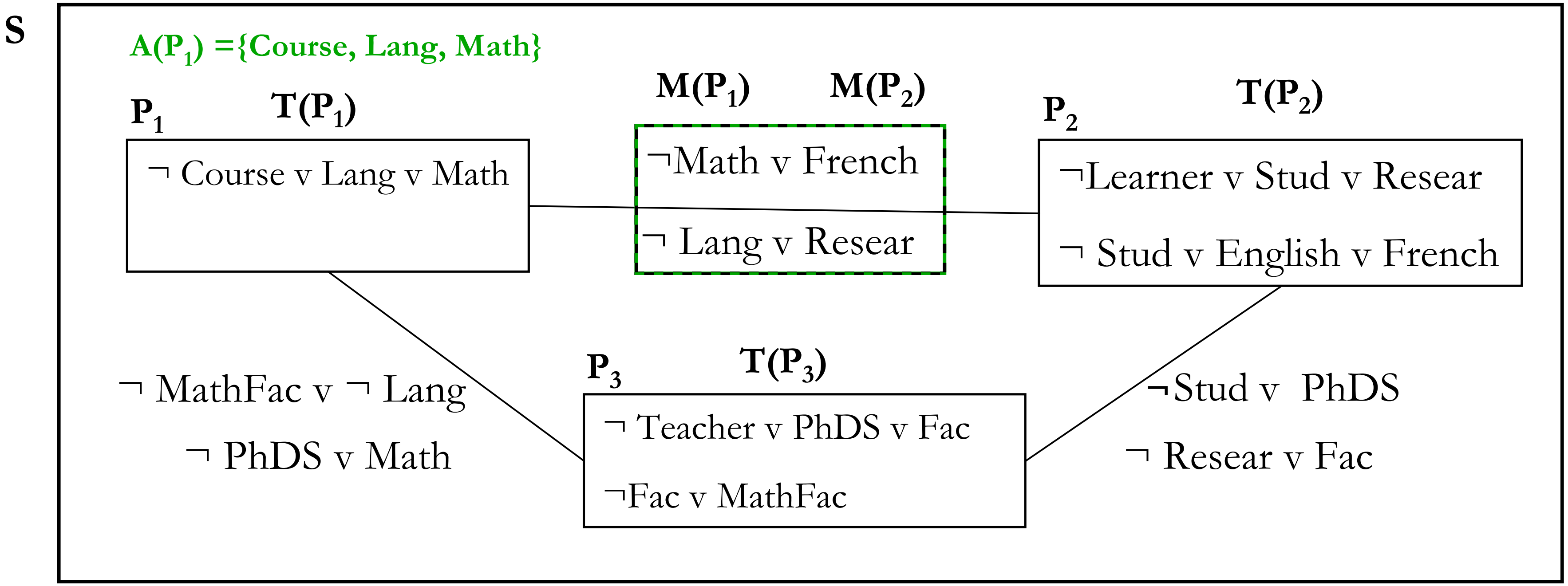
Applications in IA: Diagnosis

Applications in DB: Peer Data Management Systems for the Semantic Web

Summary of the talk

- Peer-to-Peer Inference Systems
- Reasoning Problems
- Techniques
- Conclusion

Propositional P2PISs



$T(S) = T(P_1) \cup T(P_2) \cup T(P_3) \cup M(P_1) \cup M(P_2) \cup M(P_3)$

Summary of the talk

- Peer-to-Peer Inference Systems
- Reasoning Problems
- Techniques
- Conclusion

Decision problem: CE^{dec}

INSTANCE: Let S be a P2PIS, a peer of which is P.

QUESTION: Is S a conservative extension of P?

S is a conservative extension of P:

$\forall$ **c** $\in L(P)$, if $T(S) \models$ **c** then $T(P) \models$ **c**

$3^{\#A(P)}$ clauses

$L(P) = \bigvee_{v \in A(P)} \{\square, V, \neg V\}$

coNP-complete

Complexity issue

- $\prod_2^p$-complete for unrestricted clauses
- in Δ_2^p for $K_{>=3}$-clauses
- in $\prod_1^p$ for (reverse-)Horn clauses
- in P for Krom clauses (i.e., 2-clauses)

Algorithmic issue

T(S) is a theory distributed among autonomous peers

Decentralized algorithms

Fonctional problem: CE^{fun}

INSTANCE: Let S be a P2PIS, a peer of which is P.

QUESTION: What are the witnesses to non-conservativeness of P within S?

Witness (i.e., a clausal counterexample to conservativeness)

Let S be a P2PIS, a peer of which is P. Let c be a clause of L(P).

c is a witness to non-conservativeness of P within S iff $T(S) \models c$ and $T(P) \not\models c$.

Complexity (space/time)

- CE^{fun} is at least as hard as CE^{dec}: CE^{dec} is *true* whenever $CE^{fun} \neq \{\}$ and *false* otherwise.

- CE^{fun} is strictly harder than CE^{dec} for unrestricted clauses and (reverse-)Horn clauses.

Clauses	Space	Time	CE^{dec}
Unrestricted	Exp	Exp	$\prod_2^P$-complete
$K_{>=3}$-clauses	P	Exp	in Δ_2^P
(reverse-)Horn	Exp	Exp	in $\prod_1^P$
Krom	P	P	in P

Summary of the talk

- Peer-to-Peer Inference Systems
- Reasoning Problems
- Techniques
- Conclusion

Techniques

S is a conservative extension of P:

$\forall\, \mathbf{c} \in L(P)$, if $T(S) \models \mathbf{c}$ then $T(P) \models \mathbf{c}$

What is the *necessary* and *sufficient* subset of L(P) to answer CE^{dec} /CE^{fun}?

Exact characterization

The prime implicates of S that *necessarily* follow from a mapping of P:

$\{c \mid c \in L(P) \text{ and } c \in PI(S) \text{ and } c \notin PI(S \setminus \{m\}) \text{ s.t. } m \in M(P)\}$.

S is a conservative extension of P:

$\forall\, \mathbf{m} \in M(P)$, if $T(S) \vdash^{m}_{LR} \mathbf{c}$ s.t. $\mathbf{c} \in L(P)$ then $T(P) \cup \{\neg c\} \vdash_{R} \square$

The Conservative Extension Checking Algorithm (CECA)

CECA solves both CE^{dec} and CE^{fun}

S is a conservative extension of P:

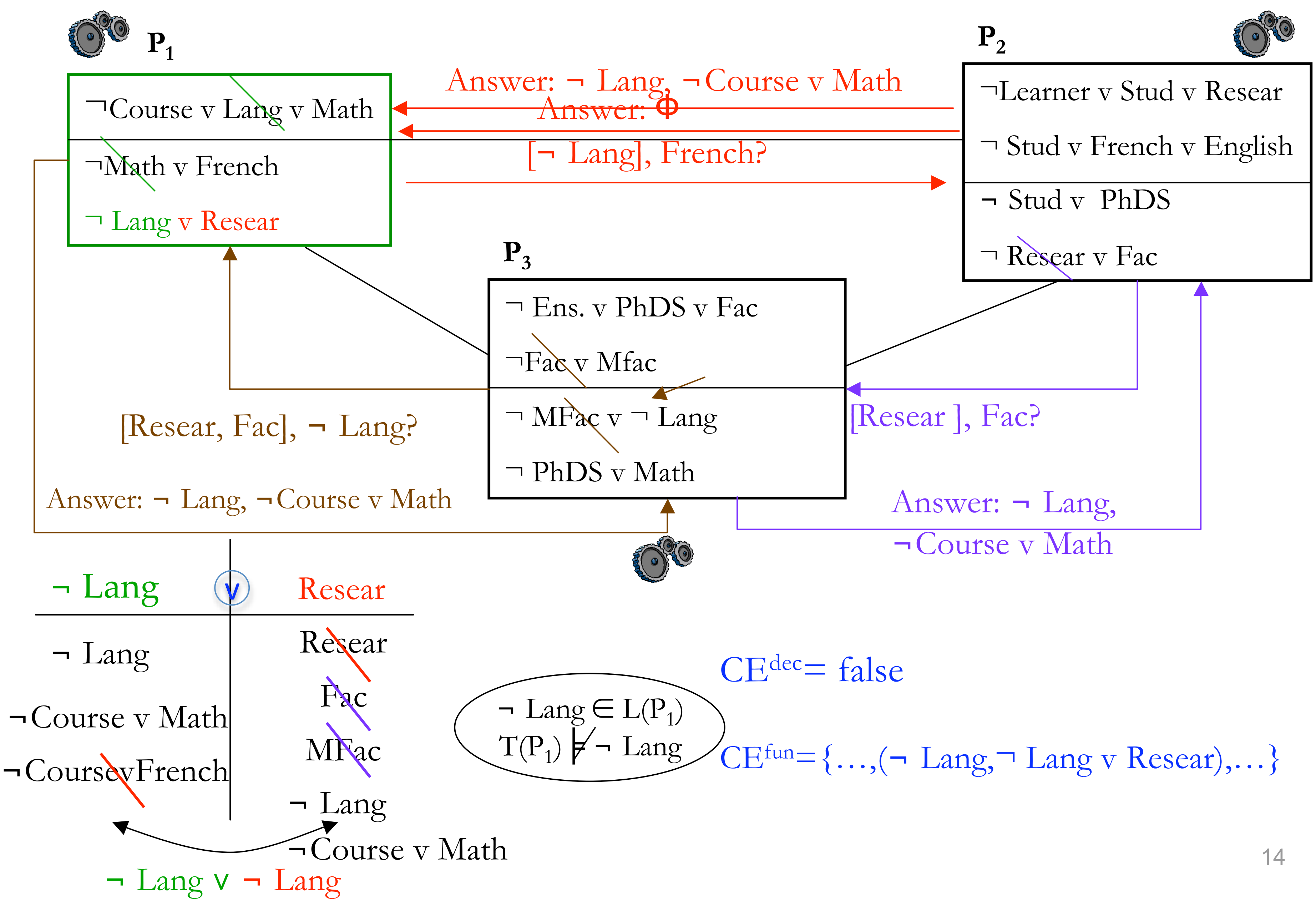
$\forall\ \mathbf{m} \in M(P)$, if $T(S) \vdash^{m}_{LR} \mathbf{c}$ s.t. $\mathbf{c} \in L(P)$ then $T(P) \cup \{\neg c\} \vdash_{R} \square$

CECA running on P	
$\forall\ \mathbf{m} \in M(P)$, $W=\{(c,m) \mid T(S) \vdash^{m}_{LR} c$ s.t. $c \in L(P)\}$ is computed	**Decentralized linear deduction** **$DECA_{LR}(m)$** *A linear virant of DECA* *IJCAI'05 & JAIR (2006).*
For $(c,m) \in W$, check whether $T(P) \cup \{\neg c\} \vdash_{R} \square$	**Centralized refutation**

CE^{dec} → Yes / No

CE^{fun} → Witnesses and their causes

CECA


P1
¬Course v Lang v Math
¬Math v French
¬ Lang v Resear
P2
¬Learner v Stud v Resear
¬ Stud v French v English
¬ Stud v PhDS
¬ Resear v Fac
P3
¬ Ens. v PhDS v Fac
¬Fac v Mfac
¬ MFac v ¬ Lang
¬ PhDS v Math
Answer: ¬ Lang, ¬Course v Math
Answer: Φ
[¬ Lang], French?
[Resear, Fac], ¬ Lang?
Answer: ¬ Lang, ¬Course v Math
[Resear], Fac?
Answer: ¬ Lang, ¬Course v Math
¬ Lang
v
Resear
¬ Lang
¬Course v Math
¬CoursevFrench
Resear
Fac
MFac
¬ Lang
¬Course v Math
¬ Lang v ¬ Lang
¬ Lang ∈ L(P1)
T(P1) ⊭ ¬ Lang
CEdec= false
CEfun={…,(¬ Lang,¬ Lang v Resear),…}

Summary of the talk

- Peer-to-Peer Inference Systems
- Reasoning Problems
- Techniques
- Conclusion

Conclusion

Our work can be directly applied to

- Propositional P2PISs
- PDMSs for the Semantic Web that are built on top of Propositional P2PISs
  - SomeOWL *(Journal of Artificial Intelligence Research, 2006)*
  - SomeRDFS *(Journal on Data Semantics, 2007)*
  - SomeDL-lite *(International Joint Conference on Artificial Intelligence, 2009)*

It can be applied to inconsistency management

It can be applied to KB/ontology mapping

A user or an expert can decide whether a new mapping is acceptable or not

- The mapping being given by someone or being automatically discovered

It may be used in decentralized diagnosis

Does the logical model respect the specifications of the components?

- NO: Is there a modeling problem of the P2P application?
- NO: Can we find a more adequate (and cheaper) component?