

Semantic data integration in P2P systems

D. Calvanese, E. Damaggio, G. De Giacomo, M. Lenzerini, R. Rosati

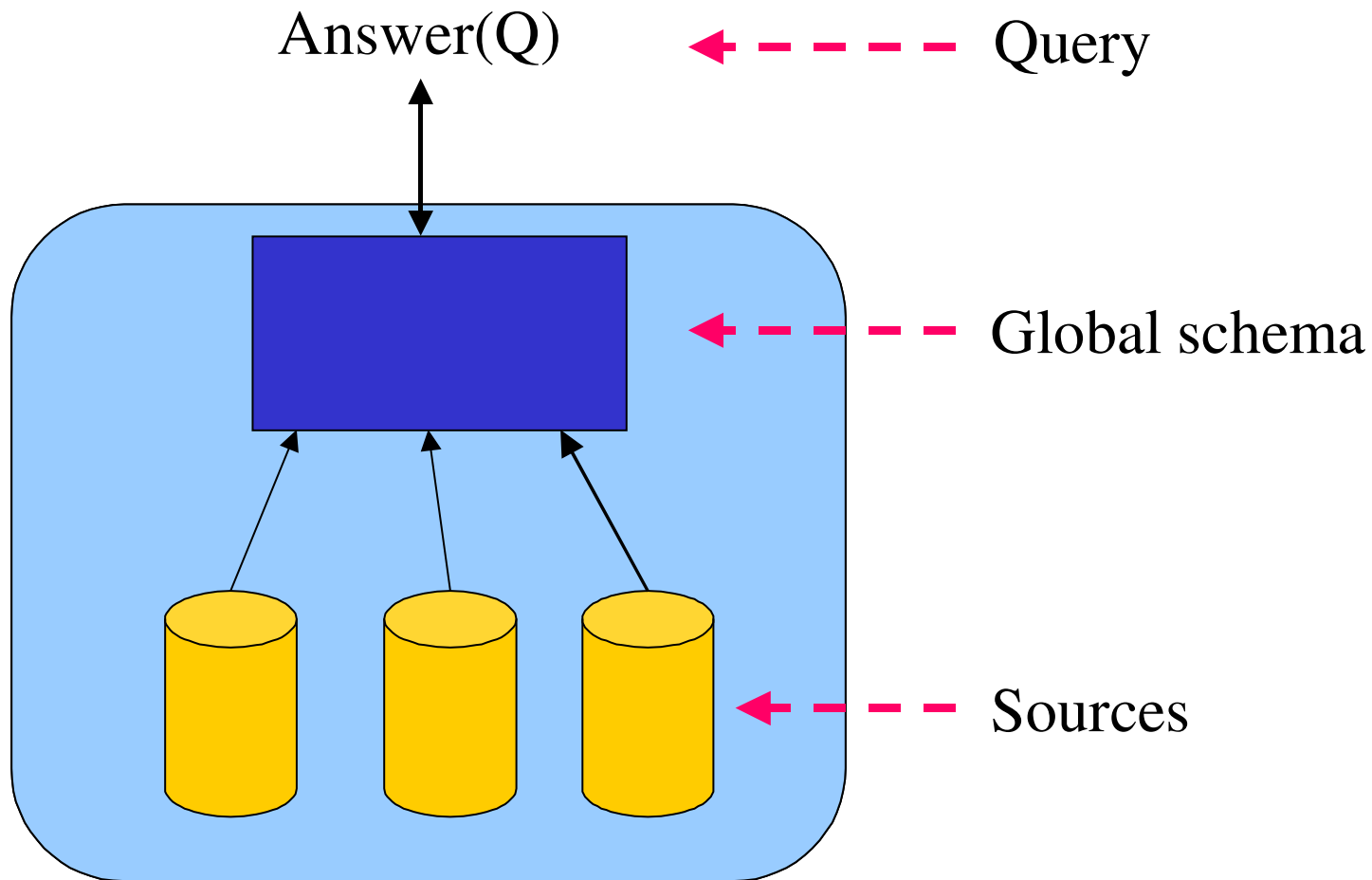
Dipartimento di Informatica e Sistemistica “Antonio Ruberti”

Università di Roma “La Sapienza”

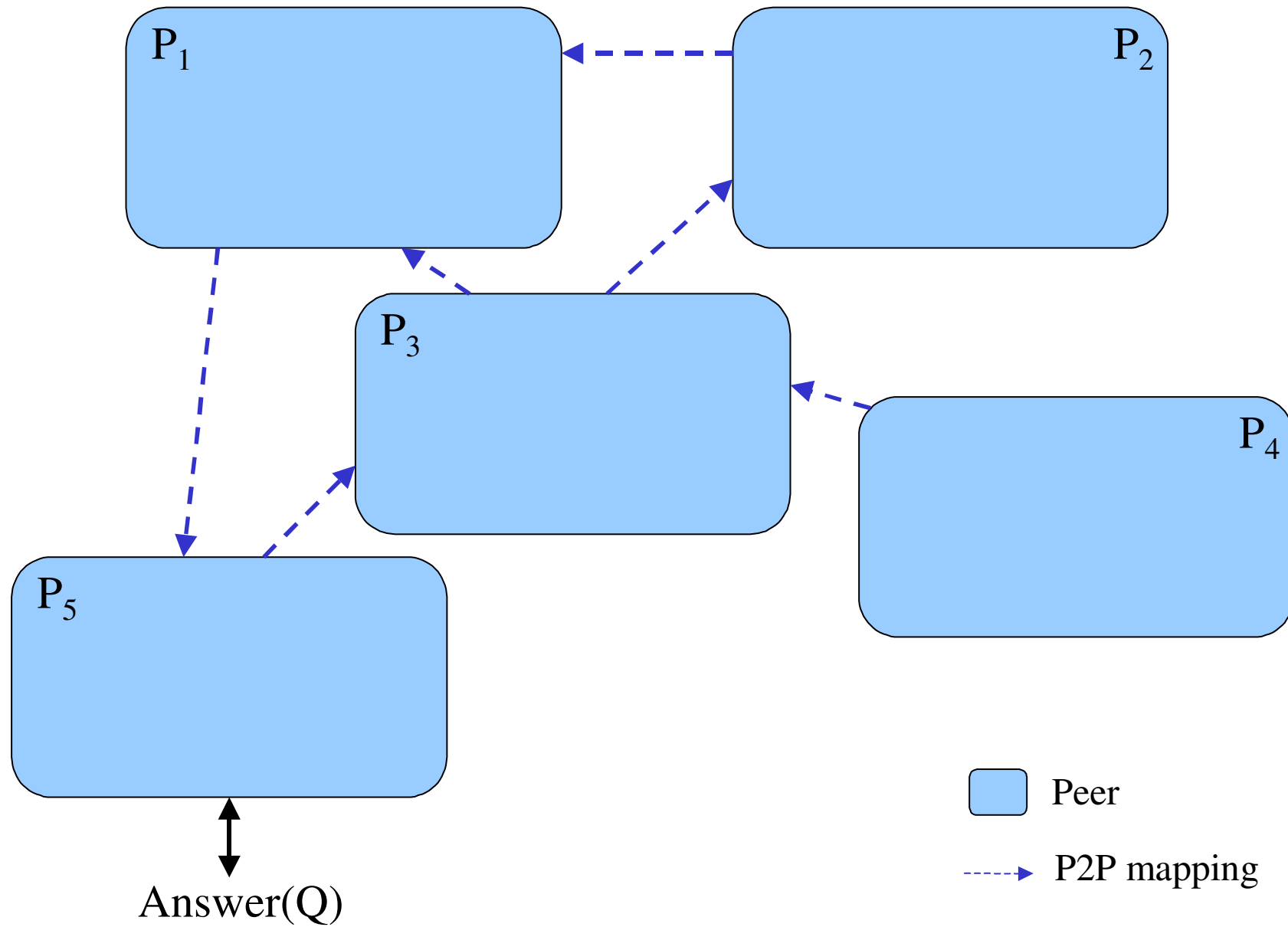
**International Workshop on
Databases, Information Systems, and Peer-to-peer computing**

Berlin, Germany – September 2003

Mediator-based data integration



P2P data integration



Objectives of our work

- Definition of a **general framework** for P2P data integration
 - Structure of one peer
 - Structure of the whole systems
- Definition of **semantics** for P2P data integration
 - Semantics of one peer
 - Semantics of the whole system
 - * based on first order logic
 - * new proposal based on **epistemic logic**
- Techniques for **query answering** in the epistemic semantics

P2P data integration: general framework

A **P2P system** $\Pi = (\mathcal{P}, \mathcal{M})$ is constituted by

- a **set \mathcal{P} of peers** $\{P_1, \dots, P_n\}$,

where each peer P_i models an autonomous information site, that

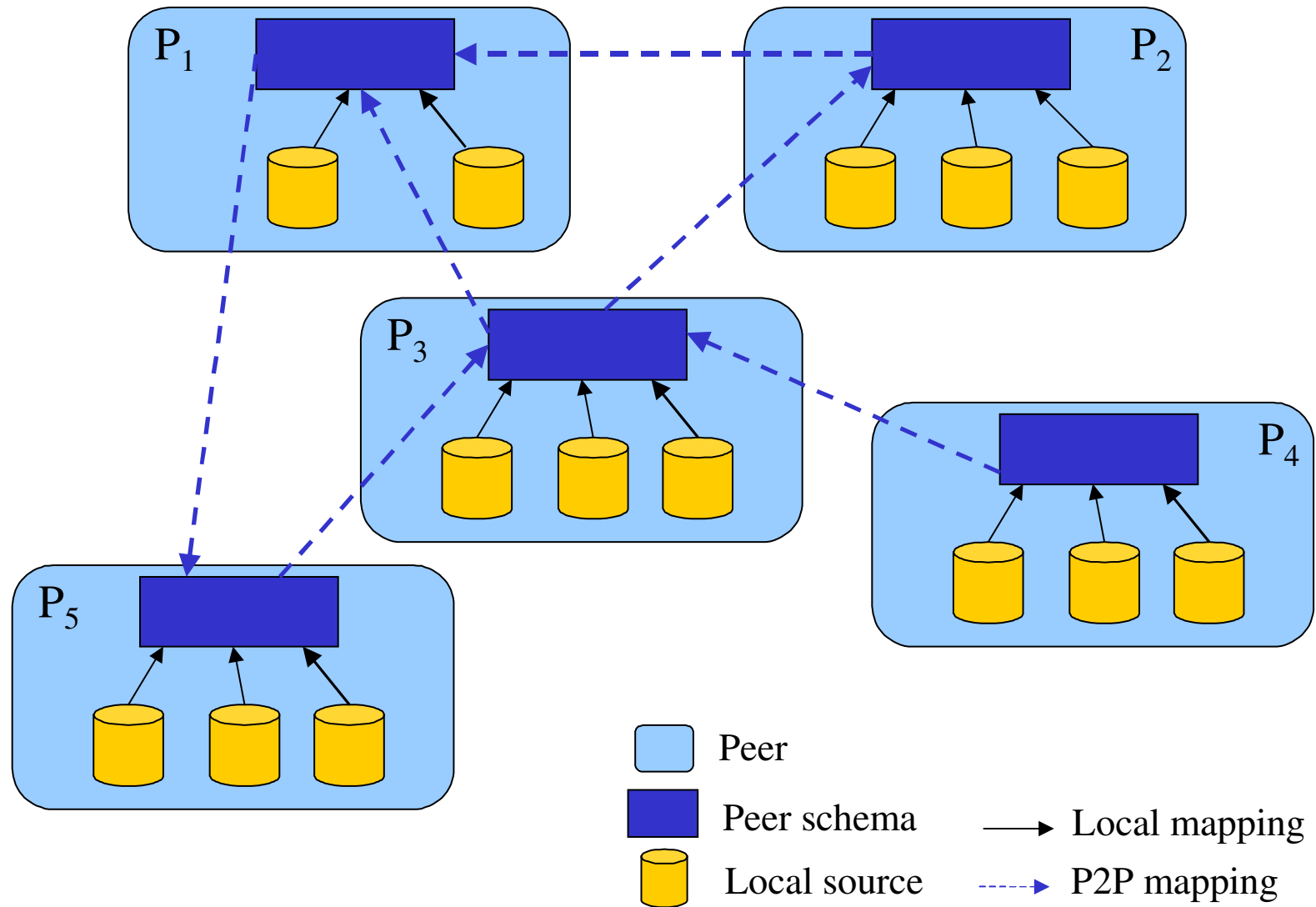
- exports its information content in terms of a schema, and
- stores actual data in a set of local sources

- a **set \mathcal{M} of P2P mappings**,

where each P2P mapping is a schema level assertion relating information between n peers (on one side) and one peer (on the other side)

Inspired by [Catarci&Lenzerini COOPIS '92], Halevy&al. ICDE'03]. Other related work: [Ghidini&Serafini FCS '98], [Bernstein&al. WebDB '02].

P2P data integration: general framework



Logic-based formal framework for P2P data integration

- Each **peer** P_i of Π is a triple (G_i, S_i, L_i) constituted by
 - a **schema** G_i , i.e., a set of FOL formulas over the peer alphabet A_{G_i}
 - a **set** S_i of **local sources** (finite relational alphabet)
 - a **set** L_i of **local GLAV mappings** from the sources S_i to the peer schema, each one of the form (ϕ_S and ϕ_G are conjunctions of atoms):

$$\exists \vec{z} \phi_S(\vec{x}, \vec{z}) \rightsquigarrow \exists \vec{y} \phi_G(\vec{x}, \vec{y})$$

- Each **P2P mapping** is an assertion of the form

$$q_1(\vec{x}) \rightsquigarrow q_2(\vec{x})$$

where

- q_1 is a FOL query over the union of the alphabets of the peers in \mathcal{P} ,
- q_2 is a FOL query over the alphabet of a single peer, and
- q_1 and q_2 are of the same arity

Formal framework for P2P data integration: semantics

- We refer to a fixed **infinite interpretation domain** Δ , **common to all peers**. We also refer to a fixed, infinite, denumerable, set Γ of constants, that act as *standard names*, i.e., Γ is isomorphic to the interpretation domain
- A **local source database** \mathcal{D} for Π is a database over Γ for the set L of all local source predicates in the various peers of Π
- A **global database for Π** is a database over Γ for the symbols in \mathcal{A}_{Π} , the alphabet of the union G of all peer schemas of Π (which are assumed to be pairwise disjoint)
- A global database for Π is said to be **legal wrt G** if it satisfies all peer schemas

Formal framework for P2P data integration: semantics

Given a local source database \mathcal{D} for Π , the **set of models of Π relative to \mathcal{D}** is:

$$sem^{\mathcal{D}}(\Pi) = \left\{ \mathcal{B} \mid \begin{array}{l} \mathcal{B} \text{ is a global database for } \Pi \text{ that is legal wrt } G, \text{ and} \\ \text{satisfies all local mapping assertions wrt } \mathcal{D}, \text{ and} \\ \text{satisfies all P2P mapping assertions} \end{array} \right\}$$

- \mathcal{B} satisfies a local mapping assertion $\exists \vec{z} \phi_S(\vec{x}, \vec{z}) \rightsquigarrow \exists \vec{y} \phi_i(\vec{x}, \vec{y})$ wrt \mathcal{D} if $(\exists \vec{z} \phi_S(\vec{x}, \vec{z}))^{\mathcal{D}} \subseteq (\exists \vec{y} \phi_i(\vec{x}, \vec{y}))^{\mathcal{B}}$
- the meaning of \mathcal{B} satisfying a P2P mapping assertion may vary in the various approaches

The set of **certain answers** to a query Q of arity n posed to a peer P of Π wrt the source database \mathcal{D} is the set

$$ans(Q^P, \Pi, \mathcal{D}) = \{ \vec{t} \in \Gamma^n \mid \forall \mathcal{B} \in sem^{\mathcal{D}}(\Pi) : \vec{t} \in Q^{\mathcal{B}} \}$$

Formalization of one peer

From the above definition, it follows that we are modeling each peer $P = (G, S, L)$ as a GLAV data integration system, in turn modeled as a **FOL theory** T_P :

- The **alphabet** of T_P is obtained as union of the alphabet of G and the alphabet of the local sources S of P ,
- The **formulas** of T_P are obtained as follows:
 - there is one formula of the form

$$\forall \vec{x} (\exists \vec{z} \phi_S(\vec{x}, \vec{z}) \rightarrow \exists \vec{y} \phi_S(\vec{x}, \vec{y}))$$

for each local mapping assertion $\exists \vec{z} \phi_S(\vec{x}, \vec{z}) \rightsquigarrow \exists \vec{y} \phi_i(\vec{x}, \vec{y})$

- T_P includes all the FOL formulas expressing the schema G .

Possible formalizations of P2P mapping

We consider two alternatives for specifying the semantics of P2P mappings:

- **Based on First Order Logic**

P2P mappings are considered as material logical implication

- **New proposal based on Epistemic Logic**

P2P mappings are considered as specifications of exchange of certain answers

First order logic semantics of P2P mappings

According to most approaches (see [Halevy&al. ICDE'03], [Bernstein&al. WebDB '02]), the semantics of P2P mapping assertions in a P2P system Π is given in terms of **first order logic** (FOL), where satisfaction of a P2P mapping assertion

$$q_1(\vec{x}) \rightsquigarrow q_2(\vec{x})$$

of Π by a global database \mathcal{B} means

- satisfaction of the FOL formula

$$\forall \vec{x} q_1(\vec{x}) \rightarrow q_2(\vec{x})$$

- which is equivalent to the condition

$$q_1^{\mathcal{B}} \subseteq q_2^{\mathcal{B}}$$

First order logic semantics of P2P mappings

We claim that the FOL semantics is not adequate for P2P data integration, because

- The system is modeled by a flat FOL theory, with no formal separation between the various peers
- The modular structure of the system is not reflected in the semantics
- Bad computational properties: computing the set of certain answers to a conjunctive query Q posed to a peer is **undecidable** (see [Halevy&al. ICDE'03], [Koch FOIKS'02]), even for **simple P2P systems**, (i.e., when all peer schemas are empty, and P2P mappings are conjunctive, see later)
- In order to recover decidability, one has to limit the expressive power of P2P mappings (e.g., acyclicity of P2P mappings is assumed in [Halevy&al. ICDE'03])

Epistemic semantics for P2P mappings: objectives

We propose a new semantics for P2P mappings, with the following aims:

- We want to take into account that peers in our context are to be considered **autonomous sites**, that exchange information
- We do not want to limit a-priori the **topology** of the mapping assertions among the peers in the system
- We seek for a semantic characterization that leads to a setting where query answering is decidable, and possibly, **polynomially tractable**

Epistemic semantics for P2P mappings: basic idea

The new semantics is based on **epistemic logic** [Reiter TARK '88]

- A P2P mapping $q_1(\vec{x}) \rightsquigarrow q_2(\vec{x})$ is interpreted as an epistemic formula which imposes that **only the certain answers** to $q_1(\vec{x})$ in the peers i_1, \dots, i_m are transferred to peer j as facts satisfying q_2 (peers i_1, \dots, i_m communicate to peer j only facts that are certain, i.e., true in every model of the P2P system)
- The modular structure of the system is now reflected in the semantics (by virtue of the modal semantics of epistemic logics)
- Good computational properties: for **simple P2P systems**, computing the set of certain answers to a conjunctive query Q wrt a local source database \mathcal{D} is not only **decidable**, but also **polynomial time** in the size of \mathcal{D} , even for cyclic mappings

Epistemic logic: basic notions

In epistemic logic, we have a new form of atoms, namely (α is again a formula):

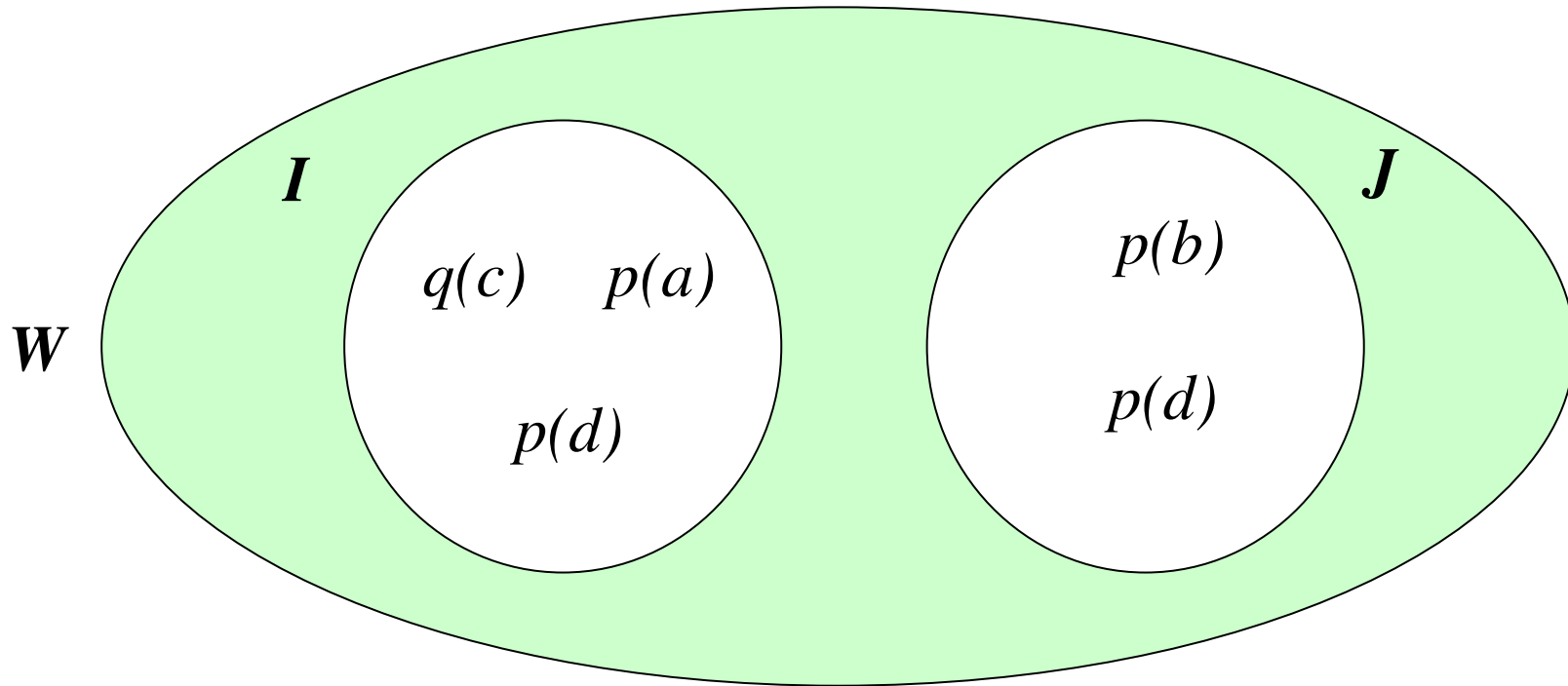
$$\mathbf{K} \alpha$$

An epistemic interpretation \mathcal{E} is a pair $(\mathcal{I}, \mathcal{W})$, where \mathcal{I} is a FOL interpretation, \mathcal{W} is a set of FOL interpretations, and $\mathcal{I} \in \mathcal{W}$.

- a FOL formula constituted by an atom $f(\vec{x})$ is satisfied in $(\mathcal{I}, \mathcal{W})$ by the tuples \vec{t} of constants such that $f(\vec{t})$ is true in \mathcal{I}
- an atom of the form $\mathbf{K}\alpha(\vec{x})$ is satisfied in $(\mathcal{I}, \mathcal{W})$ by the tuples \vec{t} of constants such that $\alpha(\vec{t})$ is satisfied in all the pairs $(\mathcal{J}, \mathcal{W})$ such that $\mathcal{J} \in \mathcal{W}$

An **epistemic model** of an epistemic logic theory $\{\phi_1, \dots, \phi_t\}$ (finite set of global axioms) is an epistemic interpretation $(\mathcal{I}, \mathcal{W})$ that satisfies every axiom of the theory, i.e., such that for every $\mathcal{J} \in \mathcal{W}$, each ϕ_i is satisfied by $(\mathcal{J}, \mathcal{W})$.

Epistemic logic: examples

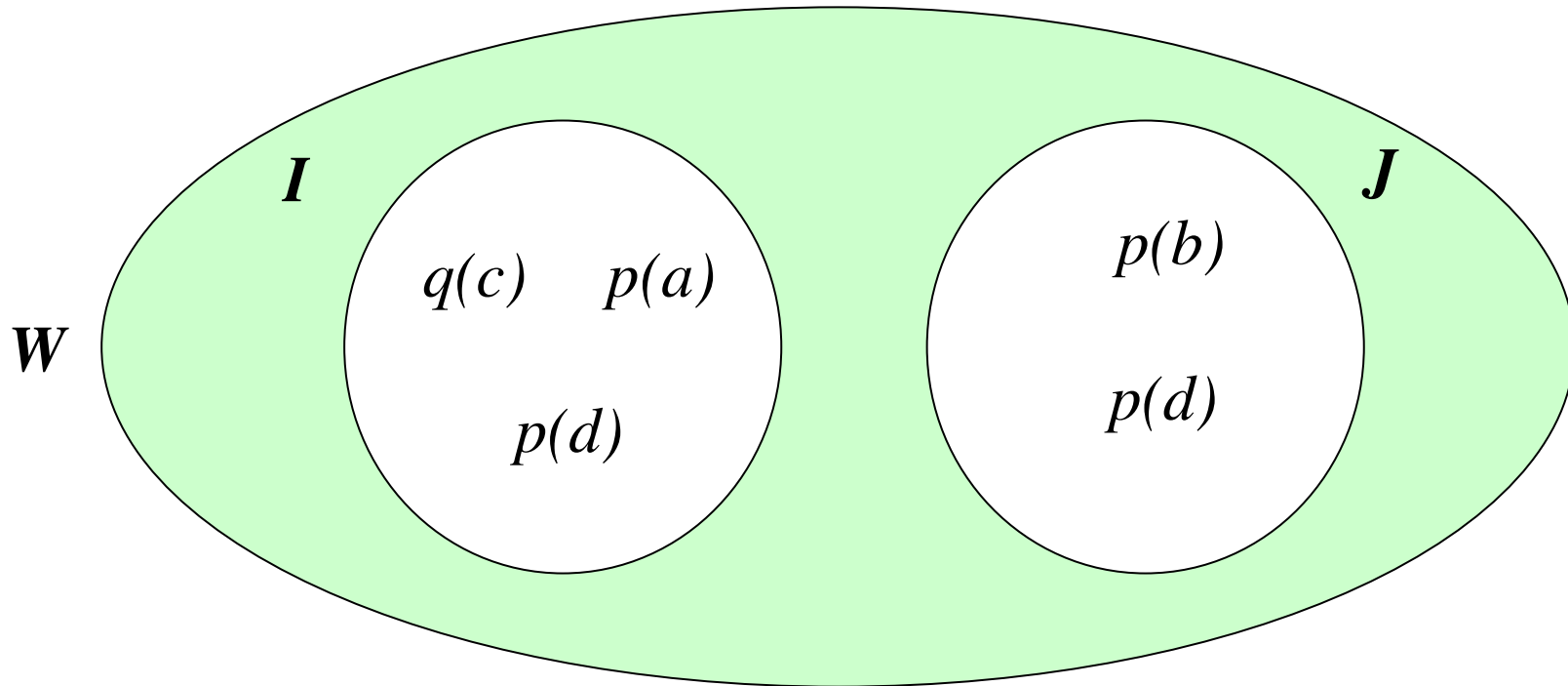


$$(I, W) \models q(c)$$

$$(J, W) \not\models q(c)$$

$$(I, W) \not\models \mathbf{K} q(c)$$

Epistemic logic: examples



$$(\mathcal{I}, \mathcal{W}) \models \mathbf{K} (p(a) \vee p(b))$$

$$(\mathcal{I}, \mathcal{W}) \not\models (\mathbf{K} p(a)) \vee (\mathbf{K} p(b))$$

$$(\mathcal{J}, \mathcal{W}) \models \mathbf{K} p(d)$$

Epistemic semantics for P2P mappings: basic idea

We formalize a P2P system $\Pi = (\mathcal{P}, \mathcal{M})$ in terms of the **epistemic logic theory** E_Π :

- the alphabet \mathcal{A}_Π is the disjoint union of the alphabets of the various peer theories, one corresponding to one peer in \mathcal{P}
- all the formulas of the various theories T_P are axioms in E_Π ,
- there is one axiom in E_Π of the form

$$\forall \vec{x} \ ((\mathbf{K} q_1(\vec{x})) \rightarrow q_2(\vec{x}))$$

for each P2P mapping assertion $q_1(\vec{x}) \rightsquigarrow q_2(\vec{x})$ in \mathcal{M} .

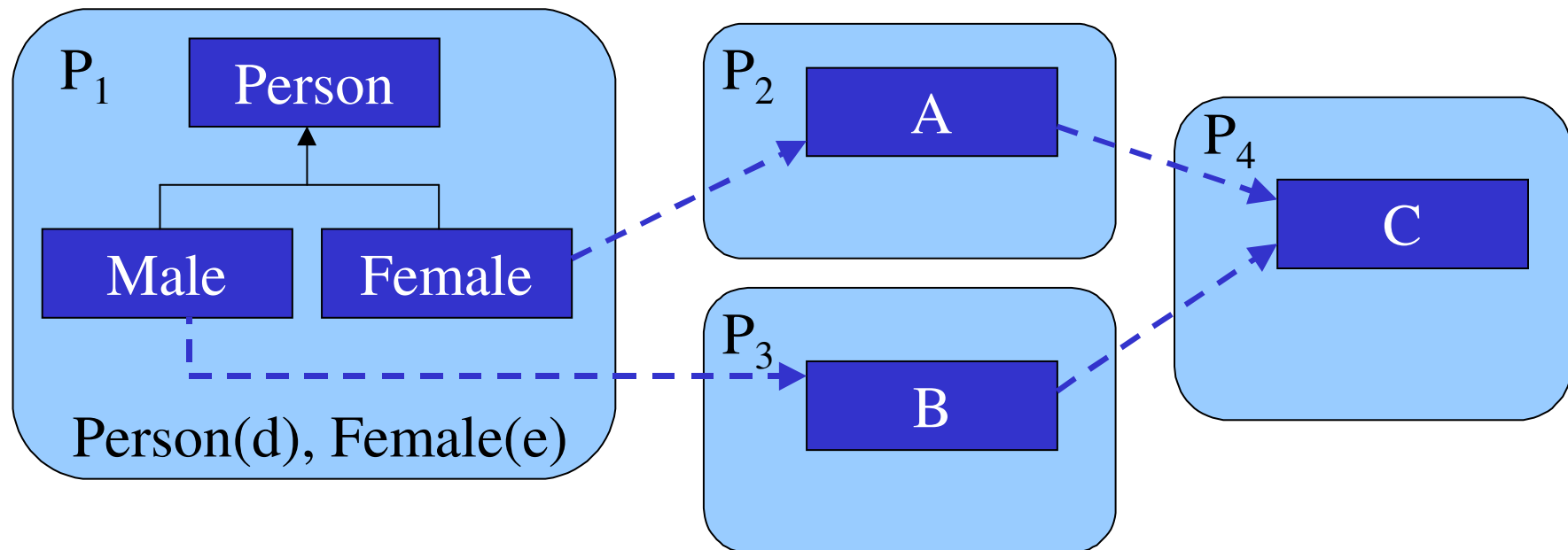
Epistemic semantics for P2P mappings: basic idea

An epistemic interpretation $(\mathcal{I}, \mathcal{W})$ for Π based on \mathcal{D} satisfies the P2P mapping assertion $q_1(\vec{x}) \rightsquigarrow q_2(\vec{x})$ if, for every $(\mathcal{J}, \mathcal{W})$ such that $\mathcal{J} \in \mathcal{W}$, for every tuple \vec{t} of objects in Γ , the fact that $q_1(\vec{t})$ is satisfied in every FOL models in \mathcal{W} implies that $q_2(\vec{t})$ is satisfied in \mathcal{J} .

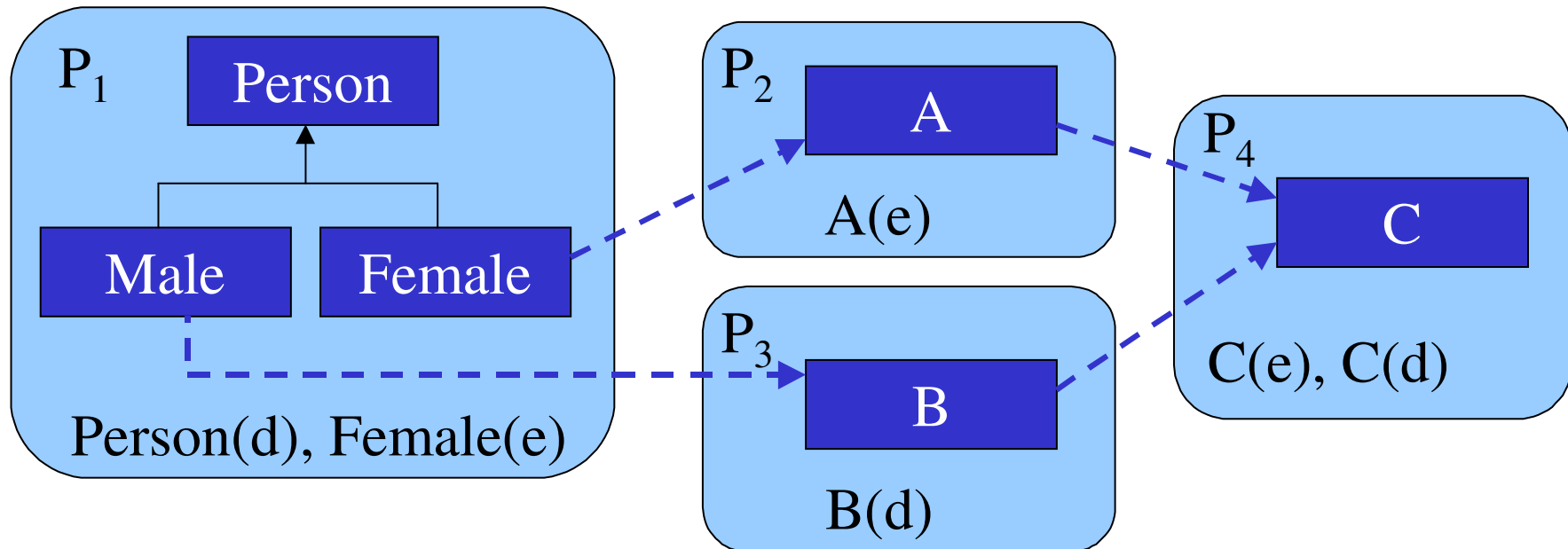
An **epistemic model** for $\Pi = (\mathcal{P}, \mathcal{M})$ based on \mathcal{D} is any epistemic interpretation for Π based on \mathcal{D} that satisfies all the axioms corresponding to the P2P mapping assertions in \mathcal{M} .

Let q be a query over one peer of Π . The certain answer $ans_k(q, \Pi, \mathcal{D})$ to q in Π based on \mathcal{D} is the set of tuples \vec{t} of objects in Γ such that $q(\vec{t})$ is satisfied in every epistemic model $(\mathcal{I}, \mathcal{W})$ of Π based on \mathcal{D} .

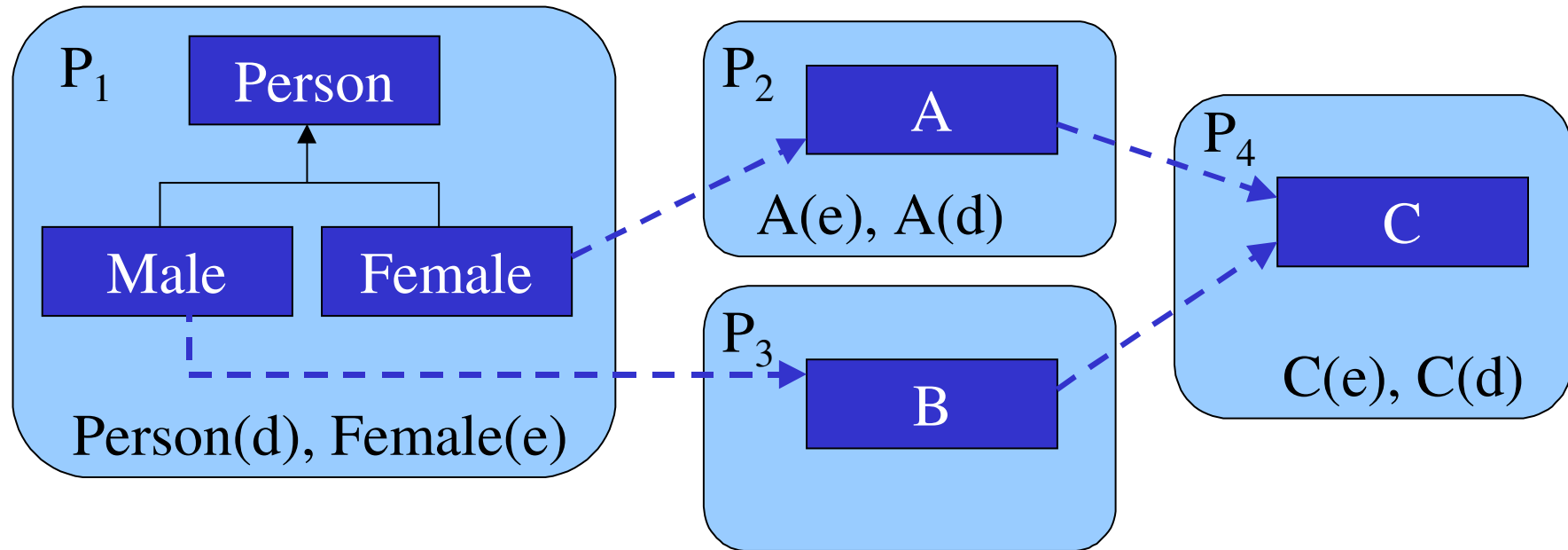
Semantics of P2P mappings: example



FOL Semantics of P2P mappings: model 1

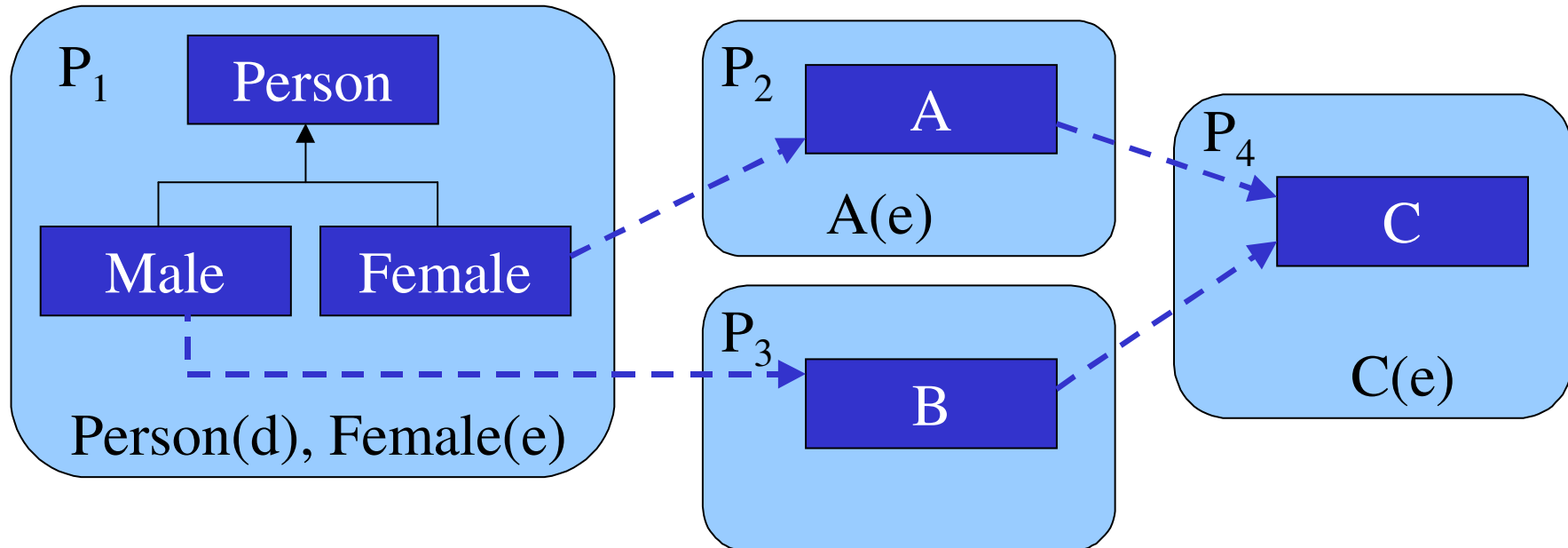


FOL Semantics of P2P mappings: model 2



According to the FOL semantics, $C(d)$ is a true in all cases, and therefore is a certain answer.

Epistemic Semantics of P2P mappings



According to the epistemic semantics, $C(d)$ is not a certain answer.

Query answering for simple P2P systems

We call a P2P system $\Pi = (\mathcal{P}, \mathcal{M})$ **simple** if it satisfies the following restrictions:

1. **peer theories are empty**, i.e., each peer schema of Π simply consists of a relational alphabet
2. P2P mapping assertions in \mathcal{M} are expressed using **conjunctive queries**, i.e., a P2P mapping assertion is an expression of the form $q_1(\vec{x}) \rightsquigarrow q_2(\vec{x})$, where q_1 and q_2 are conjunctive queries of the same arity, q_1 is expressed over the union of the alphabets of the peers, and q_2 is expressed over the alphabet of a single peer
3. the language for querying the P2P system is **union of conjunctive queries (UCQ)**, i.e., a query over a P2P system is a UCQ over the alphabet of a single peer

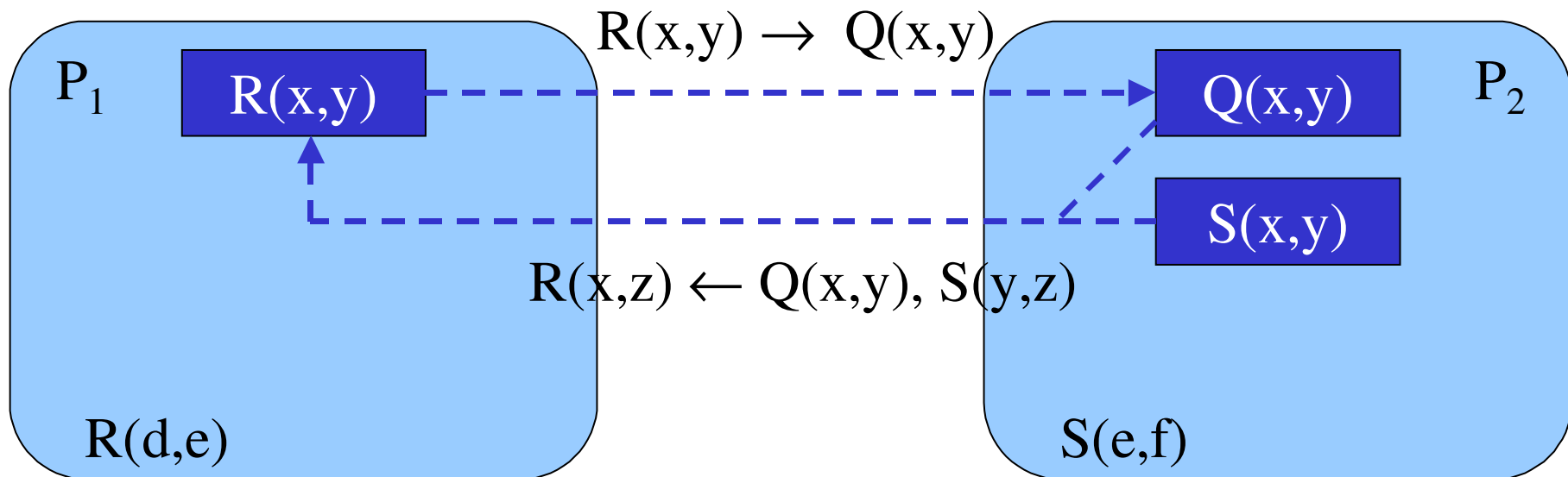
Query answering for simple P2P systems

We have proved that, given an UCQ q posed to a simple P2P system S , and given a source database \mathcal{D} for Π , one can construct a **finite database RDB** on the alphabet \mathcal{A}_Π that is the union of the alphabet of the peer schemas in Π , such that for each tuple \vec{t} of constants in Γ , $\vec{t} \in \text{ans}_k(q, \Pi, \mathcal{D})$ if and only if $\vec{t} \in q^{RDB}$.

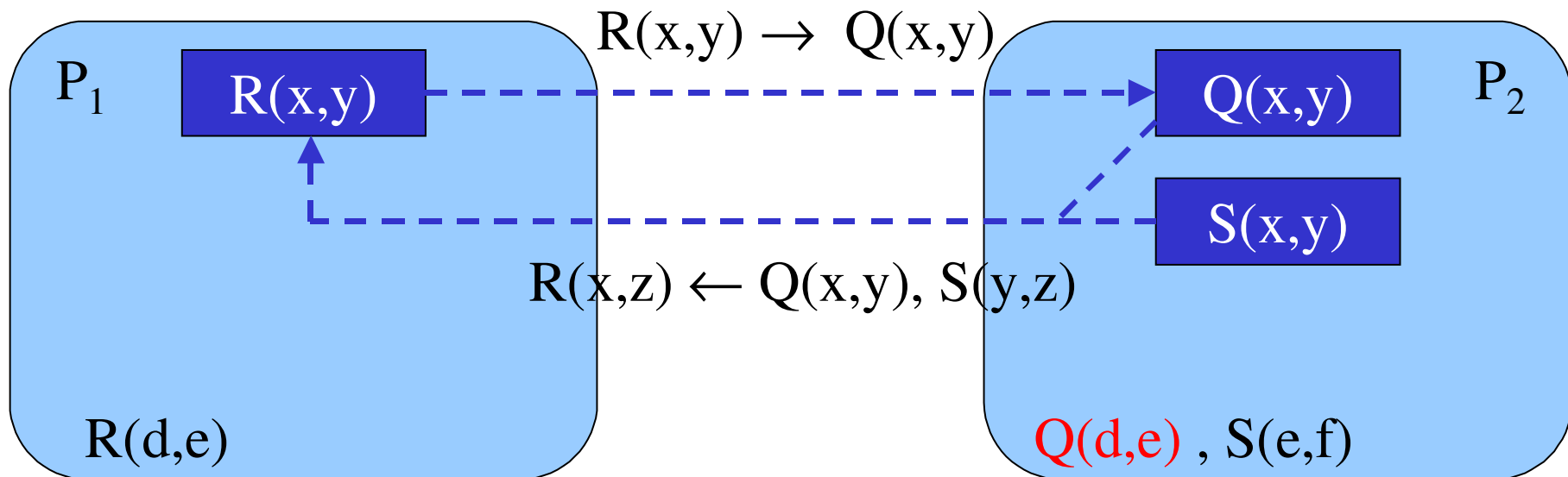
Intuitively, such a finite database *RDB* constitutes a “representative” of all the epistemic models for Π based on \mathcal{D} with respect to the query q .

The size of *RDB* is polynomial in the size of source database \mathcal{D} , and the whole process of query answering has polynomial time complexity in the size of source database \mathcal{D} .

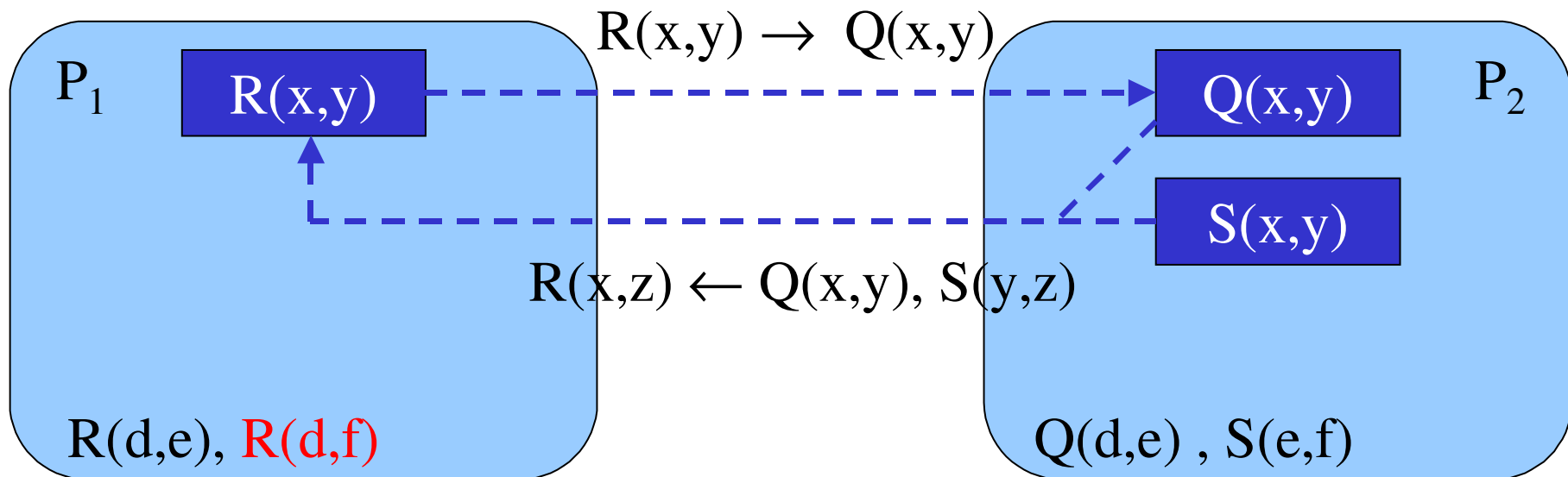
Query answering for simple P2P systems: example



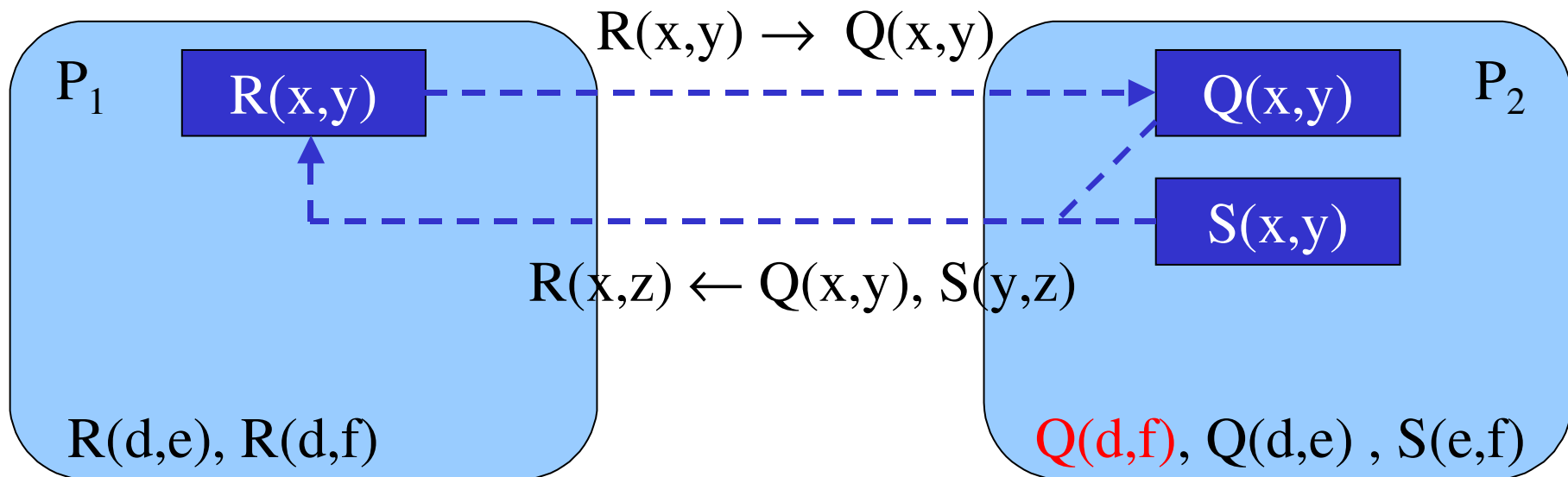
Query answering for simple P2P systems: example



Query answering for simple P2P systems: example



Query answering for simple P2P systems: example



Query answering for simple P2P systems

Algorithm answer(Π, \mathcal{D}, q)

Input simple P2P system $\Pi = (\mathcal{P}, \mathcal{M})$, with $\mathcal{P} = \{P_1, \dots, P_n\}$,
source database \mathcal{D} for Π , UCQ q over one peer P_i

Output set of tuples of objects in Γ

begin

$RDB \leftarrow \emptyset$;

for $i = 1, \dots, n$ **do** $RDB \leftarrow$ retrieve-data-by-local-mapping($\Pi, P_i, \mathcal{D}, RDB$);

repeat

$RDB' \leftarrow RDB$;

$RDB \leftarrow$ retrieve-data-by-P2P-mapping(\mathcal{M}, RDB)

until $RDB' = RDB$;

return q^{RDB}

end

Current work

- A **new algorithm** driven by the query and the structure of mappings
- More expressive peer schemas, by adding **integrity constraints** to simple P2P systems, in the line of [Calì&al. Information Systems '03]
- Dealing with **inconsistencies** between peers, in the line of [Calì&al. PODS'03]
- Dealing with peers with **limited query capabilities**
- Dealing with **different vocabularies** of constants in different peers in the line of [Bernstein&al. WebDB '02]
- **Experiments** within the **Infomix European project**, and the **Sewasie European project**