Quality-aware peer-to-peer data integration

Maurizio Lenzerini

Dipartimento di Informatica e Sistemistica “Antonio Ruberti”

Università di Roma “La Sapienza”

Joint work with D. Calvanese, G. De Giacomo, D. Lembo, R. Rosati

Invited talk at IQIS 2004

Paris, France – June 18, 2004

Outline

- Data integration architectures
- Centralized data integration
- P2P data integration: the framework
- Basic semantics for P2P data integration
- Quality-aware semantics for P2P data integration
- Conclusions

Outline

- Data integration architectures
- Centralized data integration
- P2P data integration: the framework
- Basic semantics for P2P data integration
- Quality-aware semantics for P2P data integration
- Conclusions

Three data integration architectures

- Centralized data integration

  The traditional architecture for centralized, virtual data integration

- Data exchange

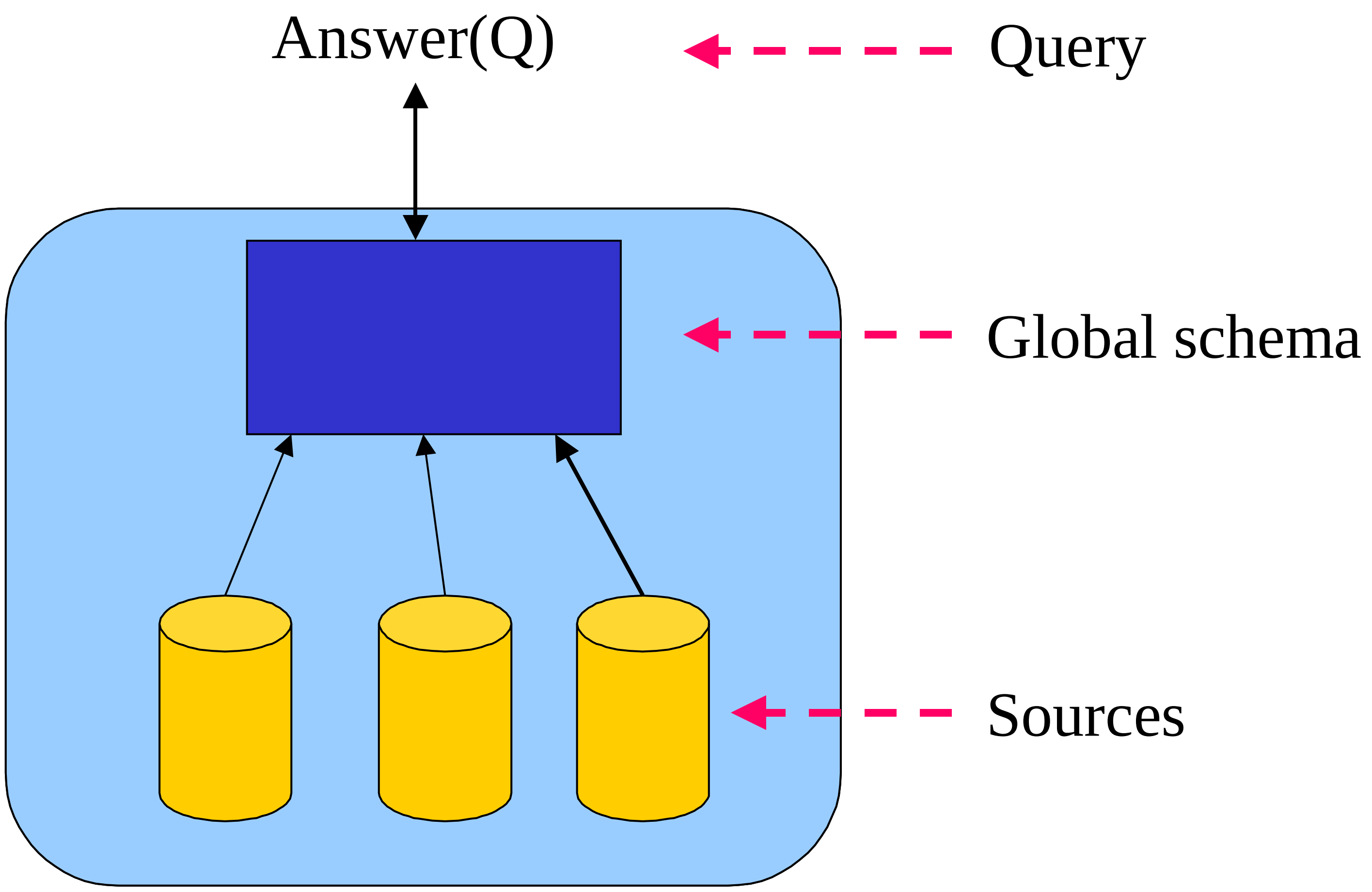
  Materialization of data from a source database to a target database

- Peer-to-peer data integration

  Decentralized, dynamic, data-centric coordination between autonomous organizations

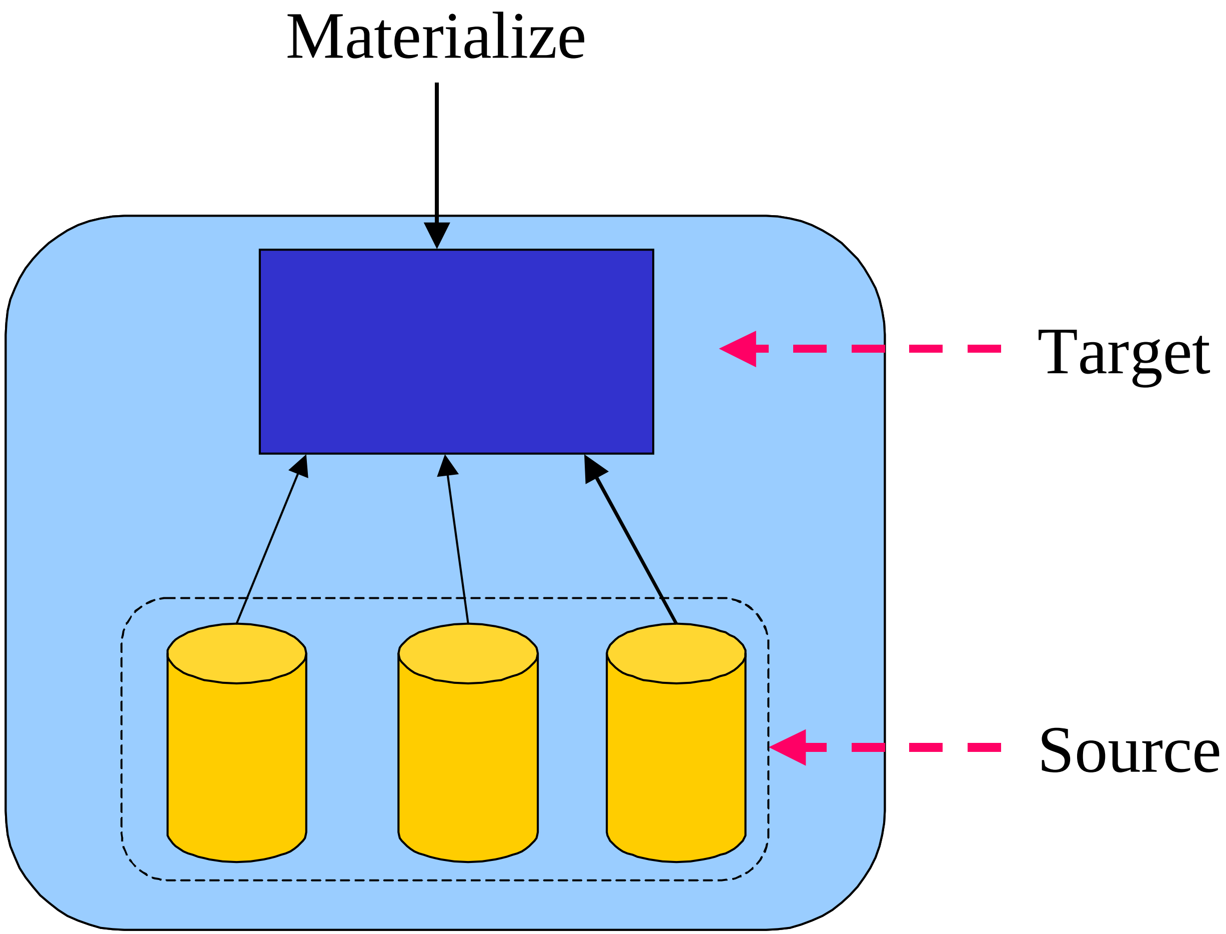
Centralized data integration

- Mapping between sources and global schema
- Queries over the global schema



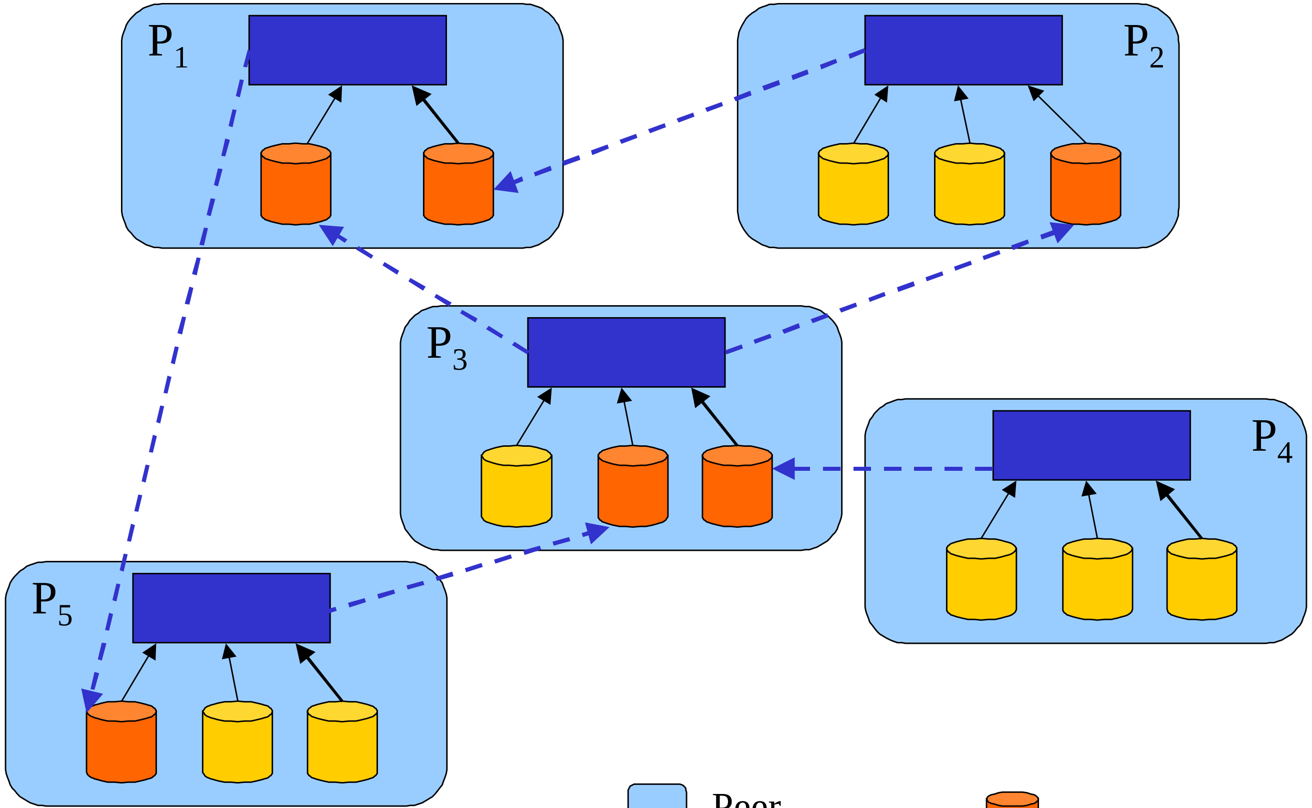
Data exchange

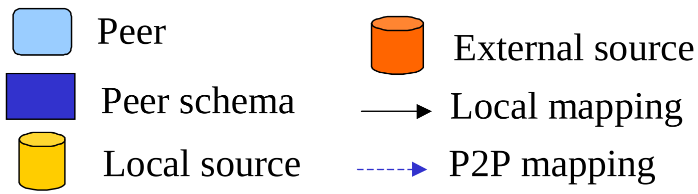
- Mapping between sources and target schema
- Materialization according to the target schema



Peer-to-peer data integration

- Several peers
- Local mappings and P2P mappings
- Each query over one peer
- Dynamic mappings





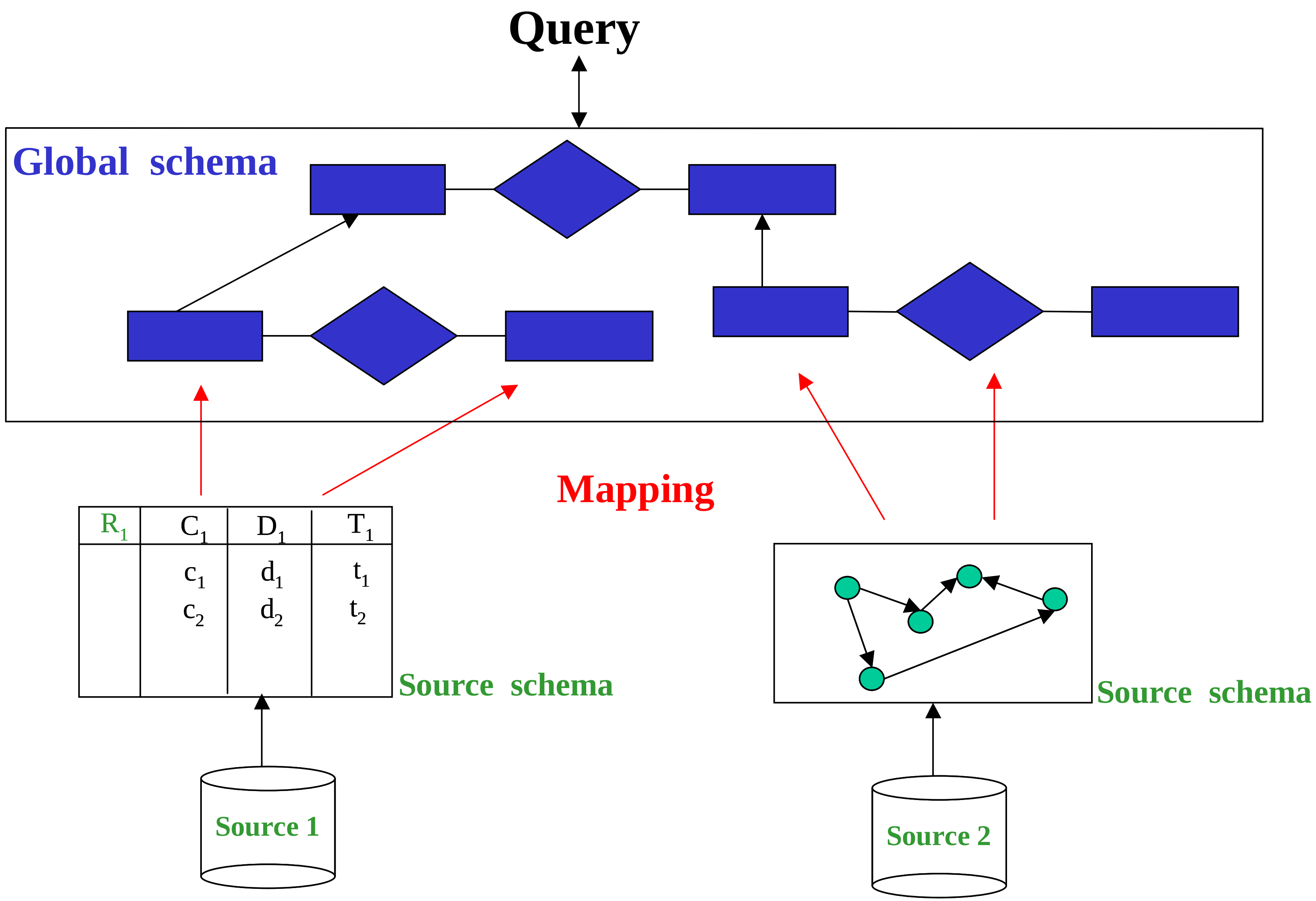
Operations:

- Answer(Q, P_i)
- Materialize(P_i)

Outline

- Data integration architectures
- Centralized data integration
- P2P data integration: the framework
- Basic semantics for P2P data integration
- Quality-aware semantics for P2P data integration
- Conclusions

Centralized data integration



Formal framework for data integration

A data integration system $\mathcal{I}$ is a triple $\langle \mathcal{G}, \mathcal{S}, \mathcal{M} \rangle$, where

- $\mathcal{G}$ is the global schema

  The global schema is a logical theory over an alphabet $\mathcal{A}_{\mathcal{G}}$

- $\mathcal{S}$ is the source schema

  The source schema is constituted simply by an alphabet $\mathcal{A}_{\mathcal{S}}$ disjoint from $\mathcal{A}_{\mathcal{G}}$

- $\mathcal{M}$ is the mapping between $\mathcal{S}$ and $\mathcal{G}$

  Different approaches to the specification of mapping

Semantics of a data integration system

Which are the databases that satisfy $\mathcal{I}$, i.e., which are the logical models of $\mathcal{I}$?

The databases that satisfy $\mathcal{I}$ are logical interpretations for $\mathcal{A}_{\mathcal{G}}$ (called global databases). We refer only to databases over a fixed infinite domain Γ of constants.

Let $\mathcal{C}$ be a source database over Γ (also called source model), fixing the extension of the predicates of $\mathcal{A}_{\mathcal{S}}$ (thus modeling the data present in the sources).

The set of models of (i.e., databases for $\mathcal{A}_{\mathcal{G}}$ that satisfy) $\mathcal{I}$ relative to $\mathcal{C}$ is:

$$sem^{\mathcal{C}}(\mathcal{I}) = \{ \mathcal{B} \mid \mathcal{B} \text{ is a global database that is legal wrt } \mathcal{G} \text{ and satisfies } \mathcal{M} \text{ wrt } \mathcal{C} \}$$

What it means to satisfy $\mathcal{M}$ wrt $\mathcal{C}$ depends on the nature of the mapping $\mathcal{M}$.

Semantics of queries to $\mathcal{I}$

A query q of arity n is a formula with n free variables.

If $\mathcal{D}$ is a database, then $q^{\mathcal{D}}$ denotes the extension of q in $\mathcal{D}$ (i.e., the set of n-tuples that are valuations in Γ for the free variables of q that make q true in $\mathcal{D}$).

If q is a query of arity n posed to a data integration system $\mathcal{I}$ (i.e., a formula over $\mathcal{A}_{\mathcal{G}}$ with n free variables), then the set of certain answers to q wrt $\mathcal{I}$ and $\mathcal{C}$ is

$$ans(q, \mathcal{I}, \mathcal{C}) = \{(c_1, \ldots, c_n) \in q^{\mathcal{B}} \mid \forall \mathcal{B} \in sem^{\mathcal{C}}(\mathcal{I})\}.$$

Note: query answering is logical implication.

Note: complexity will be mainly measured wrt the size of the source database $\mathcal{C}$, and will refer to the problem of deciding whether $\vec{\mathbf{c}} \in ans(q, \mathcal{I}, \mathcal{C})$, for a given $\vec{\mathbf{c}}$.

Databases with incomplete information, or Knowledge Bases

- Traditional database: one model of a first-order theory

  Query answering means evaluating a formula in the model

- Database with incomplete information, or Knowledge Base: set of models (specified, for example, as a restricted first-order theory)

  Query answering means computing the tuples that satisfy the query in all the models in the set

There is a strong connection between query answering in data integration and query answering in databases with incomplete information under constraints (or, query answering in knowledge bases).

The mapping

How is the mapping $\mathcal{M}$ between $\mathcal{S}$ and $\mathcal{G}$ specified?

- Are the sources defined in terms of the global schema?

  Approach called source-centric, or local-as-view, or LAV

- Is the global schema defined in terms of the sources?

  Approach called global-schema-centric, or global-as-view, or GAV

- A mixed approach?

  Approach called GLAV

Beyond GAV and LAV: GLAV

In GLAV (with sound sources), the mapping $\mathcal{M}$ is constituted by a set of assertions:

$$\phi_{\mathcal{S}} \quad \leadsto \quad \phi_{\mathcal{G}}$$

where $\phi_{\mathcal{S}}$ is a query over $\mathcal{S}$, and $\phi_{\mathcal{G}}$ is a query over $\mathcal{G}$ of the arity $\phi_{\mathcal{S}}$.

Given source database $\mathcal{C}$, a database $\mathcal{B}$ that is legal wrt $\mathcal{G}$ satisfies $\mathcal{M}$ wrt $\mathcal{C}$ if for each assertion in $\mathcal{M}$:

$$\phi_{\mathcal{S}}{}^{\mathcal{C}} \quad \subseteq \quad \phi_{\mathcal{G}}{}^{\mathcal{B}}$$

In other words, the assertion means $\quad \forall \vec{\mathbf{x}} \left(\phi_{\mathcal{S}}(\vec{\mathbf{x}}) \rightarrow \phi_{\mathcal{G}}(\vec{\mathbf{x}}) \right)$.

The mapping $\mathcal{M}$ does not provide direct information about which data satisfy the global schema: to answer a query q over $\mathcal{G}$, we have to infer how to use $\mathcal{M}$ in order to access the source database $\mathcal{C}$.

Example of GLAV

Global schema: $Work(Person, Project)$, $Area(Project, Field)$

Source 1: $HasJob(Person, Field)$

Source 2: $Teach(Professor, Course)$, $In(Course, Field)$

Source 3: $Get(Researcher, Grant)$, $For(Grant, Project)$

GLAV mapping:

$$\{ (r,f) \mid HasJob(r,f) \} \leadsto \{ (r,f) \mid Work(r,p) \wedge Area(p,f) \}$$

$$\{ (r,f) \mid Teach(r,c) \wedge In(c,f) \} \leadsto \{ (r,f) \mid Work(r,p) \wedge Area(p,f) \}$$

$$\{ (r,p) \mid Get(r,g) \wedge For(g,p) \} \leadsto \{ (r,p) \mid Work(r,p) \}$$

Query answering in different approaches

The problem of query answering comes in different forms, depending on several parameters:

- Global schema
  - without constraints (i.e., empty theory)
  - with constraints
- Mapping
  - GAV
  - LAV
  - GLAV
- Queries
  - client queries
  - queries in the mapping

LAV without constraints: basic technique

Consider conjunctive queries and conjunctive views.

$$\mathsf{r}_1(T) \quad \leadsto \quad \{\ (T) \mid \mathsf{movie}(T,Y,D) \wedge \mathsf{european}(D)\ \}$$

$$\mathsf{r}_2(T,V) \quad \leadsto \quad \{\ (T,V) \mid \mathsf{movie}(T,Y,D) \wedge \mathsf{review}(T,V)\ \}$$

$$Q(X,Y) \quad \leftarrow \quad \mathsf{movie}(X,1990,D) \wedge \mathsf{review}(X,Y) \wedge \mathsf{european}(D)$$

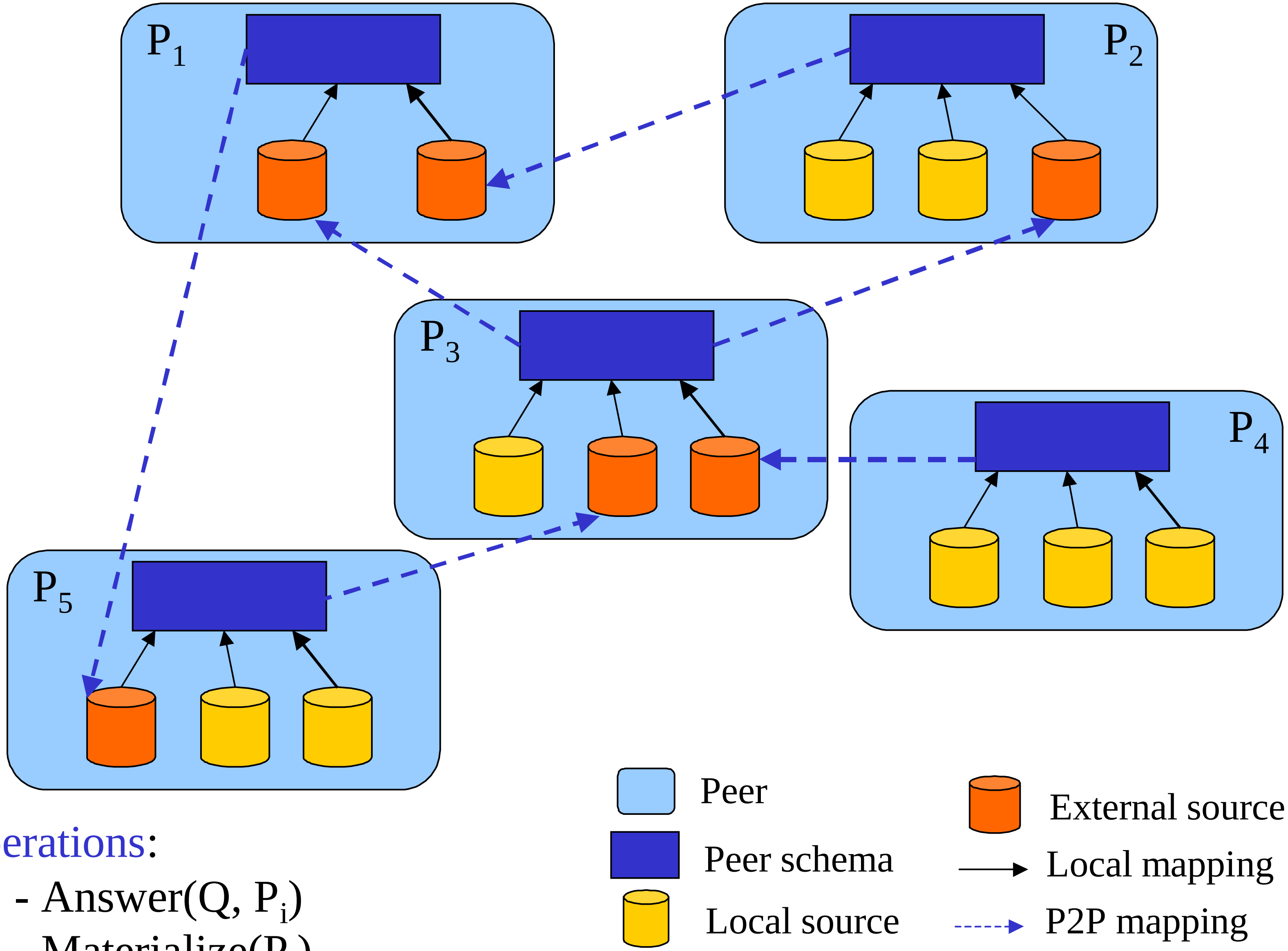
$$\begin{array}{rcl}
\mathsf{movie}(T,f_1(T),f_2(T)) & \leftarrow & \mathsf{r}_1(T) \\
\mathsf{european}(f_2(T)) & \leftarrow & \mathsf{r}_1(T) \\
\mathsf{movie}(T,f_4(T,V),f_5(T,V)) & \leftarrow & \mathsf{r}_2(T,V) \\
\mathsf{review}(T,V)) & \leftarrow & \mathsf{r}_2(T,V)
\end{array}$$

Answering query Q means evaluating the goal Q wrt to this nonrecursive logic program, i.e., this logic program is a perfect reformulation (or perfect rewriting).

Outline

- Data integration architectures
- Centralized data integration
- P2P data integration: the framework
- Basic semantics for P2P data integration
- Quality-aware semantics for P2P data integration
- Conclusions

P2P data integration: the framework



Operations:

- Answer(Q, P_i)
- Materialize(P_i)

P2P data integration: the framework

A P2P system Π is a set $\{P_1, \ldots, P_n\}$ of peers, where each peer $P_i = (G, S, L, M)$ models an autonomous information site, that

- exports its information content in terms of a peer schema G
- represents its data as a set of sources S (local sources model its own data, and external sources model data coming from other peers)
- relates sources to global schema by means of local mappings L
- is related to other peers in Π by means of a set of P2P mappings M, where each P2P mapping is a schema level assertion relating data coming from another peer P_j to one external source in P_i

Inspired by [Catarci&Lenzerini COOPIS '92], Halevy&al. ICDE'03]. Other related work: [Ghidini&Serafini FCS '98], [Bernstein&al. WebDB '02], [Franconi&al. P2PDBIS '03].

P2P data integration: local and P2P mappings

In a peer $\Pi = (G, S, L, M)$

- each local mapping in L has the form

$$\textit{ep}_S \leadsto cq_G$$

  where $\textit{ep}_S$ is an extraction program on the sources S and cq_G is a conjunctive queries over G, respectively

- each P2P mapping asserion in M has the form

$$cq \leadsto s$$

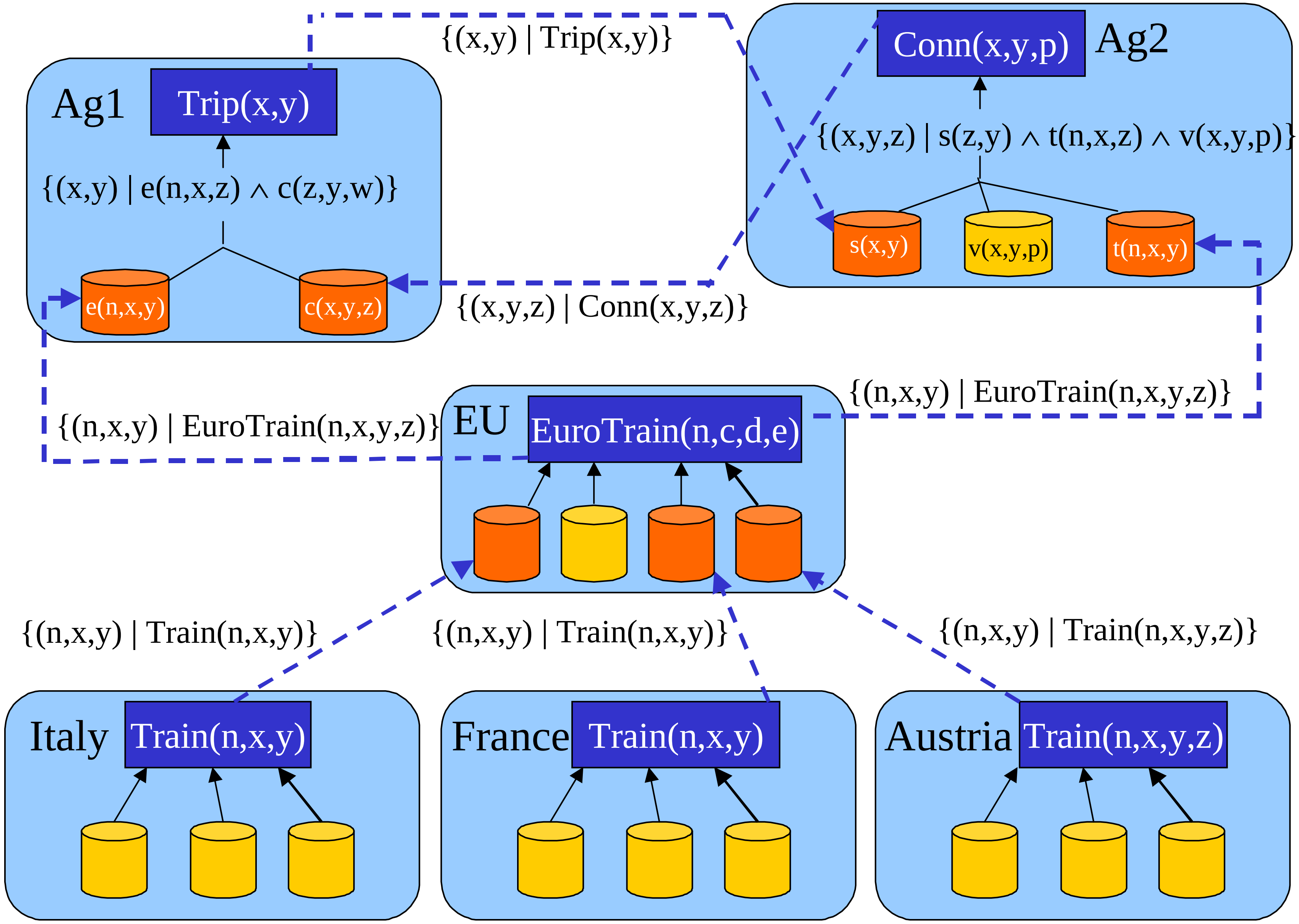
  where:
  - cq is a conjunctive query over one of the other peers in Π
  - s is an external source of the peer P
  - cq and s are of the same arity

Extraction programs

- The notion of extraction program aims at modeling computations done in order to
  - extract
  - clean
  - transform
  - reconcile

  data coming from (local and external) data sources
- We assume that, given the extensions of the sources, an extraction program extracts a set of tuples (of the same arity as the arity of the program)
- We do not deal with extraction programs, but we point out that they are accomodated in the framework

Example



{(x,y) | Trip(x,y)}
Ag1
Trip(x,y)
{(x,y) | e(n,x,z) ∧ c(z,y,w)}
e(n,x,y)
c(x,y,z)
Conn(x,y,p)
Ag2
{(x,y,z) | s(z,y) ∧ t(n,x,z) ∧ v(x,y,p)}
s(x,y)
v(x,y,p)
t(n,x,y)
{(x,y,z) | Conn(x,y,z)}
{(n,x,y) | EuroTrain(n,x,y,z)}
{(n,x,y) | EuroTrain(n,x,y,z)}
EU
EuroTrain(n,c,d,e)
{(n,x,y) | Train(n,x,y)}
{(n,x,y) | Train(n,x,y)}
{(n,x,y) | Train(n,x,y,z)}
Italy
Train(n,x,y)
France
Train(n,x,y)
Austria
Train(n,x,y,z)

Outline

- Data integration architectures
- Centralized data integration
- P2P data integration: the framework
- Basic semantics for P2P data integration
- Quality-aware semantics for P2P data integration
- Conclusions

Quality of the whole system: semantics

The client sees the whole collection of peers through the eye of one peer, and she conceives the distributed information system as a unique database

- What does this database provide to the client?
- Can the client trust the answers to queries computed by system?
- Can we prove that it is sound and/or complete in some sense?

No answers to these questions without semantics!

Semantics of one peer

For each peer $P = (G, S, L, M)$ we define a FOL theory T_P as follows:

- The alphabet of T_P is obtained as union of the alphabets of the schema G and of the sources S
- The axioms of T_P are as follows:
  - all FOL formulas in the schema G
  - for each local mapping assertion $\{\vec{\mathbf{x}} \mid ep_S(\vec{\mathbf{x}})\} \leadsto \{\vec{\mathbf{x}} \mid \exists \vec{\mathbf{z}}\, \varphi_G(\vec{\mathbf{x}}, \vec{\mathbf{z}})\}$ in L, one formula of the form

$$\forall \vec{\mathbf{x}}\ (ep_S(\vec{\mathbf{x}}) \supset \exists \vec{\mathbf{z}}\, \varphi_G(\vec{\mathbf{x}}, \vec{\mathbf{z}}))$$

Notice that T_P does not consider the P2P mappings in M

It follows that we are modeling each peer P as a GLAV data integration system, in turn modeled as a FOL theory T_P (ignoring the P2P mappings M)

Semantics of a P2P system

- A source database $\mathcal{D}$ for Π is the disjoint union of one source database for each peer P_i in Π

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

$$sem^{\mathcal{D}}(\Pi) = \{\ \mathcal{I} \mid \begin{array}{l} \mathcal{I} \text{ is a model of all peer theories } T_{P_i} \text{ based on } \mathcal{D}\text{, and} \\ \mathcal{I} \text{ satisfies all P2P mapping assertions} \end{array} \}$$

  The meaning of $\mathcal{I}$ satisfying a P2P mapping assertion may vary in the various approaches

- Given a query Q of arity n posed to a peer P_i of Π, and a source database $\mathcal{D}$, the certain answers to Q based on $\mathcal{D}$ are

$$ans(Q, \Pi, \mathcal{D}) = \{\ \vec{\mathbf{t}} \in \Gamma^n \mid \vec{\mathbf{t}} \in Q^{\mathcal{I}}, \text{ for every } \mathcal{I} \in sem^{\mathcal{D}}(\Pi)\ \}$$

Possible formalizations of P2P mappings

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

- Based on First-Order Logic

  P2P mappings are considered as material logical implications

- Based on Epistemic Logic

  P2P mappings are considered as specifications of exchange of certain answers

First-Order Logic semantics of P2P mappings

The semantics of P2P mapping assertions is given in terms of First-Order Logic [Halevy&al. ICDE'03], [Bernstein&al. WebDB '02]

An interpretation $\mathcal{I}$ satisfies a P2P mapping assertion

$$\{\vec{\mathbf{x}} \mid \exists \vec{\mathbf{y}}\, \varphi(\vec{\mathbf{x}}, \vec{\mathbf{y}})\} \leadsto s(\vec{\mathbf{x}})$$

if it satisfies the FOL formula

$$\forall \vec{\mathbf{x}}\, (\exists \vec{\mathbf{y}}\, \varphi(\vec{\mathbf{x}}, \vec{\mathbf{y}}) \equiv s(\vec{\mathbf{x}}))$$

which is equivalent to the condition

$$\{\vec{\mathbf{x}} \mid \exists \vec{\mathbf{y}}\, \varphi_1(\vec{\mathbf{x}}, \vec{\mathbf{y}})\}^{\mathcal{I}} = (s(\vec{\mathbf{x}}))^{\mathcal{I}}$$

Inadequacy of FOL semantics of P2P mappings

The FOL semantics is not adequate for P2P data integration:

- Lack of modularity
  - 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, even when all peer schemas are empty [Halevy&al. ICDE'03], [Koch FOIKS'02]
- Lack of generality

  To recover decidability, one has to limit the expressive power of P2P mappings (e.g., assume acyclicity) [Halevy&al. ICDE'03]

Epistemic semantics for P2P mappings: objectives

A new semantics for P2P mappings, with the following aims:

- 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
- Defining 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]

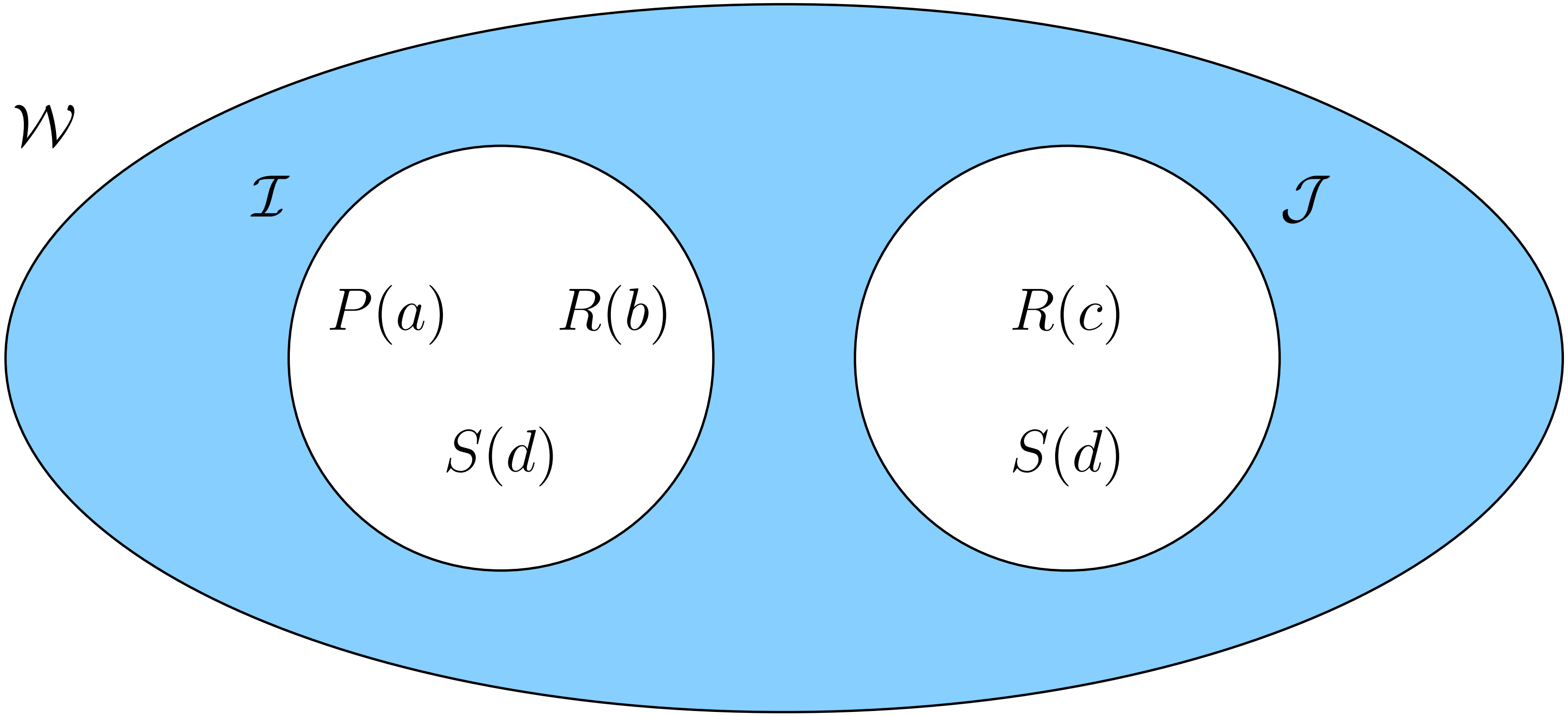
- A P2P mapping $cq_i \rightsquigarrow s_j$ (with cq_i over P_i and s_j external source of P_j) is interpreted as an epistemic formula which imposes that only the certain answers to cq_i in P_i (i.e., the facts that are known by P_i) are transferred to P_j as facts satisfying s_j.

  In other words, peer P_i communicates to peer P_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)

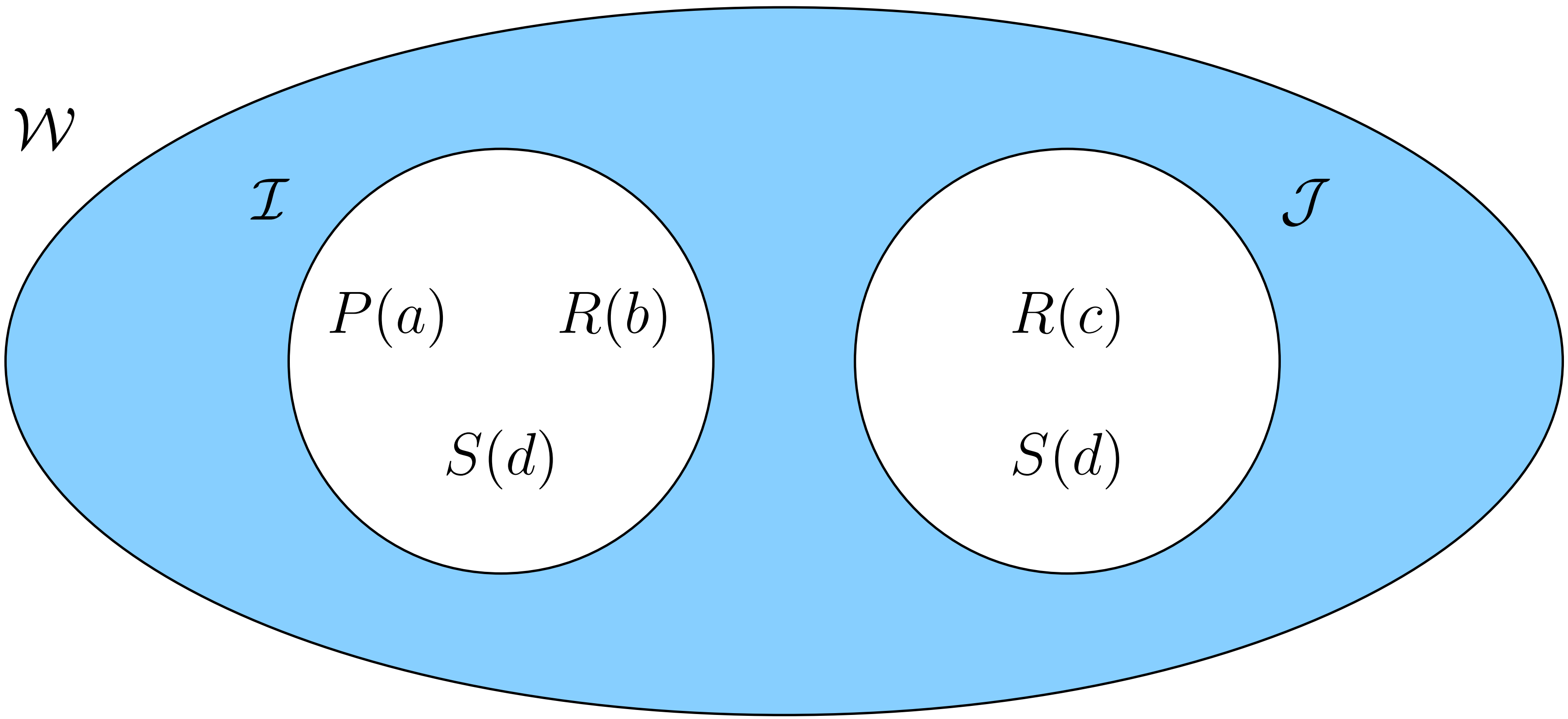
- Good computational properties: computing the certain answers to a conjunctive query Q based on a source database $\mathcal{D}$ is polynomial time in the size of $\mathcal{D}$, even for cyclic mappings

Epistemic logic semantics



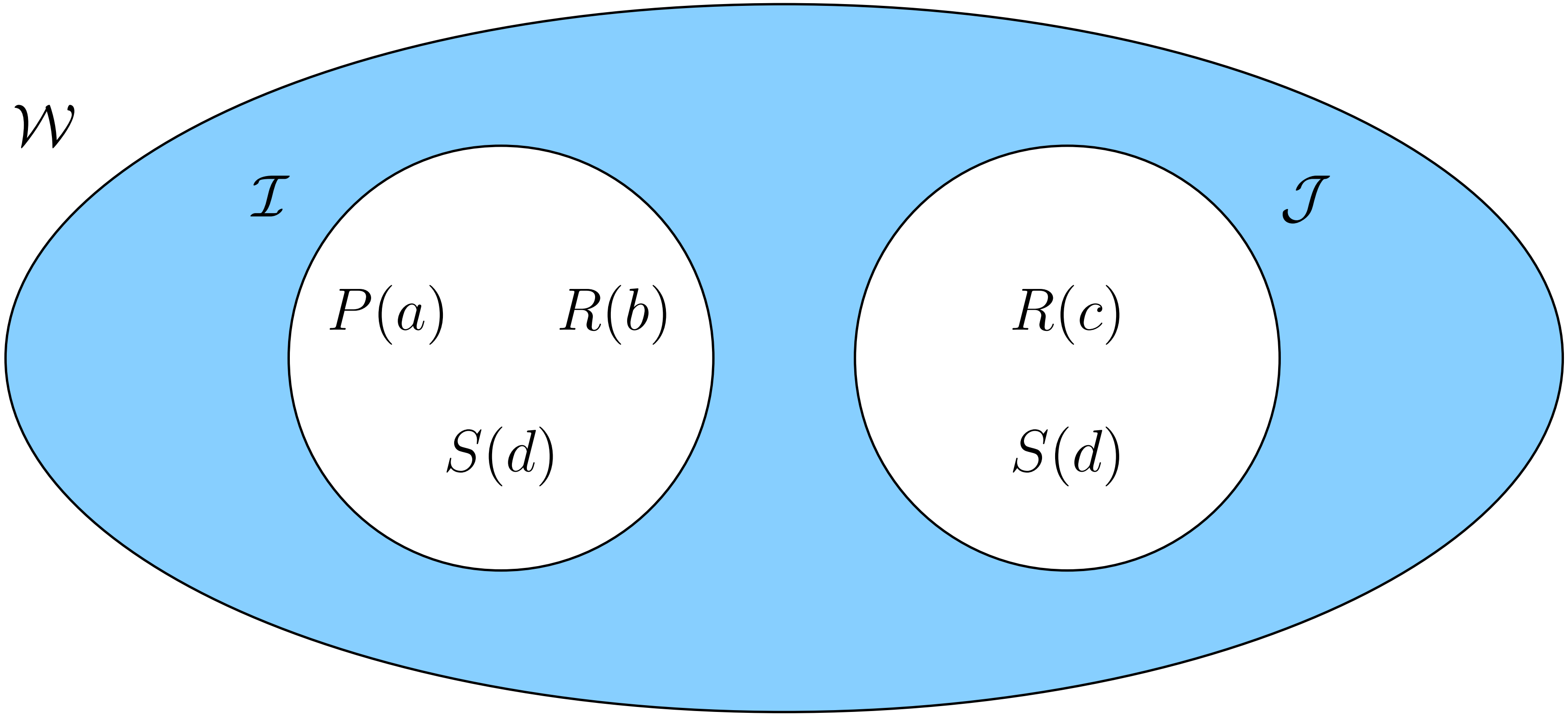
- $\mathcal{W}$ is an epistemic structure, i.e., a collection of FOL intepretations
- $\langle \mathcal{I}, \mathcal{W} \rangle$ and $\langle \mathcal{J}, \mathcal{W} \rangle$ are epistemic interpretations
- $\mathbf{K}\varphi(\vec{\mathbf{x}})$ is satisfied in $\langle \mathcal{I}, \mathcal{W} \rangle$ by the tuples $\vec{\mathbf{t}}$ of constants such that $\varphi(\vec{\mathbf{t}})$ is satisfied in all epistemic interpretations $\langle \mathcal{J}, \mathcal{W} \rangle$ with $\mathcal{J} \in \mathcal{W}$

Epistemic logic: example 1



$$
\begin{aligned}
\langle \mathcal{I}, \mathcal{W} \rangle &\models P(a) \\
\langle \mathcal{J}, \mathcal{W} \rangle &\not\models P(a) \\
\langle \mathcal{I}, \mathcal{W} \rangle &\not\models \mathbf{K}\, P(a)
\end{aligned}
$$

Epistemic logic: example 2

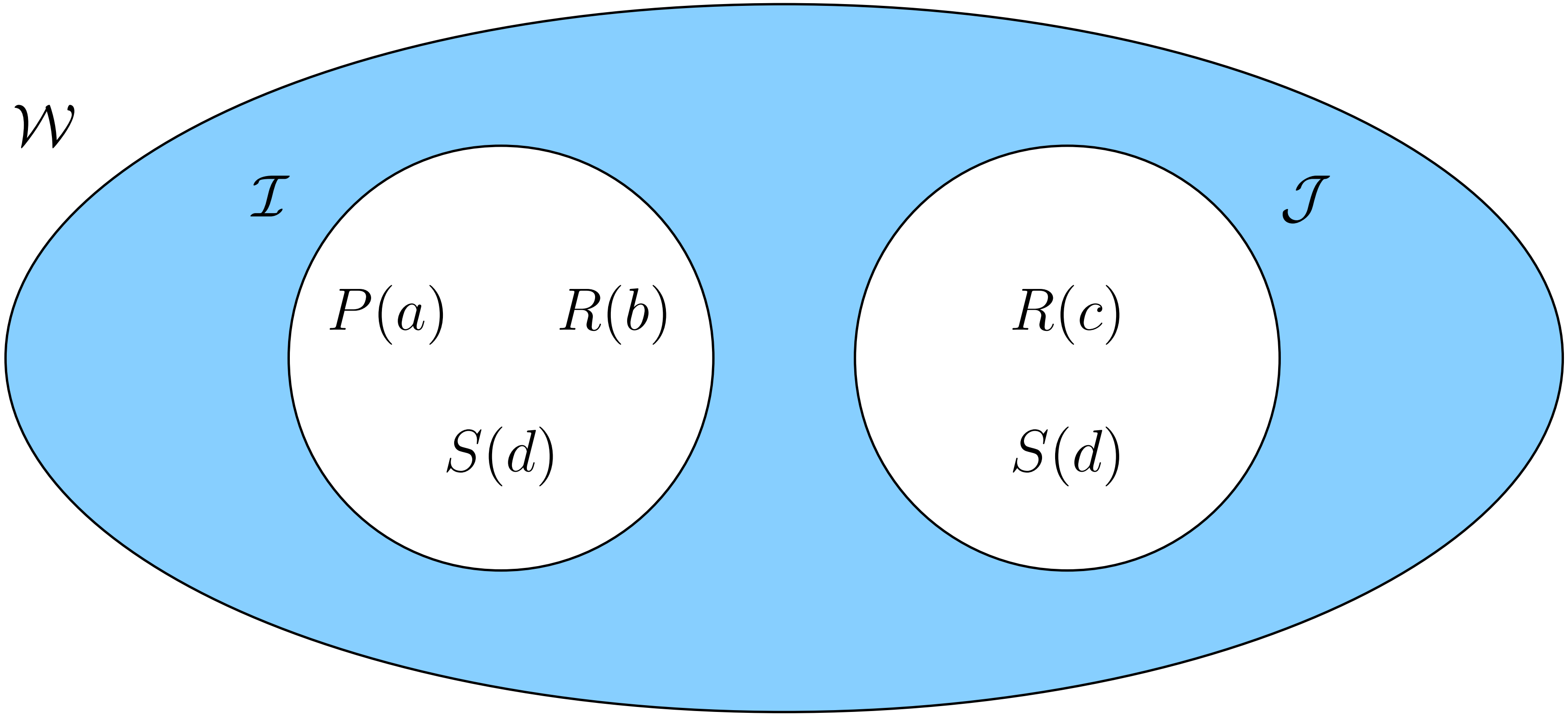


$$\langle \mathcal{I}, \mathcal{W} \rangle \models \mathbf{K}\,(R(b) \vee R(c))$$

$$\langle \mathcal{I}, \mathcal{W} \rangle \not\models (\mathbf{K}\,R(b)) \vee (\mathbf{K}\,R(c))$$

$$\langle \mathcal{I}, \mathcal{W} \rangle \models \mathbf{K}\,S(d)$$

Epistemic logic: example 3



$$\langle \mathcal{I}, \mathcal{W} \rangle \models \mathbf{K}\,(\exists x\, R(x))$$

$$\langle \mathcal{I}, \mathcal{W} \rangle \not\models \exists x\,(\mathbf{K}\, R(x))$$

$$\langle \mathcal{I}, \mathcal{W} \rangle \models \exists x\,(\mathbf{K}\, S(x))$$

Epistemic semantics for P2P mappings: basic idea

We formalize a P2P system Π in terms of the epistemic logic theory E_Π:

- the alphabet $\mathcal{A}_\Pi$ is the disjoint union of the alphabets of the various peer theories T_P, one for each peer P in Π
- all the formulas of the various theories T_P are axioms in E_Π
- for each P2P mapping assertion

$$\{\vec{\mathbf{x}} \mid \exists \vec{\mathbf{y}}\, \varphi(\vec{\mathbf{x}}, \vec{\mathbf{y}})\} \leadsto \{\vec{\mathbf{x}} \mid s(\vec{\mathbf{x}})\}$$

  in the peers of Π, there is one axiom in E_Π of the form

$$\forall \vec{\mathbf{x}}\, ((\mathbf{K}\, \exists \vec{\mathbf{y}}\, \varphi_1(\vec{\mathbf{x}}, \vec{\mathbf{y}})) \equiv s(\vec{\mathbf{x}}))$$

Epistemic semantics for P2P mappings: basic idea

In other words, $\langle \mathcal{I}, \mathcal{W} \rangle$ satisfies the P2P mapping assertion $cq \rightsquigarrow s$ if,

for every tuple $\vec{\mathbf{t}}$ of constants in Γ,

when $\vec{\mathbf{t}} \in cq^{\mathcal{J}}$ for every FOL model $\mathcal{J}$ in $\mathcal{W}$, then $\vec{\mathbf{t}} \in s^{\mathcal{I}}$

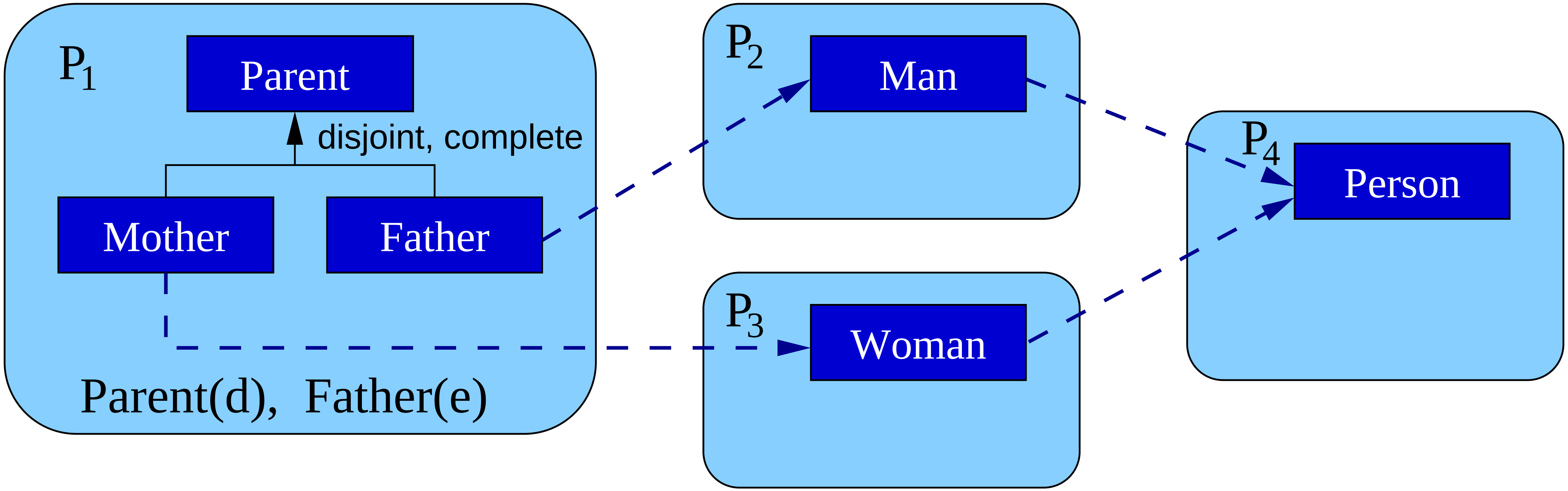
An epistemic model of Π based on $\mathcal{D}$ is an epistemic interpretation $\langle \mathcal{I}, \mathcal{W} \rangle$ such that

- $\mathcal{W}$ is a set of models of T_Π based on $\mathcal{D}$, and
- $\langle \mathcal{I}, \mathcal{W} \rangle$ satisfies all axioms corresponding to the P2P mapping assertions in the peers of Π

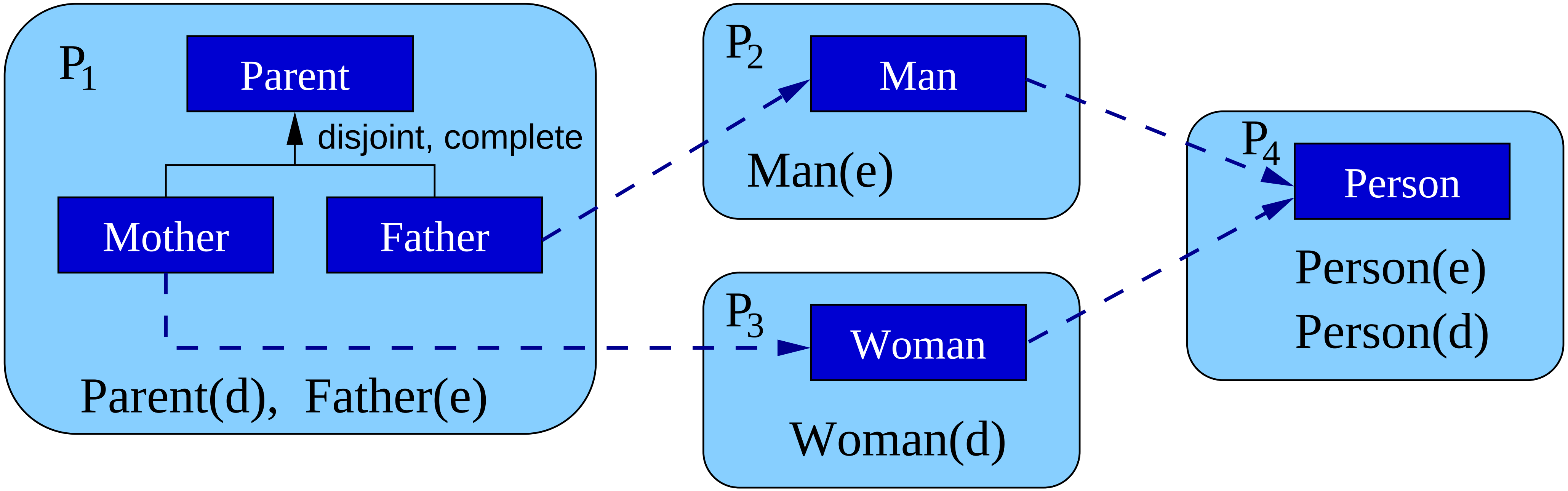
Given a query Q of arity n posed to a peer P_i of Π, and a source database $\mathcal{D}$, the certain answers to Q based on $\mathcal{D}$ under epistemic semantics are

$$\textit{ans}_{\mathbf{k}}(Q, \Pi, \mathcal{D}) \;=\; \{\, \vec{\mathbf{t}} \in \Gamma^n \mid \vec{\mathbf{t}} \in Q^{\mathcal{I}}, \text{ for every epistemic model } \langle \mathcal{I}, \mathcal{W} \rangle \text{ of } \Pi \text{ based on } \mathcal{D} \,\}$$

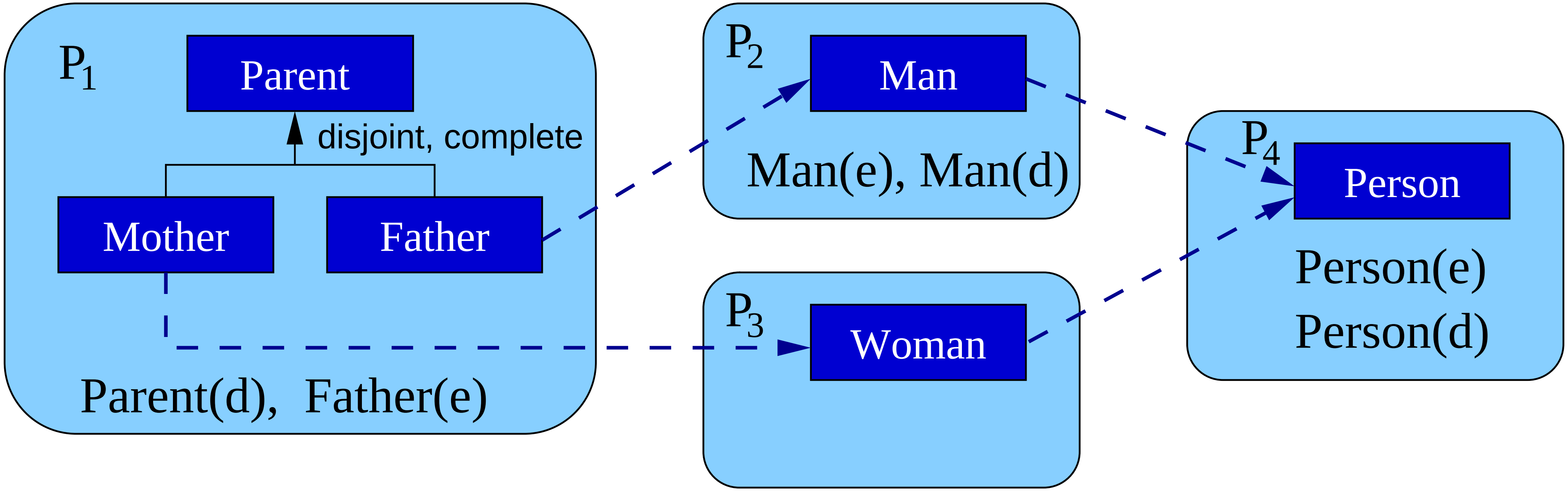
Semantics of P2P mappings: example



FOL semantics of P2P mappings: model 1

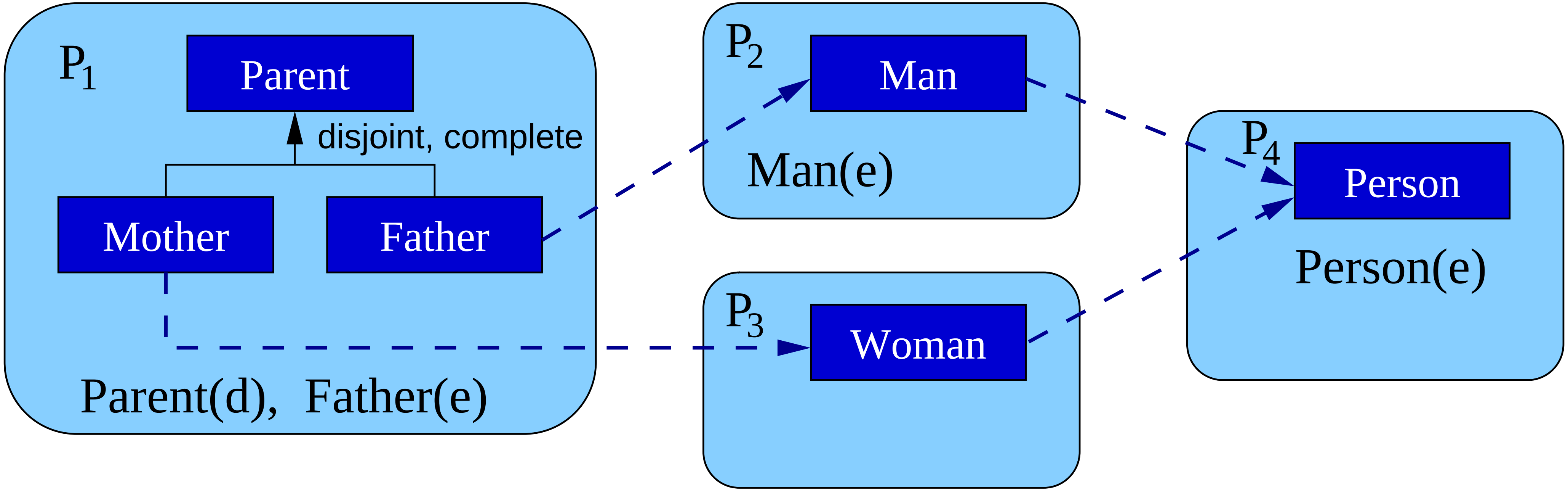


FOL semantics of P2P mappings: model 2



According to the FOL semantics, $\mathrm{Person}(d)$ is true in all cases, and therefore is a certain answer to $\{x \mid \mathrm{Person}(x)\}$

Epistemic semantics of P2P mappings



According to the epistemic semantics, $\text{Person}(d)$ is not a certain answer to $\{x \mid \text{Person}(x)\}$

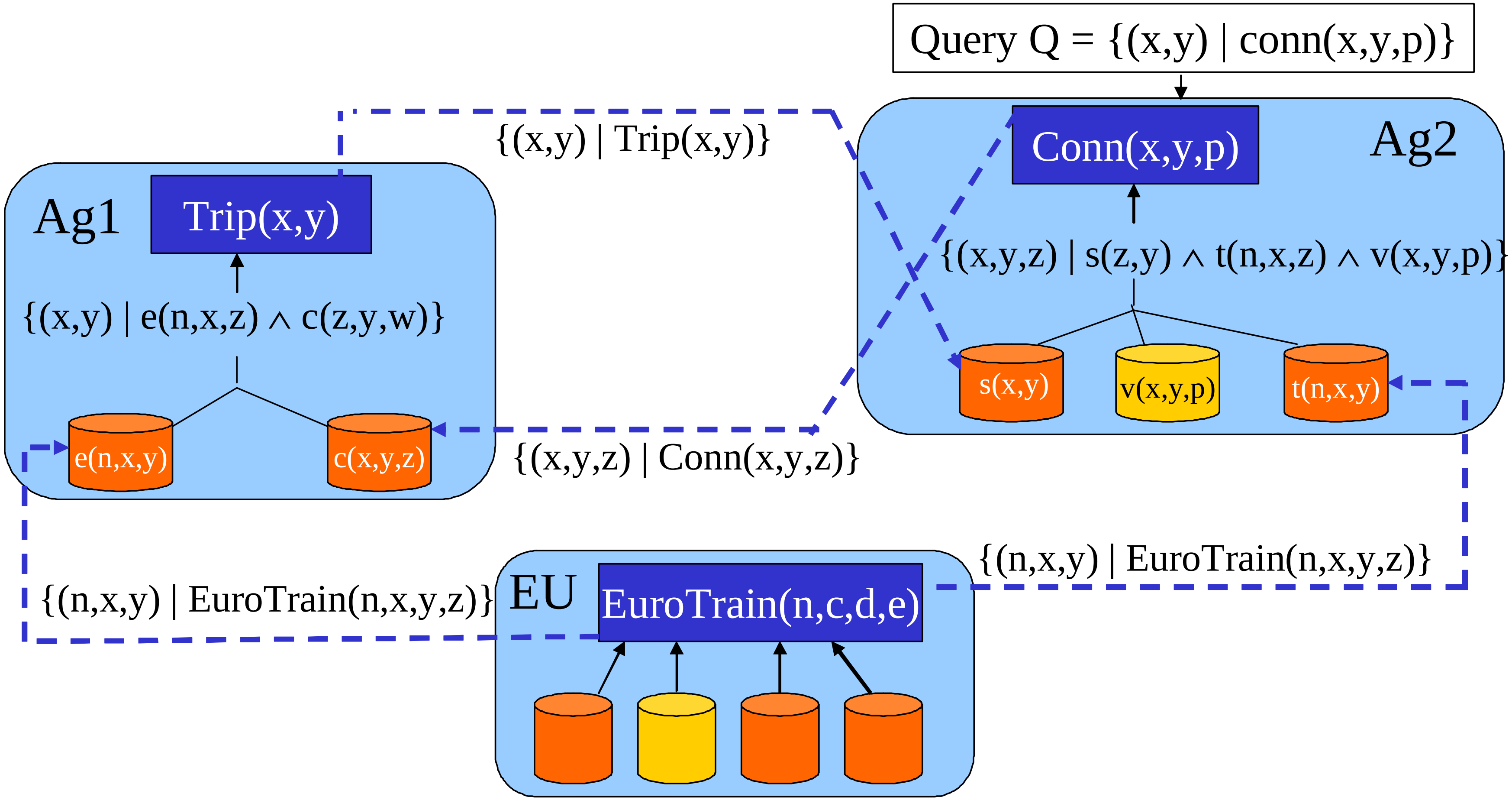
Answering queries under the epistemic semantics

- Distributed query answering
  - the query is posed to one peer in the system
  - each peer executes the same algorithm, and in doing so exchanges information only with the peers it is connected to
- Step-by-step algorithm
  - the query is posed to one peer in the system
  - each peer answers extensionally by taking into account its own data, and then answers intensionally by directing the client to other peers

In both cases, two important issues are

- Each peer is able to reformulate a query expressed over its schema in terms of the local and external sources (perfect reformulation assumption)
- Loop detection

Epistemic semantics of P2P mappings



Query Q is perfectly reformulated into:

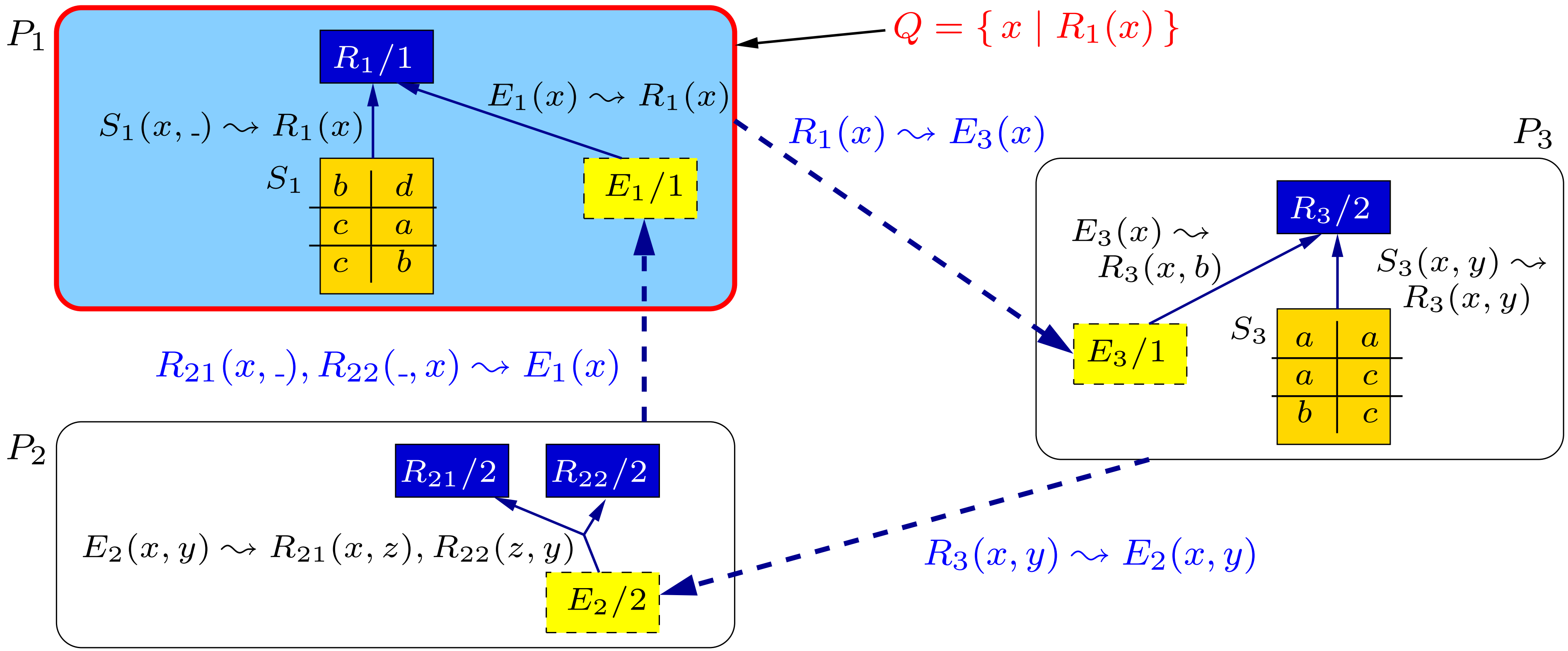
- the query $\{ (x,y) \mid Trip(x,y)\}$ to be issued to peer Ag1
- the local source v(x,y,p)
- the query $\{ (n,x,y) \mid EuroTrain(n,x,y,z)\}$ to be issued to peer EU

Query answering: distributed algorithm

[Calvanese & al PODS'04] presents a distributed query answering algorithm

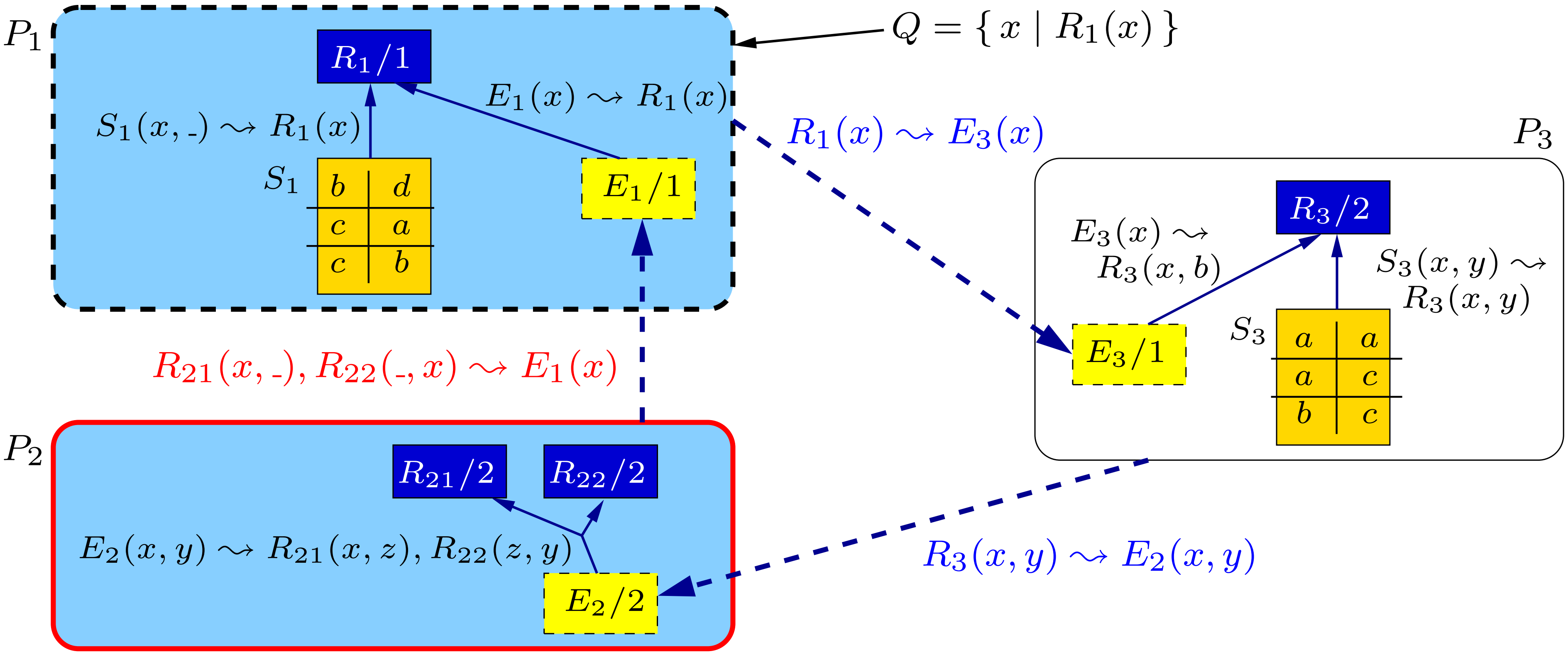
- Each peer reformulates the queries that are requested to it in terms of the local and external sources (perfect reformulation assumption)
- A reference to an external source triggers a request to the peer to which the external source is connected
- Answers to such requests consist of a Datalog program with two parts:
  - an extensional part, which is a set of facts (about source relations received from other peers)
  - an intensional part, which is a set of Datalog rules
- The final Datalog program is executed at the initiating peer
- Infinite looping is avoided by:
  - associating to each client query a unique (global) transaction id
  - avoiding requests that have already been made for the same transaction id

Query answering technique: example



$$1 \quad \left\{ \begin{array}{lcl} Q(x) & \leftarrow & S_1(x, \_) \\ Q(x) & \leftarrow & E_1(x) \end{array} \right.$$

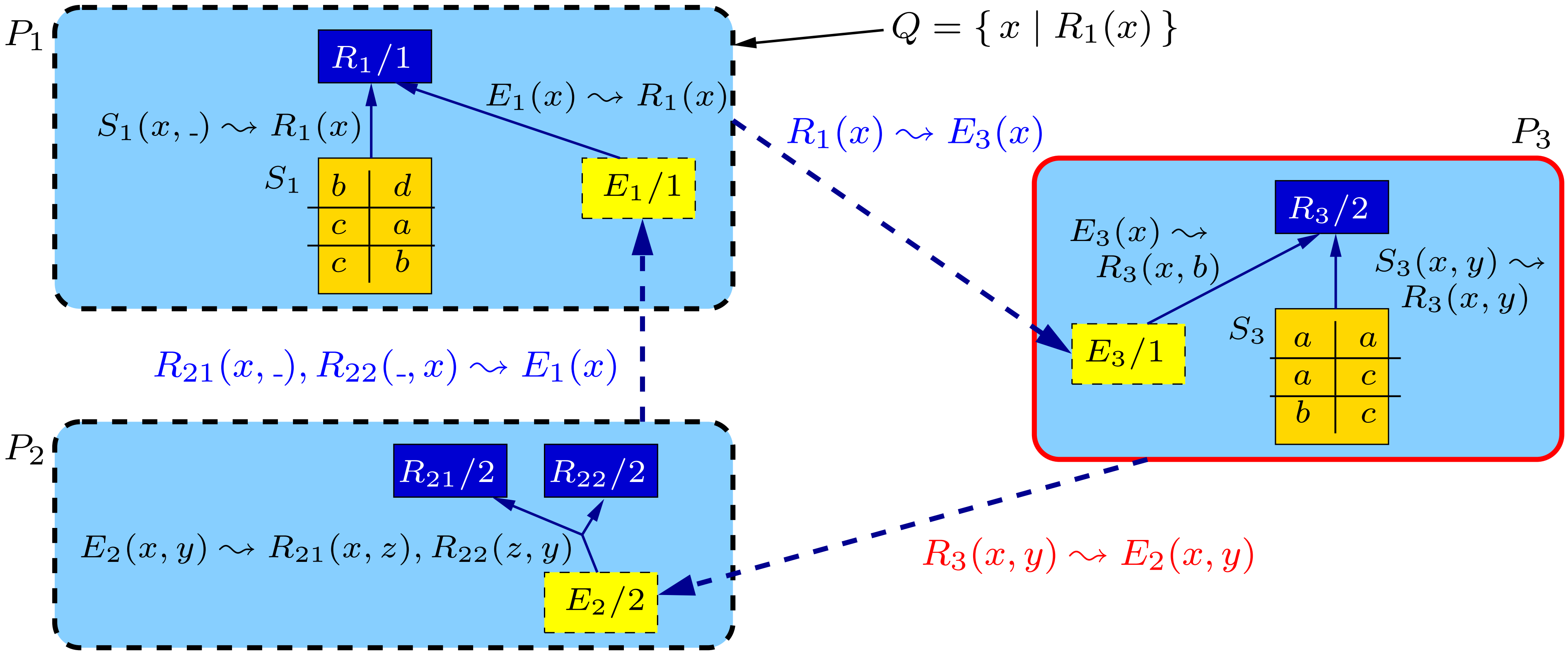
Query answering technique: example



$$1 \left\{ \begin{array}{lcl} Q(x) & \leftarrow & S_1(x, \_) \\ Q(x) & \leftarrow & E_1(x) \end{array} \right.$$

$$2 \left\{ \begin{array}{lcl} E_1(x) & \leftarrow & E_2(x, \_), E_2(\_, x) \end{array} \right.$$

Query answering technique: example

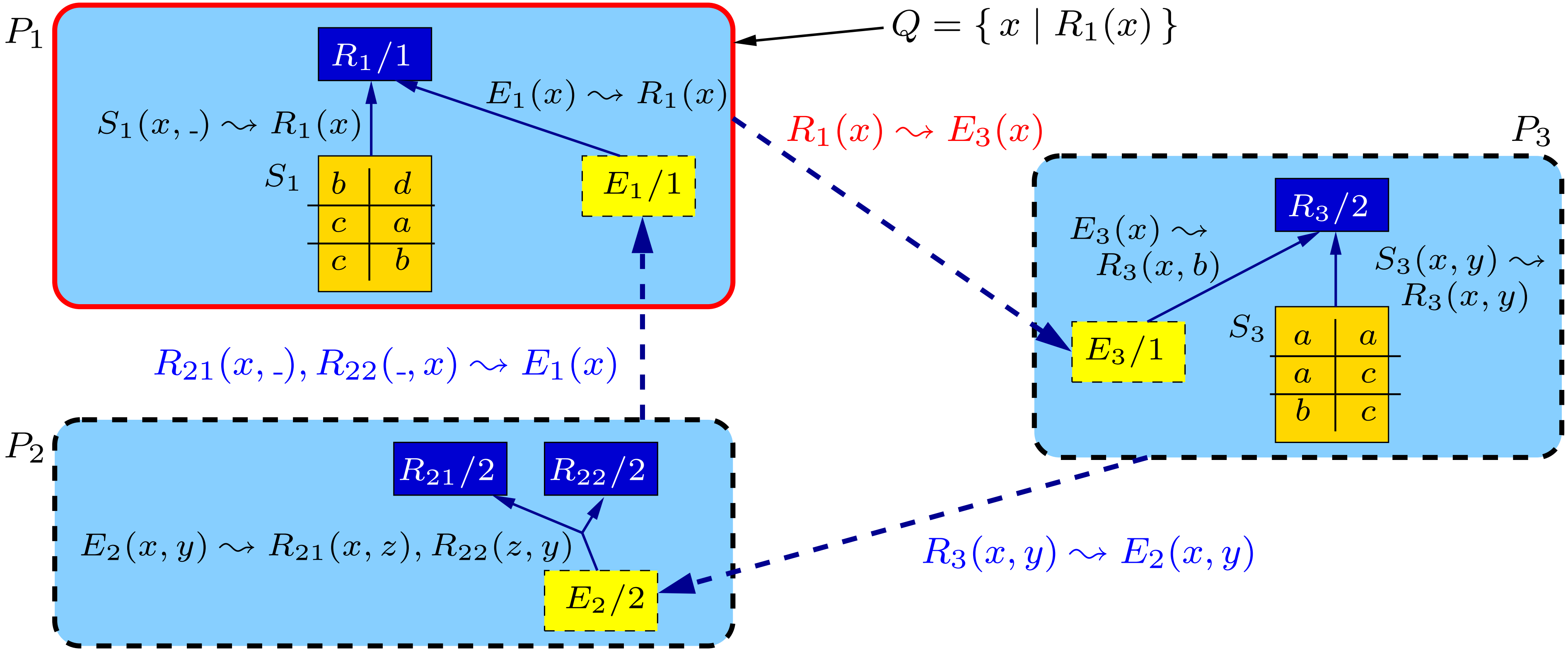


$$1 \left\{ \begin{array}{lcl} Q(x) & \leftarrow & S_1(x, \_) \\ Q(x) & \leftarrow & E_1(x) \end{array} \right.$$

$$3 \left\{ \begin{array}{lcl} E_2(x, y) & \leftarrow & S_3(x, y) \\ E_2(x, y) & \leftarrow & E_3(x), y = b \end{array} \right.$$

$$2 \left\{ \begin{array}{lcl} E_1(x) & \leftarrow & E_2(x, \_), E_2(\_, x) \end{array} \right.$$

Query answering technique: example



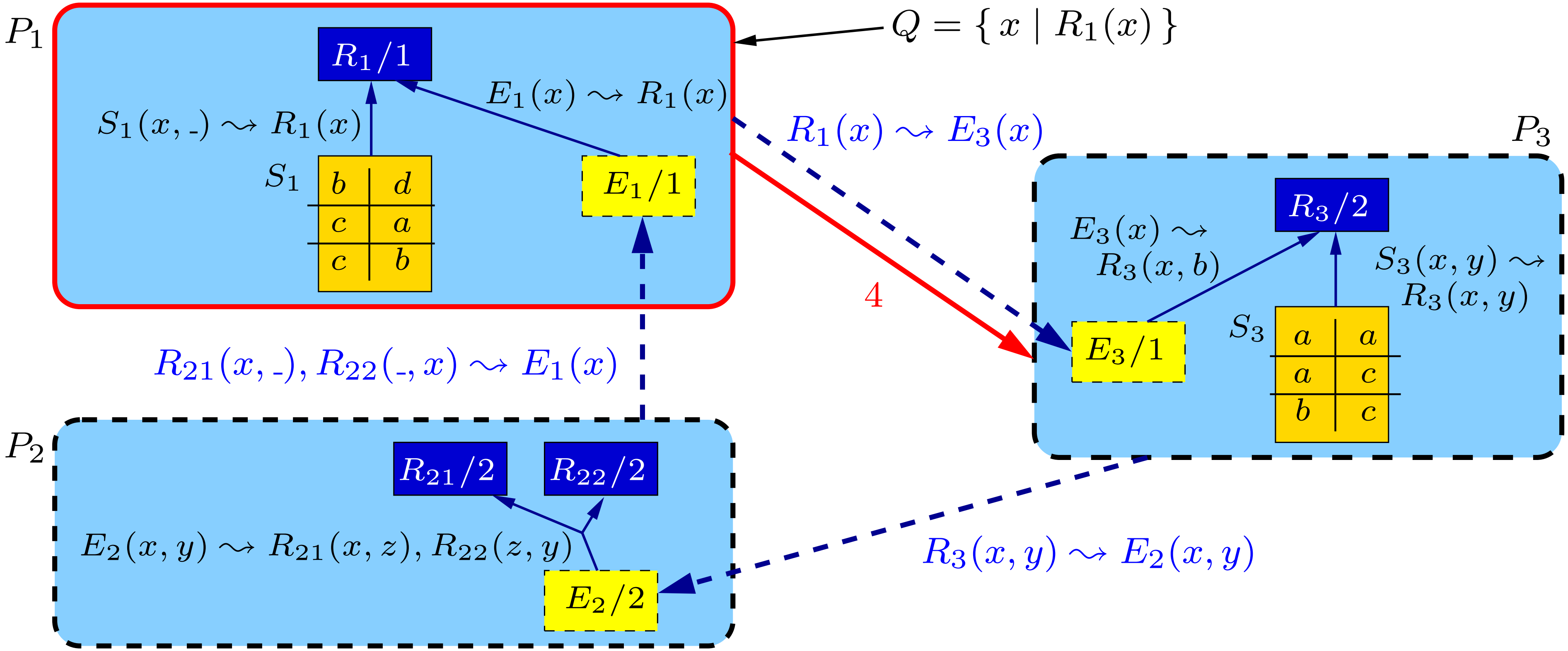
$$1 \left\{ \begin{array}{lcl} Q(x) & \leftarrow & S_1(x, \_) \\ Q(x) & \leftarrow & E_1(x) \end{array} \right.$$

$$2 \left\{ \begin{array}{lcl} E_1(x) & \leftarrow & E_2(x, \_), E_2(\_, x) \end{array} \right.$$

$$3 \left\{ \begin{array}{lcl} E_2(x, y) & \leftarrow & S_3(x, y) \\ E_2(x, y) & \leftarrow & E_3(x), y = b \end{array} \right.$$

$$4 \left\{ \begin{array}{lcl} E_3(x) & \leftarrow & S_1(x, \_) \\ E_3(x) & \leftarrow & E_1(x) \end{array} \right.$$

Query answering technique: example



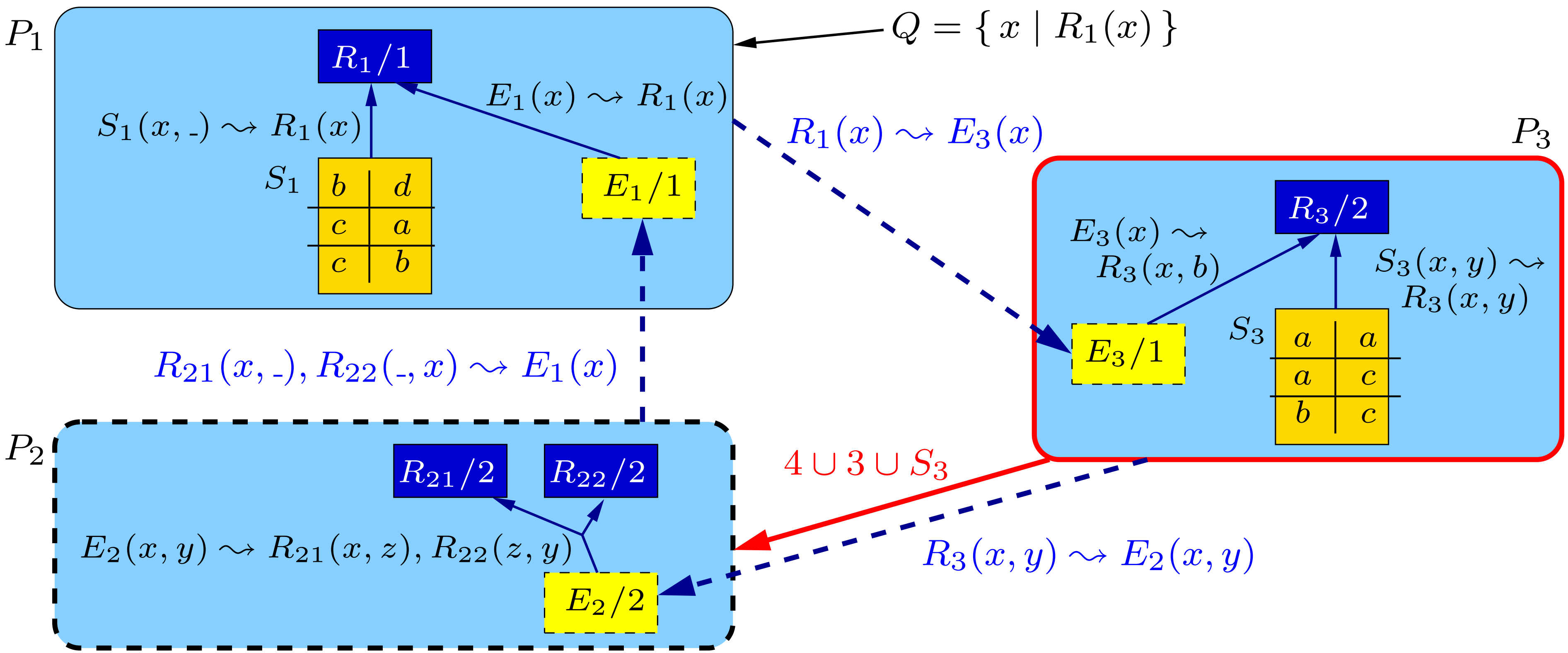
$$1 \begin{cases} Q(x) \leftarrow S_1(x, \_) \\ Q(x) \leftarrow E_1(x) \end{cases}$$

$$3 \begin{cases} E_2(x, y) \leftarrow S_3(x, y) \\ E_2(x, y) \leftarrow E_3(x), y = b \end{cases}$$

$$2 \left\{ E_1(x) \leftarrow E_2(x, \_), E_2(\_, x) \right.$$

$$4 \begin{cases} E_3(x) \leftarrow S_1(x, \_) \\ E_3(x) \leftarrow E_1(x) \end{cases}$$

Query answering technique: example



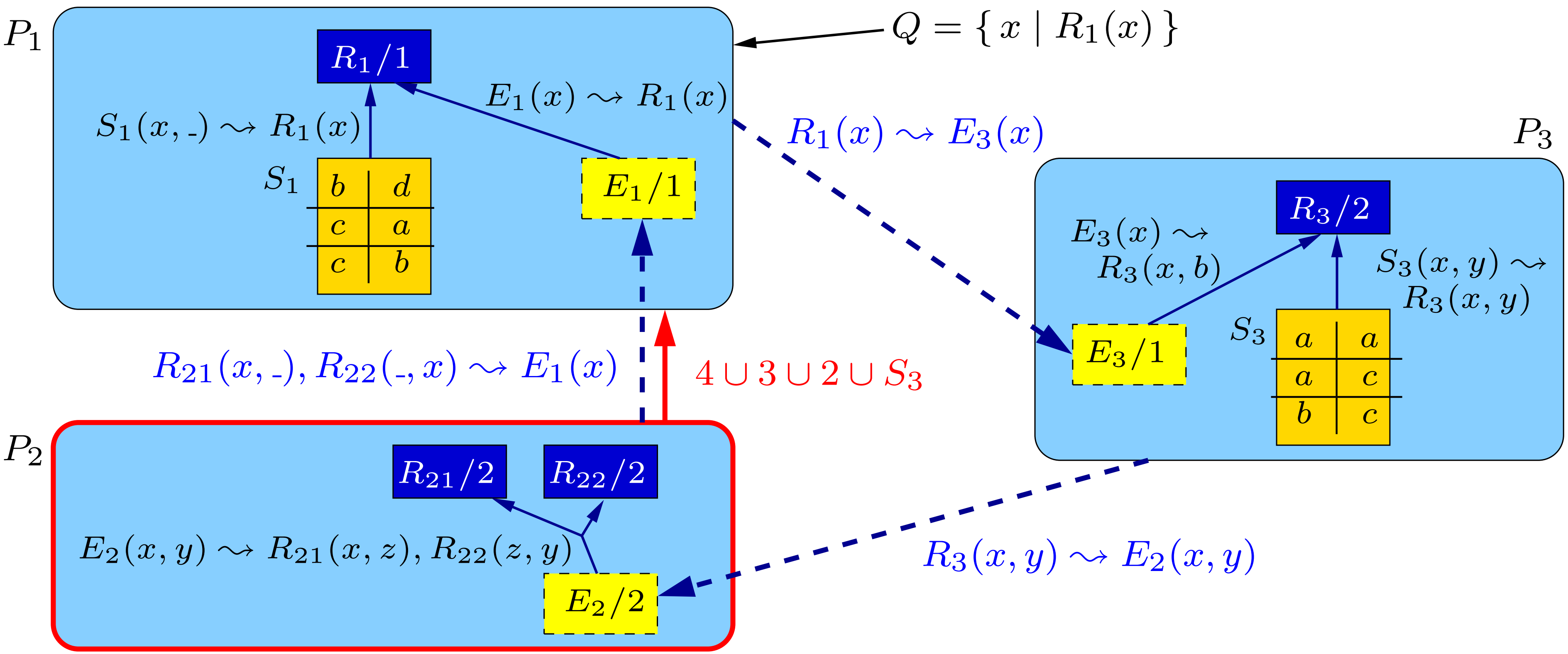
$$1 \left\{ \begin{array}{lcl} Q(x) & \leftarrow & S_1(x, \_) \\ Q(x) & \leftarrow & E_1(x) \end{array} \right.$$

$$2 \left\{ \begin{array}{lcl} E_1(x) & \leftarrow & E_2(x, \_), E_2(\_, x) \end{array} \right.$$

$$3 \left\{ \begin{array}{lcl} E_2(x, y) & \leftarrow & S_3(x, y) \\ E_2(x, y) & \leftarrow & E_3(x), y = b \end{array} \right.$$

$$4 \left\{ \begin{array}{lcl} E_3(x) & \leftarrow & S_1(x, \_) \\ E_3(x) & \leftarrow & E_1(x) \end{array} \right.$$

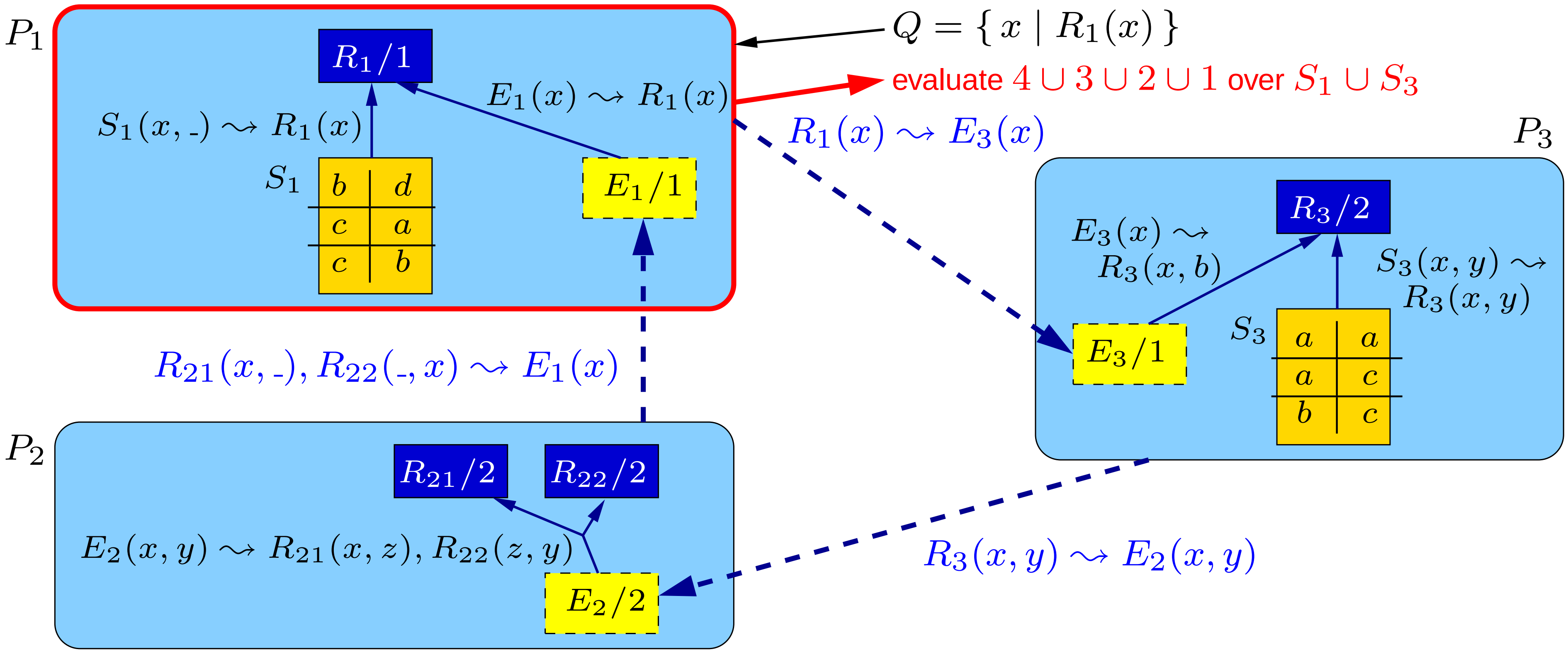
Query answering technique: example



$$1 \left\{ \begin{array}{lcl} Q(x) & \leftarrow & S_1(x, \_) \\ Q(x) & \leftarrow & E_1(x) \end{array} \right. \qquad 3 \left\{ \begin{array}{lcl} E_2(x, y) & \leftarrow & S_3(x, y) \\ E_2(x, y) & \leftarrow & E_3(x), y = b \end{array} \right.$$

$$2 \left\{ \begin{array}{lcl} E_1(x) & \leftarrow & E_2(x, \_), E_2(\_, x) \end{array} \right. \qquad 4 \left\{ \begin{array}{lcl} E_3(x) & \leftarrow & S_1(x, \_) \\ E_3(x) & \leftarrow & E_1(x) \end{array} \right.$$

Query answering technique: example



$$1 \left\{ \begin{array}{lcl} Q(x) & \leftarrow & S_1(x, \_) \\ Q(x) & \leftarrow & E_1(x) \end{array} \right.$$

$$2 \left\{ \begin{array}{lcl} E_1(x) & \leftarrow & E_2(x, \_), E_2(\_, x) \end{array} \right.$$

$$3 \left\{ \begin{array}{lcl} E_2(x, y) & \leftarrow & S_3(x, y) \\ E_2(x, y) & \leftarrow & E_3(x), y = b \end{array} \right.$$

$$4 \left\{ \begin{array}{lcl} E_3(x) & \leftarrow & S_1(x, \_) \\ E_3(x) & \leftarrow & E_1(x) \end{array} \right.$$

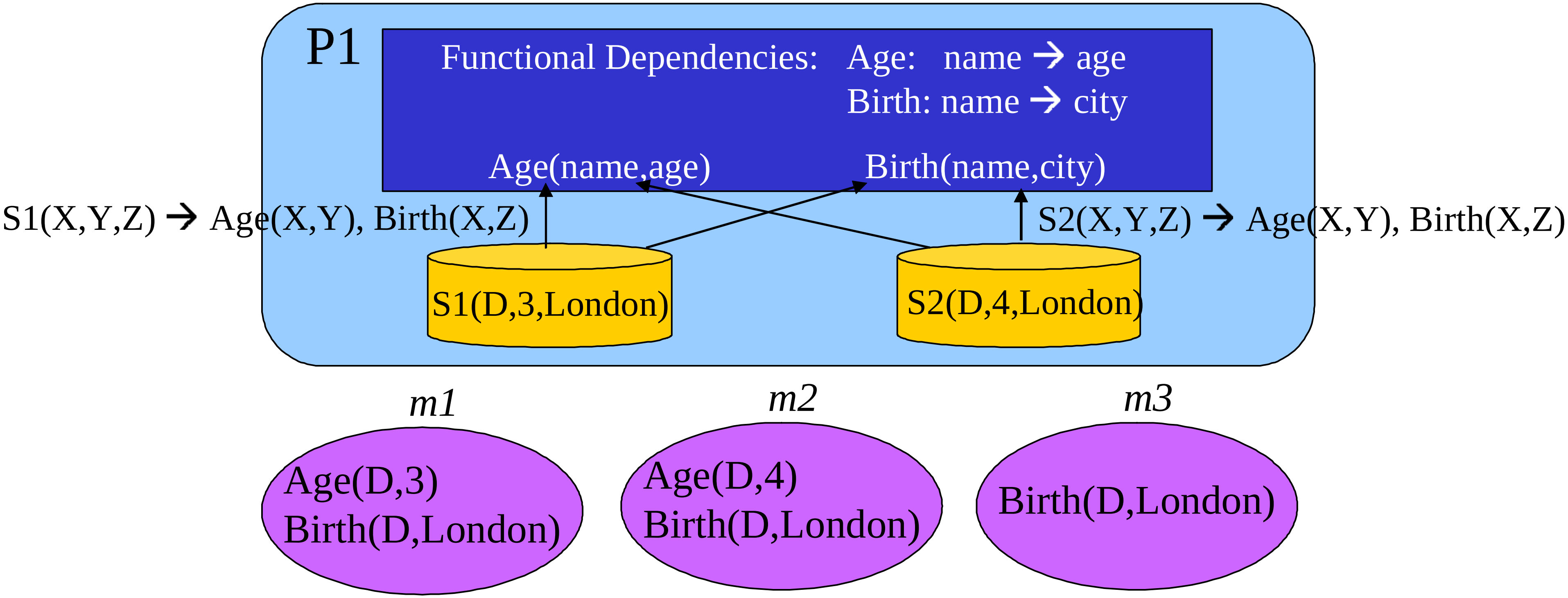
Outline

- Data integration architectures
- Centralized data integration
- P2P data integration: the framework
- Basic semantics for P2P data integration
- Quality-aware semantics for P2P data integration
- Conclusions

Quality-aware semantics P2P data integration

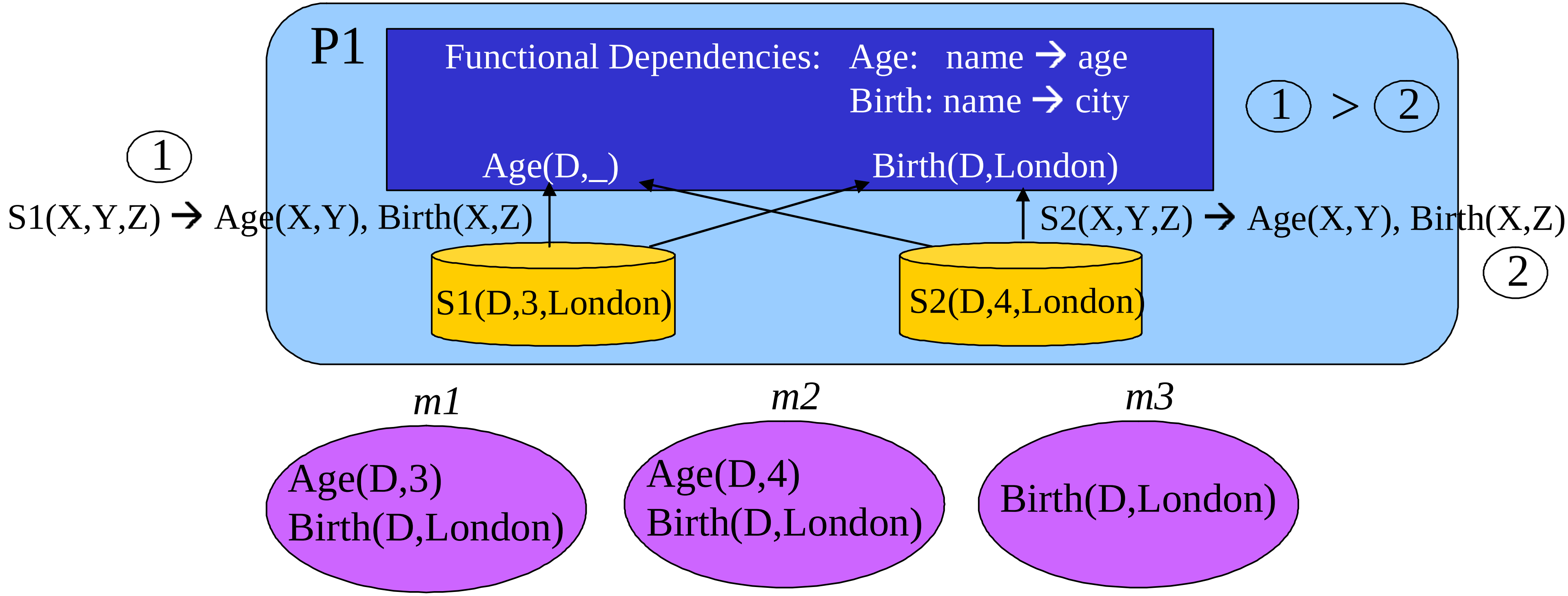
- The current formalization is ok with no inconsistency
- The whole system blows up with a single inconsistency in one peer: inacceptable quality of query answering
- We resort to quality information for dealing with inconsistencies: we conceive information on source and peer qualities as a mechanism for deciding among possible inconsistency resolutions
- If quality information do not suffice to decide, we reason disjunctively
- Our main goal is to come up with a well-defined semantics, which extends the one based on epistemic logic with new (non-monotonic) features

Dealing with inconsistencies in one peer



- A model m is preferred to model n if n misses some data from the sources that m does not miss
- In the figure, both $m1$ and $m2$ are better than $m3$
- The models of a peer are the most preferred models

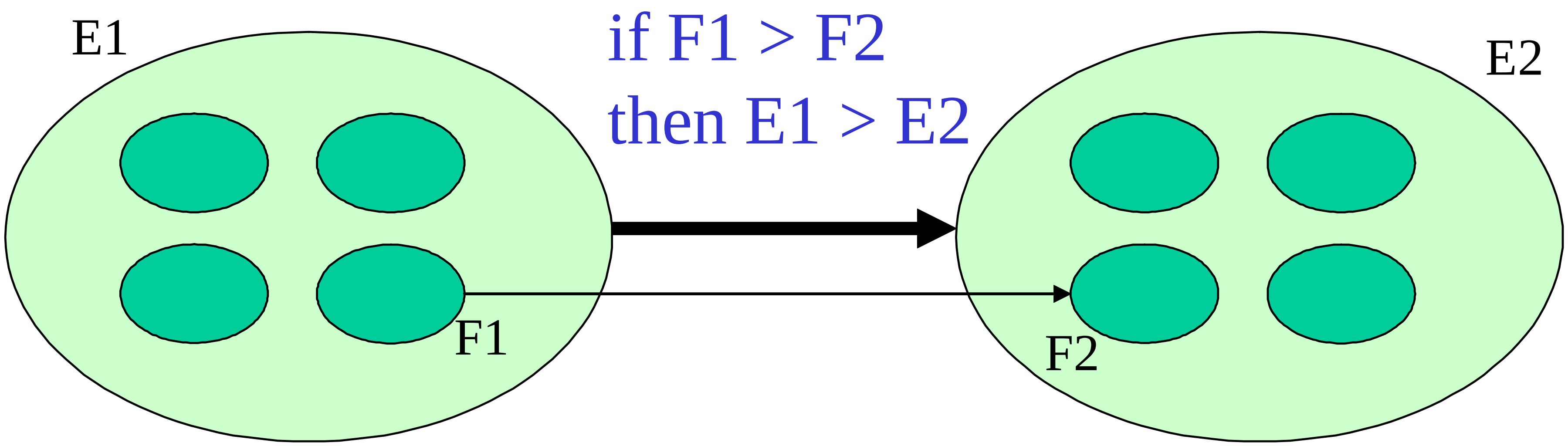
Dealing with quality-based preferences in one peer



- A model m is preferred to model n if n misses some data from the sources that m does not miss, or if m respects the preferences more than n
- In the figure, $m1$ is better than both $m2$ and $m3$
- The models of a peer are the most preferred models

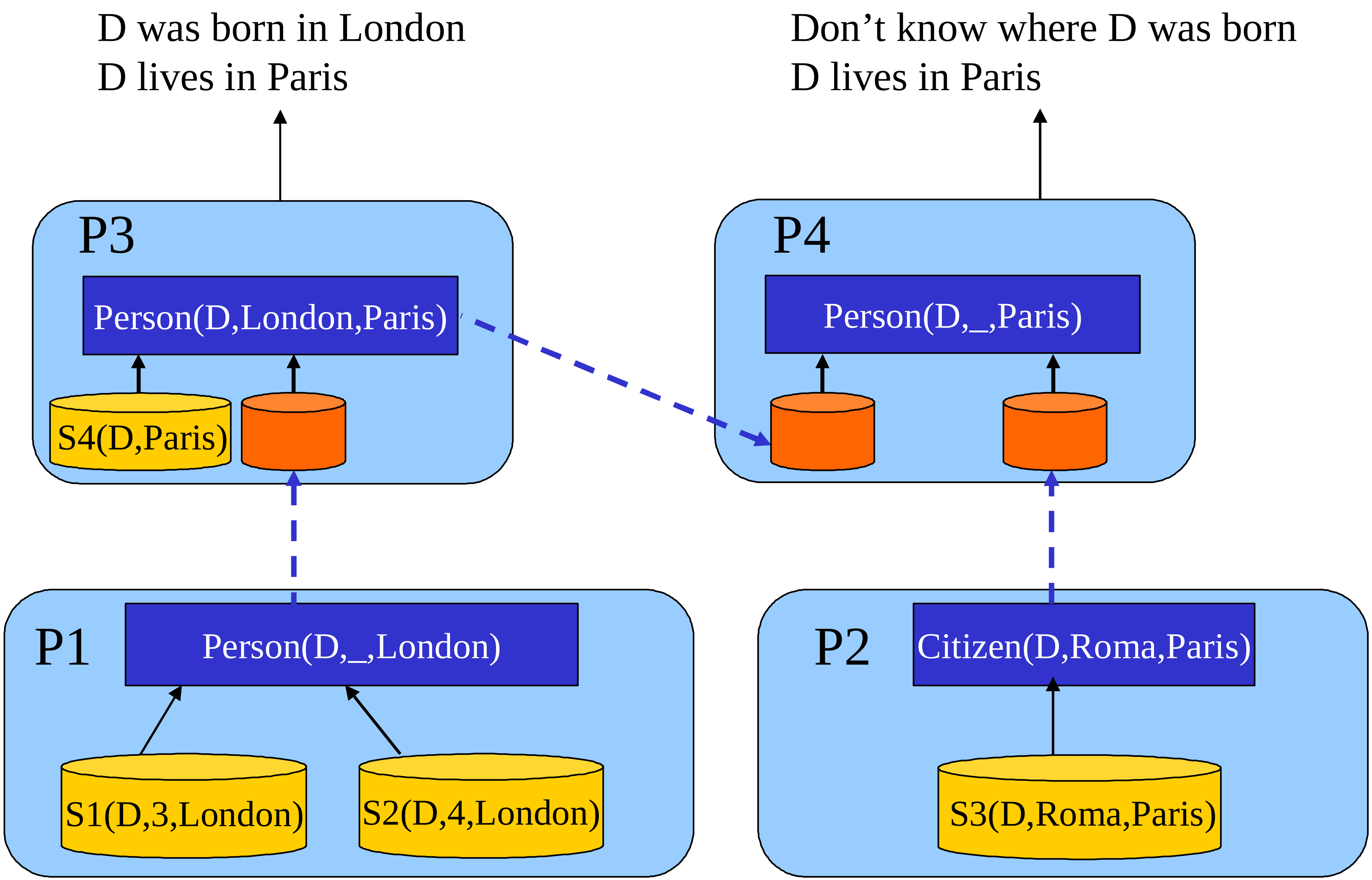
Dealing with inconsistencies in P2P data integration

- To generalize the idea to the case of multiple peers, we have to be able to compare epistemic models
- Basic idea:

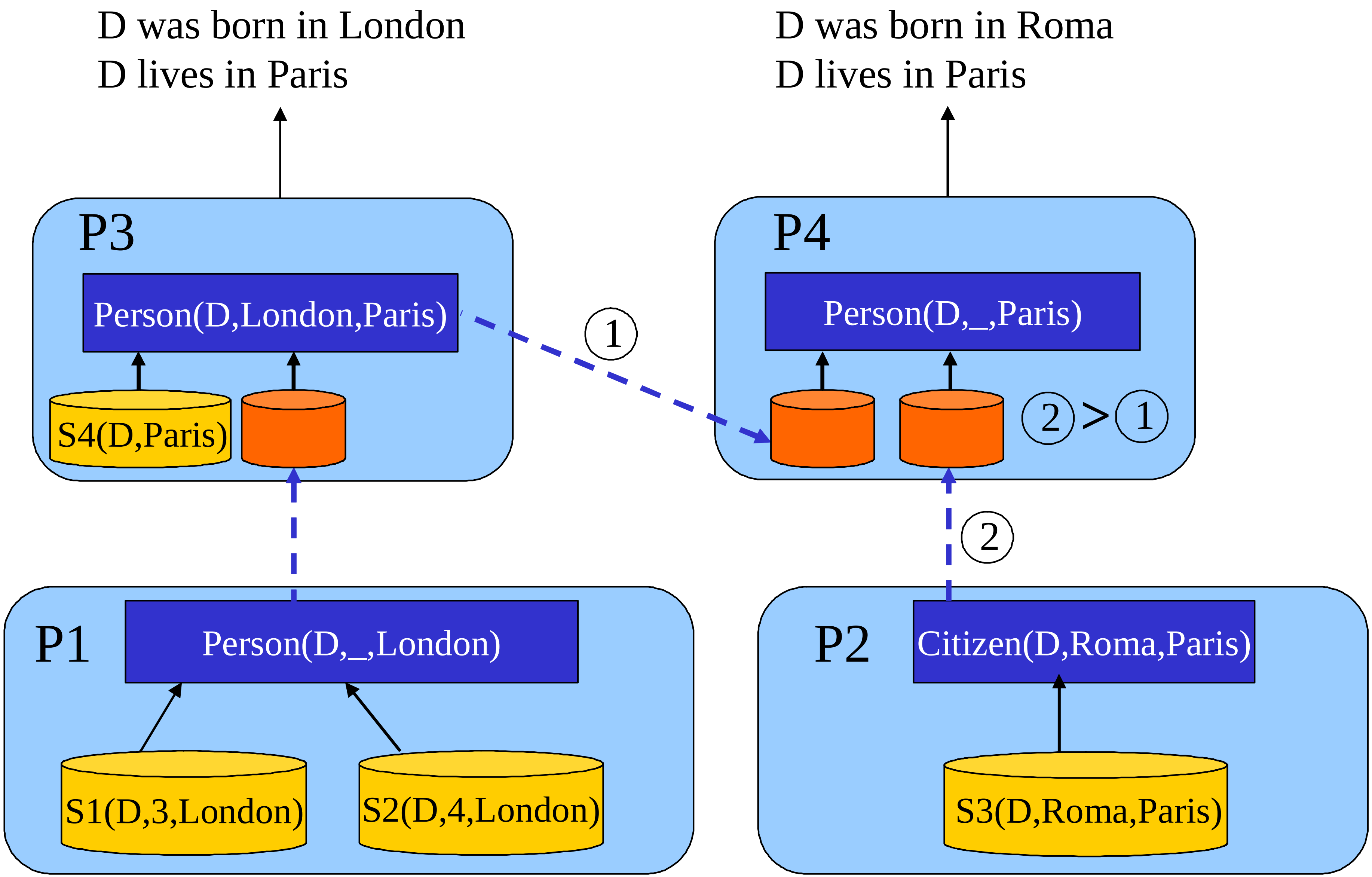


- The models of the P2P data integration system are the most preferred epsitemic models

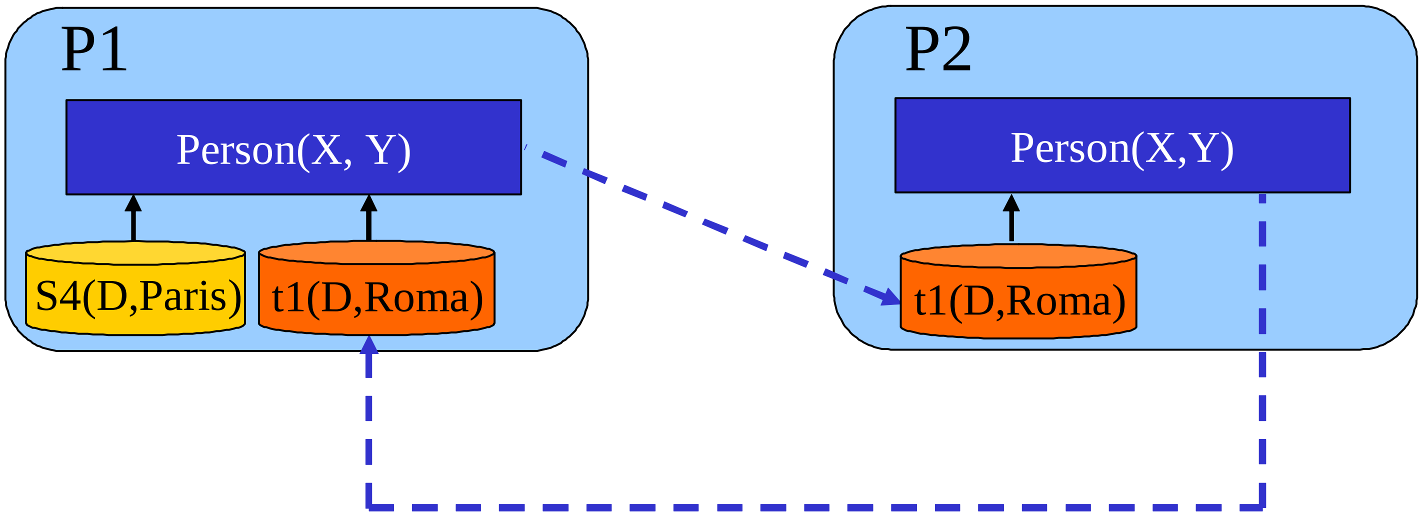
Dealing with inconsistencies in P2P data integration



Dealing with inconsistencies and quality in P2P data integration



Cycles pose new problems in the extended semantics



- The semantics should allow us to conclude Person(D,Paris), since Person(D,Roma) is not justified by any real data
- Technically, this can be accomplished by suitably defining the ordering between epistemic models

Outline

- Data integration architectures
- Centralized data integration
- P2P data integration: the framework
- Basic semantics for P2P data integration
- Quality-aware semantics for P2P data integration
- Conclusions

Conclusions

Many open problems and issues, including

- Algorithm and complexity in the extended epistemic semantics
- How to obtain information on quality and preferences
- Global schema (or target schema, or peer schemas) expressed in terms of semi-structured data (with constraints)
- Limitations in accessing the sources
- Privacy-based restrictions on peer answers
- Optimization
- Experiments (ongoing in Hyper, a joint project with IBM, and Sewasie and Infomix, two EU projects)