Logical foundations for 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 SOFSEM 2005

Liptovsky Jan, Slovak Republic – January 27, 2005

Outline

- Data integration architectures
- Centralized data integration
- P2P data integration
- Hyper: epistemic semantics for P2P data integration
- Dealing with inconsistencies in Hyper
- Conclusions

Outline

- $\boxed{\text{Data integration architectures}}$
- Centralized data integration
- P2P data integration
- Hyper: epistemic semantics for P2P data integration
- Dealing with inconsistencies in Hyper
- Conclusions

Three data integration architectures

- Centralized data integration

  The traditional architecture for centralized, virtual data integration

- Data exchange

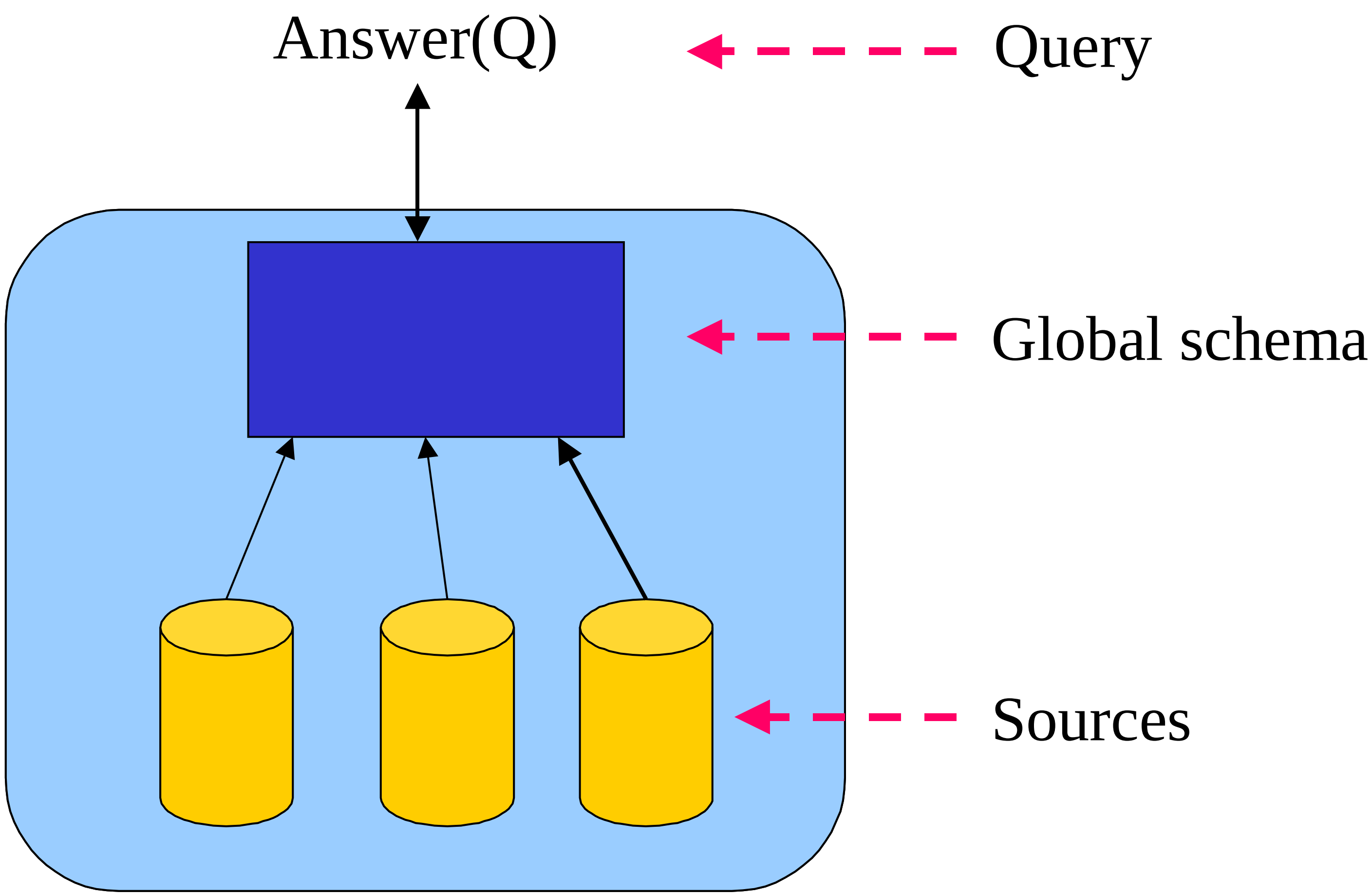
  Materialization of data from a source database to a target database

- Peer-to-peer data integration

  Decentralized, dynamic, data-centric coordination between autonomous organizations

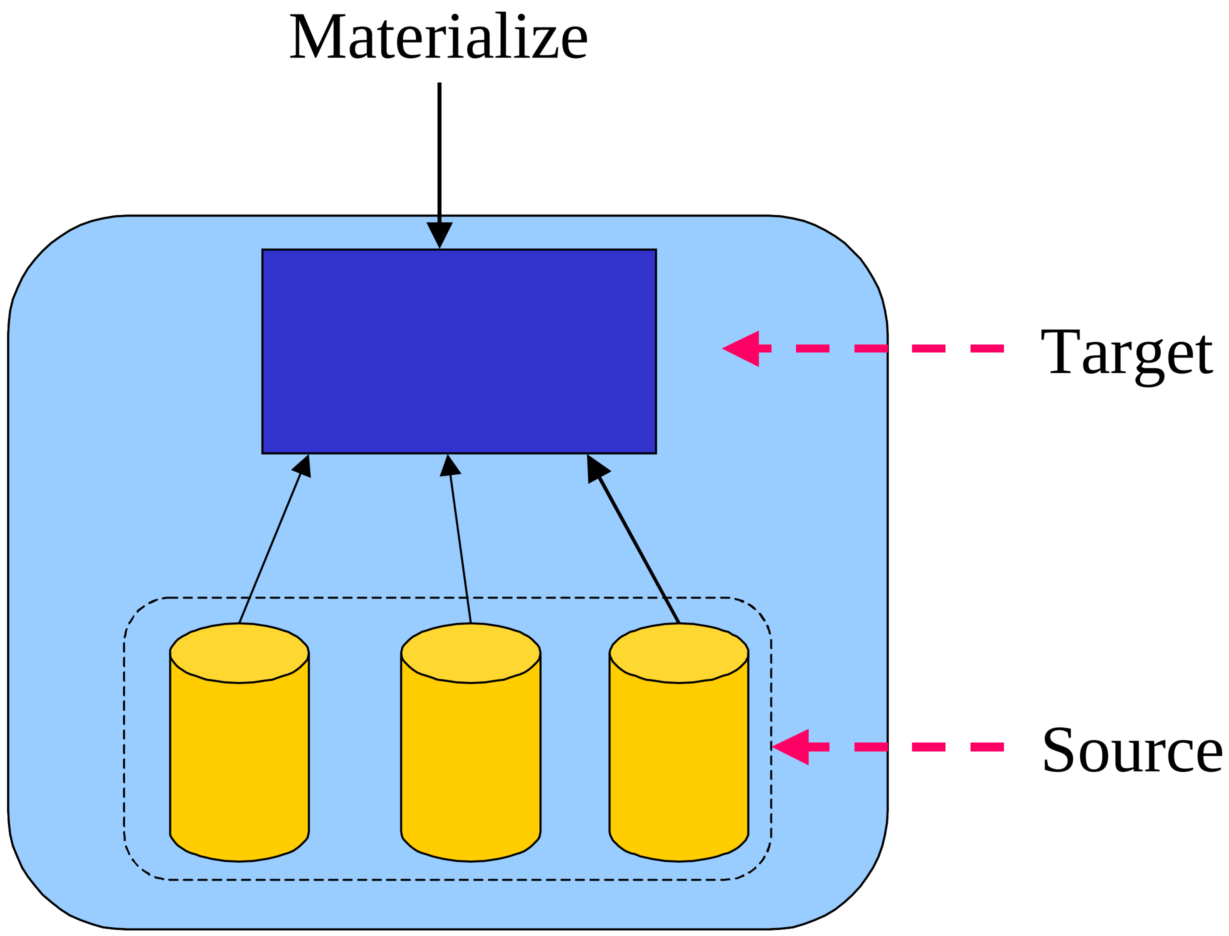
Centralized data integration

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



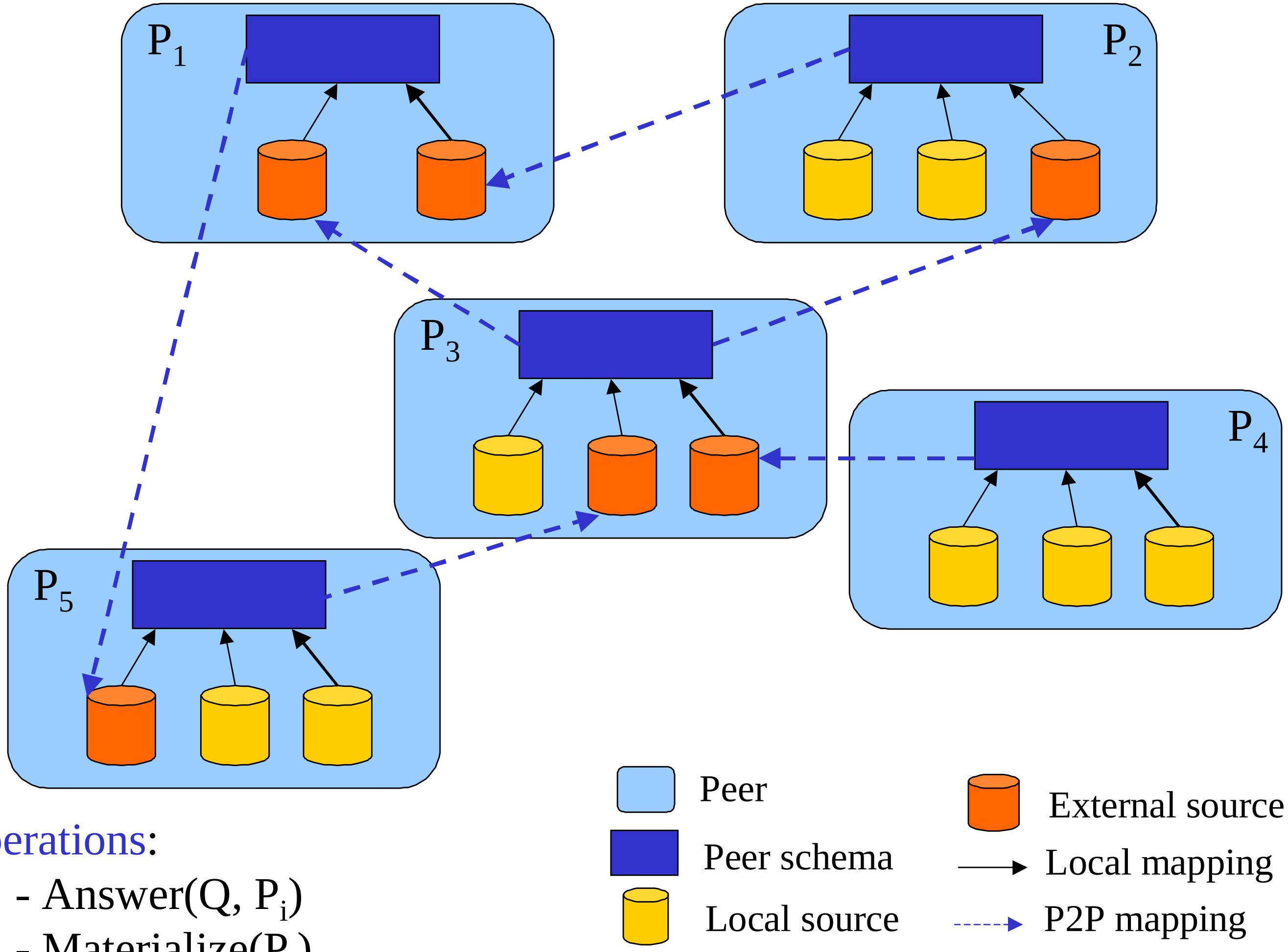
Data exchange

- 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



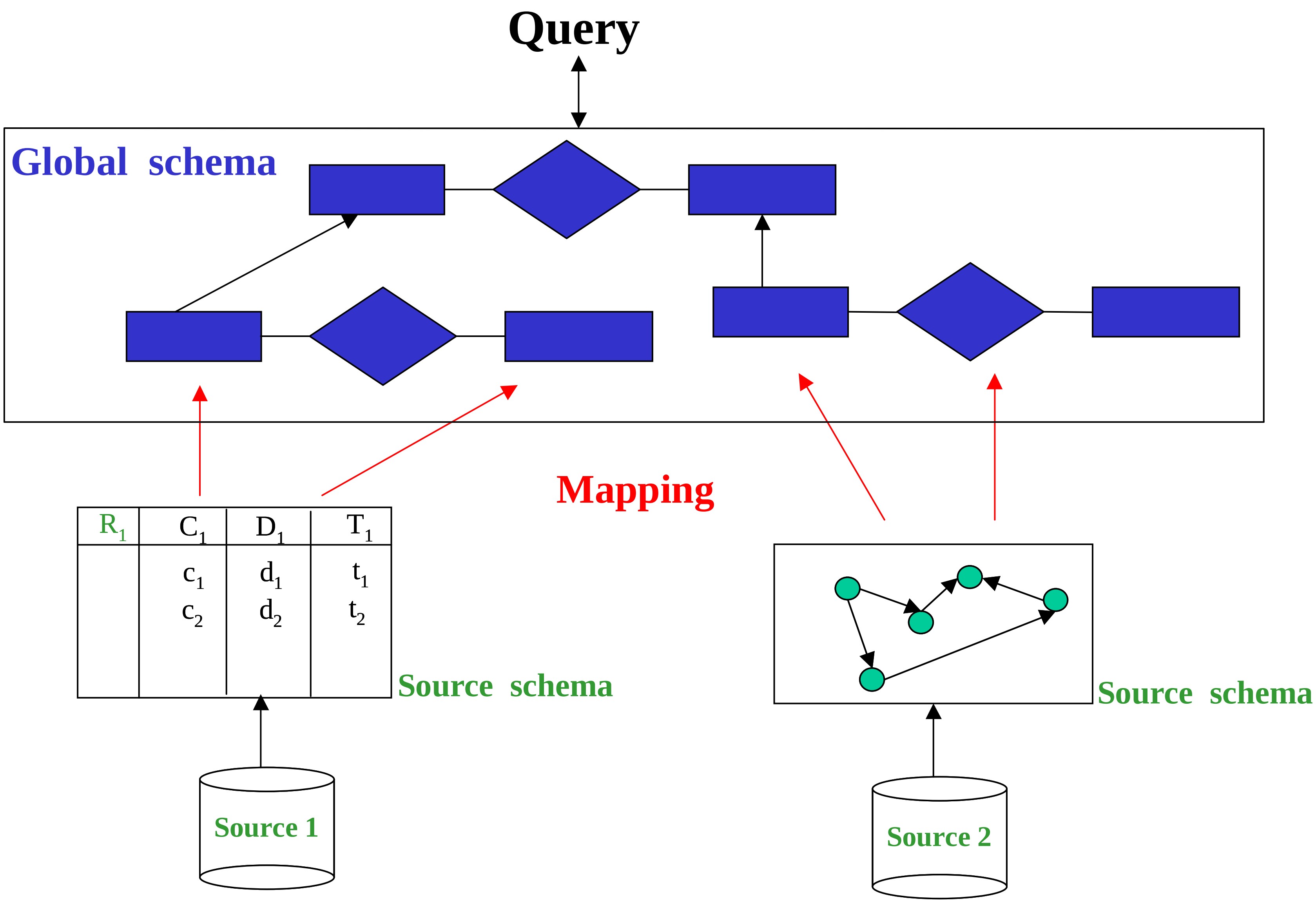
Operations:

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

Outline

- Data integration architectures
- Centralized data integration
- P2P data integration
- Hyper: epistemic semantics for P2P data integration
- Dealing with inconsistencies in Hyper
- 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 $\mathcal{I}$?

Let $\mathcal{D}$ be a source database (also called source model) over a fixed infinite domain Γ of constants, fixing the extension of the predicates of $\mathcal{A}_{\mathcal{S}}$.

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

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

What it means to satisfy $\mathcal{M}$ wrt $\mathcal{D}$ 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 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 $\mathcal{D}$ is

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

Note: query answering is logical implication.

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

GAV vs LAV – example

Global schema: movie$(Title, Year, Director)$

european$(Director)$

review$(Title, Critique)$

Source 1: r$_1(Title, Year, Director)$ since 1960, european directors

Source 2: r$_2(Title, Critique)$ since 1990

Query: Title and critique of movies in 1998

$\exists D.$ movie$(T, 1998, D) \land$ review(T, R), written

$\{ (T, R) \mid$ movie$(T, 1998, D) \land$ review$(T, R) \}$

Formalization of LAV

In LAV, the mapping $\mathcal{M}$ is constituted by a set of assertions:

$$s \quad \leadsto \quad \phi_{\mathcal{G}} \quad \text{(sound source)} \quad \forall \vec{\mathbf{x}}\, (s(\vec{\mathbf{x}}) \rightarrow \phi_{\mathcal{G}}(\vec{\mathbf{x}}))$$

one for each source element s in $\mathcal{A}_{\mathcal{S}}$, where $\phi_{\mathcal{G}}$ is a query over $\mathcal{G}$. Given source database $\mathcal{C}$, a database $\mathcal{B}$ for $\mathcal{G}$ satisfies $\mathcal{M}$ wrt $\mathcal{C}$ if for each $s \in \mathcal{S}$:

$$s^{\mathcal{C}} \quad \subseteq \quad {\phi_{\mathcal{G}}}^{\mathcal{B}} \quad \text{(sound source)}$$

The mapping $\mathcal{M}$ and the source database $\mathcal{C}$ do not provide direct information about which data satisfy the global schema. Sources are views, and we have to answer queries on the basis of the available data in the views.

LAV – example

Global schema: movie$(Title, Year, Director)$
european$(Director)$
review$(Title, Critique)$

LAV: associated to source relations we have views over the global schema

$\mathsf{r}_1(T, Y, D) \rightsquigarrow \{\, (T, Y, D) \mid \mathsf{movie}(T, Y, D) \wedge \mathsf{european}(D) \wedge Y \geq 1960 \,\}$

$\mathsf{r}_2(T, R) \rightsquigarrow \{\, (T, R) \mid \mathsf{movie}(T, Y, D) \wedge \mathsf{review}(T, R) \wedge Y \geq 1990 \,\}$

The query $\{\, (T, R) \mid \mathsf{movie}(T, 1998, D) \wedge \mathsf{review}(T, R) \,\}$ is processed by means of an inference mechanism that aims at re-expressing the atoms of the global schema in terms of atoms at the sources. In this case:

$$\{\, (T, R) \mid \mathsf{r}_2(T, R) \wedge \mathsf{r}_1(T, 1998, D) \,\}$$

Formalization of GAV

In GAV, the mapping $\mathcal{M}$ is constituted by a set of assertions:

$$g \quad \leadsto \quad \phi_{\mathcal{S}} \quad \text{(sound source)} \quad \forall \vec{\mathbf{x}}\, (\phi_{\mathcal{S}}(\vec{\mathbf{x}}) \rightarrow g(\vec{\mathbf{x}}))$$

one for each element g in $\mathcal{A}_{\mathcal{G}}$, where $\phi_{\mathcal{S}}$ is a query over $\mathcal{S}$. Given source database $\mathcal{C}$, a database $\mathcal{B}$ for $\mathcal{G}$ satisfies $\mathcal{M}$ wrt $\mathcal{C}$ if for each $g \in \mathcal{G}$:

$$g^{\mathcal{B}} \quad \supseteq \quad {\phi_{\mathcal{S}}}^{\mathcal{C}} \quad \text{(sound source)}$$

Given a source database, $\mathcal{M}$ provides direct information about which data satisfy the elements of the global schema. Relations in $\mathcal{G}$ are views, and queries are expressed over the views. Thus, it seems that we can simply evaluate the query over the data satisfying the global relations (as if we had a single database at hand).

GAV – example

Global schema: movie$(Title, Year, Director)$

european$(Director)$

review$(Title, Critique)$

GAV: associated to relations in the global schema we have views over the sources

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

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

$$\text{review}(T, R) \quad \leadsto \quad \{ \; (T, R) \mid \mathsf{r}_2(T, R) \; \}$$

GAV – example of query processing

The query $\{ \ (T, R) \mid \text{movie}(T, 1998, D) \wedge \text{review}(T, R) \ \}$ is processed by means of unfolding, i.e., by expanding the atoms according to their definitions, so as to come up with source relations. In this case:

$$\text{movie}(T,1998,D) \wedge \text{review}(T,R)$$

unfolding

$$r_1(T,1998,D) \wedge r_2(T,R)$$

GAV and LAV – comparison

LAV: (Information Manifold, DWQ, Picsel)

- Quality depends on how well we have characterized the sources
- High modularity and extensibility (if the global schema is well designed, when a source changes, only its definition is affected)
- Query processing needs reasoning (query reformulation complex)

GAV: (Carnot, SIMS, Tsimmis, IBIS, Picsel, Momis, DisAtDis, . . .)

- Quality depends on how well we have compiled the sources into the global schema through the mapping
- Whenever a source changes or a new one is added, the global schema needs to be reconsidered
- Query processing can be based on some sort of unfolding (query reformulation looks easier)

For more details, see [Ullman TCS'00], [Halevy VLDBJ'01], [Lenzerini PODS'02].

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 $\mathcal{D}$, a database $\mathcal{B}$ satisfies $\mathcal{M}$ wrt $\mathcal{D}$ if for each assertion in $\mathcal{M}$ we have ${\phi_{\mathcal{S}}}^{\mathcal{D}} \subseteq {\phi_{\mathcal{G}}}^{\mathcal{B}}$, i.e., the assertion means $\forall \vec{\mathbf{x}}\, (\phi_{\mathcal{S}}(\vec{\mathbf{x}}) \rightarrow \phi_{\mathcal{G}}(\vec{\mathbf{x}}))$.

To answer a query q over $\mathcal{G}$, we have to infer how to use $\mathcal{M}$ in order to access the source database $\mathcal{D}$.

Example of GLAV

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

Source 1: $HasJob(Person, Field)$

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

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

GLAV mapping:

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

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

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

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
- Class of queries allowed
  - client queries
  - queries in the mapping

Incompleteness and inconsistency

Constraints in $\mathcal{G}$	Type of mapping	Incompleteness	Inconsistency
no	GAV	yes/no	no
no	(G)LAV	yes	no
yes	GAV	yes	yes
yes	(G)LAV	yes	yes

LAV without constraints: basic technique

Consider conjunctive queries and conjunctive views.

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

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

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

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

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

GAV with constraints: incompleteness and inconsistency

Let us consider a system with a global schema with constraints, and with a GAV mapping $\mathcal{M}$ with sound sources, whose assertions $g \leadsto \phi_{\mathcal{S}}$ have the logical form

$$\forall \vec{\mathbf{x}} \;\; \phi_{\mathcal{S}}(\vec{\mathbf{x}}) \rightarrow g(\vec{\mathbf{x}})$$

where $\phi_{\mathcal{S}}$ is a conjunctive query, and g is an element of $\mathcal{G}$.

Basic observation: since $\mathcal{G}$ does have constraints, the global database obtained by "applying" the mapping may not be legal for $\mathcal{G}$.

GAV with constraints: example

Global schema $\mathcal{G}$:

student$(Scode, Sname, Scity)$, $\quad key\{Scode\}$

university$(Ucode, Uname)$, $\quad key\{Ucode\}$

enrolled$(Scode, Ucode)$, $\quad key\{Scode, Ucode\}$

$$\text{enrolled}[Scode] \subseteq \text{student}[Scode]$$
$$\text{enrolled}[Ucode] \subseteq \text{university}[Ucode]$$

Sources $\mathcal{S}$: database relations $\mathsf{s}_1(X, Y, Z), \mathsf{s}_2(X, Y), \mathsf{s}_3(X, Y)$

Mapping $\mathcal{M}$:

$$\text{student}(X, Y, Z) \rightsquigarrow \{ (X, Y, Z) \mid \mathsf{s}_1(X, Y, Z, W) \}$$
$$\text{university}(X, Y) \rightsquigarrow \{ (X, Y) \mid \mathsf{s}_2(X, Y) \}$$
$$\text{enrolled}(X, Y) \rightsquigarrow \{ (X, Y) \mid \mathsf{s}_3(X, Y) \}$$

GAV with constraints: example

University

code	name
AF	*bocconi*
BN	*ucla*

Student

code	name	city
15	*bill*	*oslo*
12	*anne*	*florence*

Enrolled

Scode	Ucode
12	*AF*
16	*BN*

$s_1^{\mathcal{C}}$

12	*anne*	*florence*	21
15	*bill*	*oslo*	24

$s_2^{\mathcal{C}}$

AF	*bocconi*
BN	*ucla*

$s_3^{\mathcal{C}}$

12	AF
16	BN

Example of source database and corresponding retrieved global database

GAV with constraints: example of incompleteness

Source database $\mathcal{C}$:

$s_1^{\mathcal{C}}$

12	*anne*	*florence*	21
15	*bill*	*oslo*	24

$s_2^{\mathcal{C}}$

AF	*bocconi*
BN	*ucla*

$s_3^{\mathcal{C}}$

12	AF
16	BN

$s_3^{\mathcal{C}}(16, BN)$ and the mapping imply enrolled$^{\mathcal{B}}(16, BN)$, for all $\mathcal{B} \in sem^{\mathcal{C}}(\mathcal{I})$.

Due to the integrity constraints in the global schema, 16 is the code of some student in all $\mathcal{B} \in sem^{\mathcal{C}}(\mathcal{I})$.

Since $\mathcal{C}$ says nothing about the name and the city of the student with code 16, we must accept as legal for $\mathcal{I}$ wrt $\mathcal{C}$ all virtual global databases that differ in such attributes.

GAV with constraints: unfolding is not sufficient

Mapping $\mathcal{M}$:

$$\text{student}(X,Y,Z) \leadsto \{ (X,Y,Z) \mid \text{s}_1(X,Y,Z,W) \}$$

$$\text{university}(X,Y) \leadsto \{ (X,Y) \mid \text{s}_2(X,Y) \}$$

$$\text{enrolled}(X,Y) \leadsto \{ (X,Y) \mid \text{s}_3(X,Y) \}$$

$s_1^{\mathcal{C}}$

12	*anne*	*florence*	21
15	*bill*	*oslo*	24

$s_2^{\mathcal{C}}$

AF	*bocconi*
BN	*ucla*

$s_3^{\mathcal{C}}$

12	AF
16	BN

Query: $\{ (X) \mid \text{student}(X,Y,Z), \text{enrolled}(X,W) \}$

Unfolding wrt $\mathcal{M}$: $\{ (X) \mid s_1(X,Y,Z,V), s_3(X,W) \}$

retrieves only the answer $\{12\}$ from $\mathcal{C}$, although $\{12, 16\}$ is the correct answer. The simple unfolding strategy is not sufficient in our context.

Most GAV systems use the simple unfolding strategy!

GAV with constraints: example of inconsistency

Source database $\mathcal{C}$:

$s_1^{\mathcal{C}}$

12	*anne*	*florence*	21
12	*bill*	*oslo*	24

$s_2^{\mathcal{C}}$

AF	*bocconi*
BN	*ucla*

$s_3^{\mathcal{C}}$

12	AF
16	BN

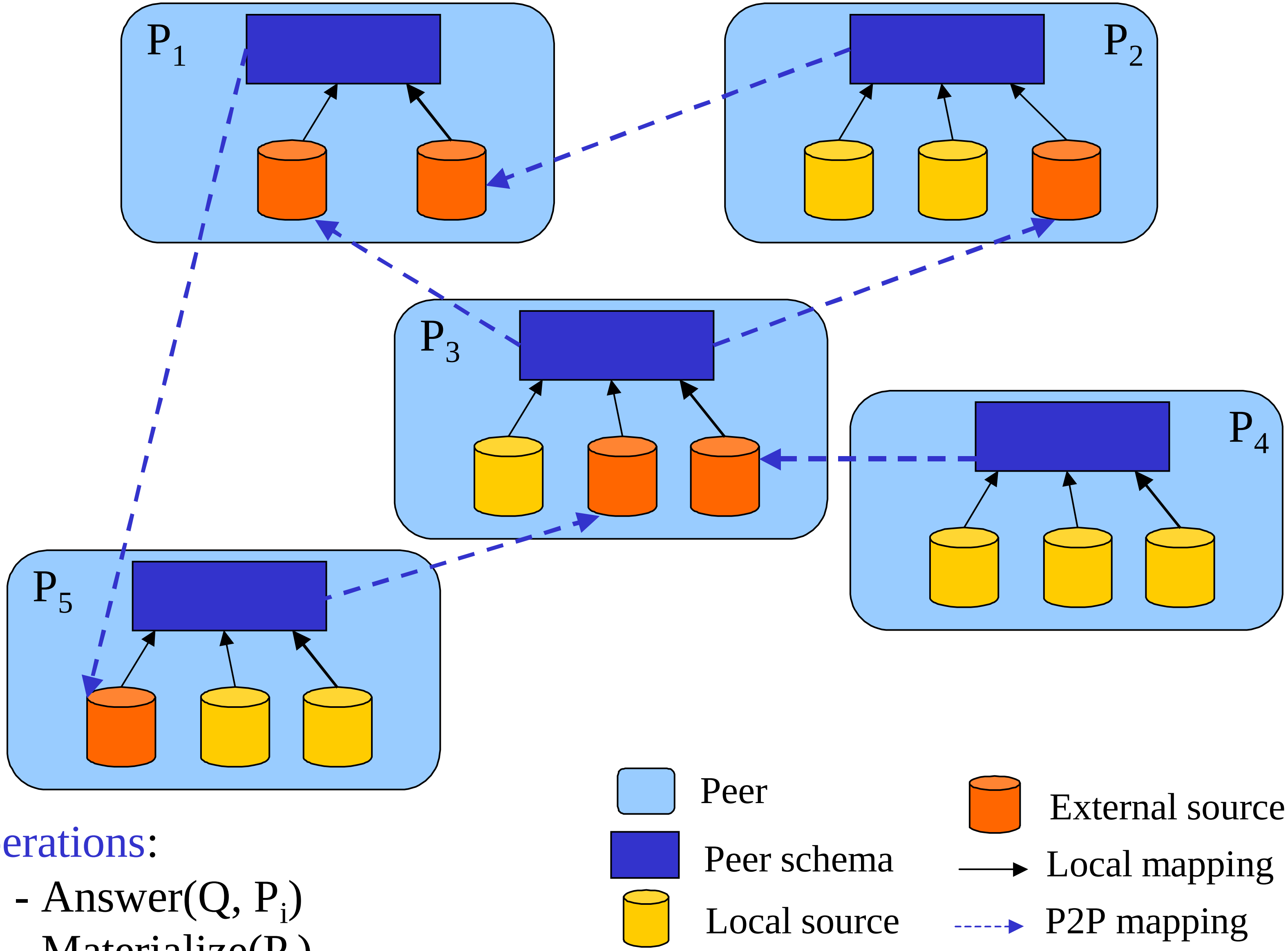
$s_1^{\mathcal{C}}$ imply student$^{\mathcal{B}}(12,$ *anne*$,$ *florence*$, 21)$, and student$^{\mathcal{B}}(12,$ *bill*$,$ *oslo*$, 24)$, for all $\mathcal{B}$ that satisfies the mapping.

Due to the integrity constraints in the global schema, it follows that there is no database that satisfies both the mapping and the global schema, i.e., $sem^{\mathcal{C}}(\mathcal{I}) = \emptyset$.

Outline

- Data integration architectures
- Centralized data integration
- $\boxed{\text{P2P data integration}}$
- Hyper: epistemic semantics for P2P data integration
- Dealing with inconsistencies in Hyper
- Conclusions

P2P data integration: the general picture



Operations:

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

P2P data integration: what's in a peer

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, whose alphabet is considered disjoint from that of the other peers
- 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 of another peer P_j to one external source in P_i

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

P2P data integration: local and P2P mappings

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

- each local mapping in L has the form

$$\mathit{ep}_S \;\leadsto\; cq_G$$

  where ep_S is an extraction program on the sources S and cq_G is a conjunctive queries over G, respectively

- each P2P mapping assertion 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

Importance of the semantics

The client sees the whole collection of peers through the eyes of one peer, and (s)he 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 formulas 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 a set of local source databases, one local 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:

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

  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 relative to $\mathcal{D}$ are

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

Outline

- Data integration architectures
- Centralized data integration
- P2P data integration
- Hyper: epistemic semantics for P2P data integration
- Dealing with inconsistencies in Hyper
- Conclusions

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 implications in logic

- 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 usually given in terms of First-Order Logic, e.g., [Halevy&al. ICDE'03]

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 multi-agent epistemic logic [Reiter TARK'88], [Fagin&al 1995]

- 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 known to satisfy s_j.

  In other words, peer P_i communicates to peer P_j only facts that are known, 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 relative to a source database $\mathcal{D}$ is polynomial time in the size of $\mathcal{D}$, even for cyclic mappings

Epistemic logic: basic notions

In multi-agent epistemic logic we want to model the knowledge of n agents

From a language point of views, we have a new form of atom, namely (φ is again a formula, and $i \in \{1, \ldots, n\}$):

$$\mathbf{K}_i\, \varphi$$

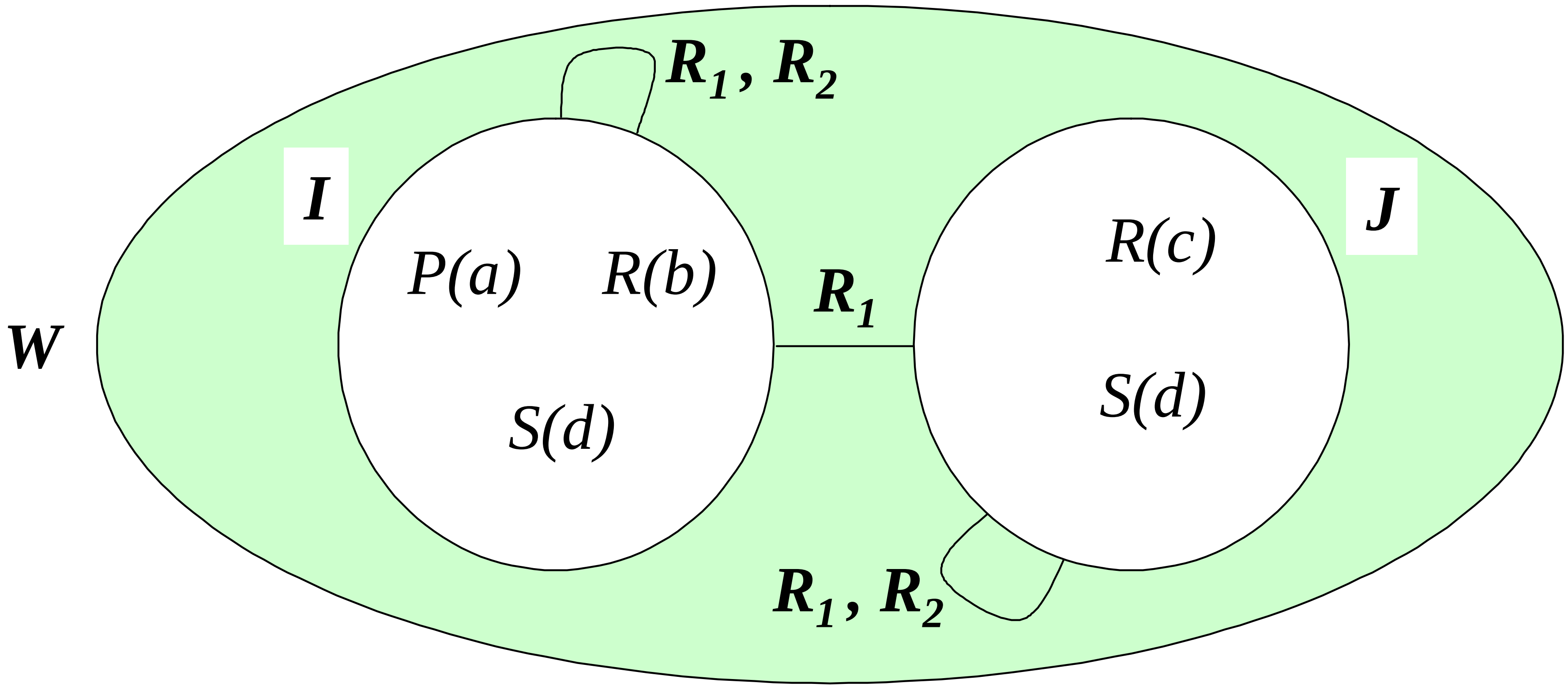
An epistemic interpretation is a tuple $\langle \mathcal{I}, \mathcal{W}, \mathcal{R}_1, \ldots, \mathcal{R}_n \rangle$, where

- $\mathcal{W}$ is a set of FOL interpretations and $\mathcal{I} \in \mathcal{W}$
- each $\mathcal{R}_i$ is a binary relation over $\mathcal{W}$

We use the $S5_n$ modal system, and therefore, we also impose that

- each $\mathcal{R}_i$ is an equivalence relation, i.e., it is reflexive, symmetric, and transitive

Epistemic interpretation: example



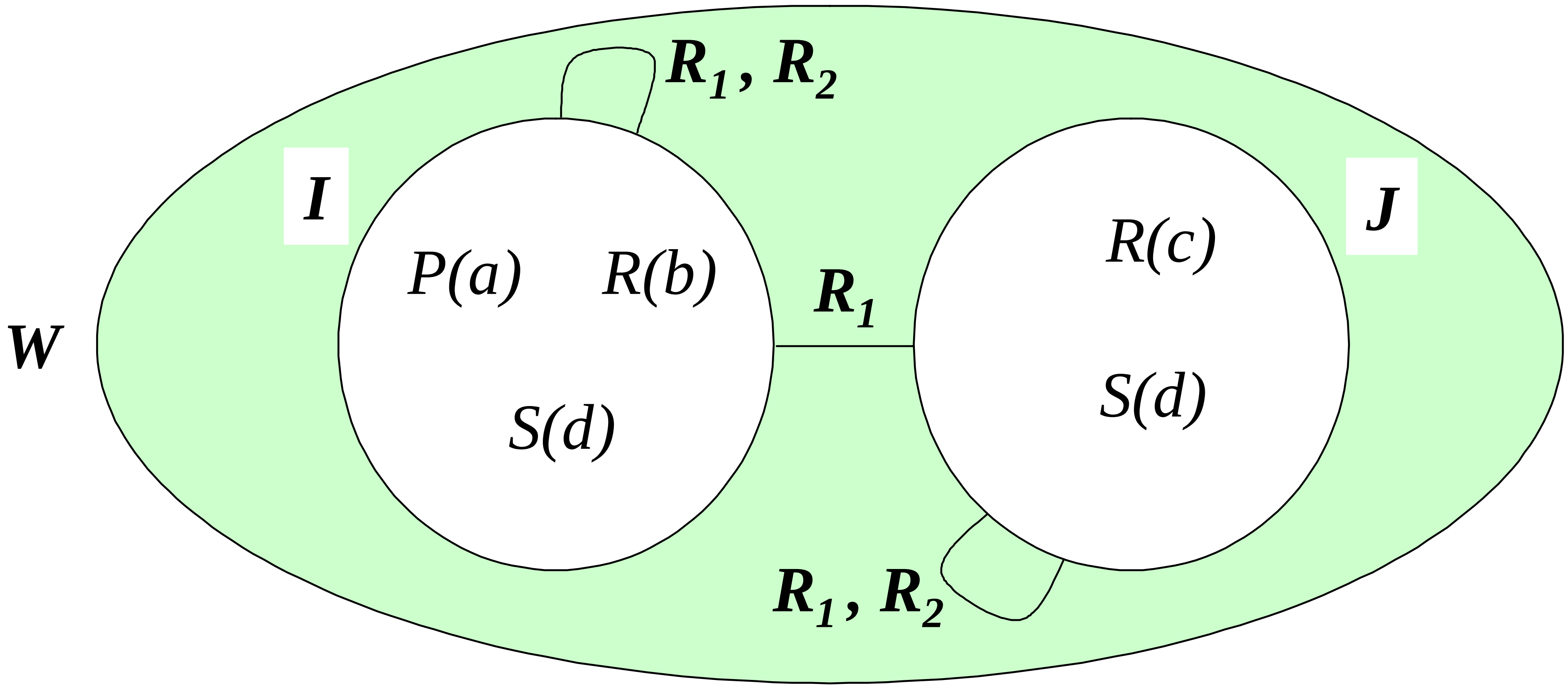
$\langle \mathcal{I}, \mathcal{W}, \mathcal{R}_1, \mathcal{R}_2 \rangle$ and $\langle \mathcal{J}, \mathcal{W}, \mathcal{R}_1, \mathcal{R}_2 \rangle$ are epistemic interpretations

Epistemic logic: basic notions

- a FOL formula constituted by an atom $a(\vec{\mathbf{x}})$ is satisfied in $\langle \mathcal{I}, \mathcal{W}, \mathcal{R}_1, \ldots, \mathcal{R}_n \rangle$ by the tuples $\vec{\mathbf{t}}$ of constants such that $a(\vec{\mathbf{t}})$ is true in $\mathcal{I}$
- an atom of the form $\mathbf{K}_i \varphi(\vec{\mathbf{x}})$ is satisfied in $\langle \mathcal{I}, \mathcal{W}, \mathcal{R}_1, \ldots, \mathcal{R}_n \rangle$ by the tuples $\vec{\mathbf{t}}$ of constants such that $\varphi(\vec{\mathbf{t}})$ is satisfied in all epistemic interpretations $\langle \mathcal{J}, \mathcal{W}, \mathcal{R}_1, \ldots, \mathcal{R}_n \rangle$ with $(\mathcal{I}, \mathcal{J}) \in \mathcal{R}_i$

An epistemic model of an epistemic logic theory $\{\varphi_1, \ldots, \varphi_t\}$ is an epistemic interpretation that satisfies every formula $\varphi_i \in \{\varphi_1, \ldots, \varphi_t\}$.

Epistemic logic: example 1

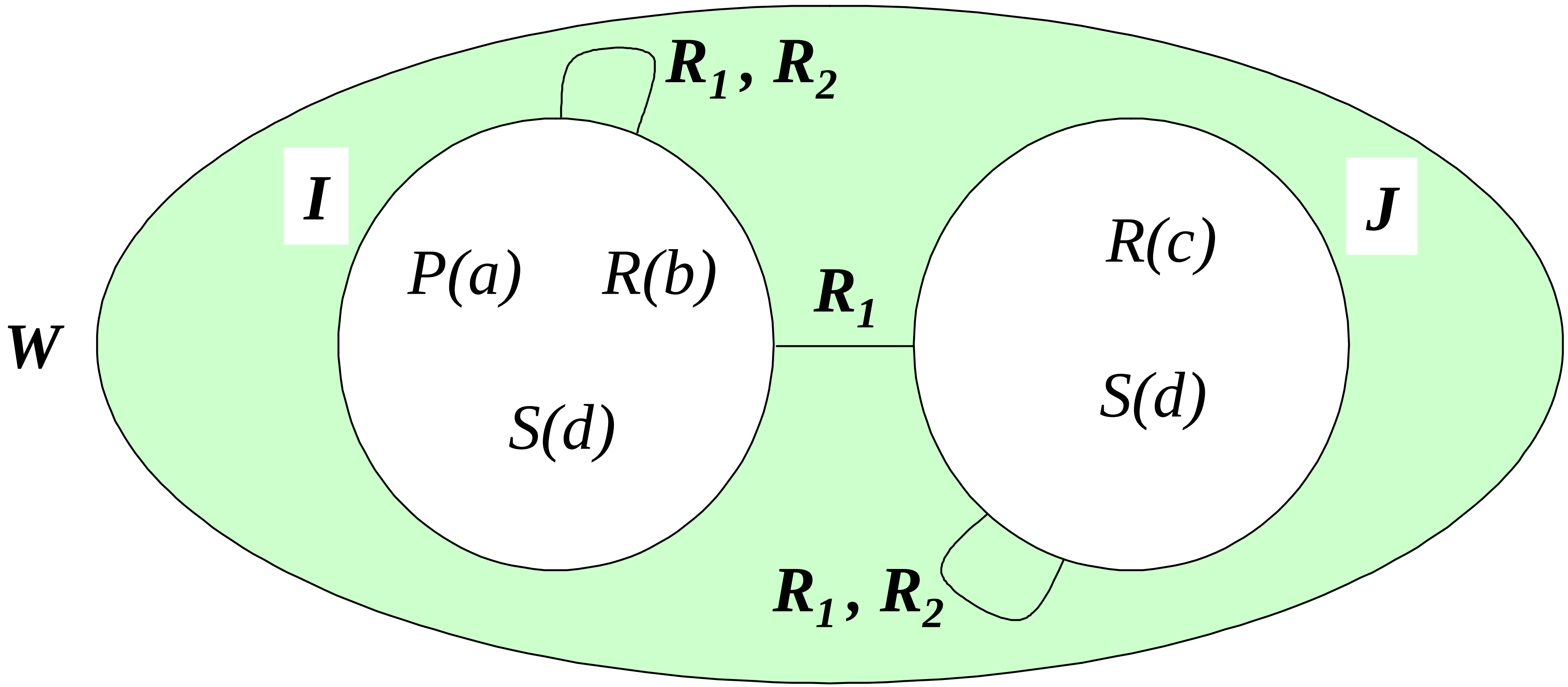


$$\langle \mathcal{I}, \mathcal{W}, \mathcal{R}_1, \mathcal{R}_2 \rangle \models P(a)$$

$$\langle \mathcal{J}, \mathcal{W}, \mathcal{R}_1, \mathcal{R}_2 \rangle \not\models P(a)$$

$$\langle \mathcal{I}, \mathcal{W}, \mathcal{R}_1, \mathcal{R}_2 \rangle \not\models \mathbf{K}_1\, P(a)$$

Epistemic logic: example 2

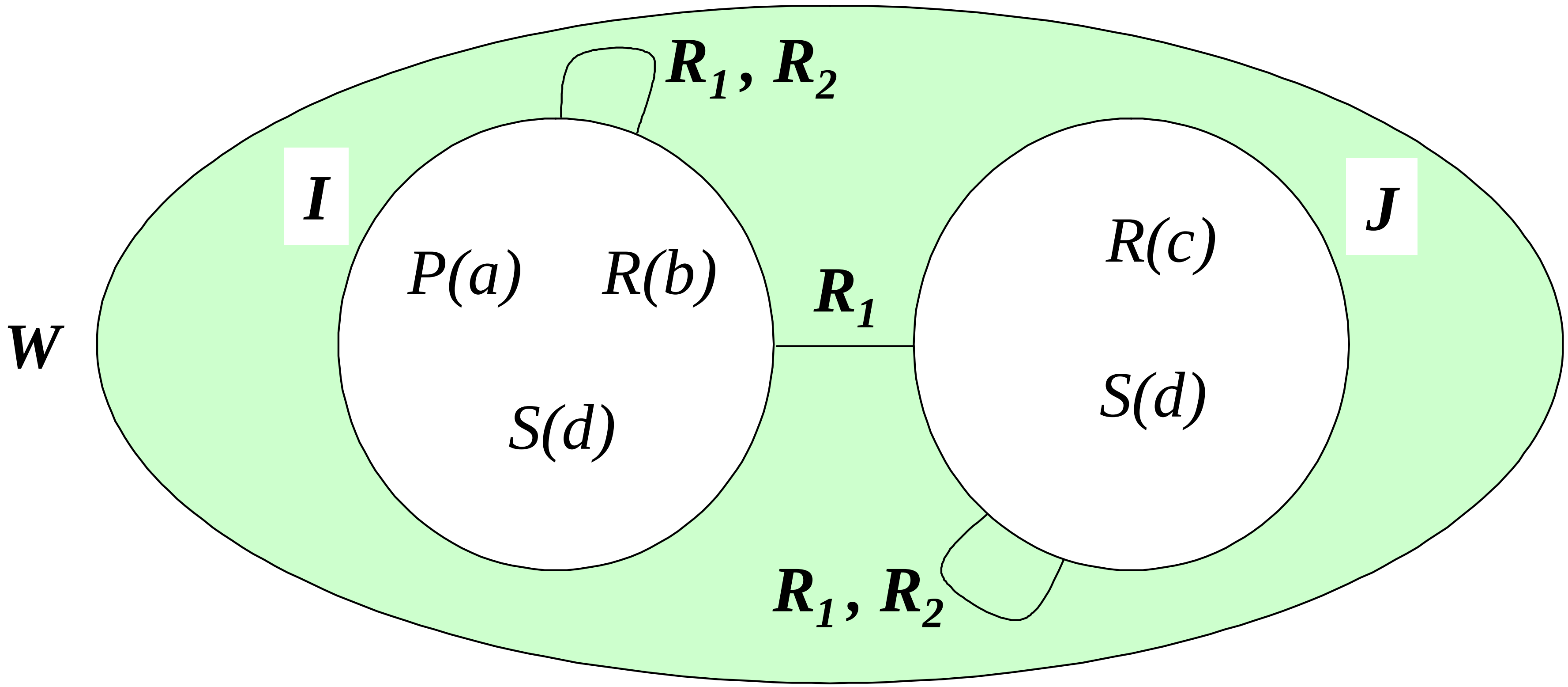


$$\langle \mathcal{I}, \mathcal{W}, \mathcal{R}_1, \mathcal{R}_2 \rangle \models \mathbf{K}_1 \left(R(b) \vee R(c)\right)$$

$$\langle \mathcal{I}, \mathcal{W}, \mathcal{R}_1, \mathcal{R}_2 \rangle \not\models (\mathbf{K}_1 R(b)) \vee (\mathbf{K}_1 R(c))$$

$$\langle \mathcal{J}, \mathcal{W}, \mathcal{R}_1, \mathcal{R}_2 \rangle \models \mathbf{K}_2 \mathbf{K}_1 S(d)$$

Epistemic logic: example 3



$$\langle \mathcal{I}, \mathcal{W}, \mathcal{R}_1, \mathcal{R}_2 \rangle \models \mathbf{K}_1 (\exists x\, R(x))$$

$$\langle \mathcal{I}, \mathcal{W}, \mathcal{R}_1, \mathcal{R}_2 \rangle \not\models \exists x\, (\mathbf{K}_1\, R(x))$$

$$\langle \mathcal{J}, \mathcal{W}, \mathcal{R}_1, \mathcal{R}_2 \rangle \models \exists x\, (\mathbf{K}_2\, S(x))$$

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
- For each peer theory T_{P_i}, for each formula $\varphi \in T_{P_i}$, the following formula is in E_Π

$$\mathbf{K}_i\, \varphi$$

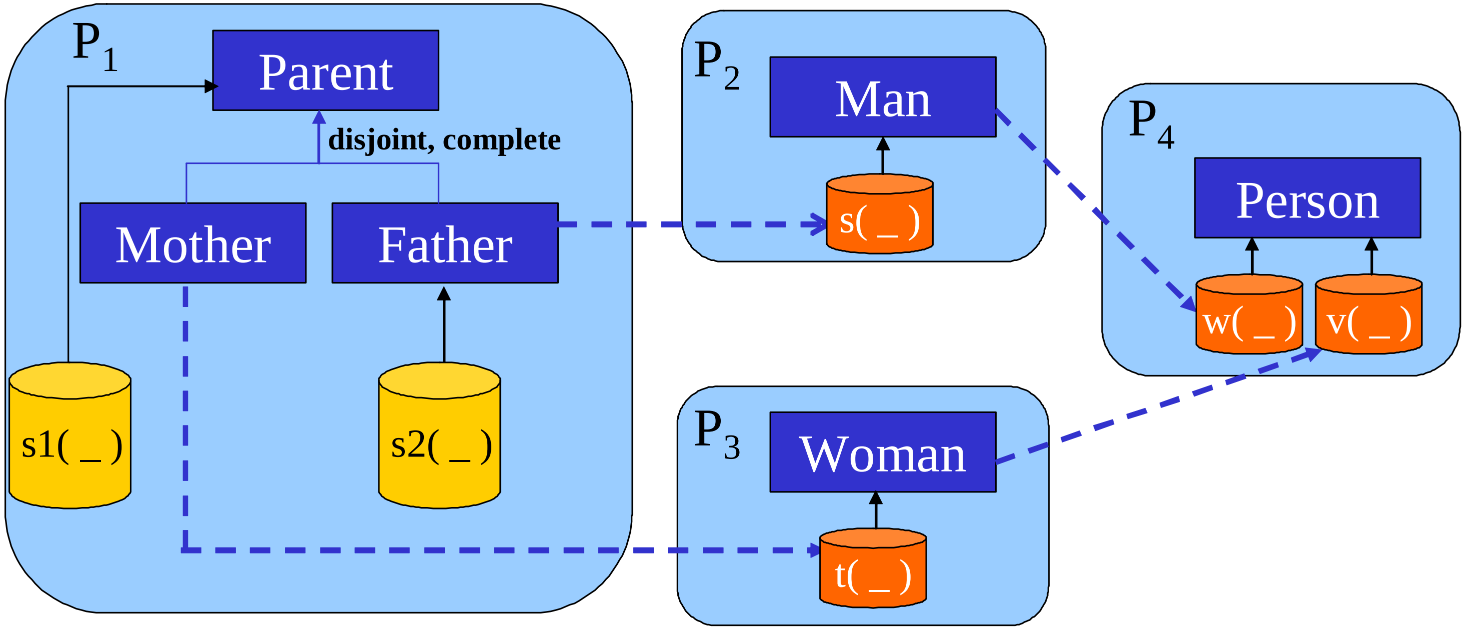
- for each P2P mapping assertion from peer P_i to peer P_j

$$\{\vec{\mathbf{x}} \mid \exists \vec{\mathbf{y}}\, \varphi(\vec{\mathbf{x}}, \vec{\mathbf{y}})\}_i \rightsquigarrow \{\vec{\mathbf{x}} \mid s(\vec{\mathbf{x}})\}_j$$

in the peers of Π, the following formula is in E_Π

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

Epistemic semantics for P2P mappings: example of theory



$$\mathbf{K}_1\, \forall x((Parent(x) \equiv Mother(x) \vee Father(x)) \wedge (Mother(x) \supset \neg Father(x)))$$

$$\mathbf{K}_1\, \forall x(s1(x) \supset Parent(x)) \qquad \mathbf{K}_1\, \forall x(s2(x) \supset Father(x))$$

$$\mathbf{K}_2\, \forall x(s(x) \supset Man(x)) \qquad \mathbf{K}_3\, \forall x(t(x) \supset Woman(x))$$

$$\mathbf{K}_4\, \forall x(w(x) \supset Person(x)) \qquad \mathbf{K}_4\, \forall x(v(x) \supset Person(x))$$

$$\forall x(\mathbf{K}_1 Mother(x) \equiv \mathbf{K}_3 t(x)) \qquad \forall x(\mathbf{K}_1 Father(x) \equiv \mathbf{K}_2 s(x))$$

$$\forall x(\mathbf{K}_2 Man(x) \equiv \mathbf{K}_4 w(x)) \qquad \forall x(\mathbf{K}_3 Woman(x) \equiv \mathbf{K}_4 v(x))$$

Epistemic semantics for P2P mappings: basic idea

An epistemic model of Π relative to the source database $\mathcal{D}$ is an epistemic model $\langle \mathcal{I}, \mathcal{W}, \mathcal{R}_1, \ldots, \mathcal{R}_n \rangle$ of E_Π such that, in every $\mathcal{J} \in \mathcal{W}$, the extension of all local sources in all peers is the one sanctioned by $\mathcal{D}$.

If $\langle \mathcal{I}, \mathcal{W}, \mathcal{R}_1, \ldots, \mathcal{R}_n \rangle$ is an epistemic model of Π relative to $\mathcal{D}$, then $\langle \mathcal{I}, \mathcal{W}, \mathcal{R}_1, \ldots, \mathcal{R}_n \rangle$ satisfies all formulas of E_Π, and therefore

- for each i, $\mathcal{I}$ is a model of T_{P_i} relative to $\mathcal{D}$, which means that $\mathcal{I}$ satisfies all constraints in the various peer schemas, and
- $\langle \mathcal{I}, \mathcal{W}, \mathcal{R}_1, \ldots, \mathcal{R}_n \rangle$ satisfies all formulas corresponding to the P2P mapping assertions in the peers of Π

Epistemic semantics for P2P mappings: basic idea

Note that $\langle \mathcal{I}, \mathcal{W}, \mathcal{R}_1, \ldots, \mathcal{R}_n \rangle$ satisfying the P2P mapping assertion $cq_h \leadsto s_k$ means that

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

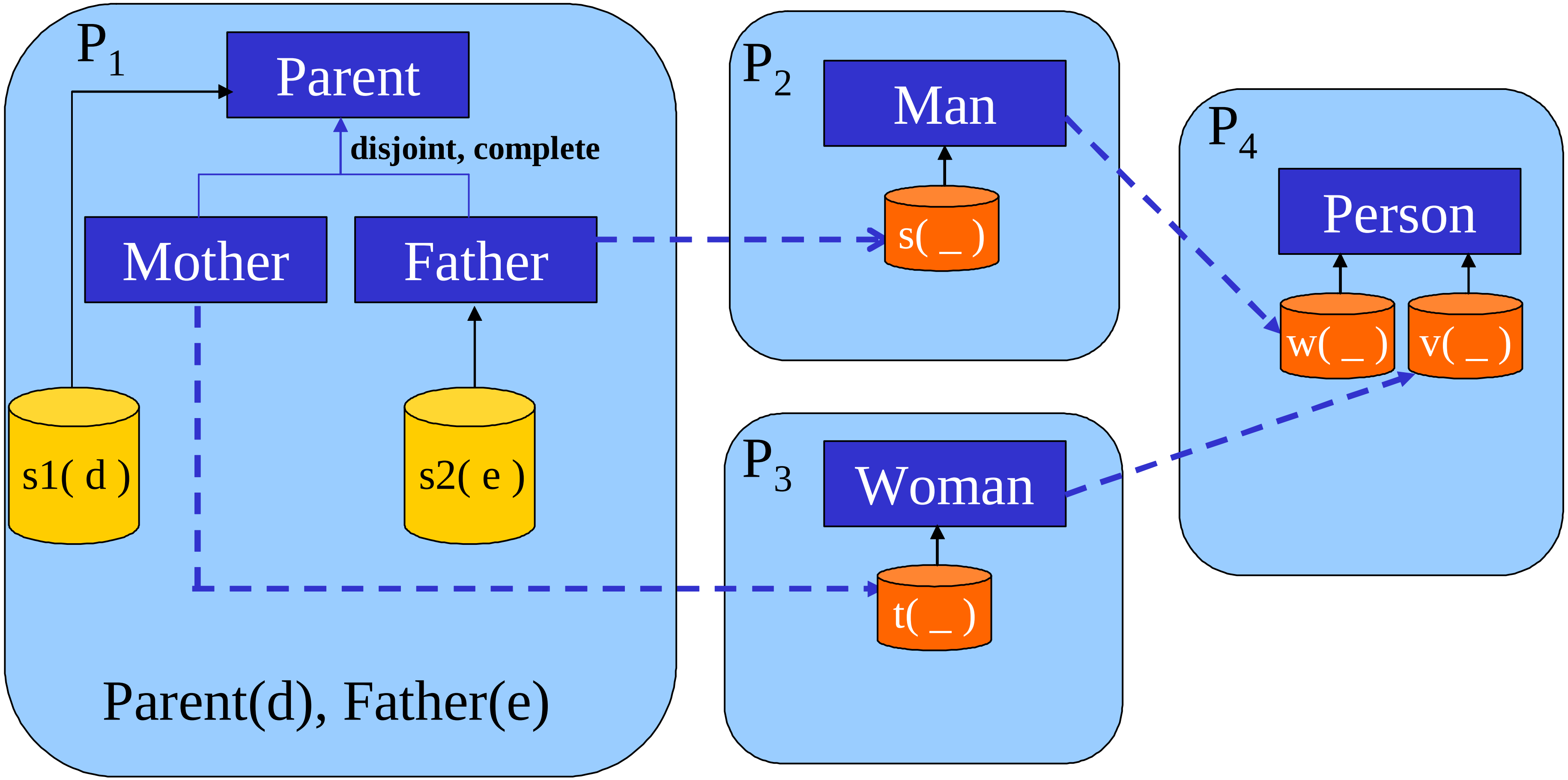
if $\vec{\mathbf{t}} \in cq^{\mathcal{J}}$ for every FOL model $\mathcal{J}$ such that $(\mathcal{I}, \mathcal{J}) \in \mathcal{R}_h$,

then $\vec{\mathbf{t}} \in s^{\mathcal{J}}$ for every FOL model $\mathcal{J}$ such that $(\mathcal{I}, \mathcal{J}) \in \mathcal{R}_k$

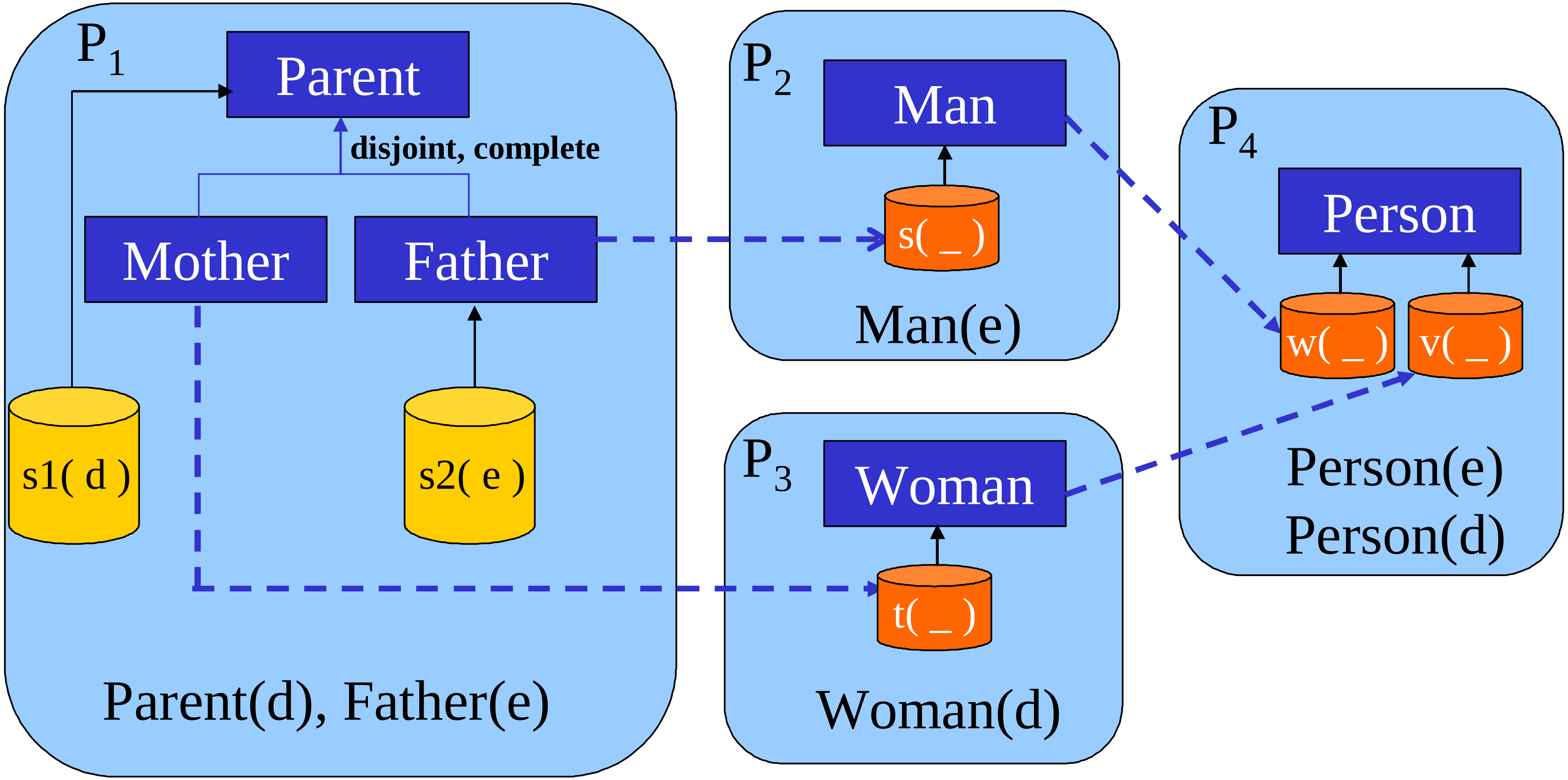
Given a query Q of arity n posed to a peer P_i of Π, and a source database $\mathcal{D}$, the certain answers to Q relative to $\mathcal{D}$ under the epistemic semantics are

$$ans_{\mathbf{k}}(Q, \Pi, \mathcal{D}) = \{ \vec{\mathbf{t}} \in \Gamma^n \mid \vec{\mathbf{t}} \in Q^{\mathcal{I}}, \text{ for every epistemic model } \langle \mathcal{I}, \mathcal{W}, \mathcal{R}_1, \ldots, \mathcal{R}_n \rangle \text{ of } \Pi \text{ relative to } \mathcal{D} \}$$

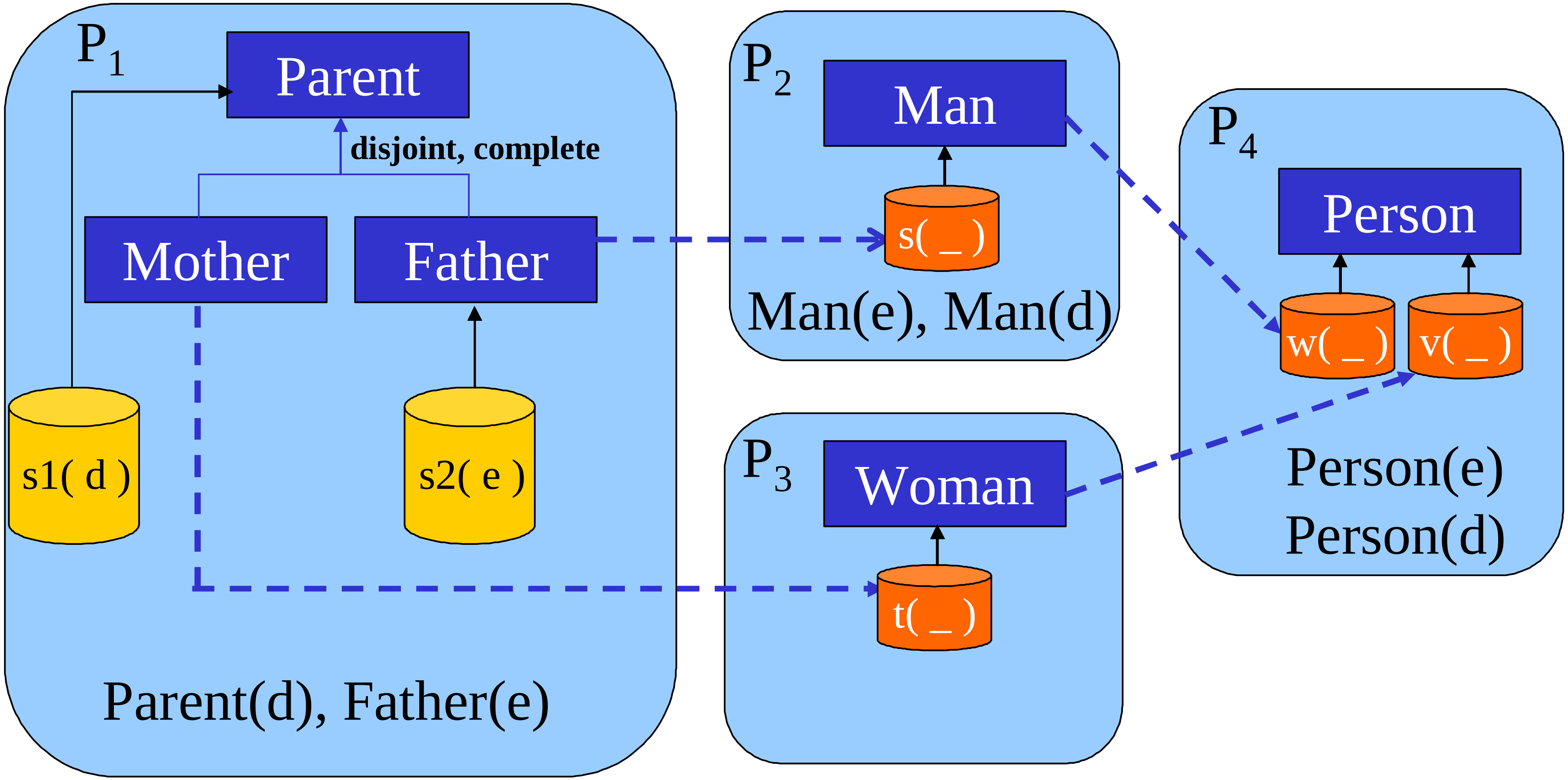
Semantics of P2P mappings: example



FOL semantics of P2P mappings: model 1

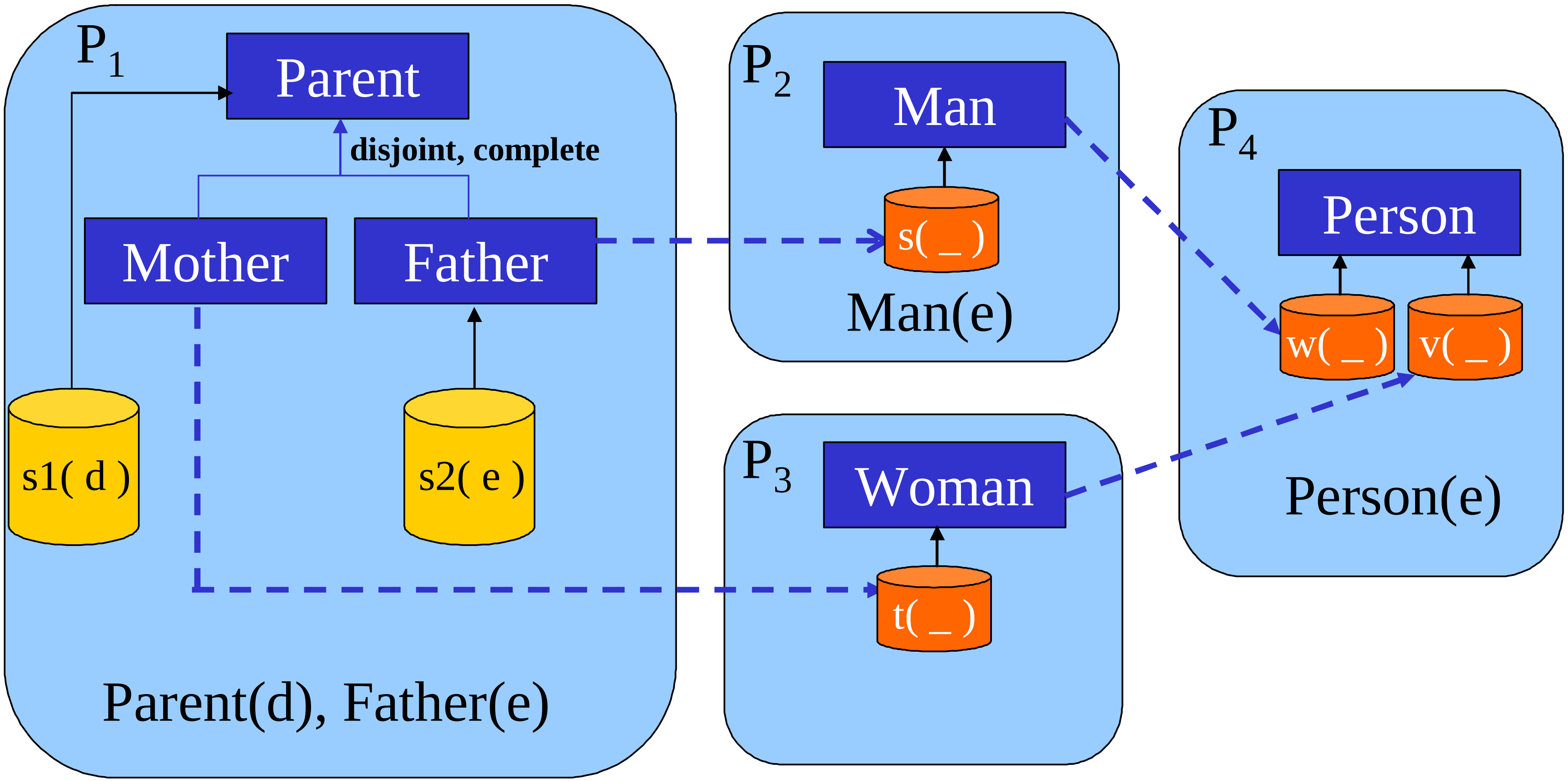


FOL semantics of P2P mappings: model 2



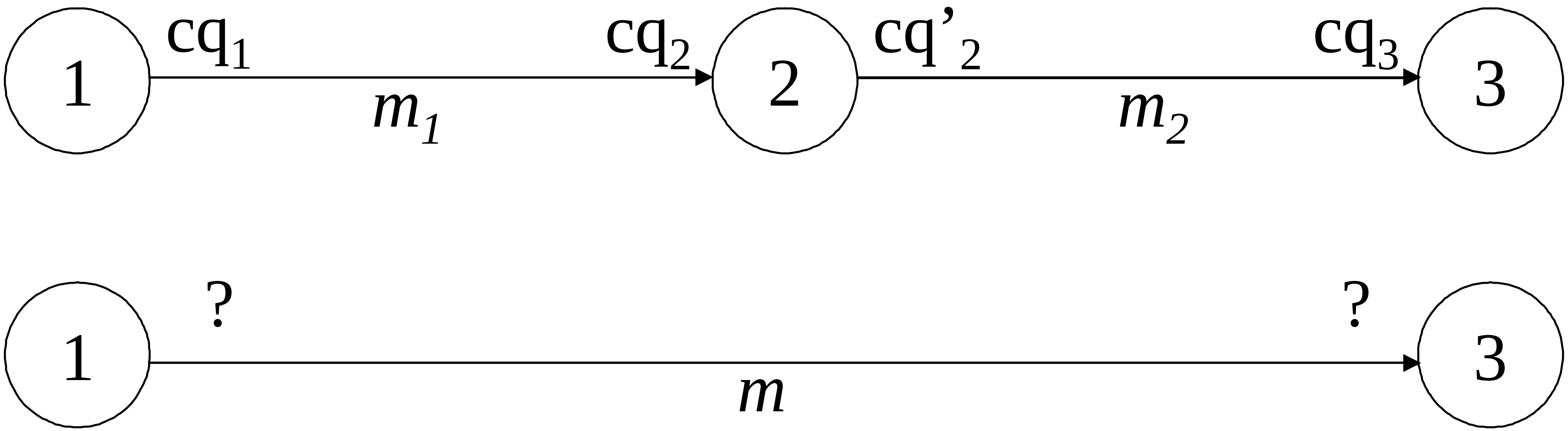
According to the FOL semantics, $\mathrm{Person}(\mathrm{d})$ is true in all cases, and therefore is a certain answer to $\{\mathrm{x} \mid \mathrm{Person}(\mathrm{x})\}$

Epistemic semantics of P2P mappings



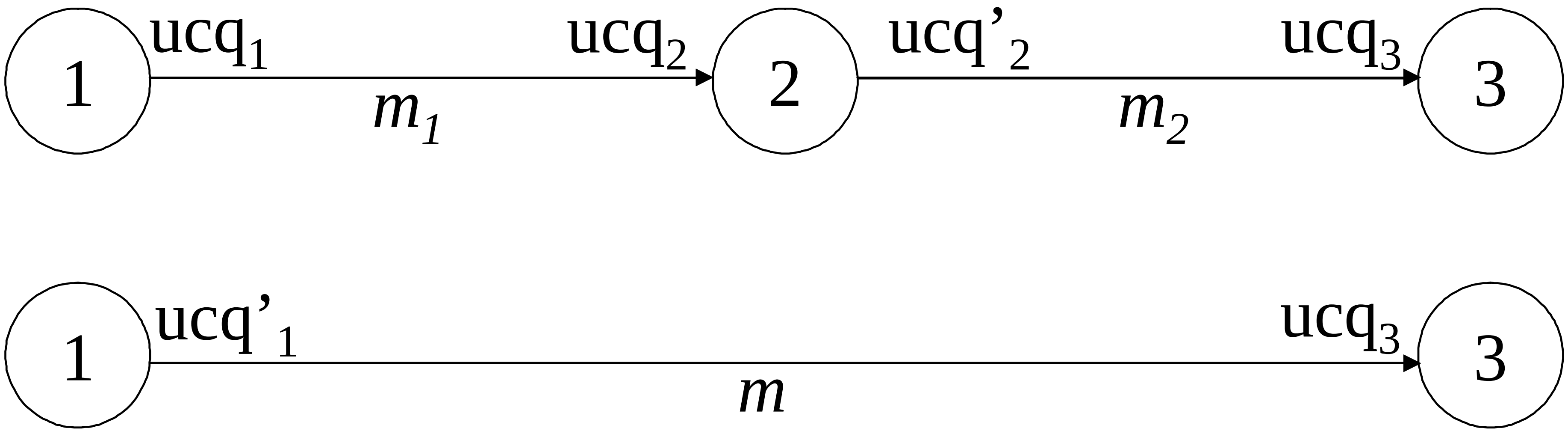
According to the epistemic semantics, $\mathrm{Person}(\mathrm{d})$ is not a certain answer to $\{x \mid \mathrm{Person}(\mathrm{x})\}$, since neither $\mathbf{K}_2\mathrm{Man}(\mathrm{d})$ nor $\mathbf{K}_3\mathrm{Woman}(\mathrm{d})$ is valid, and therefore, $\mathbf{K}_4\mathrm{Person}(\mathrm{d})$ is not valid too.

Epistemic semantics of P2P mappings: composition



In general, in FOL, m is not captured by any finite set of GLAV conjunctive mapping assertions.

Epistemic semantics of P2P mappings: composition



In Hyper, m is always captured by a finite set of GLAV mapping assertions.

Answering queries in P2P data integration

- 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

- How to reformulate a query expressed over a peer schema in terms of the local and external sources
- Loop detection

Query answering in P2P systems under epistemic semantics

- We are interested in an algorithm for distributed query answering, with no central coordination or centralized data structures

- We assume that peers accept queries in a query language $\mathcal{L}$ (subsuming at least conjunctive queries)

- We require that each peer, given a query Q in $\mathcal{L}$, is able to compute a Datalog query Q' that is a perfect rewriting of Q

Perfect rewriting

- Given peer P, and query Q in $\mathcal{L}$, a query Q_1 is a perfect rewriting of Q in P if
  - Q_1 is expressed over the source alphabet of P
  - for each source database D for P (i.e., over the local and the external sources), we have that

$$Q_1^D = ans(Q, P, D)$$

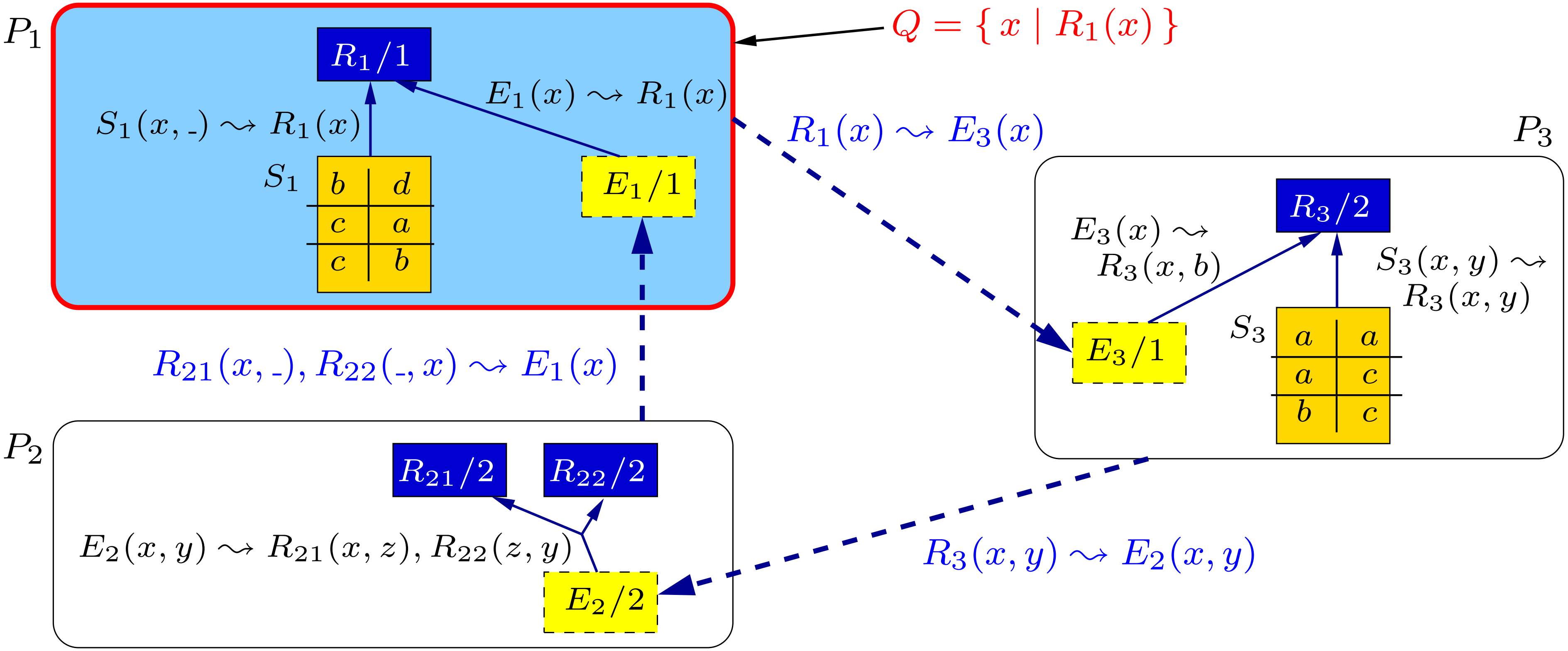
- Perfect rewritings exists in several settings, such as conjunctive client queries, and
  - conjunctive GLAV mappings
  - conjunctive GLAV mappings, plus key and foreign key constraints
  - conjunctive GLAV mappings, plus inclusion dependencies

Query answering: distributed algorithm

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

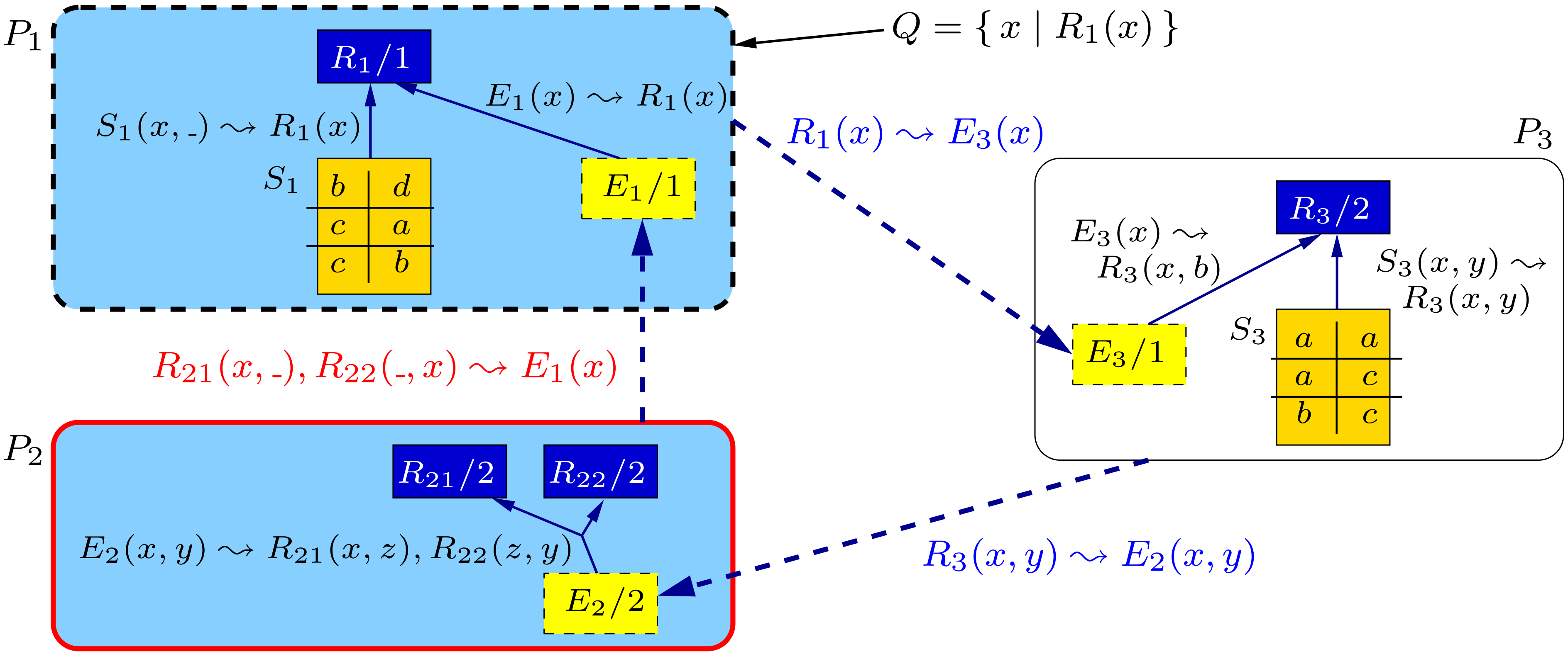
- Each peer rewrites the queries that are requested to it in terms of the local and external sources
- A reference to an external source triggers a request to the peer to which the external source is connected
- Answers to such requests consist of a Datalog program with two parts:
  - an extensional part, which is a set of facts (about local 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



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

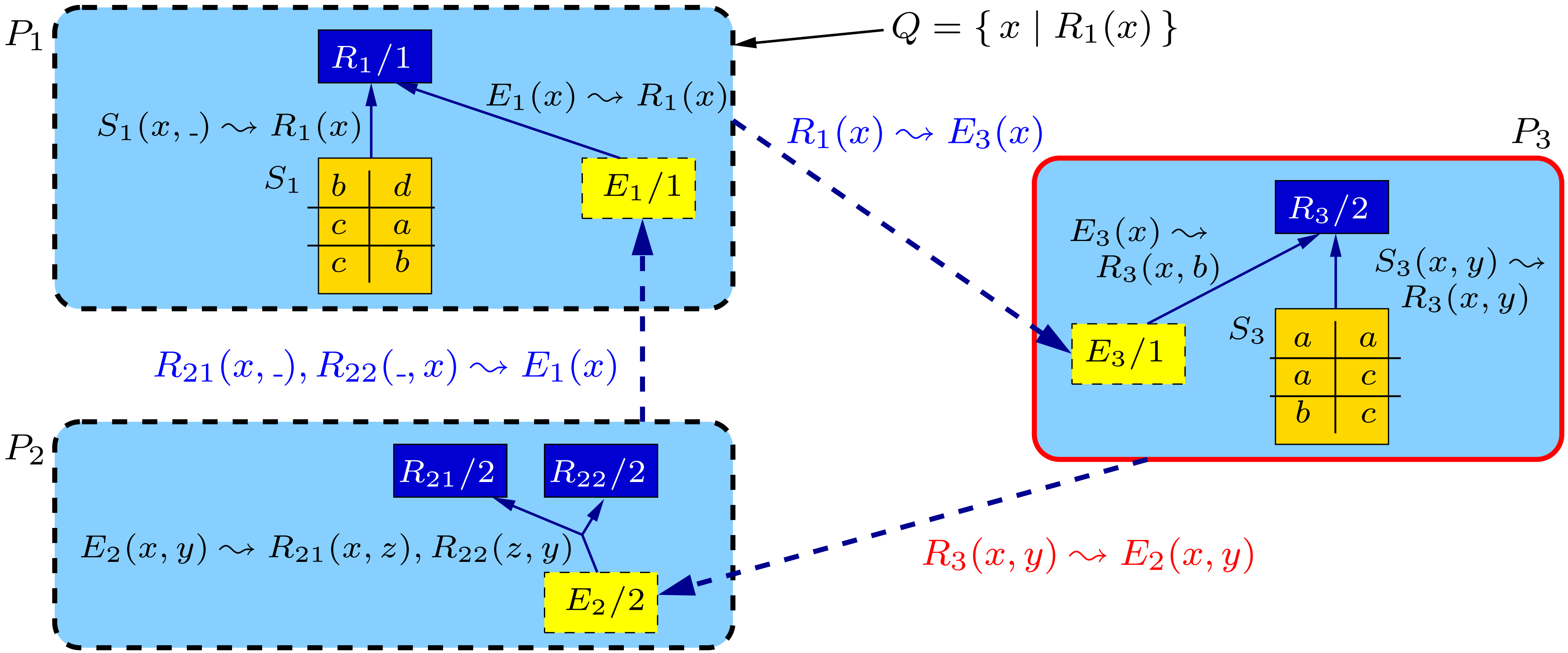
Query answering technique: example



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

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

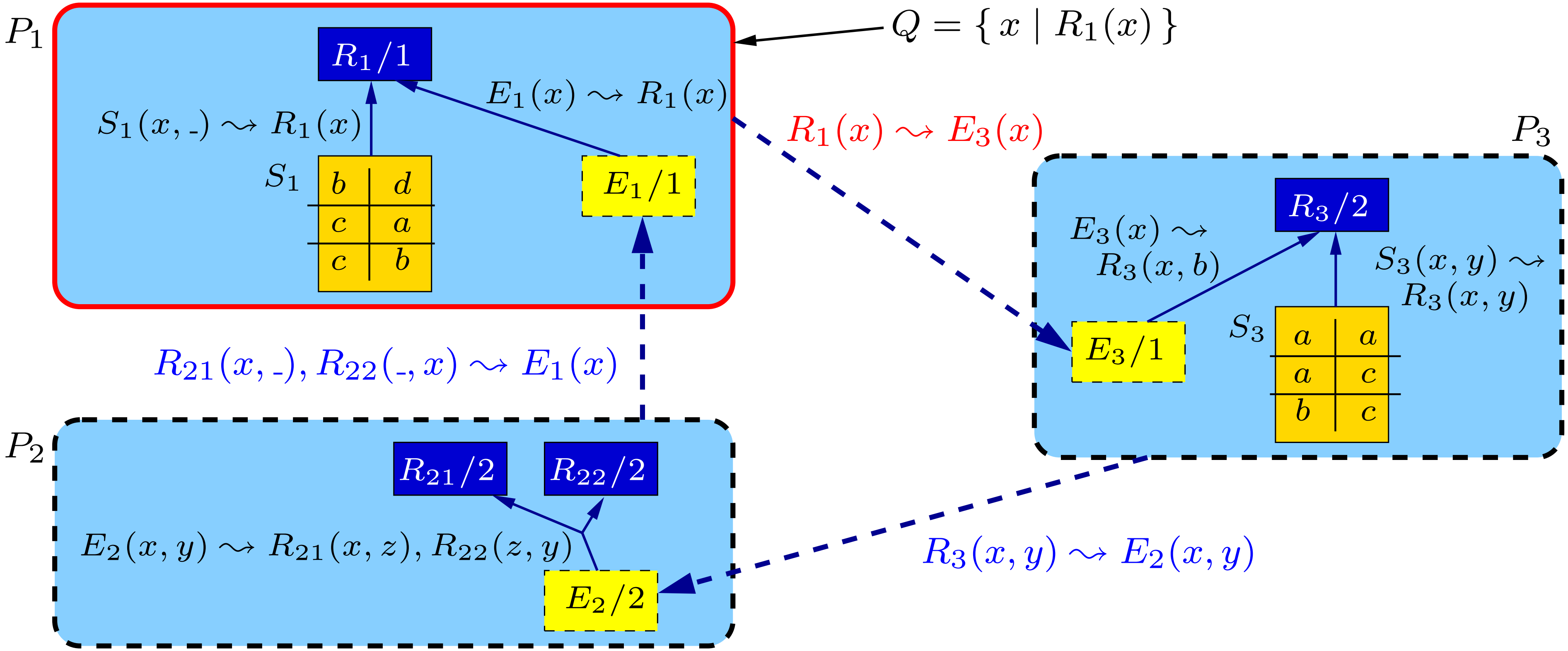
Query answering technique: example



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

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

Query answering technique: example



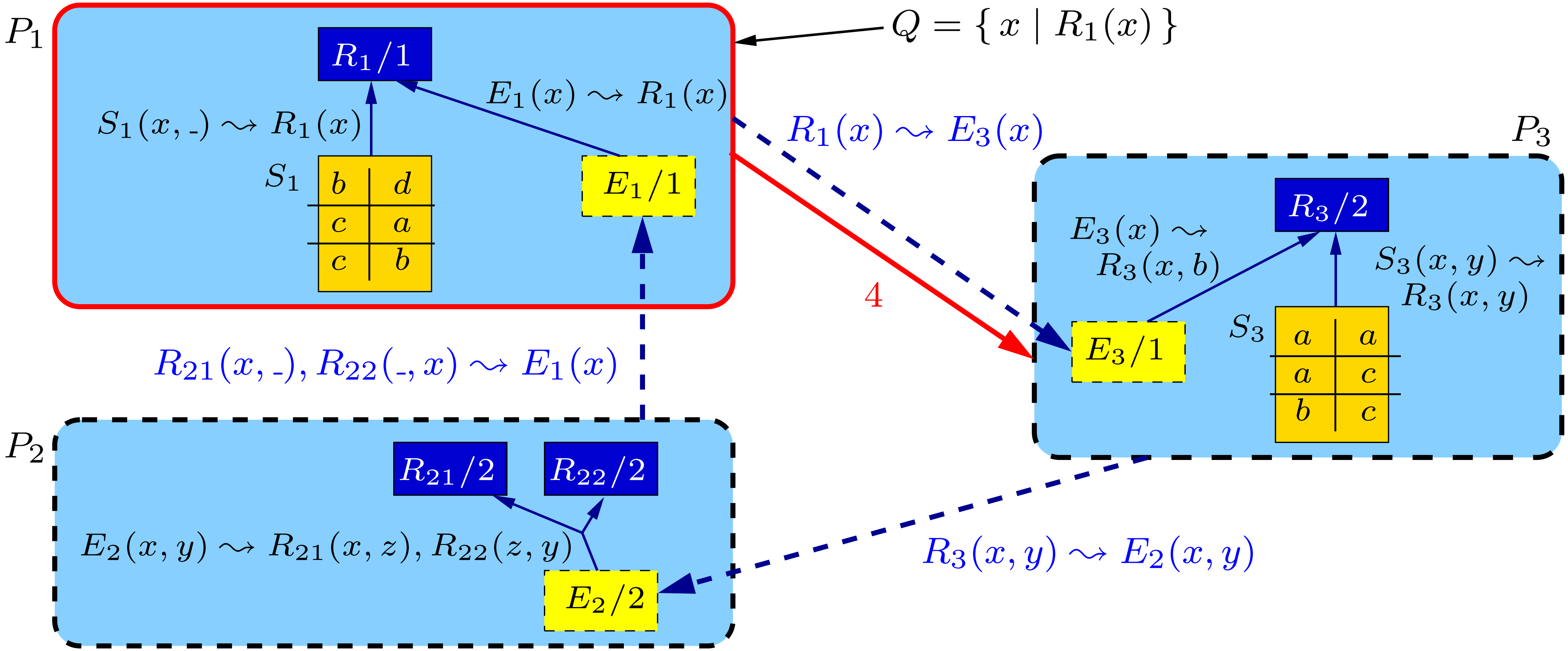
$$1 \left\{ \begin{array}{lcl} Q(x) & \leftarrow & S_1(x, \_) \\ Q(x) & \leftarrow & E_1(x) \end{array} \right.$$

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

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

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

Query answering technique: example



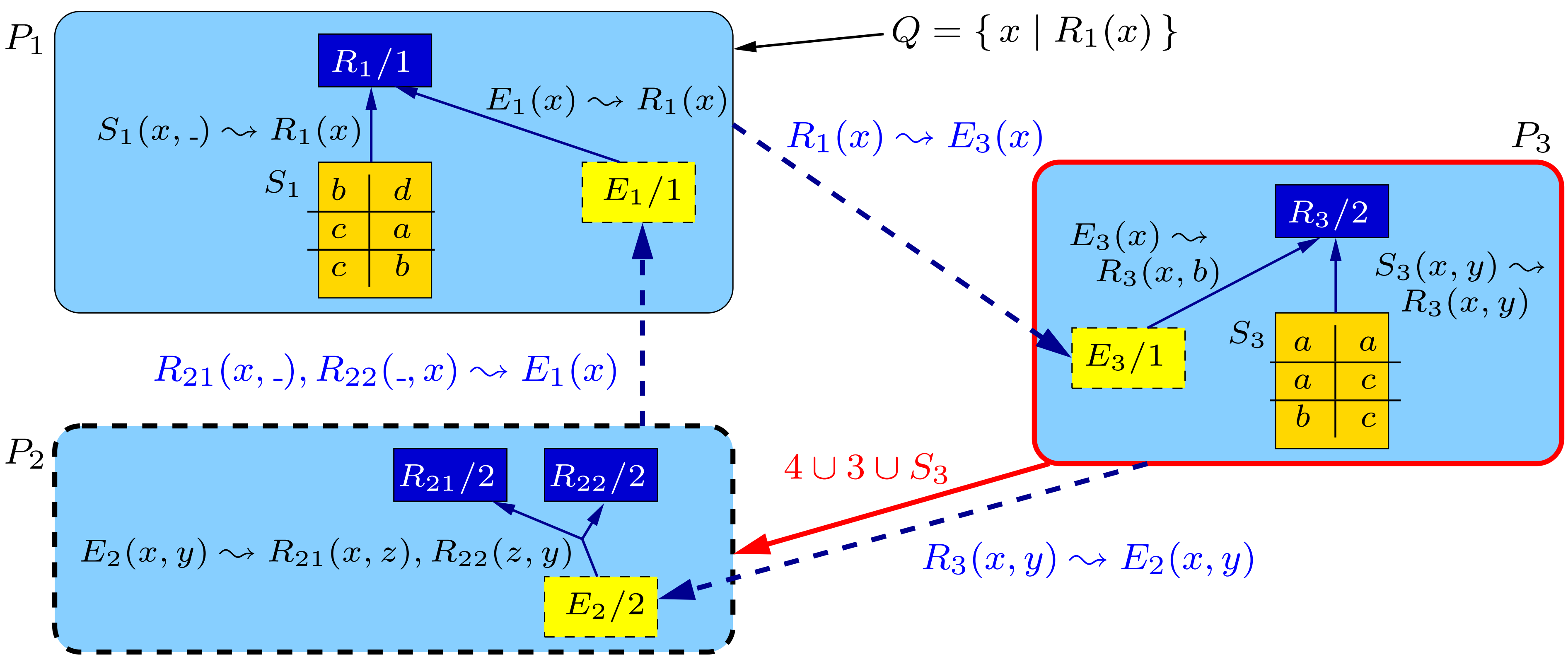
$$1 \begin{cases} Q(x) & \leftarrow \quad S_1(x, \_) \\ Q(x) & \leftarrow \quad E_1(x) \end{cases}$$

$$2 \begin{cases} E_1(x) & \leftarrow \quad E_2(x, \_), E_2(\_, x) \end{cases}$$

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

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

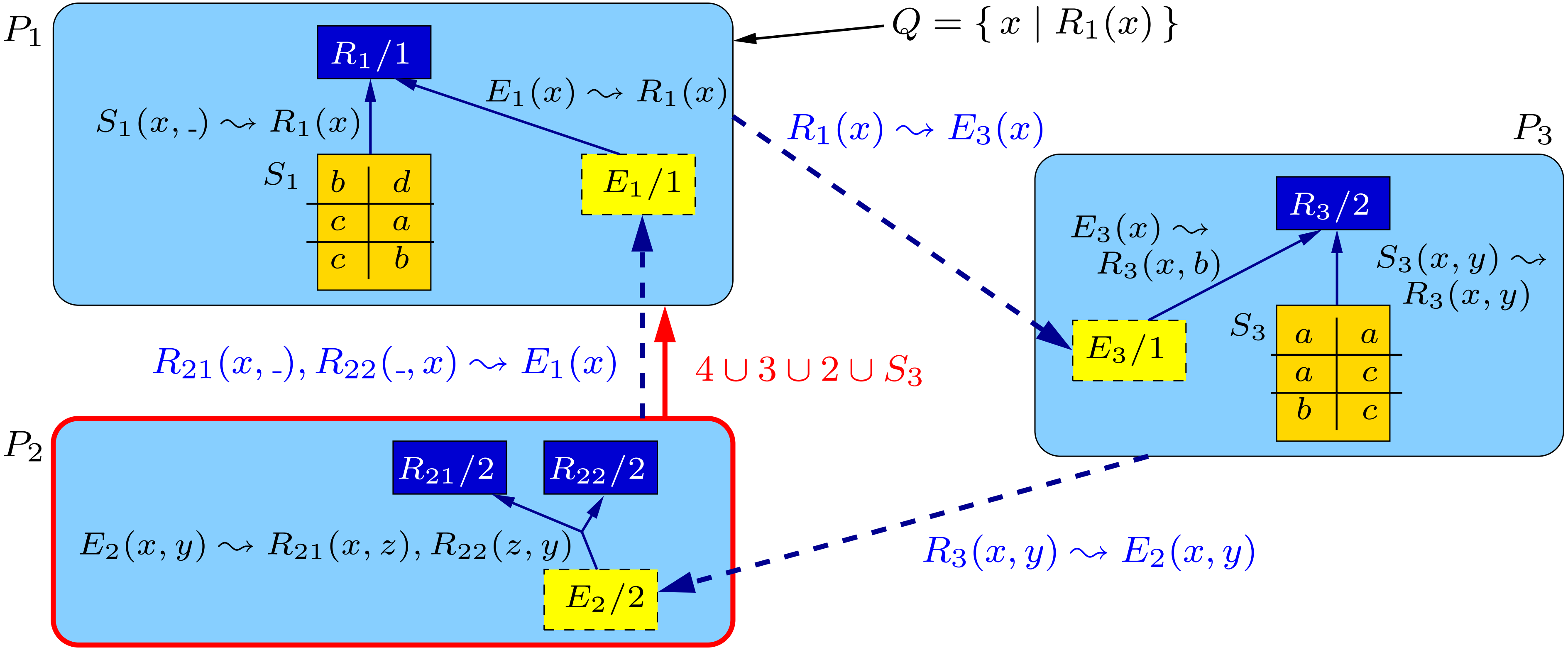
Query answering technique: example



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

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

Query answering technique: example



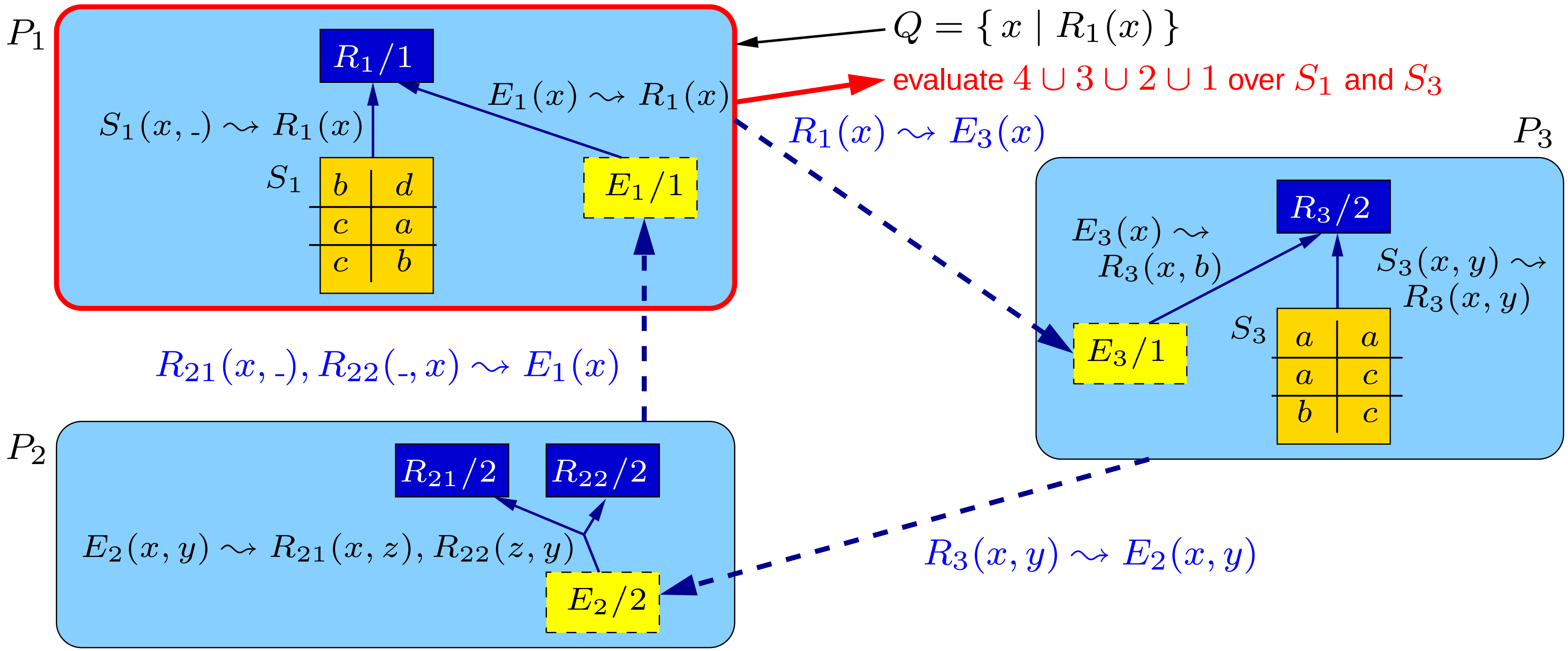
$$1 \left\{ \begin{array}{lcl} Q(x) & \leftarrow & S_1(x, \_) \\ Q(x) & \leftarrow & E_1(x) \end{array} \right.$$

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

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

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

Query answering technique: example



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

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

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

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

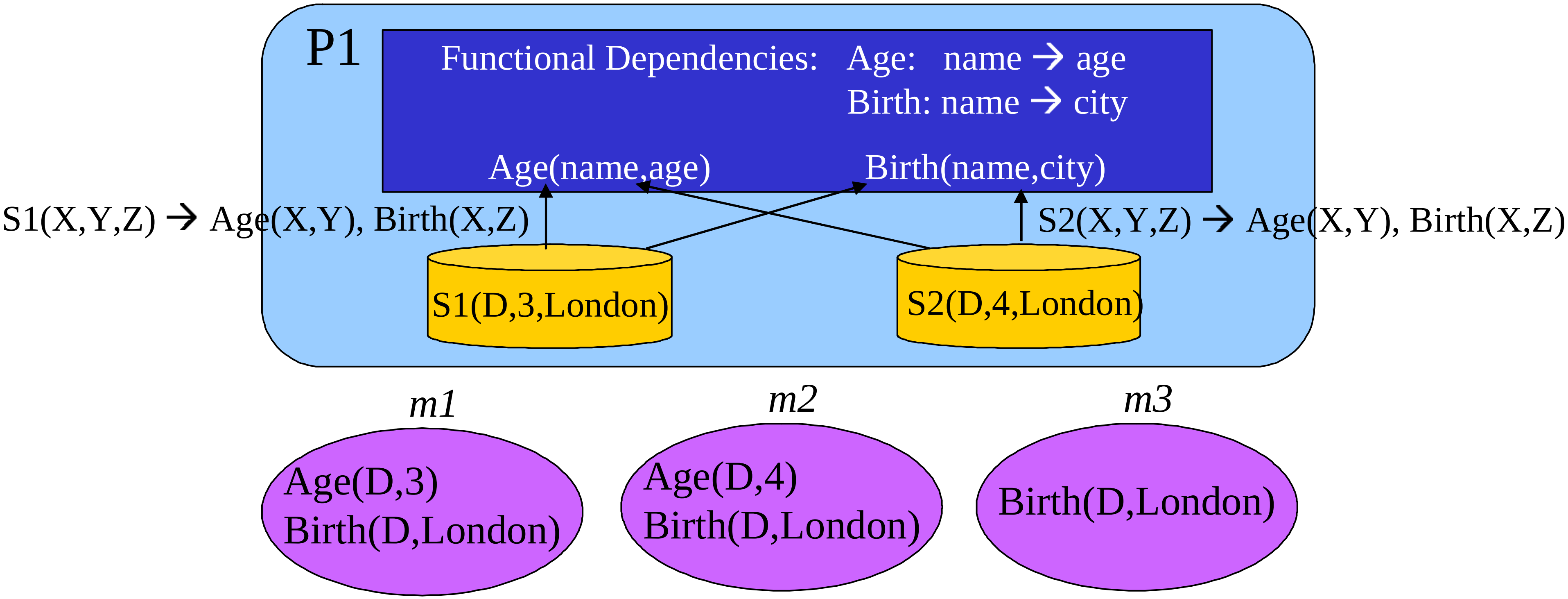
Outline

- Data integration architectures
- Centralized data integration
- P2P data integration
- Hyper: epistemic semantics for P2P data integration
- Dealing with inconsistencies in Hyper
- Conclusions

Dealing with inconsistencies in P2P data integration

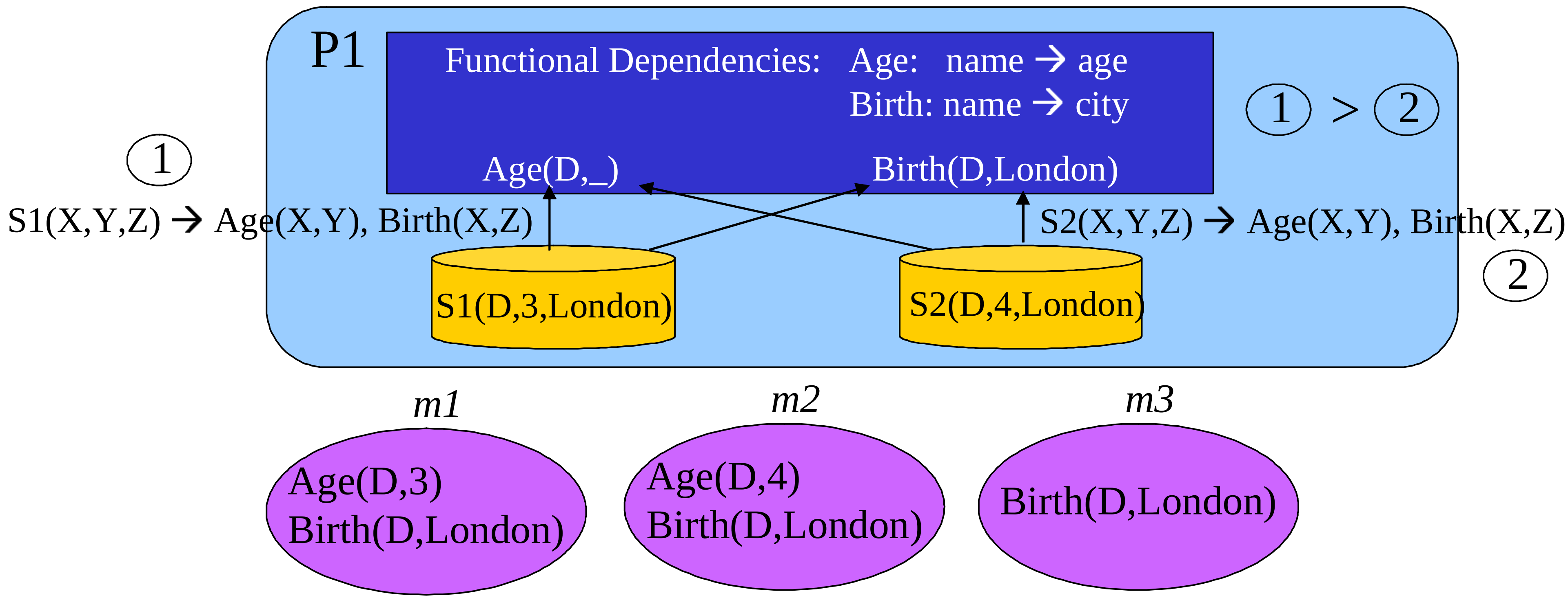
- The whole system blows up with a single inconsistency in one peer: unacceptable quality of query answering
- Extraction programs are supposed to perform data cleaning, but still may miss the resolution of some inconsistencies
- We resort to quality information for dealing with inconsistencies: we conceive 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
- We are using the notion of repair [Faign&al 1983, Arenas&al 1999, Arenas&al 2000, Chomicki&al 2002, Calì&al 2003]

Dealing with quality-based preferences in one peer

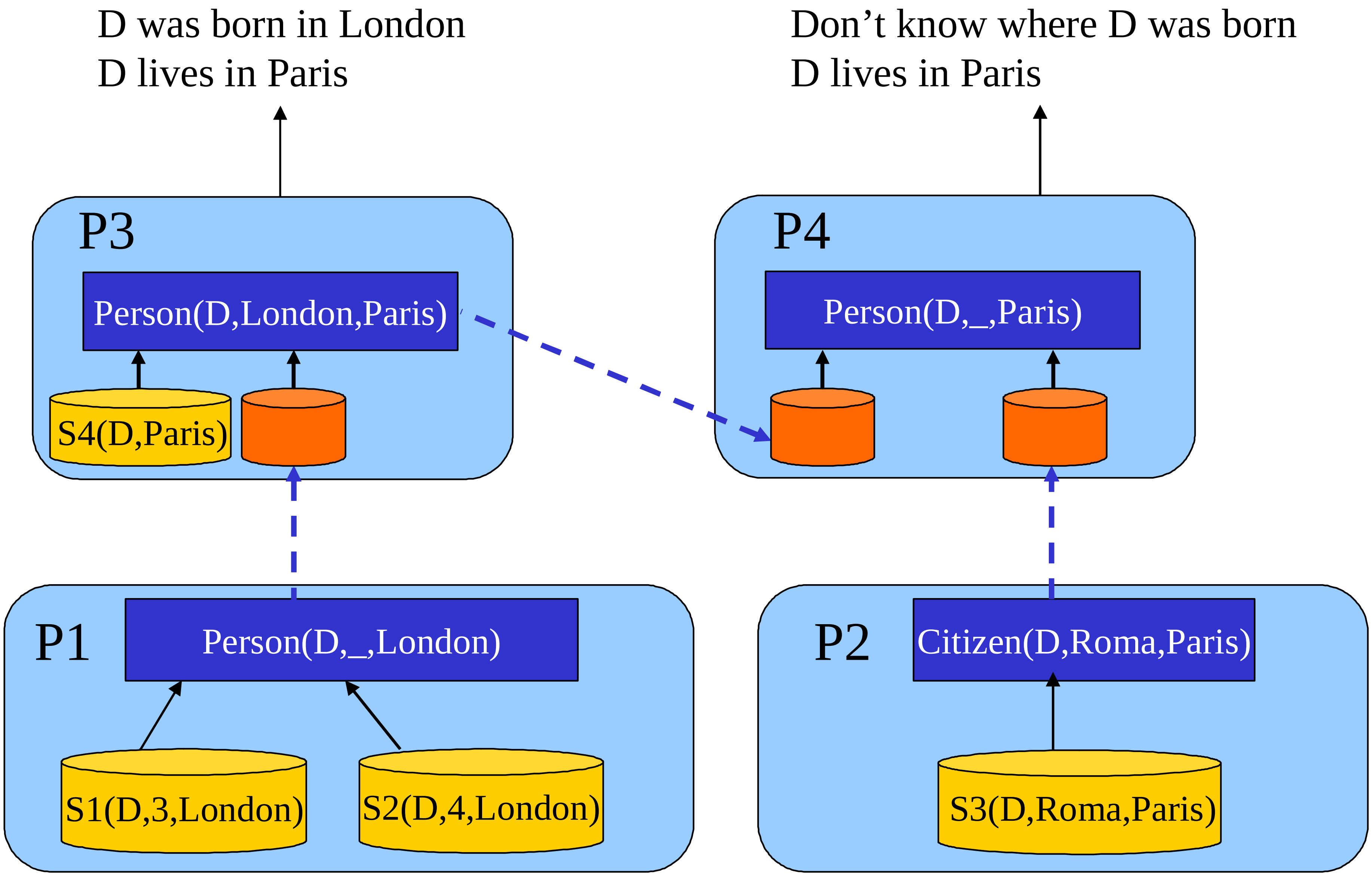


- A model m is preferred to model n if n misses some data from the sources that m does not miss, or if m respects the preferences more than n
- In the figure, $m1$ is better than both $m2$ and $m3$
- The models of a peer are the most preferred models

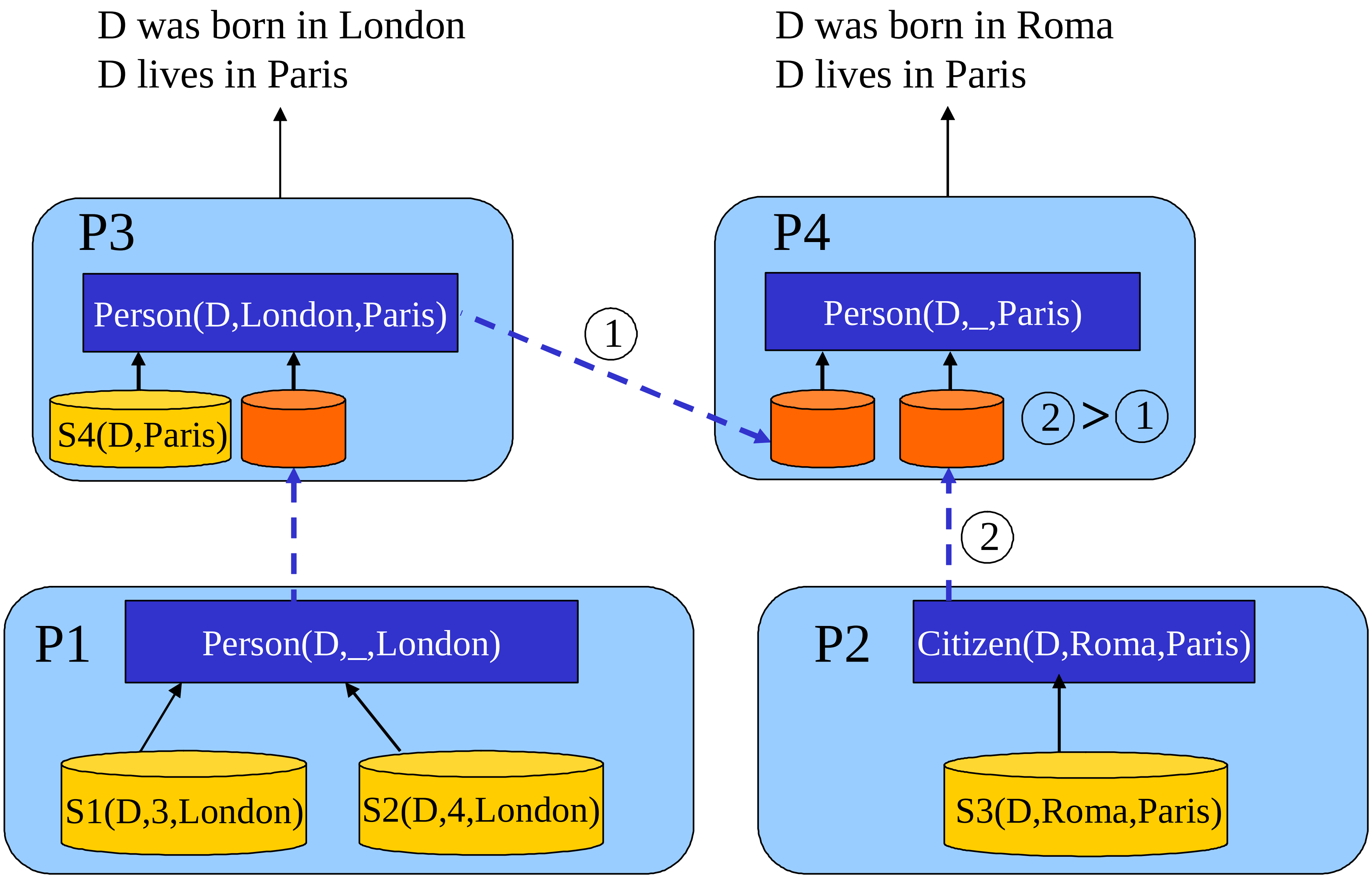
Dealing with inconsistencies in P2P data integration

- To generalize the idea to the case of multiple peers, we have to be able to compare epistemic models
- The models of the P2P data integration system are the most preferred epistemic models

Dealing with inconsistencies in P2P data integration



Dealing with inconsistencies and quality in P2P data integration



Outline

- Data integration architectures
- Centralized data integration
- P2P data integration
- Hyper: epistemic semantics for P2P data integration
- Dealing with inconsistencies in Hyper
- Conclusions

Conclusions

Many open problems and issues, including

- Object identification and mapping
- How to obtain information on mapping, quality and preferences
- Algorithm and complexity in the extended epistemic semantics
- 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 and dynamicity during query processing
- Ongoing experiments within
  - Hyper, an IBM Shared University Research program, see `http://www.dis.uniroma1.it/~lenzerini/progetti/hyper/`
  - the EU project Sewasie