

A Logic Programming Approach to Querying and Integrating P2P Deductive Databases

L. Caroprese, S. Greco and E. Zumpano

DEIS, Univ. della Calabria

87030 Rende, Italy

{lcaroprese, greco, zumpano}@deis.unical.it

Abstract

The paper proposes a logic framework for modeling the interaction among deductive databases and computing consistent answers to logic queries in a P2P environment. As usual, data are exchanged among peers by using logical rules, called mapping rules. The novelty of our approach is that only data not violating integrity constraints are exchanged. The (declarative) semantics of a P2P system is defined in terms of weak models. Under this semantics only facts not making the local databases inconsistent are imported, and the preferred weak models are those in which peers import maximal sets of facts not violating integrity constraints. An equivalent and alternative characterization of preferred weak model semantics, in terms of prioritized logic programs, is also introduced and the computational complexity of P2P logic queries is investigated.

Introduction

The spread of Internet and the possibility for its users for sharing knowledge from a large number of informative sources, have enabled the development of new methods for data integration easily usable for processing distributed and autonomous data. In traditional data integration systems queries are posed through a central *mediated schema*. Data is stored locally in each source and the two main formalisms managing the mapping between the mediated schema and the local sources are the *global-as-view* (GAV) and the *local-as-view* (LAV) approach (Lenzerini 2002). The main drawbacks of traditional integration systems are due to the lack of flexibility: i) the centralized mediated schema, that controls and manages the interaction among distributed sources, must be defined looking at the global system; ii) the insertion of a new source or the modification of an existing one may cause a violation of the mappings to the mediated schema. Recently, there have been several proposals which consider the integration of information and the computation of queries in an open ended

network of distributed peers (Bernstein *et al.* 2002; Calvanese *et al.* 2004; Franconi *et al.* 2003) as well as the problem of schema mediation and query optimization in P2P environments (Halevy *et al.* 2003; Madhavan & Halevy 2003; Tatarinov & Halevy. 2004; Gribble *et al.* 2001). Generally, peers can both provide or consume data and the only information a peer participating in a P2P system has is about neighbors, i.e. information about the peers that are reachable and can provide data of interest. More specifically, each peer joining a P2P system exhibits a set of *mapping rules*, i.e. a set of semantic correspondences to a set of peers which are already part of the system (neighbors). Thus, in a P2P system the entry of a new source, *peer*, is extremely simple as it just requires the definition of the mapping rules. By using mapping rules as soon as it enters the system a peer can participate and access all data available in its neighborhood, and through its neighborhood it becomes accessible to all the other peers in the system.

As stated before, the problem of integrating and querying databases in P2P environments has been investigated in (Calvanese *et al.* 2004; Franconi *et al.* 2003). In both works peers are modeled as autonomous agents which can export only data belonging to their knowledge, i.e. data which are *true* in all possible scenarios (models).

In (Calvanese *et al.* 2004) a new semantics for a P2P system, based on epistemic logic, is proposed. The paper also shows that the semantics is more suitable than traditional semantics based on FOL (First Order Logic) and proposes a sound, complete and terminating procedure that returns the certain answers to a query submitted to a peer.

In (Franconi *et al.* 2003) a characterization of P2P database systems and a model-theoretic semantics dealing with inconsistent peers is proposed. The basic idea is that if a peer does not have models all (ground) queries submitted to the peer are *true* (i.e. are *true* with respect to all models). Thus, if some databases

are inconsistent it does not mean that the entire system is inconsistent.

An interesting approach for answering queries in a Peer to Peer data exchange system has been recently proposed in (Bertossi & Bravo 2004). Given a peer P in a P2P system a solution for P is a database instance that respects the exchange constraints and trust relationship P has with its ‘immediate neighbors’ and stays as close as possible to the available data in the system.

In (Halevy *et al.* 2003) the problem of schema mediation in a Peer Data Management System (PDMS) is investigated. A flexible formalism, *PPL*, for mediating peer schemas, which uses the GAV and LAV formalism to specify mappings, is proposed. The semantics of query answering for a PDMS is defined by extending the notion of certain answer.

In (Tatarinov & Halevy. 2004) several techniques for optimizing the reformulation of queries in a PDMS are presented. In particular the paper presents techniques for pruning semantic paths of mappings in the reformulation process and for minimizing the reformulated queries.

The design of optimization methods for query processing over a network of semantically related data is investigated in (Madhavan & Halevy 2003).

Motivation. The motivation of this work stems from the observation that previously proposed approaches result not to be *sound* with respect to query answering when some peer is inconsistent.

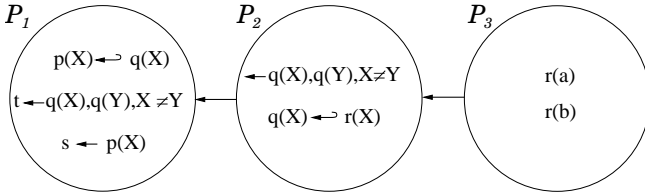


Figure 1: A P2P system

Example 1 Consider the P2P system depicted in Figure 1 consisting of three peers \mathcal{P}_1 , \mathcal{P}_2 and \mathcal{P}_3 where

- \mathcal{P}_3 contains two atoms: $r(a)$ and $r(b)$,
- \mathcal{P}_2 imports data from \mathcal{P}_3 using the (mapping) rule $q(X) \leftrightarrow r(X)$ ¹. Moreover imported atoms must satisfy the constraint $\leftarrow q(X), q(Y), X \neq Y$ stating that the relation q may contain at most one tuple, and
- \mathcal{P}_1 imports data from \mathcal{P}_2 , using the (mapping) rule $p(X) \leftrightarrow q(X)$. \mathcal{P}_1 also contains the rules $s \leftarrow p(X)$ stating that s is *true* if the relation p contains at least one tuple, and $t \leftarrow p(X), p(Y), X \neq Y$, stating that t

¹Please, note the special syntax we use for mapping rules.

is *true* if the relation p contains at least two distinct tuples.

The intuition is that, with $r(a)$ and $r(b)$ being *true* in \mathcal{P}_3 , either $q(a)$ or $q(b)$ could be imported in \mathcal{P}_2 (but not both, otherwise the integrity constraint is violated) and, consequently, only one tuple is imported in the relation p of the peer \mathcal{P}_1 . Note that whatever is the derivation in \mathcal{P}_2 , s is derived in \mathcal{P}_1 while t is not derived. Therefore, the atoms s and t are, respectively, *true* and *false* in \mathcal{P}_1 . It is worth noting that the approaches above mentioned do not capture such a semantics. Indeed, the epistemic semantics proposed in (Calvanese *et al.* 2004) states that both the atoms $q(a)$ and $q(b)$ are imported in the peer \mathcal{P}_2 which becomes inconsistent. In this case the semantics assumes that the whole P2P system is inconsistent and every atom is *true* as it belongs to all minimal models. Consequently, t and s are *true*. The semantics proposed in (Franconi *et al.* 2003) assumes that only \mathcal{P}_2 is inconsistent as it has no model. Thus, as the atoms $q(a)$ and $q(b)$ are *true* in \mathcal{P}_2 (they belong to all models of \mathcal{P}_2), then the atoms $p(a)$ and $p(b)$ can be derived in \mathcal{P}_1 and finally t and s are *true*. \square

The idea we propose in this paper consists in importing in each peer maximal sets of atoms not violating integrity constraints. A similar approach, using epistemic logic, has been recently proposed in (Calvanese *et al.* 2005).

Contributions. The paper presents a logic-based framework for modeling the interaction among peers. It is assumed that each peer consists of a database, a set of standard logic rules, a set of mapping rules and a set of integrity constraints. In such a context, a query can be posed to any peer in the system and the answer is provided by using locally stored data and all the information that can be consistently imported from its neighbors. In synthesis, the main contributions are:

1. A formal declarative semantics for P2P systems, called *Preferred Weak Model* semantics, which uses the mapping rules between peers to import only maximal sets of atoms not violating integrity constraints.
2. An alternative equivalent semantics, called *Preferred Stable Model* semantics, based on the rewriting of mapping rules into standard logic rules with priorities.
3. A generalization of the preferred weak model semantics which allows to compute consistent queries also in the presence of possible inconsistent source databases.
4. Results on the complexity of answering queries.

Background

We assume there are finite sets of predicate symbols, constants and variables. A *term* is either a constant or a variable. An *atom* is of the form $p(t_1, \dots, t_n)$ where p is a predicate symbol and t_1, \dots, t_n are terms. A *literal* is either an atom A or its negation $\text{not } A$. A *rule* is of the form $H \leftarrow \mathcal{B}$, where H is an atom (*head* of the rule) and \mathcal{B} is a conjunction of literals (*body* of the rule). A program \mathcal{P} is a finite set of rules. \mathcal{P} is said to be positive if it is negation free. The definition of a predicate p consists of all rules having p in the head. A ground rule with empty body is a *fact*. A rule with empty head is a *constraint*. It is assumed that programs are *safe*, i.e. variables appearing in the head or in negated body literals are range restricted as they appear in some positive body literal. The ground instantiation of a program \mathcal{P} , denoted by $\text{ground}(\mathcal{P})$ is built by replacing variables with constants in all possible ways. An interpretation is a set of ground atoms. The truth value of ground atoms, literals and rules with respect to an interpretation M is as follows: $\text{val}_M(A) = A \in M$, $\text{val}_M(\text{not } A) = \text{not } \text{val}_M(A)$, $\text{val}_M(L_1, \dots, L_n) = \min\{\text{val}_M(L_1), \dots, \text{val}_M(L_n)\}$ and $\text{val}_M(A \leftarrow L_1, \dots, L_n) = \text{val}_M(A) \geq \text{val}_M(L_1, \dots, L_n)$, where A is an atom, L_1, \dots, L_n are literals and $\text{true} > \text{false}$. An interpretation M is a model for a program \mathcal{P} , if all rules in $\text{ground}(\mathcal{P})$ are *true* w.r.t. M . A model M is said to be minimal if there is no model N such that $N \subset M$. We denote the set of minimal models of a program \mathcal{P} with $\mathcal{MM}(\mathcal{P})$. Given an interpretation M and a predicate symbol g , $M[g]$ denotes the set of g -tuples in M . The semantics of a positive program \mathcal{P} is given by its unique minimal model which can be computed by applying the *immediate consequence operator* $\mathbf{T}_{\mathcal{P}}$ until the fixpoint is reached ($\mathbf{T}_{\mathcal{P}}^\infty(\emptyset)$). The semantics of a program with negation \mathcal{P} is given by the set of its stable models, denoted as $\mathcal{SM}(\mathcal{P})$. An interpretation M is a *stable model* (or *answer set*) of \mathcal{P} if M is the unique minimal model of the positive program \mathcal{P}^M , where \mathcal{P}^M is obtained from $\text{ground}(\mathcal{P})$ by (i) removing all rules r such that there exists a negative literal $\text{not } A$ in the body of r and A is in M and (ii) removing all negative literals from the remaining rules (Gelfond & Lifschitz 1988). It is well known that stable models are minimal models (i.e. $\mathcal{SM}(\mathcal{P}) \subseteq \mathcal{MM}(\mathcal{P})$) and that for negation free programs, minimal and stable model semantics coincide (i.e. $\mathcal{SM}(\mathcal{P}) = \mathcal{MM}(\mathcal{P})$).

Prioritized logic programs

Several recent works have investigated the introduction of various forms of priorities into logic languages (Brewka & Eiter 1999; Brewka, Niemela, & Truszczyn-

ski 2003; Delgrande, Schaub, & Tompits 2003; Sakama & Inoue 2000). In this paper we refer to the extension proposed in (Sakama & Inoue 2000). A (*partial*) *preference relation* \succeq among atoms is defined as follows. Given two atoms e_1 and e_2 , the statement $e_2 \succeq e_1$ is a *priority* stating that for each a_2 instance of e_2 and for each a_1 instance of e_1 , a_2 has higher priority than a_1 . If $e_2 \succeq e_1$ and $e_1 \not\succeq e_2$ we write $e_2 \succ e_1$. If $e_2 \succ e_1$ the sets of ground instantiations of e_1 and e_2 have empty intersection. The relation \succeq is transitive and reflexive. A *prioritized logic program* (PLP) is a pair (\mathcal{P}, Φ) where \mathcal{P} is a program and Φ is a set of priorities. Φ^* denotes the set of priorities which can be reflexively or transitively derived from Φ . Given a prioritized logic program (\mathcal{P}, Φ) , the relation \sqsupseteq is defined over the stable models of \mathcal{P} as follows. For any stable models M_1, M_2 and M_3 of \mathcal{P} : (i) $M_1 \sqsupseteq M_1$; (ii) $M_2 \sqsupseteq M_1$ if a) $\exists e_2 \in M_2 - M_1$, $\exists e_1 \in M_1 - M_2$ such that $(e_2 \succeq e_1) \in \Phi^*$ and b) $\nexists e_3 \in M_1 - M_2$ such that $(e_3 \succ e_2) \in \Phi^*$; (iii) if $M_2 \sqsupseteq M_1$ and $M_1 \sqsupseteq M_0$, then $M_2 \sqsupseteq M_0$.

If $M_2 \sqsupseteq M_1$ holds, then we say that M_2 is *preferable* to M_1 w.r.t. Φ . Moreover, we write $M_2 \sqsupset M_1$ if $M_2 \sqsupseteq M_1$ and $M_1 \not\sqsupseteq M_2$. An interpretation M is a *preferred* stable model of (\mathcal{P}, Φ) if M is a stable model of \mathcal{P} and there is no stable model N such that $N \sqsupset M$. The set of preferred stable models of (\mathcal{P}, Φ) will be denoted by $\mathcal{PSM}(\mathcal{P}, \Phi)$.

P2P systems: syntax and FOL semantics

A (*peer*) *predicate* symbol is a pair $i : p$ where i is a *peer identifier* and p is a predicate symbol. A (*peer*) *atom* is of the form $i : A$ where A is a standard atom. A (*peer*) *literal* is of the form $\text{not } A$ where A is a peer atom. A (*peer*) *rule* is of the form $A \leftarrow A_1, \dots, A_n$ where A is a peer atom and A_1, \dots, A_n are peer atoms or built-in atoms. A (*peer*) *integrity constraint* is of the form $\leftarrow L_1, \dots, L_m$ where L_1, \dots, L_m are peer literals or built-in atoms. Whenever the peer is understood, the peer identifier can be omitted. The definition of a predicate $i : p$ consists of all rules having as head predicate symbol $i : p$. In the following, we assume that for each peer \mathcal{P}_i there are three distinct sets of predicates called, respectively, *base*, *derived* and *mapping predicates*. A base predicates is defined by ground facts; a derived predicate $i : p$ is defined by *standard rules*, i.e. peer rules using in the body only predicates defined in the peer \mathcal{P}_i ; a mapping predicate $i : p$ is defined by *mapping rules*, i.e. peer rules using in the body only predicates defined in other peers. Without loss of generality, we assume that every mapping predicate is defined by only one rule of the form $i : p(X) \leftarrow j : q(X)$ with $j \neq i$. The

definition of a mapping predicate $i : p$ consisting of n rules of the form $i : p(X_k) \leftarrow \mathcal{B}_k$ with $1 \leq k \leq n$, can be rewritten into $2n$ rules of the form $i : p_k(X_k) \leftarrow \mathcal{B}_k$ and $i : p(X) \leftarrow i : p_k(X)$, with $1 \leq k \leq n$.

Definition 1 A peer \mathcal{P}_i is a tuple $\langle \mathcal{D}_i, \mathcal{LP}_i, \mathcal{MP}_i, \mathcal{IC}_i \rangle$ where (i) \mathcal{D}_i is a (local) database consisting of a set of facts; (ii) \mathcal{LP}_i is a set of standard rules; (iii) \mathcal{MP}_i is a set of mapping rules and (iv) \mathcal{IC}_i is a set of constraints over predicates defined in \mathcal{D}_i , \mathcal{LP}_i and \mathcal{MP}_i . A P2P system \mathcal{PS} is a set of peers $\{\mathcal{P}_1, \dots, \mathcal{P}_n\}$. \square

Given a P2P system $\mathcal{PS} = \{\mathcal{P}_1, \dots, \mathcal{P}_n\}$ where $\mathcal{P}_i = \langle \mathcal{D}_i, \mathcal{LP}_i, \mathcal{MP}_i, \mathcal{IC}_i \rangle$, we denote as $\mathcal{D}, \mathcal{LP}, \mathcal{MP}$ and \mathcal{IC} respectively the global sets of ground facts, standard rules, mapping rules and integrity constraints: $\mathcal{D} = \mathcal{D}_1 \cup \dots \cup \mathcal{D}_n$, $\mathcal{LP} = \mathcal{LP}_1 \cup \dots \cup \mathcal{LP}_n$, $\mathcal{MP} = \mathcal{MP}_1 \cup \dots \cup \mathcal{MP}_n$ and $\mathcal{IC} = \mathcal{IC}_1 \cup \dots \cup \mathcal{IC}_n$. With a little abuse of notation we shall also denote with \mathcal{PS} both the tuple $\langle \mathcal{D}, \mathcal{LP}, \mathcal{MP}, \mathcal{IC} \rangle$ and the set $\mathcal{D} \cup \mathcal{LP} \cup \mathcal{MP} \cup \mathcal{IC}$. Given a peer \mathcal{P}_i , $MPred(\mathcal{P}_i)$, $DPred(\mathcal{P}_i)$ and $BPred(\mathcal{P}_i)$ denote, respectively, the sets of mapping, derived and base predicates defined in \mathcal{P}_i . Analogously, $MPred(\mathcal{PS})$, $DPred(\mathcal{PS})$ and $BPred(\mathcal{PS})$ define the sets of mapping, derived and base predicates in \mathcal{PS} .

FOL semantics. The FOL semantics of a P2P system $\mathcal{PS} = \{\mathcal{P}_1, \dots, \mathcal{P}_n\}$ is given by the minimal model semantics of $\mathcal{PS} = \mathcal{D} \cup \mathcal{LP} \cup \mathcal{MP} \cup \mathcal{IC}$. For a given P2P system \mathcal{PS} , $\mathcal{MM}(\mathcal{PS})$ denotes the set of minimal models of \mathcal{PS} . As $\mathcal{D} \cup \mathcal{LP} \cup \mathcal{MP}$ is a positive program, \mathcal{PS} may admit zero or one minimal model. In particular, if $\mathcal{MM}(\mathcal{D} \cup \mathcal{LP} \cup \mathcal{MP}) = \{M\}$ then $\mathcal{MM}(\mathcal{PS}) = \{M\}$ if $M \models \mathcal{IC}$, otherwise $\mathcal{MM}(\mathcal{PS}) = \emptyset$. The problem with such a semantics is that local inconsistencies make the global system inconsistent.

Weak model semantics

This section introduces a new semantics, called *weak model* semantics, based on a new interpretation of mapping rules, which will now be denoted with a different syntax of the form $H \leftrightarrow \mathcal{B}$. Intuitively, $H \leftrightarrow \mathcal{B}$ means that if the body conjunction \mathcal{B} is *true* in the source peer the atom H could be imported in the target peer, that is H is *true* in the target peer only if it does not imply (directly or indirectly) the violation of some constraints. The following example should make the meaning of mapping rules crystal clear.

Example 2 Consider the P2P system depicted in Figure 2. \mathcal{P}_2 contains the fact $q(b)$ whereas \mathcal{P}_1 contains the fact $s(a)$, the mapping rule $p(X) \leftrightarrow q(X)$, the constraint $\leftarrow r(X), r(Y), X \neq Y$ and the standard rules $r(X) \leftarrow p(X)$ and $r(X) \leftarrow s(X)$. In this case the fact

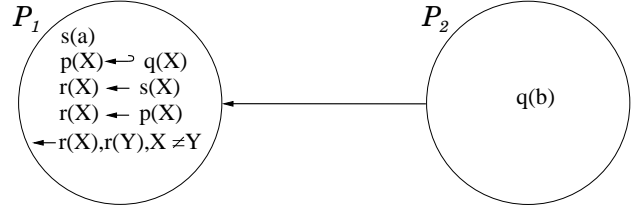


Figure 2: A P2P system

$p(b)$ cannot be imported in \mathcal{P}_1 as it indirectly violates its integrity constraint.

Before formally presenting the weak model semantics we introduce some notation. Given a mapping rule $r = A \leftrightarrow \mathcal{B}$, with $St(r)$ we denote the corresponding logic rule $A \leftarrow \mathcal{B}$. Analogously, given a set of mapping rules \mathcal{MP} , $St(\mathcal{MP}) = \{St(r) \mid r \in \mathcal{MP}\}$ and given a P2P system $\mathcal{PS} = \mathcal{D} \cup \mathcal{LP} \cup \mathcal{MP} \cup \mathcal{IC}$, $St(\mathcal{PS}) = \mathcal{D} \cup \mathcal{LP} \cup St(\mathcal{MP}) \cup \mathcal{IC}$.

In the next two subsections we present two alternative and equivalent characterizations of the weak model semantics. The first semantics is based on a different satisfaction of mapping rules, whereas the second one is based on the rewriting of mapping rules into prioritized rules (Brewka, Niemela, & Truszczyński 2003; Sakama & Inoue 2000). For the rest of this section we assume that all peers are *locally consistent*, i.e. for each peer \mathcal{P}_i , the database \mathcal{D}_i and the standard rules in \mathcal{LP}_i are consistent w.r.t. \mathcal{IC}_i , $(\mathcal{D}_i \cup \mathcal{LP}_i \models \mathcal{IC}_i)$. In such a case we say that the P2P system is consistent and inconsistencies may be introduced when peers import data from other peers. The generalization for inconsistent peers will be considered in the next section.

Preferred weak models

Informally, the idea is that for a ground mapping rule $A \leftrightarrow \mathcal{B}$, the atom A could be *inferred* only if the body \mathcal{B} is *true*. Formally, given an interpretation M , a ground standard rule $D \leftarrow \mathcal{C}$ and a ground mapping rule $A \leftrightarrow \mathcal{B}$, $val_M(\mathcal{C} \leftarrow \mathcal{D}) = val_M(\mathcal{C}) \geq val_M(\mathcal{D})$ whereas $val_M(A \leftrightarrow \mathcal{B}) = val_M(A) \leq val_M(\mathcal{B})$.

Definition 2 Given a P2P system $\mathcal{PS} = \mathcal{D} \cup \mathcal{LP} \cup \mathcal{MP} \cup \mathcal{IC}$, an interpretation M is a *weak model* for \mathcal{PS} if $\{M\} = \mathcal{MM}(St(\mathcal{PS}^M))$, where \mathcal{PS}^M is the program obtained from $ground(\mathcal{PS})$ by removing all mapping rules whose head is *false* w.r.t. M . \square

We shall denote with $M[\mathcal{D}]$ (resp. $M[\mathcal{LP}]$, $M[\mathcal{MP}]$) the set of ground atoms of M which are defined in \mathcal{D} (resp. \mathcal{LP} , \mathcal{MP}).

Definition 3 Given two weak models M and N , we say that M is *preferable* to N , and we write $M \sqsupseteq N$, if $M[\mathcal{MP}] \supseteq N[\mathcal{MP}]$. Moreover, if $M \sqsupseteq N$ and $N \not\sqsupseteq M$

M we write $M \sqsupseteq N$. A weak model M is said to be *preferred* if there is no weak model N such that $N \sqsupseteq M$. \square

The set of weak models for a P2P \mathcal{PS} system will be denoted by $\mathcal{WM}(\mathcal{PS})$, whereas the set of preferred weak models will be denoted by $\mathcal{PWM}(\mathcal{PS})$.

Proposition 1 For any P2P system \mathcal{PS} , \sqsupseteq defines a partial order on the set of weak models of \mathcal{PS} . \square

The next theorem shows that P2P systems always admit preferred weak models.

Theorem 1 For every consistent P2P system \mathcal{PS} , $\mathcal{PWM}(\mathcal{PS}) \neq \emptyset$. \square

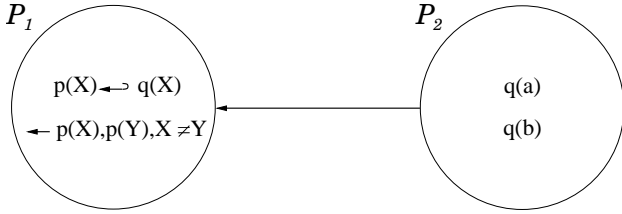


Figure 3: The system \mathcal{PS}

Example 3 Consider the P2P system \mathcal{PS} depicted in Figure 3. \mathcal{P}_2 contains the facts $q(a)$ and $q(b)$, whereas \mathcal{P}_1 contains the mapping rule $p(X) \leftarrow q(X)$ and the constraint $\leftarrow p(X), p(Y), X \neq Y$. The weak models of the system are $M_0 = \{q(a), q(b)\}$, $M_1 = \{q(a), q(b), p(a)\}$ and $M_2 = \{q(a), q(b), p(b)\}$ whereas the preferred weak models are M_1 and M_2 .

We conclude this section showing how a classical problem can be expressed using the preferred weak model semantics.

Example 4 Three-colorability. We are given two peers \mathcal{P}_1 , containing a set of nodes, defined by a unary relation *node*, and a set of colors, defined by the unary predicate *color*, and \mathcal{P}_2 , containing the mapping rule

$$\text{colored}(X, C) \leftarrow 1:\text{node}(X), 1:\text{color}(C)$$

and the integrity constraints

$$\begin{aligned} &\leftarrow \text{colored}(X, C_1), \text{colored}(X, C_2), C_1 \neq C_2 \\ &\leftarrow \text{edge}(X, Y), \text{colored}(X, C), \text{colored}(Y, C) \end{aligned}$$

stating, respectively, that a node cannot be colored with two different colors and two connected nodes cannot be colored with the same color. The mapping rule states that the node x can be colored with the color c , only if in doing this no constraint is violated, that is if the node x is colored with a unique color and there is no adjacent node colored with the same color. Each preferred weak model computes a maximal subgraph which is three-colorable. \square

Prioritized programs and preferred stable models

We now present an alternative semantics based on the rewriting of mapping rules into prioritized rules (Brewka, Niemela, & Truszczyński 2003; Sakama & Inoue 2000). For the sake of notation we consider exclusive disjunctive rules of the form $A \oplus A' \leftarrow \mathcal{B}$ whose meaning is that if \mathcal{B} is *true* then exactly one of A or A' must be *true*. Note that the rule $A \oplus A' \leftarrow \mathcal{B}$ is just shorthand for $A \leftarrow \mathcal{B}, \text{not } A', A' \leftarrow \mathcal{B}, \text{not } A$ and $\leftarrow A, A'$.

This illustrates the possibility of eliminating ‘exclusive’ disjunctions from a logic program.

Definition 4 Given a P2P system $\mathcal{PS} = \mathcal{D} \cup \mathcal{LP} \cup \mathcal{MP} \cup \mathcal{IC}$ and a mapping rule $r = i : p(x) \leftarrow \mathcal{B}$, then

- $Rew(r)$ denotes the pair $(i : p(x) \oplus i : p'(x) \leftarrow \mathcal{B}, i : p(x) \succeq i : p'(x))$, consisting of a disjunctive mapping rule and a priority statement,
- $Rew(\mathcal{MP}) = (\{Rew(r)[1] \mid r \in \mathcal{MP}\}, \{Rew(r)[2] \mid r \in \mathcal{MP}\})$ and
- $Rew(\mathcal{PS}) = (\mathcal{D} \cup \mathcal{LP} \cup Rew(\mathcal{MP})[1] \cup \mathcal{IC}, Rew(\mathcal{MP})[2])$. \square

In the above definition the atom $i : p(x)$ (resp. $i : p'(x)$) means that the fact $p(x)$ is imported (resp. not imported) in the peer \mathcal{P}_i . For a given mapping rule r , $Rew(r)[1]$ (resp. $Rew(r)[2]$) denotes the first (resp. second) component of $Rew(r)$.

Example 5 Consider again the system analyzed in Example 3. The rewriting of the system is $Rew(\mathcal{PS}) = (\{q(a), q(b), p(X) \oplus p'(X) \leftarrow q(X), \leftarrow p(X), p(Y), X \neq Y, \{p(X) \succeq p'(X)\}$.

$Rew(\mathcal{PS})[1]$ has three stable models:

$$\begin{aligned} M_0 &= \{q(a), q(b), p'(a), p'(b)\}, \\ M_1 &= \{q(a), q(b), p(a), p'(b)\}, \\ M_2 &= \{q(a), q(b), p'(a), p(b)\}. \end{aligned}$$

The set of preferred stable models are $\{M_1, M_2\}$. \square

Example 6 The rewriting of the mapping rules of Example 4 consists of the ‘disjunctive’ rules

$$\begin{aligned} \text{colored}(X, \text{red}) \oplus \text{colored}'(X, \text{red}) &\leftarrow \text{node}(X) \\ \text{colored}(X, \text{blue}) \oplus \text{colored}'(X, \text{blue}) &\leftarrow \text{node}(X) \\ \text{colored}(X, \text{yellow}) \oplus \text{colored}'(X, \text{yellow}) &\leftarrow \text{node}(X) \end{aligned}$$

plus the preferences:

$$\begin{aligned} \text{colored}(X, \text{red}) &\succeq \text{colored}'(X, \text{red}) \\ \text{colored}(X, \text{blue}) &\succeq \text{colored}'(X, \text{blue}) \\ \text{colored}(X, \text{yellow}) &\succeq \text{colored}'(X, \text{yellow}) \end{aligned}$$

Given a P2P system \mathcal{PS} and a preferred stable model M for $Rew(\mathcal{PS})$ we denote with $St(M)$ the subset of non-primed atoms of M and we say that $St(M)$ is a preferred stable model of \mathcal{PS} . We denote the set of

preferred stable models of \mathcal{PS} as $\mathcal{PSM}(\mathcal{PS})$. The following theorem shows the equivalence of preferred stable models and preferred weak models.

Theorem 2 *For every P2P system \mathcal{PS} , $\mathcal{PSM}(\mathcal{PS}) = \mathcal{PWM}(\mathcal{PS})$. \square*

For the system of the previous example, $\mathcal{PSM}(\mathcal{PS}) = \{\{q(a), q(b), p(a)\}, \{q(a), q(b), p(b)\}\}$.

Query answers and complexity

We consider now the computational complexity of calculating preferred weak models and answers to queries. As a P2P system may admit more than one preferred weak model, the answer to a query is given by considering *brave* or *cautious* reasoning (also known as *possible* and *certain* semantics).

Definition 5 *Given a P2P system $\mathcal{PS} = \{\mathcal{P}_1, \dots, \mathcal{P}_n\}$ and a ground peer atom A , then A is true under*

- *brave reasoning if $A \in \bigcup_{M \in \mathcal{PWM}(\mathcal{PS})} M$,*
- *cautious reasoning if $A \in \bigcap_{M \in \mathcal{PWM}(\mathcal{PS})} M$. \square*

The following lemma states that for every P2P system \mathcal{PS} an atom is *true* in some of its preferred weak models if and only if it is *true* in some of its weak models.

Lemma 1 $\bigcup_{M \in \mathcal{PWM}(\mathcal{PS})} M = \bigcup_{N \in \mathcal{WM}(\mathcal{PS})} N$ \square

The upper bound results can be immediately fixed by considering analogous results on stable model semantics for prioritized logic programs. For disjunction-free (\vee -free)² prioritized programs deciding whether an atom is *true* in some preferred model is Σ_2^P -complete, whereas deciding whether an atom is *true* in every preferred model is Π_2^P -complete (Sakama & Inoue 2000).

Theorem 3 *Let \mathcal{PS} be a P2P system, then*

1. *Deciding whether an interpretation M is a preferred weak model of \mathcal{PS} is coNP -complete.*
2. *Deciding whether an atom A is true in some preferred weak model of \mathcal{PS} is NP -complete.*
3. *Deciding whether an atom A is true in every preferred weak model of \mathcal{PS} is in Π_2^P and coNP -hard. \square*

Conclusion

In this paper we have introduced a logic programming based framework for P2P deductive databases. The new semantics, called preferred weak model semantics, is based on a new interpretation of mapping rules. We

²The symbol \vee denotes inclusive disjunction and is different from \oplus as the latter denotes exclusive disjunction. It should be recalled that inclusive disjunction allows more than one atom to be true while exclusive disjunction allows only one atom to be true.

have presented a different characterization of the semantics based on preferred stable models for prioritized logic programs. Moreover, we have also provided some preliminary results on the complexity of answering queries in different contexts.

References

- Bernstein, P. A.; Giunchiglia, F.; Kementsietsidis, A.; Mylopoulos, J.; Serafini, L.; and Zaihrayen, I. 2002. Data management for peer-to-peer computing: A vision. *WebDB* 89–94.
- Bertossi, L., and Bravo, L. 2004. Query answering in peer-to-peer data exchange systems. *EDBT Workshops* 476–485.
- Brewka, G., and Eiter, T. 1999. Preferred answer sets for extended logic programs. *AI* 109(1-2):297–356.
- Brewka, G.; Niemela, I.; and Truszczyński, M. 2003. Answer set optimization. *IJCAI* 867–872.
- Calvanese, D.; De Giacomo, G.; Lenzerini, M.; and Rosati, R. 2004. Logical foundations of peer-to-peer data integration. *IJCAI* 241–251.
- Calvanese, D.; De Giacomo, G.; Lembo, D.; Lenzerini, M.; and Rosati, R. 2005. Inconsistency tolerance in p2p data integration: an epistemic logic approach. *DBPL* 90–105.
- Delgrande, J., P.; Schaub, T.; and Tompits, H. 2003. A framework for compiling preferences in logic programs. *TPLP* 3(2):129–187.
- Franconi, E.; Kuper, G. M.; Lopatenko, A.; and Zaihrayeu, I. 2003. A robust logical and computational characterisation of peer to-peer database systems. *DBISP2P* 64–76.
- Gelfond, M., and Lifschitz, V. 1988. The stable model semantics for logic programming. *ICLP* 1070–1080.
- Gribble, S.; Halevy, A. and Ives, Z.; Rodrig, M.; and Suciu, D. 2001. What can databases do for peer-to-peer? *WebDB* 31–36.
- Halevy, A.; Ives, Z.; Suciu, D.; and Tatarinov, I. 2003. Schema mediation in peer data management systems. *ICDT* 505–516.
- Lenzerini, M. 2002. Data integration: A theoretical perspective. *PODS* 233–246.
- Madhavan, J., and Halevy, A. Y. 2003. Composing mappings among data sources. *VLDB* 572–583.
- Sakama, C., and Inoue, K. 2000. Prioritized logic programming and its application to commonsense reasoning. *AI* 123(1-2):185–222.
- Tatarinov, I., and Halevy, A. 2004. Efficient query reformulation in peer data management systems. *SIGMOD* 539–550.