

Inconsistency Tolerance in P2P Data Integration: an Epistemic Logic Approach

*Diego Calvanese¹, Giuseppe De Giacomo², Domenico Lembo²,
Maurizio Lenzerini², and Riccardo Rosati²*

¹ Faculty of Computer Science, Free University of Bolzano/Bozen,
Piazza Domenicani 3, I-39100 Bolzano, Italy
calvanese@inf.unibz.it

² Dipartimento di Informatica e Sistemistica, SAPIENZA Università di Roma,
Via Ariosto 25, 00185 Roma, Italy
degiacomo, lembo, lenzerini, rosati@dis.uniroma1.it

Abstract. We study peer-to-peer data integration, where each peer models an autonomous system that exports data in terms of its own schema, and data interoperation is achieved by means of mappings among the peer schemas, rather than through a unique global schema. We propose a multi-modal epistemic logical formalization based on the idea that each peer is conceived as a rational agent that exchanges knowledge/belief with other peers, thus nicely modeling the modular structure of the system. We then address the issue of dealing with possible inconsistencies, and distinguish between two types of inconsistencies, called local and P2P, respectively. We define a nonmonotonic extension of our logic that is able to reason on the beliefs of peers under both local and P2P inconsistency tolerance. Tolerance to local inconsistency essentially means that the presence of inconsistency within one peer does not affect the consistency of the whole system. Tolerance to P2P inconsistency means being able to resolve inconsistencies arising from the interaction between peers. We study query answering in the new nonmonotonic logic, with the main goal of establishing its decidability and its computational complexity. Indeed, we show that, under reasonable assumptions on peer schemas, query answering is decidable, and is coNP-complete with respect to data complexity, i.e., the size of the data stored at the peers.

1 Introduction

In this paper we study data integration in a peer-to-peer (P2P) architecture. In a P2P data integration system (P2PDIS), each peer is an autonomous information system providing part of the overall information available from a distributed environment, and acts both as a client and as a server. Information integration in these systems does not rely on a single global view, as in traditional data integration [35]. Instead, it is achieved by establishing mappings between peers, and by exploiting such mappings to collect and merge data from the various peers when answering user queries.

P2P data integration has been the subject of several investigations in the last years. Recent papers focused on providing techniques for evolving from basic P2P networks supporting only file exchanges to more complex systems like schema-based P2P networks, capable of supporting the exchange of structured contents. From papers like [29,

6, 27, 14, 23, 44] the idea of peer data management emerges: every peer is characterized by a schema that represents the domain of interest from the peer perspective, and is equipped with mappings to other peers [40], each mapping providing a semantic relationship between pairs of peers. Data integration in such systems is typically virtual: data stored in one peer is not replicated in other peers, and when a query is posed to a peer, query processing is done by both looking at local data, and collecting relevant data from other peers according to the mappings. Cycles in the mappings pose challenging problems, and various proposals have been put forward to deal with them. For example, in [14], starting from the observation that query answering in P2PDISs in the presence of cycles in the mappings is undecidable under a first-order interpretation of such mappings, an epistemic semantics is proposed that weakens the usual semantics based on first-order logic [29], and allows for both a better modeling of the modular structure of the system, and decidable query answering (even polynomially tractable w.r.t. data complexity, under common assumptions on the various peer schemas). Some papers look at peer data management under the perspective of exchanging data between peers. Peers are again interconnected by means of mappings, but in this case, the focus is on materializing the data flowing from one peer to another [21, 4].

In this paper we are interested in virtual P2P data integration, and thus we do not deal with the issue of materializing exchanged data. In particular, we aim at addressing an important problem that is still unexplored in formal approaches to P2P data integration, namely inconsistency tolerance, i.e., how to deal with inconsistencies in the data stored by the peers.

The problem of dealing with inconsistency has its roots in studies in *belief revision and update* [3, 25] in Artificial Intelligence, which deal with the problem of integrating new information with previous knowledge. In the context of databases, where the underlying theory takes the form of a database schema and the revision process focuses on data [22], the general goal is to provide informative answers even when a database does not satisfy its integrity constraints. Most of this work relies on the notion of *repair* as introduced in [5]: a repair of a database is a new database that satisfies the constraints in the schema, and minimally differs from the original one (see, e.g., [5, 11]). Recently, some papers (see, e.g., [12, 9]) have tackled data inconsistency in a data integration setting, where the basic idea is to apply the repairs to data retrieved from the sources, again according to some minimality criteria. Instead, only few works deal with inconsistencies in P2P architectures. The approach in [8] is based on the notion of “solution” for a peer P , i.e., an instance for the peer database schema that respects both the mappings and the trust relationships that P has with other peers, and stays as close as possible to the available data in the system. This mechanism characterizes how each peer locally repairs data collected from other peers. Instead, [16] proposes to identify those mappings (called *nogoods*) causing inconsistencies with the local peer constraints, and use them to compute those facts that are consequences of some consistent subset of the global peer theory. We refer to Section 8 for a more detailed discussion on these approaches.

Differently from previous work, we provide here a formal semantics to the whole P2PDIS that does not rely on a particular repairing strategy adopted by the peers. Specifically, we follow the approach of [14], and we extend it in different ways:

- We want to stress the modularity of P2P architectures, i.e., the fact that each peer is autonomous. To this end, we formalize a P2P data integration system in terms of a multi-modal epistemic logic, namely $K45_n$ [31, 36], where each peer is modeled as a rational agent that exchanges knowledge/belief with other peers. This is in line with the idea of modeling a distributed information system in terms of multi-agent modal logic [20]. Our formalization nicely models the modular structure of the system, without resorting to any assumptions, such as acyclicity, on its topology.
- We want the P2PDIS to be inconsistency tolerant in two ways. First, we want a P2PDIS to be able to “isolate” peers that are locally inconsistent, i.e., that contain inconsistent data. Second, we aim at a system that is tolerant to P2P inconsistency, i.e., is able to repair inconsistent data coming from different peers. In order to deal with both types of tolerance, we introduce a novel nonmonotonic epistemic logic, called $K45_n^A$, which extends $K45_n$ with suitable nonmonotonic modal operators. Within this logic, a P2PDIS can be formalized in such a way that (i) each locally inconsistent peer is isolated, (ii) each locally consistent peer believes its own data, and (iii) each locally consistent peer maximizes information coming from other peers, but without falling into inconsistency.
- We want query answering in the P2PDIS to be decidable. To this aim, we consider a (relatively simple) case of practical interest in which inconsistency may arise in a P2PDIS, and exhibit an algorithm for this case that is sound and complete with respect to our $K45_n^A$ -formalization of P2PDISs, thus showing that query answering in such a setting is decidable. More precisely, we consider the setting in which P2P mappings are GAV mappings [35], and each peer schema is a relational schema with only key dependencies. Our algorithm works in coNP data complexity (i.e., the complexity with respect to the size of the data stored at the peers). We also observe that the problem in the above setting is coNP-hard, thus showing that query answering in our $K45_n^A$ -formalization of P2PDISs is coNP-complete already in the simple case considered. We argue that our technique may be generalized to more complex scenarios, and that actually query answering is always decidable in all those cases in which query answering over a single peer is decidable.

The paper is organized as follows. In Section 2, we introduce the P2PDIS framework that we will use in the rest of the paper. In Section 3, we illustrate the multi-modal epistemic logic $K45_n$, and in Section 4 we show how to formalize our P2P framework in such a logic. In Section 5, we present $K45_n^A$, which is an extension of $K45_n$ with nonmonotonic features. In Section 6, we illustrate how to use such a logic to provide an inconsistency tolerant formalization of the P2PDIS framework, and we argue about the effectiveness of our formalization, by illustrating some of its basic formal properties. In Section 7, we show that query answering in the new framework is decidable, and discuss its computational complexity under GAV mappings and key dependencies on the peer schemas. Finally, in Section 8 we discuss related work, and in Section 9 we conclude the paper.

The present paper is an extended version of [13].

2 Framework

In this section we describe the framework for P2P data integration adopted in the present paper³. We refer to a fixed, infinite, denumerable set Γ of constants. Such constants are shared by all peers, and denote the data items managed by the P2PDIS. Moreover, given a relational alphabet A , we denote with \mathcal{L}_A the set of function-free first-order logic (FOL) formulas whose relation symbols are in A and whose constants are in Γ . A FOL query over a relational alphabet A is a FOL open formula over A . A *conjunctive query* (CQ) of arity n over A is a special kind of FOL query, written in the form

$$\{\mathbf{x} \mid \exists \mathbf{y}. \text{body}_{cq}(\mathbf{x}, \mathbf{y})\}$$

where $\text{body}_{cq}(\mathbf{x}, \mathbf{y})$ is a conjunction of atoms of \mathcal{L}_A involving the free variables (also called the *distinguished* variables of the query) $\mathbf{x} = x_1, \dots, x_n$, the existentially quantified variables (also called the *non-distinguished* variables of the query) $\mathbf{y} = y_1, \dots, y_m$, and constants from Γ .

A *P2P data integration system* $\mathcal{P} = \{P_1, \dots, P_n\}$ is constituted by a set of n peers, each with its own identifier, that is unique in \mathcal{P} . In the following, we assume that a peer P_i is identified by its subscript i .

Each peer $P_i \in \mathcal{P}$ (cf. [29]) is specified by means of a tuple (G, S, L, M) , where:

- G is the *schema* of P_i , which is a finite set of formulas of \mathcal{L}_{A_G} (representing local integrity constraints), where A_G is a relational alphabet (disjoint from the other alphabets in \mathcal{P}) called the *alphabet* of P_i . For convenience, we include in the language \mathcal{L}_{A_G} of peer P_i the special sentence \perp_i that is false in every interpretation for \mathcal{L}_{A_G} . Intuitively, the peer schema provides an intensional view of the information managed by the peer.
- S is the *local source schema* of P_i , which is simply a finite relational alphabet (again disjoint from the other alphabets in \mathcal{P}), called the *local alphabet* of P_i . Intuitively, the local source schema describes the structure of the data sources of the peer (possibly obtained by wrapping physical sources), i.e., the sources where the real data managed by the peer are stored.
- L is a set of *local mapping assertions* between G and S . Each local mapping assertion is an expression of the form

$$cq_S \rightsquigarrow cq_G,$$

where cq_S and cq_G are two conjunctive queries of the same arity, respectively over the local source schema S and over the peer schema G . The local mapping assertions establish the connection between the elements of the local source schema and those of the peer schema in P_i . In particular, an assertion of the form $cq_S \rightsquigarrow cq_G$ specifies that all the data satisfying the query cq_S over the sources also satisfy the concept in the peer schema represented by the query cq_G . In the terminology used in data integration, the combination of peer schema, local source schema, and local mapping assertions constitutes a *GLAV data integration system* [35] managing a set of sound data sources S defined in terms of a (virtual) global schema G .

³ Our framework basically corresponds to the one presented in [14].

- M is a set of *P2P mapping assertions*, which specify the semantic relationships that the peer P_i has with the other peers. Each assertion in M is an expression of the form

$$cq_j \rightsquigarrow cq_i,$$

where cq_i , called the *head* of the assertion, is a conjunctive query over the peer (schema of) P_i , while cq_j , called the *tail* of the assertion, is a conjunctive query of the same arity as cq_i over (the schema of) one of the other peers in \mathcal{P} . A P2P mapping assertion $cq_j \rightsquigarrow cq_i$ from peer P_j to peer P_i expresses the fact that the P_j -concept represented by cq_j is mapped to the P_i -concept represented by cq_i . From an extensional point of view, the assertion specifies that every tuple that can be retrieved from P_j by issuing query cq_j satisfies cq_i in P_i . Observe that no limitation is imposed on the topology of the whole set of P2P mapping assertions in the system \mathcal{P} , and hence the set of all P2P mappings may be cyclic.

For each peer $P_i \in \mathcal{P}$, the tuple (G, S, L, M) is intended to provide the specification of the peer at the intensional level. To model the data managed by the system, we now introduce the notion of *extension* for a P2PDIS $\mathcal{P} = \{P_1, \dots, P_n\}$. Namely, an extension for \mathcal{P} is simply a collection of extensions, one for each peer of \mathcal{P} , i.e., a collection $\mathcal{D} = \{D_1, \dots, D_n\}$, where each D_i is an extension of (i.e., the set of tuples satisfying) the predicates in the local source schema of peer P_i .

As already said, in our formalization of a P2PDIS, a single peer is seen as a data integration system [35] equipped with a set M of P2P mappings assertions. This characterization allows us to properly represent the typical scenario in which an organization, which has its own data sources within its own information systems, wants to connect itself with other organizations in a network of peers to both export and import data, still keeping hidden how information is internally managed. Hence, each organization shares in the peer network only its global view of the information it manages, expressed in terms of a peer schema. Obviously, our formalization captures also those situations in which peers have a simpler structure (e.g., are database systems exporting their schema). On the other hand, one peer would like to allow (some) other peers to access only portions of its schema, and to extract therefore only part of its own data, thus setting the stage for the issues of privacy and authorization. These aspects are however outside the scope of the present paper.

Each peer in a P2PDIS can be queried by an external user or by another peer (both acting as peer's client). Queries to a peer P_i must be posed over the peer schema in a query language that the peer can process, and which we call *the language accepted by P_i* . In principle, each peer may have its own accepted query language. However, for simplicity we assume that all peers in a P2PDIS accept the same query language, and that such a language is a fragment of FOL that contains at least the class of conjunctive queries (indeed, since for each P_j -to- P_i mapping assertion $cq_j \rightsquigarrow cq_i$, by definition, cq_j is a conjunctive query, it is reasonable to require that such queries are accepted by P_j).

A P2PDIS, together with one extension, is intended to be queried by a client. A client enquires the whole system by accessing any peer P of \mathcal{P} , and by issuing a *query* q to P . The query q is processed by P if and only if q is expressed over the schema of P and is accepted by P .

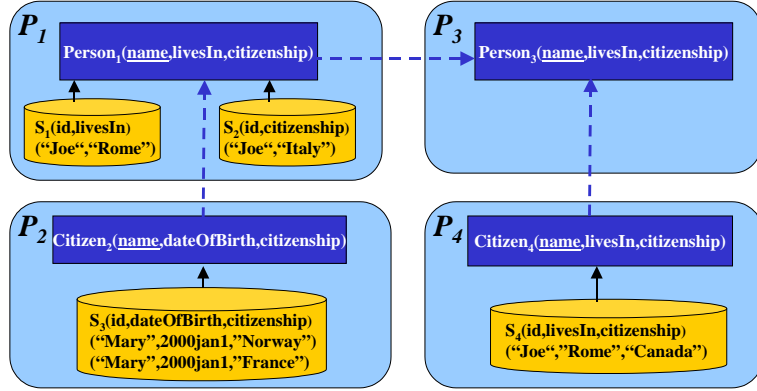


Fig. 1. A P2P system

Example 1. Let us consider the P2PDIS in Figure 1, in which we have 4 peers P_1 , P_2 , P_3 , and P_4 , whose schemas contain only FOL formulas specifying key dependencies on relations. For ease of presentation we use relation symbols with attribute names, and underline the attributes corresponding to the key of the relation (when present).

The schema of peer P_1 is formed by the relation symbol $\text{Person}_1(\underline{\text{name}}, \text{livesIn}, \text{citizenship})$, where name is the key. P_1 contains the local source $S_1(\text{id}, \text{livesIn})$, mapped to the peer schema by the local mapping assertion $\{x, y \mid S_1(x, y)\} \rightsquigarrow \{x, y \mid \exists z. \text{Person}_1(x, y, z)\}$, and the local source $S_2(\text{id}, \text{citizenship})$, mapped to the peer schema by the local mapping assertion $\{x, y \mid S_2(x, y)\} \rightsquigarrow \{x, z \mid \exists y. \text{Person}_1(x, y, z)\}$. Moreover, it has a P2P mapping assertion $\{x, z \mid \exists y. \text{Citizen}_2(x, y, z)\} \rightsquigarrow \{x, z \mid \exists y. \text{Person}_1(x, y, z)\}$ relating information in peer P_2 to those in peer P_1 . Finally, P_1 has an extension D_1 represented in Figure 1 by the facts $S_1(\text{"Joe"}, \text{"Rome"})$, $S_2(\text{"Joe"}, \text{"Italy"})$.

The schema of P_2 is composed by the relation symbol $\text{Citizen}_2(\underline{\text{name}}, \text{dateOfBirth}, \text{citizenship})$, whereas the P_2 local source schema contains $S_3(\text{id}, \text{dateOfBirth}, \text{citizenship})$, mapped to the peer schema through the local mapping $\{x, y, z \mid S_3(x, y, z)\} \rightsquigarrow \{x, y, z \mid \text{Citizen}_2(x, y, z)\}$. P_2 has no P2P mappings, whereas it has an extension D_2 represented by the facts $S_3(\text{"Mary"}, \text{2000jan1}, \text{"Norway"})$, $S_3(\text{"Mary"}, \text{2000jan1}, \text{"France"})$.

P_3 has $\text{Person}_3(\underline{\text{name}}, \text{livesIn}, \text{citizenship})$ as schema, contains no local sources (and therefore has no local mapping assertions and no local extensions), and has a P2P mapping $\{x, y, z \mid \text{Person}_1(x, y, z)\} \rightsquigarrow \{x, y, z \mid \text{Person}_3(x, y, z)\}$ with P_1 , and a P2P mapping $\{x, y, z \mid \text{Citizen}_4(x, y, z)\} \rightsquigarrow \{x, y, z \mid \text{Person}_3(x, y, z)\}$ with P_4 .

P_4 has $\text{Citizen}_4(\underline{\text{name}}, \text{livesIn}, \text{citizenship})$ as schema, and a local source $S_4(\text{id}, \text{livesIn}, \text{citizenship})$ mapped to the peer schema through the local mapping $\{x, y, z \mid S_4(x, y, z)\} \rightsquigarrow \{x, y, z \mid \text{Citizen}_4(x, y, z)\}$. P_4 has no P2P mappings, whereas it has an extension D_4 represented by the fact $S_4(\text{"Joe"}, \text{"Rome"}, \text{"Canada"})$.

Obviously, the P2PDIS extension \mathcal{D} is given by the union of D_1 , D_2 and D_4 . \square

3 The multi-modal epistemic logic $K45_n$

One of the goals of this paper is to present a multi-model epistemic formalization of the framework described in the previous section. To this end, we will use a specific modal epistemic logic, called $K45_n$, which is the multi-modal version of the epistemic logic $K45$ with n modal operators [33, 31, 36]. The aim of this section is to introduce such logic.

The language $\mathcal{L}(K45_n)$ of $K45_n$ is a first-order multi-modal language over a relation alphabet A (and fixed set Γ of constants) with a set $\mathbf{K}_1, \dots, \mathbf{K}_n$ of modal operators. $K45_n$ formulas are inductively defined as follows:

- an atom $r(\mathbf{c})$, where $r \in A$ and \mathbf{c} is a tuple of variable or constant symbols, is a $K45_n$ formula;
- an equality $t_1 = t_2$, where t_1 and t_2 are variable or constant symbols, is a $K45_n$ formula;
- if ϕ is a $K45_n$ formula, $\neg\phi$ and $\mathbf{K}_i\phi$, where $i \in \{1, \dots, n\}$, are $K45_n$ formulas;
- if ψ is a $K45_n$ formula with open variables \mathbf{x} , $\exists\mathbf{x}.\psi$ is a $K45_n$ formula;
- if ϕ_1 and ϕ_2 are $K45_n$ formulas, $\phi_1 \wedge \phi_2$ is a $K45_n$ formula.

Formulas without occurrences of \mathbf{K}_i are said to be *objective formulas* since they talk about what is true. Instead, formulas of the form $\mathbf{K}_i\phi$ are said to be *subjective formulas* since they are used to formalize the epistemic state of an agent. Obviously there are formulas that are neither objective nor subjective. Informally, a subjective formula $\mathbf{K}_i\phi$ should be read as “ ϕ is known to hold by the agent i ”. In fact, in $K45_n$, we do not have that what is known by an agent must hold in the real world: the agent can have inaccurate knowledge of what is true, i.e., believe something to be true although in reality it is false. Often this kind of knowledge is referred to as belief. On the other hand, $K45_n$ states that the agent has complete information on what it knows, i.e., if agent i knows ϕ then it knows of knowing ϕ , and if agent i does not know ϕ , then it knows that it does not know ϕ . In other words, the following assertions hold for every $K45_n$ formula ϕ (in such assertions, \supset denotes material implication):

$$\begin{aligned} \mathbf{K}_i\phi \supset \mathbf{K}_i(\mathbf{K}_i\phi) & \quad \text{known as the axiom schema 4,} \\ \neg\mathbf{K}_i\phi \supset \mathbf{K}_i(\neg\mathbf{K}_i\phi) & \quad \text{known as the axiom schema 5.} \end{aligned}$$

On the other hand, the assertion $\mathbf{K}_i\phi \supset \phi$ does not hold, i.e., what is known is not necessarily true.

To define the semantics of $K45_n$, we start from first-order interpretations. In particular, we restrict our attention to first-order interpretations that share a fixed infinite domain Δ . We further assume that for each domain element $d \in \Delta$, we have a unique constant $c_d \in \Gamma$ that denotes exactly d , and, vice versa, that every constant $c_d \in \Gamma$ denotes exactly one domain element $d \in \Delta$ ⁴. In particular this implies that equality never holds between two distinct constants (i.e., we are imposing the *unique name assumption*).

We adopt the so-called *possible-worlds* semantics (see e.g., [32, 31]): in a given world (initial world) each agent believes a set of worlds (not necessarily containing the

⁴ In other words, the constants in Γ act as *standard names* [36].

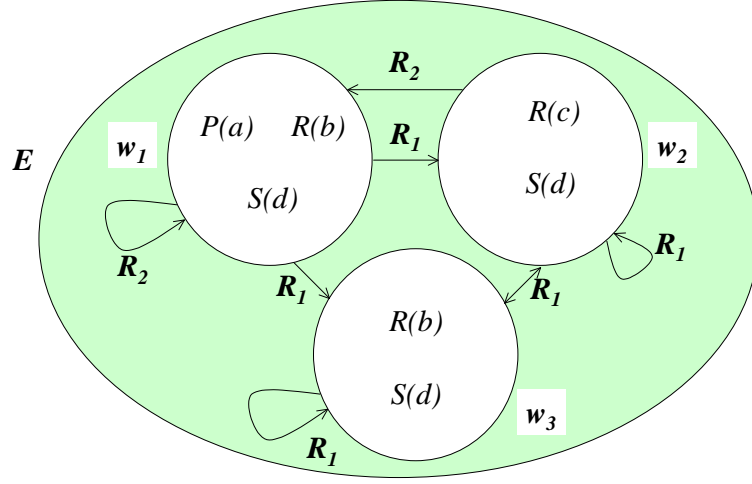


Fig. 2. A $K45_n$ -structure

initial world) be possibly the real world, and it believes that a sentence ϕ is true if ϕ is true in all the worlds in this set. Conversely, the agent does not believe that ϕ is true if there is a world in the set in which ϕ is not true.

As formal model for possible world semantics we consider $K45_n$ -structures. A $K45_n$ -structure is a Kripke structure E of the form $(W, \{R_1, \dots, R_n\}, V)$, where:

- W is a set whose elements are the *possible worlds*;
- V is a function assigning to each $w \in W$ a first-order interpretation $V(w)$;
- each R_i , called the *accessibility relation* for the modality \mathbf{K}_i , is a binary relation over W , with the following constraints:

if $(w_1, w_2) \in R_i$ and $(w_2, w_3) \in R_i$ then $(w_1, w_3) \in R_i$, i.e., R_i is transitive;
if $(w_1, w_2) \in R_i$ and $(w_1, w_3) \in R_i$ then $(w_2, w_3) \in R_i$, i.e., R_i is Euclidean.

Intuitively, $(w_k, w_j) \in R_i$ specifies that, in world w_k , the agent i believes that w_j is a possible world.

In Figure 2, we give an example of a simple $K45_n$ -structure $E = (W, \{R_1, R_2\}, V)$, which is represented as a labelled directed graph in which each node is a world of W , and there is an edge labelled with R_i from w_j to w_k for each $(w_j, w_k) \in R_i$. In the example, $W = \{w_1, w_2, w_3\}$, and in world w_1 we have that $P^{V(w_1)} = \{a\}$, $R^{V(w_1)} = \{b\}$, $S^{V(w_1)} = \{d\}$, represented by facts $P(a)$, $R(b)$ and $S(d)$ in node w_1 . Analogously, $V(w_2)$ is represented by facts $R(c)$ and $S(d)$ in node w_2 and $V(w_3)$ is represented by facts $R(b)$ and $S(d)$ in node w_3 . Furthermore, $R_1 = \{(w_1, w_2), (w_2, w_2), (w_2, w_3), (w_3, w_2), (w_3, w_3), (w_1, w_3)\}$, and $R_2 = \{(w_2, w_1), (w_1, w_1)\}$.

A projection π_i of a graph G representing a $K45_n$ -structure is the sub-graph of G containing all the edges labelled with R_i and the nodes that they connect, i.e., it

is the sub-graph which represents only the accessibility relation R_i . Then, since each accessibility relation of a $K45_n$ -structure is transitive and Euclidean, in each connected component of a projection π_i , each node is either (a) a node with only outgoing edges, i.e., it is a *root*, or (b) a node connected via a direct edge to every other non-root node of the projection π_i , itself included (note that, by Euclidean property, if $(w_j, w_k) \in R_i$, then also $(w_k, w_k) \in R_i$). In the example of Figure 2 each projection has exactly one root, namely, w_1 for the projection π_1 corresponding to accessibility relation R_1 and w_2 for the projection π_2 corresponding to R_2 . Obviously, in general a projection may have more than one root, or also none.

A $K45_n$ -interpretation is a pair (E, w) , where $E = (W, \{R_1, \dots, R_n\}, V)$ is a $K45_n$ -structure, and w is a world in W , called the initial world. A sentence (i.e., a closed formula) ϕ is true in an interpretation (E, w) (or, is true on world $w \in W$ in E), written $E, w \models \phi$ iff:⁵

$$\begin{aligned} E, w \models P(c_1, \dots, c_n) & \text{ iff } V(w) \models P(c_1, \dots, c_n) \\ E, w \models \phi_1 \wedge \phi_2 & \text{ iff } E, w \models \phi_1 \text{ and } E, w \models \phi_2 \\ E, w \models \neg\phi & \text{ iff } E, w \not\models \phi \\ E, w \models \exists x. \psi & \text{ iff } E, w \models \psi_c^x \text{ for some constant } c \\ E, w \models \mathbf{K}_i\phi & \text{ iff } E, w' \models \phi \text{ for every } w' \text{ such that } (w, w') \in R_i \end{aligned}$$

Informally, an objective formula ϕ_0 is true in (E, w) if ϕ_0 is true in the initial world w , no matter if it is true in all the other worlds of W , whereas a subjective formula $\mathbf{K}_i\phi$ is true in (E, w) if ϕ is true in all the worlds of W which are accessible from w , according to R_i . In other words, $\mathbf{K}_i\phi$ is true in (E, w) if ϕ is true in all worlds that the agent i believes possible in the initial world w .

For the $K45_n$ -structure shown in Figure 2, we have, for example, that

$$\begin{aligned} E, w_1 \models P(a) & \quad E, w_2 \not\models P(a) \\ E, w_1 \not\models \mathbf{K}_1R(b) & \quad E, w_1 \models \mathbf{K}_1S(d) \\ E, w_2 \models \mathbf{K}_2P(a) & \quad E, w_2 \models \mathbf{K}_2(\mathbf{K}_1S(d)). \end{aligned}$$

We say that a sentence ϕ is *satisfiable* if there exists a $K45_n$ -model for ϕ , i.e., a $K45_n$ -interpretation (E, w) such that $E, w \models \phi$, *unsatisfiable* otherwise. A *model* for a set Σ of sentences is a model for every sentence in Σ . A sentence ϕ is *logically implied* by a set Σ of sentences, written $\Sigma \models_{K45_n} \phi$, if and only if in every $K45_n$ -model (E, w) of Σ , we have that $E, w \models \phi$.

To the aim of the present paper, it is sufficient to consider in the following only sentences form $\mathcal{L}(K45_n)$ that are sentences of *modal depth 1*, i.e., such that there is no nested occurrence of a modal operator (namely, there is no occurrence of a modal operator within the scope of another modal operator). It is known that, in order to establish satisfiability or logical implication of $\mathcal{L}(K45_n)$ sentences of modal depth of at most 1, we can, without loss of generality, restrict our attention to canonical interpretations (and therefore from now on we will only refer to such interpretations). A *canonical $K45_n$ -interpretation* (E, w) , is such that the (canonical) $K45_n$ -structure $E = (W, \{R_1, \dots, R_n\}, V)$ satisfies the following conditions:

⁵ We have used ψ_c^x to denote the formula obtained from ψ by substituting each free occurrence of the variable x with the constant c .

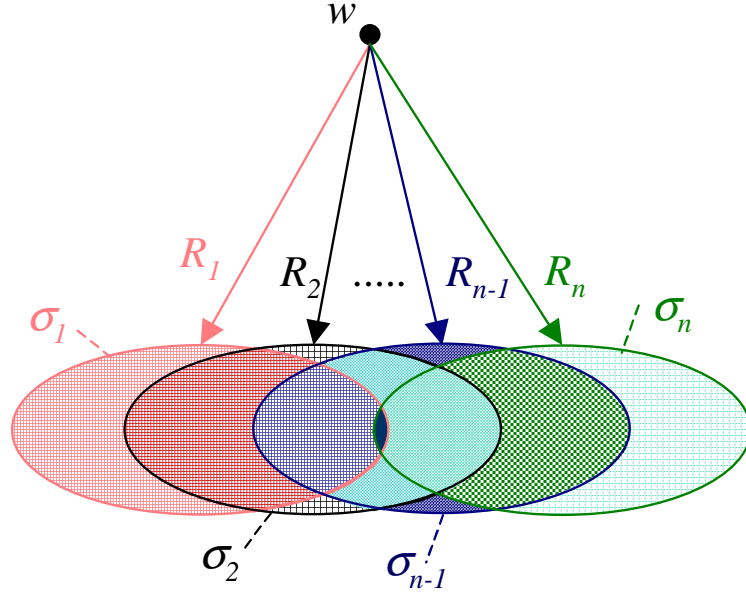


Fig. 3. A Canonical Interpretation

1. for each $w' \in W$ such that $w' \neq w$ there exists $R_i \in \{R_1, \dots, R_n\}$ such that $(w, w') \in R_i$.
2. for each $w' \in W$ there exists no $R_i \in \{R_1, \dots, R_n\}$ such that $(w', w) \in R_i$;
3. for each $w', w'' \in W$ such that $w' \neq w$ and $w'' \neq w$, and for each $R_i \in \{R_1, \dots, R_n\}$, if $(w', w'') \in R_i$ then $(w'', w') \in R_i$.

In other words, each world $w' \in W$, different from w , is accessible from w via at least one accessibility relation R_i , while w is not accessible by any world (including itself). Moreover, give a relation R_i , the set $\sigma_i = \{w' \mid (w, w') \in R_i\}$ (which does not contain w) forms a strongly connected graph, called the K_i -cluster of the structure E .

In Figure 3, such a graph is given in a compact form. The edge R_i is a representative of all the edges from w to nodes in σ_i . Furthermore, the nodes in the K_i -cluster σ_i are collectively depicted as a cloud to render that they are a strongly connected component. Notice that different K_i -clusters may be overlapping, since different accessibility relations may have pairs of worlds in common.

4 Formalization of P2P data integration systems in $K45_n$

By virtue of the characteristics mentioned in the previous section, and based on the premise that each peer in the system can be seen as a rational agent, we argue that $K45_n$ is well-suited to formalize P2PDISs of the kind presented in Section 2. The goal of this section is to present such a formalization.

Let $\mathcal{P} = \{P_1, \dots, P_n\}$ be a P2PDIS. For each peer $P_i = (G, S, L, M)$ we use a modal operator \mathbf{K}_i to formalize its epistemic state, i.e., specify the sentences that P_i believes to hold. To this aim, we transform the specification of P_i in a such way that each formula expressed on its alphabet or on its local alphabet, is put in the scope of the modality \mathbf{K}_i . Formally, for each P_i we define the $K45_n$ theory $\mathcal{T}_K(P_i)$ as follows:

- Schema G of P_i : for each sentence ϕ in G , $\mathcal{T}_K(P_i)$ contains the sentence

$$\mathbf{K}_i\phi$$

Observe that ϕ is a function-free first-order sentence expressed in the alphabet of P_i , which is disjoint from the alphabets of all the other peers in \mathcal{P} . The intended meaning of $\mathbf{K}_i\phi$ is that peer P_i believes that the sentence ϕ holds, and for an epistemic interpretation (E, w) to satisfy $\mathbf{K}_i\phi$, ϕ has to be true in all worlds believed possible from w according to the accessibility relation R_i in E . Therefore, we ascribe to P_i the characteristic of believing all assertions that specify the corresponding peer schema.

- Local mapping assertions L between G and the local source schema S : for each mapping assertion $\{\mathbf{x} \mid \exists \mathbf{y}. \text{body}_{cq_S}(\mathbf{x}, \mathbf{y})\} \rightsquigarrow \{\mathbf{x} \mid \exists \mathbf{z}. \text{body}_{cq_G}(\mathbf{x}, \mathbf{z})\}$ in L , $\mathcal{T}_K(P_i)$ contains the sentence

$$\mathbf{K}_i(\forall \mathbf{x}. \exists \mathbf{y}. \text{body}_{cq_S}(\mathbf{x}, \mathbf{y}) \supset \exists \mathbf{z}. \text{body}_{cq_G}(\mathbf{x}, \mathbf{z}))$$

Analogously to sentences in G , local mapping assertions are considered local knowledge of the peer and therefore are put in the scope of \mathbf{K}_i . In other words, each peer believes its own mappings to its local sources.

- P2P mapping assertions M : for each P2P mapping assertion $\{\mathbf{x} \mid \exists \mathbf{y}. \text{body}_{cq_j}(\mathbf{x}, \mathbf{y})\} \rightsquigarrow \{\mathbf{x} \mid \exists \mathbf{z}. \text{body}_{cq_i}(\mathbf{x}, \mathbf{z})\}$ between the peer j and the peer i in M , $\mathcal{T}_K(P_i)$ contains the sentence

$$\forall \mathbf{x}. \mathbf{K}_j(\exists \mathbf{y}. \text{body}_{cq_j}(\mathbf{x}, \mathbf{y})) \supset \mathbf{K}_i(\exists \mathbf{z}. \text{body}_{cq_i}(\mathbf{x}, \mathbf{z})) \quad (1)$$

In words, this sentence captures the following intuition: for each tuple of values \mathbf{t} , if peer j believes the sentence $\exists \mathbf{y}. \text{body}_{cq_j}(\mathbf{t}, \mathbf{y})$, then peer i believes the sentence $\exists \mathbf{z}. \text{body}_{cq_i}(\mathbf{t}, \mathbf{z})$.

We denote by $\mathcal{T}_K(\mathcal{P})$ the theory corresponding to the P2PDIS \mathcal{P} , i.e., $\mathcal{T}_K(\mathcal{P}) = \bigcup_{i=1, \dots, n} \mathcal{T}_K(P_i)$. Now, in order to take into account also the extensions of the system \mathcal{P} in our $K45_n$ formalization, we specify an additional axiom to be added to $\mathcal{T}_K(\mathcal{P})$ for modeling the data stored at the various peers. In particular, an extension $\mathcal{D} = \{D_1, \dots, D_n\}$ for a P2PDIS \mathcal{P} is modeled as a $K45_n$ sentence $DB(\mathcal{D})$ representing all facts corresponding to the tuples stored in the peer sources, i.e., $DB(\mathcal{D}) = \bigwedge_{i=1}^n DB(D_i)$ where $DB(D_i) = \mathbf{K}_i(\bigwedge_{t \in r^{D_i}} r(t))$.

To sum up, the pair $(\mathcal{P}, \mathcal{D})$ constituted by a P2PDIS \mathcal{P} and one extension $\mathcal{D} = \{D_1, \dots, D_n\}$ for \mathcal{P} , is formalized as the $K45_n$ theory $\mathcal{T}_K(\mathcal{P}) \cup DB(\mathcal{D})$. Notice that all sentences in the theory have modal depth 1.

Example 2. We provide now the formalization of the P2PDIS of Example 1. The theory $\mathcal{T}_K(P_1)$ modeling peer P_1 is the conjunction of:

$$\begin{aligned} & \mathbf{K}_1(\forall x, y, y', z, z'. \text{Person}_1(x, y, z) \wedge \text{Person}_1(x, y', z') \supset y = y' \wedge z = z') \\ & \mathbf{K}_1(\forall x, y. \mathbf{S}_1(x, y) \supset \exists z. \text{Person}_1(x, y, z)) \\ & \mathbf{K}_1(\forall x, z. \mathbf{S}_2(x, z) \supset \exists y. \text{Person}_1(x, y, z)) \\ & \forall x, z. \mathbf{K}_2(\exists y. \text{Citizen}_2(x, y, z)) \supset \mathbf{K}_1(\exists y. \text{Person}_1(x, y, z)) \end{aligned}$$

Notice that in the first row, the FOL sentence in the scope of the modal operator \mathbf{K}_1 encodes the key dependencies specified over the peer schema (the same will be done in the following for the other peers). Furthermore, at the extensional level, the peer P_1 is modeled by the formula

$$DB(D_1) = \mathbf{K}_1(\mathbf{S}_1(\text{"Joe"}, \text{"Rome"}) \wedge \mathbf{S}_2(\text{"Joe"}, \text{"Italy"})).$$

The theory $\mathcal{T}_K(P_2)$ modeling peer P_2 is the conjunction of:

$$\begin{aligned} & \mathbf{K}_2(\forall x, y, y', z, z'. \text{Citizen}_2(x, y, z) \wedge \text{Citizen}_2(x, y', z') \supset y = y' \wedge z = z') \\ & \mathbf{K}_2(\forall x, y, z. \mathbf{S}_3(x, y, z) \supset \text{Citizen}_2(x, y, z)) \end{aligned}$$

At the extensional level, the peer P_1 is modeled by the formula

$$DB(D_2) = \mathbf{K}_2(\mathbf{S}_3(\text{"Mary"}, \text{"2000jan1"}, \text{"Norway"}) \wedge \mathbf{S}_3(\text{"Mary"}, \text{"2000jan1"}, \text{"France"})).$$

The theory $\mathcal{T}_K(P_3)$ modeling peer P_3 is the conjunction of:

$$\begin{aligned} & \mathbf{K}_3(\forall x, y, y', z, z'. \text{Person}_3(x, y, z) \wedge \text{Person}_3(x, y', z') \supset y = y' \wedge z = z') \\ & \forall x, y, z. \mathbf{K}_1(\text{Person}_1(x, y, z)) \supset \mathbf{K}_3(\text{Person}_3(x, y, z)) \\ & \forall x, y, z. \mathbf{K}_4(\text{Citizen}_4(x, y, z)) \supset \mathbf{K}_3(\text{Person}_3(x, y, z)) \end{aligned}$$

No extension is given for peer P_3 , and hence no formula modeling such extension is needed.

The theory $\mathcal{T}_K(P_4)$ modeling peer P_4 is the conjunction of:

$$\begin{aligned} & \mathbf{K}_4(\forall x, y, y', z, z'. \text{Citizen}_4(x, y, z) \wedge \text{Citizen}_4(x, y', z') \supset y = y' \wedge z = z') \\ & \mathbf{K}_4(\forall x, y, z. \mathbf{S}_4(x, y, z) \supset \text{Citizen}_4(x, y, z)) \end{aligned}$$

At the extensional level, the peer P_1 is modeled by the formula

$$DB(D_2) = \mathbf{K}_2(\mathbf{S}_4(\text{"Joe"}, \text{"Rome"}, \text{"Canada"})).$$

Finally, $DB(\mathcal{D}) = DB(D_1) \wedge DB(D_2) \wedge DB(D_4)$. \square

As we said in Section 2, a client of the P2PDIS interacts with one of the peers, say peer P_i , posing a *query* to it, where a query q is an open formula $q(\mathbf{x})$ with free variables \mathbf{x} expressed in the language accepted by P_i (we recall that such a language is a subset of first-order logic).

It is immediate to specify what is the meaning in our $K45_n$ formalization of a query q posed to a peer $P_i = (G, S, L, M)$ of \mathcal{P} with respect to an extension \mathcal{D} . In particular, the semantics of query q is defined as the set of tuples that satisfy such query in every model of the $K45_n$ theory representing $(\mathcal{P}, \mathcal{D})$, i.e., $ANS_{K45_n}(q, i, \mathcal{P}, \mathcal{D}) = \{t \mid T_K(\mathcal{P}) \cup DB(\mathcal{D}) \models_{K45_n} \mathbf{K}_i q(t)\}$, where $q(t)$ denotes the sentence obtained from the open formula $q(x)$ by replacing all occurrences of the free variables in x with the corresponding constants in t .

As we said at the beginning of this section, we argue that $K45_n$ is well-suited to formalize P2PDISs. Indeed, one possible choice for formalizing such systems is classical first order logic (FOL). In this case, P2P mappings would be simply represented as logical implication, analogously to local mappings. However, in [14] we present several motivations for resorting to epistemic logic. One of the main motivations, is that query answering in cyclic P2PDISs is undecidable, even for empty peer schemas, whereas, due to the fact epistemic logic weakens the semantics of P2P mappings, query answering becomes decidable (and, actually, it can be solved in polynomial time in the size of the extension of the system, for commonly adopted forms of peer schemas). So, it is interesting to comment on how $K45_n$ weakens the meaning of P2P mappings wrt classical FOL. The basic idea is that, by using the $K45_n$ formalization of a P2P mapping (see sentence 1 above), only tuples that are *believed* to satisfy $\{x \mid body_{cq_j}(x, y)\}$ are forced to satisfy $\{x \mid body_{cq_i}(x, y)\}$, and therefore only such tuples flow from peer P_j to peer P_i . This is somehow coherent with the following intuitive reading of the above mapping: in order for peer P_i to deduce which are the tuples satisfying $\{x \mid body_{cq_i}(x, y)\}$, it should issue query $\{x \mid body_{cq_j}(x, y)\}$ to peer P_j , and conclude that all the corresponding answers will satisfy query $\{x \mid body_{cq_i}(x, y)\}$.

Another interesting observation on the difference between the FOL semantics and the epistemic semantics has to do with the “direction” of the P2P mapping. While in FOL, a P2P mapping from peer P_j to peer P_i may cause knowledge flowing from peer P_i to peer P_j , this cannot happen in epistemic logic. Indeed in FOL an implication of the form $\alpha_j \supset \beta_i$ is equivalent to $\neg\beta_i \supset \neg\alpha_j$. Now, if $\neg\beta_i$ can be deduced in the peer P_i then $\neg\alpha_j$ holds in peer P_j and therefore, together with the formulas in the schema of P_j , $\neg\alpha_j$ may affect answers to queries posed to P_j . On the contrary, in the $K45_n$ formalization, the above mapping would be represented by the formula $\mathbf{K}_j\alpha_j \supset \mathbf{K}_i\beta_i$, whose converse is $\neg\mathbf{K}_i\beta_i \supset \neg\mathbf{K}_j\alpha_j$. Now, if $\mathbf{K}_i\neg\beta_i$ holds in peer P_i , then the above mapping only implies that $\neg\mathbf{K}_j\alpha_j$ holds in peer P_j (and it does not imply $\mathbf{K}_j\neg\alpha_j$). However, since both the schema of P_j and the queries to P_j are formalized through formulas of the form $\mathbf{K}_j\phi$, where ϕ is objective, the above conclusion $\neg\mathbf{K}_j\alpha_j$ does not affect answers to queries posed to P_j . The only exception to this is if the peer P_j logically implies $\mathbf{K}_j\alpha_j$: in this case we get inconsistency of both peers. We will deal with such an issue in the next section.

Finally, observe that the formalization presented above originates from the one proposed in [14], but extends it in two ways. First, we have moved from a logic that makes use of a single modal operator to multi-modal epistemic logic, so as to stress that we are modeling each peer as an autonomous agent. Second, we have moved from the epistemic logic $S5$ to $K45$, hence dropping the assumption that what is believed by an agent is actually true. This allows for having models of the whole system even if one or more

peers are inconsistent and hence the system has no FOL models. These changes set the stage for the treatment of inconsistencies to be presented next.

5 Nonmonotonic extension of $K45_n$

The P2PDIS formalization presented in the previous section is not well suited for managing the presence of inconsistent data. Indeed, as shown in the next section, query answering under our $K45_n$ formalization becomes meaningless (i.e., any tuple is in the answer to any query), when a peer in the system is locally inconsistent, i.e., its local data contradict the peer schema, or when data coming into a peer from other peers contradict the peer schema when combined together, or when combined with data locally managed by the peer. In order to provide a new formalization of P2PDISs, suited to deal with inconsistencies, in this section we introduce a nonmonotonic extension of the multi-modal logic $K45_n$. The new formalization of inconsistency tolerant P2PDISs based on such a nonmonotonic logic will then be given in the next section.

Informally, we extend $K45_n$ by adding a new set of modal operators $\mathbf{A}_1, \dots, \mathbf{A}_n$ to the modal language. Then, following (and generalizing) the semantic construction of the logic MKNF [37], the modal operators $\mathbf{K}_1, \dots, \mathbf{K}_n$ are interpreted as epistemic operators of minimal knowledge, and the modal operators $\mathbf{A}_1, \dots, \mathbf{A}_n$ are interpreted as epistemic operators of *justified assumption* [39], which corresponds to (the complement of) the well-known notion of *negation as failure* [38].

5.1 Adding modal operators of negation as failure

First, we introduce the language $\mathcal{L}(K45_n^A)$, which is an extension of $\mathcal{L}(K45_n)$ obtained by adding to the first-order modal language a new set of modal operators, $\mathbf{A}_1, \dots, \mathbf{A}_n$.

In order to define the semantics of $\mathcal{L}(K45_n^A)$ sentences, we first give the notion of canonical $K45_n^A$ -interpretation. Such a notion is similar to the one given for the logic $K45_n$, but presents the restriction that both the set of worlds W and the world interpretation function V are now fixed. This restriction is introduced for technical reasons, in order to allow for a well-founded definition of a preference order between structures, which will be introduced in Section 5.2 (see e.g., [37]). However, such a restriction does not affect the semantics of $K45_n^A$ (with respect to satisfiability of a formula of modal depth 1).

Let \mathcal{I} be the set of all FOL interpretations (over our relational alphabet) with fixed domain Δ , we define the set of worlds $\mathcal{W}_c = \mathcal{W}_0 \cup \mathcal{W}_1$, where:

$$\begin{aligned}\mathcal{W}_0 &= \{(I, 0) \mid I \in \mathcal{I}\} \\ \mathcal{W}_1 &= \{(I, 1) \mid I \in \mathcal{I}\}\end{aligned}$$

That is, \mathcal{W}_c contains two distinct elements for each FOL interpretation $I \in \mathcal{I}$. The worlds from \mathcal{W}_0 will be the ones from which the initial world of $K45_n^A$ interpretations will be picked, while the worlds from \mathcal{W}_1 will be used for all the other possible worlds in $K45_n^A$ interpretations.

Moreover, we define the following world interpretation function $V_c : \mathcal{W}_c \rightarrow \mathcal{I}$:

$$\text{for each } j \in \{0, 1\} \text{ and for each } w = (I, j) \in \mathcal{W}_c, \quad V_c(w) = I.$$

Namely, I is the interpretation that V_c associates with a world (I, j) in \mathcal{W}_c .

A *canonical $K45_n^A$ -interpretation* (E, w) , is such that the (canonical) $K45_n^A$ -structure $E = (\mathcal{W}_c, \{R_1, \dots, R_n, R_1^a, \dots, R_n^a\}, V_c)$ satisfies the following conditions:

- \mathcal{W}_c and V_c are defined as above;
- $w \in \mathcal{W}_0$;
- if $(w', w'') \in R_i$ or $(w', w'') \in R_i^a$, then $w'' \in \mathcal{W}_1$;
- each R_i and each R_i^a are binary relations over W satisfying the conditions imposed on canonical $K45_n$ -interpretations (see conditions 1, 2, and 3 in Section 3).

Notice that, from the above definition, it follows that *all* canonical $K45_n$ -structures are defined over the *same* set of worlds \mathcal{W}_c and the *same* world interpretation function V_c . Furthermore, with respect to $K45_n$ -structures, $K45_n^A$ -structures have n additional accessibility relations R_1^a, \dots, R_n^a . Such relations account for the additional modal operators $\mathbf{A}_1, \dots, \mathbf{A}_n$.

Under the above conditions, we can alternatively (and more compactly) represent a canonical $K45_n^A$ -interpretation (E, w) (with $E = (\mathcal{W}_c, \{R_1, \dots, R_n, R_1^a, \dots, R_n^a\}, V_c)$) by a pair (E', w) where E' is the tuple $(\sigma_1, \dots, \sigma_n, \sigma_1^a, \dots, \sigma_n^a)$ such that, for every $i \in \{1, \dots, n\}$, σ_i is the K_i -cluster of E , i.e., $\sigma_i = \{w' \mid (w, w') \in R_i\}$ and σ_i^a is the A_i -cluster of E , i.e., $\sigma_i^a = \{w' \mid (w, w') \in R_i^a\}$ (cf. the definition of cluster of a canonical $K45_n$ -structure given in Section 3). That is, a canonical $K45_n^A$ -structure can be represented by $2n$ set of worlds, where each such set is a subset of \mathcal{W}_1 . In the following, when considering a canonical $K45_n^A$ -interpretation E , we implicitly refer to its compact representation E' .

The notion of truth of an $\mathcal{L}(K45_n^A)$ sentence in a world of a $K45_n^A$ -interpretation is analogous to the notion given in Section 3 for $\mathcal{L}(K45_n)$ sentences, with the addition of the following rule:

- $E, w \models \mathbf{A}_i \phi$ iff $E, w' \models \phi$ for each w' such that $(w, w') \in R_i^a$

Analogously to the $K45_n$ logic, it can be shown that, for a formula $\varphi \in \mathcal{L}(K45_n^A)$ of modal depth 1, φ is true in an arbitrary $K45_n^A$ -interpretation iff φ is true in a canonical $K45_n^A$ -interpretation. In other words, restricting to the set of worlds \mathcal{W}_c and interpreting \mathcal{W}_c according to V_c does not change satisfiability of formulas of modal depth 1. Consequently, from now on we restrict our attention to canonical $K45_n^A$ -interpretations only.

5.2 Nonmonotonic semantics

So far, the logic $K45_n^A$ does not appear as a significant extension of the logic $K45_n$: in particular, according to the above notion of truth, the new modal operators \mathbf{A}_i are treated just like any \mathbf{K}_i operator in $K45_n$, so there is no apparent reason to distinguish the \mathbf{A}_i 's operators from the \mathbf{K}_i 's.

Actually, the different (nonmonotonic) meaning of the two sets of modal operators in $K45_n^A$ with respect to $K45_n$ is due to the following notion of $K45_n^A$ -model for a sentence ϕ , which is obtained by imposing a preference order over $K45_n^A$ -structures satisfying ϕ .

Below we define a relation \leq_K between canonical $K45_n^A$ -structures which agree on their A -clusters, i.e., on the accessibility relations R_i^a 's.

Definition 1. Let $E = (\sigma_1, \dots, \sigma_n, \sigma_1^a, \dots, \sigma_n^a)$ and $E' = (\sigma'_1, \dots, \sigma'_n, \sigma_1^a, \dots, \sigma_n^a)$ be canonical $K45_n^A$ -structures. We say that E is \mathbf{K} -contained in E' (denoted by $E \leq_K E'$) if, for each $i \in \{1, \dots, n\}$, $\sigma_i \subseteq \sigma'_i$.

Intuitively, if E is \mathbf{K} -contained in E' , then E' has less (or equal) knowledge with respect to the modal operators \mathbf{K}_i than E , since adding possible worlds (by adding worlds to the K -clusters σ_i) enlarges the relations R_i interpreting the \mathbf{K}_i 's operators.

For instance, it can be immediately verified that, if E is \mathbf{K} -contained in E' , then, for each first-order sentence ϕ and for each $w \in W$, if $E', w \models \mathbf{K}_i \phi$ then $E, w \models \mathbf{K}_i \phi$, but not necessarily vice-versa.

We now prove that the relation \leq_K between $K45_n^A$ -structures is well-defined, since it constitutes a partial order.

Proposition 1. *The relation \leq_K between $K45_n^A$ -structures constitutes a partial order.*

Proof. It is immediate to see that, from the definition of canonical $K45_n^A$ -interpretations and Definition 1, reflexivity, antisymmetry and transitivity of \leq_K hold. Consequently, \leq_K is a partial order. \square

Definition 2. Let $\phi \in \mathcal{L}(K45_n^A)$ be a formula of modal depth 1, let $E = (\sigma_1, \dots, \sigma_n, \sigma_1^a, \dots, \sigma_n^a)$ be a canonical $K45_n^A$ -structure, and let $w \in \mathcal{W}_0$. The canonical $K45_n^A$ -interpretation (E, w) is a $K45_n^A$ -model for ϕ if the following conditions hold:

1. $E, w \models \phi$;
2. $\sigma_i = \sigma_i^a$ for each $i \in \{1, \dots, n\}$;
3. there exists no canonical $K45_n^A$ -structure E' such that $E' \neq E$, $E', w \models \phi$, and $E \leq_K E'$.

A $K45_n^A$ -model for a set Σ of sentences is a $K45_n^A$ -model for every sentence in Σ . A sentence ϕ is $K45_n^A$ -entailed by a set Σ of sentences, written $\Sigma \models_{K45_n^A} \phi$, if and only if $E, w \models \phi$ in every $K45_n^A$ -model (E, w) of Σ .

The above semantics formalizes the idea of selecting $K45_n^A$ -structures that satisfy two intuitive principles:

1. *knowledge is minimal*, which is realized through the notion of preference between structures;
2. *assumptions are justified by knowledge*, which is realized by the fact that, for each i , the meaning of the operators \mathbf{A}_i and \mathbf{K}_i is the same, since $\sigma_i = \sigma_i^a$.

Such semantic principles of minimal knowledge and justified assumptions are well-known in nonmonotonic reasoning [39, 38, 42]. In particular, we recall that the principle of justified assumption exactly corresponds to the semantics of the modal operator in Moore's autoepistemic logic [42]. Moreover, as illustrated in [37–39], the justified assumption operator exactly formalizes the complement of the notion of *negation as failure* in logic programming under the stable model semantics.

Remark. From the technical viewpoint, the above preference semantics for the logic $K45_n^A$ is a non-trivial extension of analogous semantic constructions underlying other

nonmonotonic modal logics. The main difference with respect to such previous constructions is that here, due to the presence of multiple modal operators, we cannot impose the condition that the preferred models of a theory always correspond to structures in which each accessibility relation is total (which has a syntactic counterpart in the so-called *stable sets* of modal formulas [43]). Consequently, minimality of knowledge in the preferred models is imposed via a different, although simple, condition (formally stated by Definition 1), which can be seen as a generalization of analogous minimality criteria in previous nonmonotonic modal formalisms like MKNF [37] or ground nonmonotonic modal logics [19].

To gain some intuition on the use of the operators \mathbf{K}_i and \mathbf{A}_i under the nonmonotonic semantics, let us look at a few examples.

Example 3. Consider the formula

$$\Phi_1 = \mathbf{K}_i \alpha$$

where α is an objective sentence (i.e., a sentence without occurrences of the modal operators), which can be read as “*peer*” i knows α . Then, the only $K45_n^A$ -models of the above formula Φ_1 according to Definition 2 are the ones whose \mathbf{K}_i -cluster includes all the worlds whose associated FOL interpretation satisfies α . Intuitively, this realizes a *minimal knowledge* semantics for the modal operator \mathbf{K}_i , since, in all $K45_n^A$ -models of Φ_1 , peer P_i only knows α , and therefore, for every objective sentence β such that β is not a logical consequence of α in FOL, $\mathbf{K}_i \alpha \models_{K45_n^A} \neg \mathbf{K}_i \beta$, i.e., peer P_i does not know β .⁶ \square

Example 4. Consider the formula

$$\Phi_2 = \neg \mathbf{A}_j \perp_j \supset \mathbf{K}_i \alpha$$

where α is an objective sentence, which can be read as *if peer j is consistent then peer i knows α* . Indeed, the above formula Φ_2 is equivalent to $\mathbf{A}_j \perp_j \vee \mathbf{K}_i \alpha$. Now, consider a canonical $K45_n^A$ interpretation (E, w) that satisfies Φ_2 . Then, either $E, w \models \mathbf{K}_i \alpha$ or $E, w \models \mathbf{A}_j \perp_j$. In the first case, the $K45_n^A$ interpretation is a $K45_n^A$ -model of Φ_2 if it has the form described in the previous example. In the latter case, the \mathbf{A}_j -cluster of E is empty. Now, from Definition 2, (E, w) can be a $K45_n^A$ -model of Φ_2 only if (i) the \mathbf{K}_j -cluster of E is also empty, and (ii) every canonical $K45_n^A$ interpretation obtained from (E, w) by extending the \mathbf{K}_j -cluster of E does not satisfy Φ_2 . However, it is immediate to see that the last condition is false, since Φ_2 does not impose any condition on the \mathbf{K}_j -cluster of E . Hence, (E, w) can not be a $K45_n^A$ -model of Φ_2 . Therefore, the formula Φ_2 is actually equivalent to $\mathbf{K}_i \alpha$ (since it has the same $K45_n^A$ -models of $\mathbf{K}_i \alpha$). Conversely, if we conjoin Φ_2 with the formula $\mathbf{K}_j \perp_j$ (i.e., peer P_j is inconsistent), then the $K45_n^A$ -models of $\Phi_2 \wedge \mathbf{K}_j \perp_j$ coincide with the $K45_n^A$ -models of $\mathbf{K}_j \perp_j$, hence Φ_2 becomes vacuous and has no impact of the knowledge of peer P_i . \square

⁶ Observe that this new semantics for the operators \mathbf{K}_i does not actually affect per se the answers to the queries allowed in our framework, as explained in Section 4.

Example 5. Consider the formula

$$\Phi_3 = \neg \mathbf{A}_i \neg \alpha \supset \mathbf{K}_i \alpha$$

where α is a (FOL-satisfiable) objective sentence, which can be read as *if it is consistent for peer i to assume α , then peer i knows α* . Observe that this corresponds to a well-known form of *default rule* [41]. Following the line of reasoning in the previous example, it can be shown that the formula Φ_3 is actually equivalent to $\mathbf{K}_i \alpha$, since it has the same $K45_n^A$ -models of $\mathbf{K}_i \alpha$. But if we conjoin Φ_3 with the formula $\mathbf{K}_i \neg \alpha$, then the $K45_n^A$ -models of $\Phi_3 \wedge \mathbf{K}_i \neg \alpha$ coincide with the $K45_n^A$ -models of $\mathbf{K}_i \neg \alpha$, hence Φ_3 becomes vacuous and does not lead to inconsistency of peer P_i . \square

Example 6. Finally, to further explain the differences between the operators \mathbf{K}_i and \mathbf{A}_i , we show that the two modalities are not equivalent. In particular, suppose that α is an objective sentence. We now prove that adding the formula

$$\Phi_4 = \mathbf{K}_i \alpha \equiv \mathbf{A}_i \alpha$$

to a theory T actually changes the set of $K45_n^A$ -models of T . Since $\mathbf{K}_i \alpha \equiv \mathbf{A}_i \alpha$ corresponds to the conjunction of the two formulas $\mathbf{K}_i \alpha \supset \mathbf{A}_i \alpha$ and $\mathbf{A}_i \alpha \supset \mathbf{K}_i \alpha$, we consider such two formulas:

- first, given any theory T , it is easy to see that the set of $K45_n^A$ -models of T and the set of $K45_n^A$ -models of $T \cup \{\mathbf{K}_i \alpha \supset \mathbf{A}_i \alpha\}$ coincide;
- conversely, we now show that the formula $\mathbf{A}_i \alpha \supset \mathbf{K}_i \alpha$ in general does not preserve the set of $K45_n^A$ -models. The only $K45_n^A$ -models of the empty theory are $K45_n^A$ interpretations of the form (E, w) where E is the structure in which both the \mathbf{K}_i -cluster and the \mathbf{A}_i -cluster of E coincide with the entire set of worlds \mathcal{W}_1 . Conversely, the set of $K45_n^A$ -models of $\mathbf{A}_i \alpha \supset \mathbf{K}_i \alpha$ also contains all the $K45_n^A$ interpretations of the form (E', w) where E' is such that both the \mathbf{K}_i -cluster and the \mathbf{A}_i -cluster of E' coincide with the set of worlds from \mathcal{W}_1 whose associated FOL interpretation satisfies α .

From the above argument, it follows that adding the formula $\mathbf{K}_i \alpha \equiv \mathbf{A}_i \alpha$ to a theory T in general changes the set of $K45_n^A$ -models of T . \square

6 Inconsistency tolerance

We now modify our basic framework so as to be able to handle inconsistency. In particular, we want the P2PDIS to be inconsistency-tolerant in the following sense:

1. When a peer is *locally inconsistent*, i.e., data at the sources in P_i contradict, via the local mapping, the peer schema, making the whole peer inconsistent, the P2PDIS should be equivalent to the one obtained by eliminating the peer P_i from the system. In other words, an inconsistent peer should be “isolated” from the other peers: in this way, a local inconsistency does not affect the overall consistency (and meaning) of the system.

2. In the presence of *P2P inconsistency*, i.e., when in a peer P_i the data coming from another peer P_j (through a P2P mapping) contradict the local data of P_i (or the data coming to P_i from another peer P_k), the peer P_i should not reach an inconsistent state: rather, it should discard a *minimal* amount of the data retrieved from the other peers in order to preserve consistency.

We point out that the focus of this paper is how to deal with the inconsistency that may arise in P2PDISs due to peer interactions. More precisely:

1. We do not specifically study inconsistency that may locally arise in a peer because its own data contradict local constraints specified on the peer schema. According to this vision, we do not want to impose any particular assumption on the ability of the peer to deal with local inconsistency, hence we consider each peer as a black box. Under this assumption of modularity, the most natural way to deal with the presence of an inconsistent peer in the overall P2P system is to isolate it;
2. Our treatment of P2P inconsistency is based on the assumption that each peer prefers its local data to the data coming from other peers, while it does not make any preference between data coming from different peers. We believe that these are reasonable assumptions, which may reflect the intended behavior of a P2PDIS in many application scenarios. Of course, these assumptions may not always be the appropriate ones: in particular, they might be refined and/or generalized (e.g., by using meta information on the reliability of different peers). The study of such more involved forms of P2P inconsistency tolerance is outside the scope of the present paper.

Formally, the above notions of local inconsistency and P2P inconsistency can be stated as follows. Let $\mathcal{P} = \{P_1, \dots, P_n\}$ be a P2PDIS and $\mathcal{D} = \{D_1, \dots, D_n\}$ be an extension \mathcal{D} for \mathcal{P} . We say that:

- A peer $P_i \in \mathcal{P}$ is *locally inconsistent wrt* D_i if $\mathcal{T}_{\bar{K}}(P_i) \cup DB(D_i) \models_{K45_n} \mathbf{K}_i \perp_i$, where $\mathcal{T}_{\bar{K}}(P_i)$ is obtained from $\mathcal{T}_K(P_i)$ by dropping the sentences formalizing the P2P mappings (otherwise we say that P_i is *locally consistent wrt* D_i).
- A peer $P_i \in \mathcal{P}$ is *P2P inconsistent wrt* \mathcal{D} if P_i is locally consistent wrt D_i and $\mathcal{T}_K(\mathcal{P}) \cup DB(\mathcal{D}) \models_{K45_n} \mathbf{K}_i \perp_i$.

To capture systems that are inconsistency-tolerant we move from a formalization based on the logic $K45_n$ to a new one given in terms of the nonmonotonic multi-modal logic $K45_n^A$. Indeed, $K45_n^A$ is particularly well suited for the treatment of both local and P2P inconsistency.

Handling local inconsistency. To capture tolerance w.r.t. local inconsistency, we need to refine the epistemic formalization of P2P mapping assertions presented in Section 4 as follows: for each P2P mapping assertion of peer P_i , we replace in $\mathcal{T}_K(P_i)$ the sentence (1) with

$$\forall \mathbf{x}. \neg \mathbf{A}_j \perp_j \wedge \mathbf{K}_j(\exists \mathbf{y}. \text{body}_{cq_j}(\mathbf{x}, \mathbf{y})) \supset \mathbf{K}_i(\exists \mathbf{z}. \text{body}_{cq_i}(\mathbf{x}, \mathbf{z})).$$

Informally, the above sentence captures the following intuition: for each tuple of values \mathbf{t} , if peer P_j knows the sentence $\exists \mathbf{y}. \text{body}_{cq_j}(\mathbf{t}, \mathbf{y})$ and P_j is *not locally inconsistent*, then peer P_i knows the sentence $\exists \mathbf{z}. \text{body}_{cq_i}(\mathbf{t}, \mathbf{z})$. In other words, information

flows from P_j to peer P_i through a P2P mapping assertion only if P_j is locally consistent. A part of the modification in the P2P mapping assertions described above, the new formalization of a P2PDIS coincides with the $K45_n$ one given in Section 4 (both at the intensional and extensional level).

Notice that, if a peer P_j is locally inconsistent, the P2PDIS system has $K45_n^A$ -models anyway. Formally, in a $K45_n^A$ -model (E, w) of the P2PDIS, in which $E = (\sigma_1, \dots, \sigma_n, \sigma_1^a, \dots, \sigma_n^a)$ is a $K45_n^A$ -canonical structure, we have that $\sigma_j = \sigma_j^a = \emptyset$, i.e., there are no worlds accessible from the initial world w for the modality K_j and the modality A_j . This implies that any n -tuple of values \mathbf{t} is in the answer to any query of arity n posed to P_j , but implies that also $\neg \mathbf{A}_j \perp_j$ evaluates to true. Therefore, the addition of $\neg \mathbf{A}_j \perp_j$ in the formalization of any P_j -to- P_i mapping assertion prevents the peer P_i to retrieve meaningless data from peer P_j . In other words, the above formalization makes the P2PDIS tolerant to local inconsistency, in the sense that it isolates the peers that are locally inconsistent, by simply dropping the P2P mapping assertion, whose $K45_n^A$ formalization given above indeed becomes the trivial sentence “true”. Obviously, if a client directly queries an inconsistent peer it gets contradicting, hence meaningless, answers.

We finally remark that for a P2PDIS \mathcal{P} without locally inconsistent peers, the new formalization of \mathcal{P} coincides with the formalization in the logic $K45_n$ (see Proposition 2 below).

Example 7. Consider the P2PDIS of Example 1. The $K45_n^A$ formalization, limited to the treatment of local inconsistency, is easily obtained from the $K45_n$ one by substituting the P2P mapping assertions in $\mathcal{T}_K(P_1)$ and $\mathcal{T}_K(P_3)$ of Example 2 with the following assertions:

$$\begin{aligned} \forall x, z. \neg \mathbf{A}_2 \perp_2 \wedge \mathbf{K}_2(\exists y. \text{Citizen}_2(x, y, z)) \supset \mathbf{K}_1(\exists y. \text{Person}_1(x, y, z)) \\ \forall x, y, z. \neg \mathbf{A}_1 \perp_1 \wedge \mathbf{K}_1(\text{Person}_1(x, y, z)) \supset \mathbf{K}_3(\text{Person}_3(x, y, z)) \\ \forall x, y, z. \neg \mathbf{A}_4 \perp_4 \wedge \mathbf{K}_4(\text{Citizen}_4(x, y, z)) \supset \mathbf{K}_3(\text{Person}_3(x, y, z)). \end{aligned}$$

It is easy to see that P_2 is locally inconsistent, since from tuples stored in its local source S_2 it concludes facts $\text{Citizen}_2(\text{"Mary"}, \text{"2000jan1"}, \text{"Norway"})$ and $\text{Citizen}_2(\text{"Mary"}, \text{"2000jan1"}, \text{"France"})$, which violate the key dependency in Citizen_2 . However, thanks to the above formalization, P_2 turns out to be isolated from the other peers, and therefore the P2P mapping in $\mathcal{T}_K(P_1)$ connecting P_2 to P_1 has no effects in the P2PDIS. \square

Handling both local and P2P inconsistency. We now take into account P2P inconsistency. In particular, we formalize, in $K45_n^A$, P2PDISs that are inconsistency-tolerant wrt both local and P2P mappings. Again, the $K45_n^A$ theory representing the P2PDIS \mathcal{P} , denoted by $\mathcal{T}_A(\mathcal{P})$, is similar to the theory $\mathcal{T}_K(\mathcal{P})$ defined in Section 4, but with an important difference on how to formalize P2P mapping assertions: we replace each sentence of the form (1) with

$$\forall \mathbf{x}. \neg \mathbf{A}_j \perp_j \wedge \mathbf{K}_j(\exists \mathbf{y}. \text{body}_{cq_j}(\mathbf{x}, \mathbf{y})) \wedge \neg \mathbf{A}_i(\neg \exists \mathbf{z}. \text{body}_{cq_i}(\mathbf{x}, \mathbf{z})) \supset \mathbf{K}_i(\exists \mathbf{z}. \text{body}_{cq_i}(\mathbf{x}, \mathbf{z}))$$

Informally, the above sentence captures the following intuition: for each tuple of values \mathbf{t} , if peer P_j is consistent and knows the sentence $\exists \mathbf{y}. \text{body}_{cq_j}(\mathbf{t}, \mathbf{y})$, and the sentence $\exists \mathbf{z}. \text{body}_{cq_i}(\mathbf{t}, \mathbf{z})$ is consistent with what peer P_i knows, then P_i knows the sentence $\exists \mathbf{z}. \text{body}_{cq_i}(\mathbf{t}, \mathbf{z})$. In other words, information flows from P_j to peer P_i through a P2P mapping assertion only if adding such information to P_i does not give rise to a P2P inconsistency in peer i . More precisely, the meaning of the above sentence in $K45_n^A$ is that exactly a *maximal* amount of information (i.e., a maximal set of tuples) consistent with peer i flows from peer j to peer i through the P2P mapping assertion.

We remark that the above semantics implies that: (i) when inconsistency arises between local data and non-local data in a peer, i.e., when data coming from the peer sources through the local mapping contradicts the data retrieved by a peer through a P2P mapping, then the peer always prefers the local data. Formally, in this case there is one $K45_n^A$ -model for the P2PDIS, which represents the situation in which non-local data is discarded; (ii) when inconsistency arises between two different pieces of non-local data, i.e., when a piece of data retrieved by a peer through a P2P mapping contradicts another piece of data retrieved through the P2P mappings, then no preference is made between these two pieces of information, in the sense that in this case there are two $K45_n^A$ -models for the P2PDIS, each of which represents the situation in which one of the two pieces of data is discarded.

Finally, the semantics $ANS_{K45_n^A}(q, i, \mathcal{P}, \mathcal{D})$ of a query q posed to a peer P_i of a P2PDIS \mathcal{P} wrt an extension \mathcal{D} is defined as for $K45_n$, except that now we have to take into account the $K45_n^A$ formalization of the \mathcal{P} .

Example 8. Consider again the P2PDIS of Example 1. The $K45_n^A$ formalization can be now obtained from the $K45_n$ one by substituting the P2P mapping assertions in $\mathcal{T}_K(P_1)$ and $\mathcal{T}_K(P_3)$ of Example 2 with the following assertions:

$$\begin{aligned} \forall x, z. \neg \mathbf{A}_2 \perp_2 \wedge \mathbf{K}_2(\exists y. \text{Citizen}_2(x, y, z)) \wedge \neg \mathbf{A}_1(\neg \exists y. \text{Person}_1(x, y, z)) \supset \\ \mathbf{K}_1(\exists y. \text{Person}_1(x, y, z)) \\ \forall x, y, z. \neg \mathbf{A}_1 \perp_1 \wedge \mathbf{K}_1(\text{Person}_1(x, y, z)) \wedge \neg \mathbf{A}_3(\neg \text{Person}_3(x, y, z)) \supset \\ \mathbf{K}_3(\text{Person}_3(x, y, z)) \\ \forall x, y, z. \neg \mathbf{A}_4 \perp_4 \wedge \mathbf{K}_4(\text{Citizen}_4(x, y, z)) \wedge \neg \mathbf{A}_3(\neg (\text{Person}_3(x, y, z))) \supset \\ \mathbf{K}_3(\text{Person}_3(x, y, z)). \end{aligned}$$

It is easy to see that P_3 gets from P_1 that $\text{Person}_3(\text{"Joe"}, \text{"Rome"}, \text{"Italy"})$ and from P_4 that $\text{Person}_3(\text{"Joe"}, \text{"Rome"}, \text{"Canada"})$, but since name is a key for Person_3 , taking together such two facts would give rise to an inconsistency. In fact, according to our new formalization, in each $K45_n^A$ -model of the P2PDIS, we have that either the sentence $\mathbf{K}_3(\text{Person}_3(\text{"Joe"}, \text{"Rome"}, \text{"Italy"}))$ holds or the sentence $\mathbf{K}_3(\text{Person}_3(\text{"Joe"}, \text{"Rome"}, \text{"Canada"}))$ holds, and hence P_3 does not know the citizenship of "Joe". However, P_3 still knows that "Joe" lives in "Rome". In Figure 4 we present the two possible forms that each $K45_n^A$ -model may assume. In each model (E, w) , where $E = (\mathcal{W}_n, \{R_1, R_2, R_3, R_4, R_1^a, R_2^a, R_3^a, R_4^a\}, V_n)$, we have that $R_2 = R_2^a = \emptyset$ (since P_2 is locally inconsistent), in the projection π_1 , representing both the accessibility relation R_1 and R_1^a , the worlds belonging to the completely connected

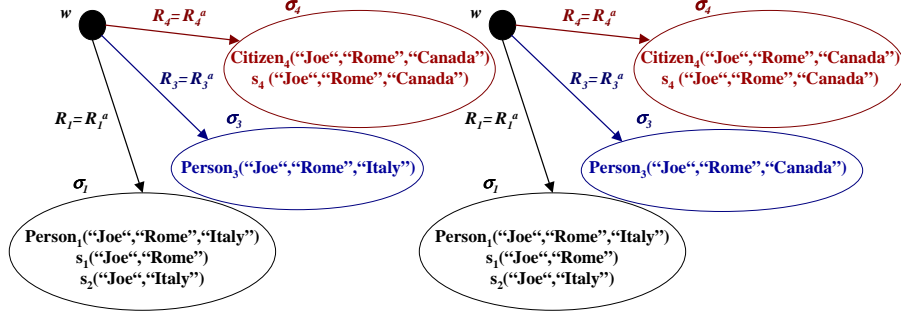


Fig. 4. Canonical Interpretations for the P2PDIS of Example 8

subgraph of π_1 are represented by the set

$$\sigma_1 = \{w \in W_n \mid V_n(w) \models \text{Person}_1(\text{"Joe"}, \text{"Rome"}, \text{"Italy"}) \wedge s_1(\text{"Joe"}, \text{"Rome"}) \wedge s_2(\text{"Joe"}, \text{"Italy"})\},$$

the worlds in the completely connected subgraph of the projection π_4 are represented by the set

$$\sigma_4 = \{w \in W_n \mid V_n(w) \models \text{Citizen}_4(\text{"Joe"}, \text{"Rome"}, \text{"Canada"}) \wedge s_4(\text{"Joe"}, \text{"Rome"}, \text{"Canada"})\},$$

whereas the worlds in the completely connected subgraph of the projection π_3 are represented by either the set

$$\sigma_3 = \{w \in W_n \mid V_n(w) \models \text{Person}_4(\text{"Joe"}, \text{"Rome"}, \text{"Italy"}),$$

for models of the first form, or

$$\sigma_3 = \{w \in W_n \mid V_n(w) \models \text{Person}_4(\text{"Joe"}, \text{"Rome"}, \text{"Canada"}),$$

for models of the second form. Notice that the interpretation associated to the initial world $w \in W_n$ is actually of no matter for establishing that (E, w) is a $K45_n^A$ -model of the $\mathcal{L}(K45_n^A)$ theory formalizing the P2PDIS together with its extension.

Given the query $q = \{x \mid \exists y. \text{Person}_3(\text{"Joe"}, x, y)\}$ posed to P_3 , we have that $\text{ANS}_{K45_n^A}(q, \mathcal{P}, \mathcal{D}) = \{\text{"Rome"}\}$, while for the query $q' = \{y \mid \exists x. \text{Person}_3(\text{"Joe"}, x, y)\}$ we have $\text{ANS}_{K45_n^A}(q, \mathcal{P}, \mathcal{D}) = \emptyset$. \square

We finally remark that due to the fact that, in the presence of inconsistency, each peer prefers its local data to the data coming from other peers, situations may arise in which apparently equivalent queries posed to different peers produce different answers. Assume for instance a simple setting \mathcal{P} with peers P_1 and P_2 . P_1 has a relation L_1 in its local schema, and G_1 in its global schema. P_2 has a relation L_2 in its local schema, and G_2 in its global schema. All relations have two attributes (Name and

City), and Name is the key in both global relations. The local mappings of P_1 and P_2 simply copy the local data to the global schemas. In addition, we have the following P2P mappings $\{n, c \mid G_1(n, c)\} \rightsquigarrow \{n, c \mid G_2(n, c)\}$ and $\{n, c \mid G_2(n, c)\} \rightsquigarrow \{n, c \mid G_1(n, c)\}$. Finally, let the extension of the local sources \mathcal{D} be composed of $D_1 = \{L_1(\text{"Joe"}, \text{"Norway"})\}$ and $D_2 = \{L_2(\text{"Joe"}, \text{"Italy"})\}$, and let q_1 be the query $q_1(x, y) = G_1(x, y)$ and $q_2(x, y) = G_2(x, y)$. Notice that G_1 and G_2 are “conceptually equivalent” (due to the form of the P2P mappings), however we have that $ANS_{K45_n^A}(q_1, 1, \mathcal{P}, \mathcal{D}) = \{\text{"Joe"}, \text{"Norway"}\}$ and $ANS_{K45_n^A}(q_2, 2, \mathcal{P}, \mathcal{D}) = \{\text{"Joe"}, \text{"Italy"}\}$, which is due to the fact that each peer prefer its local data to the data coming from the other peer. This behavior is in fact not surprising in the light of the principle of modularity underlying our semantics. Indeed, in many application scenarios, in the presence of inconsistent data, it is perfectly reasonable to get different answers (to “equivalent” queries) from different peers.

Fundamental properties of the $K45_n^A$ formalization. Next, we report some properties of the $K45_n^A$ formalization of P2PDISs that clarify from a formal point of view how such a formalization captures the notions of local inconsistency tolerance and P2P inconsistency tolerance.

We start by emphasizing that the formalization of a P2PDIS based on $K45_n^A$ is a “conservative extension” of the one based on $K45_n$, in the sense that, if no peer is locally inconsistent, and the data at the sources do not give rise to P2P inconsistencies, then the semantics of queries is the same in the two logics.

Proposition 2. *Let \mathcal{P} be a P2PDIS and let \mathcal{D} be an extension for \mathcal{P} such that each peer in \mathcal{P} is neither locally inconsistent, nor P2P inconsistent wrt \mathcal{D} . Then, for each peer $P_i \in \mathcal{P}$ and for each query q posed to P_i , $ANS_{K45_n^A}(q, i, \mathcal{P}, \mathcal{D}) = ANS_{K45_n}(q, i, \mathcal{P}, \mathcal{D})$.*

Then, we turn our attention to local inconsistency tolerance. The following proposition shows that the P2PDIS is tolerant to local inconsistency, in the sense that it isolates the peers that are locally inconsistent.

Proposition 3. *Let \mathcal{P} be a P2PDIS, let \mathcal{D} be an extension for \mathcal{P} , let $P_i \in \mathcal{P}$ be a peer locally inconsistent wrt D_i , and let $\mathcal{P}' = \mathcal{P} - \{P_i\}$. Then, for each query q posed to a peer $P_j \in \mathcal{P}$ different from P_i , we have that $ANS_{K45_n^A}(q, j, \mathcal{P}, \mathcal{D}) = ANS_{K45_n^A}(q, j, \mathcal{P}', \mathcal{D})$.*

Moreover, the following proposition shows that the new formalization enjoys the basic property for being tolerant to P2P inconsistency, namely that locally consistent peers always provide meaningful answers.

Proposition 4. *Let \mathcal{P} be a P2PDIS and let \mathcal{D} be an extension for \mathcal{P} . If $P_i \in \mathcal{P}$ is locally consistent wrt D_i , then $\mathcal{T}_A(\mathcal{P}) \cup DB(\mathcal{D}) \not\models_{K45_n^A} \mathbf{K}_i \perp_i$.*

7 Decidability and complexity of query answering

In this section we study decidability and complexity of query answering in the framework of P2PDISs defined above. We do so by focusing on a specific class of P2PDISs,

which we call GAV_{KD} -P2PDISs. Such a class is characterized by simple peer schemas (i.e., relational schemas with key dependencies) and a simple kind of local mappings (which are indeed GAV mappings [35]). Such peers are one of the simplest kinds of peers in which inconsistency may arise. As for P2P mappings, we consider them in their full generality, without posing any restriction on their form.

Specifically, for such a case we devise below an algorithm that is based directly on the multimodal epistemic semantics, making use of the notion of first-order extension (FOE) typical of nonmonotonic epistemic logics (see e.g. [18]).⁷

We start by defining formally the class of GAV_{KD} -P2PDISs.

Definition 3. A GAV_{KD} -P2PDIS is a P2PDIS such that:

- each peer schema is a relational schema with key dependencies;
- in each peer, the local mappings are global-as-view (GAV) mappings, i.e., mappings of the form

$$\{\mathbf{x} \mid \exists \mathbf{y}. \text{body}_{cq_r}(\mathbf{x}, \mathbf{y})\} \rightsquigarrow \{\mathbf{x} \mid r(\mathbf{x})\}$$

where r is a relation of the peer schema of P_i . In other words, a GAV mapping defines a relation of P_i as a view (conjunctive query cq_r) over the sources of P_i .

From now on, we restrict our attention to the class of GAV_{KD} -P2PDISs, and study query answering in such systems.

We now present an algorithm to solve the decision problem associated with query answering in GAV_{KD} -P2PDISs. We start by giving some auxiliary definitions.

Definition 4. Let m be the following P2P mapping assertion:

$$\{\mathbf{x} \mid \exists \mathbf{y}. \text{body}_{cq_j}(\mathbf{x}, \mathbf{y})\} \rightsquigarrow \{\mathbf{x} \mid \exists \mathbf{z}. \text{body}_{cq_i}(\mathbf{x}, \mathbf{z})\} \quad (2)$$

and let \mathbf{t} be a tuple of constants. Then:

- we denote by $\text{prec}(m, \mathbf{t})$ the first-order sentence $\exists \mathbf{y}. \text{body}_{cq_j}(\mathbf{t}, \mathbf{y})$;
- we denote by $\text{cons}(m, \mathbf{t})$ the first-order sentence $\exists \mathbf{z}. \text{body}_{cq_i}(\mathbf{t}, \mathbf{z})$.

Definition 5. Let \mathcal{P} be a GAV_{KD} -P2PDIS and let \mathcal{D} be an extension for \mathcal{P} . For each peer $P_i \in \mathcal{P}$, we denote by $T_i(\mathcal{P}, \mathcal{D})$ the following set of facts:

$$T_i(\mathcal{P}, \mathcal{D}) = \{r(\mathbf{t}) \mid r \text{ is a global relation in the schema of } P_i \text{ and } \mathbf{t} \in q_r^{\mathcal{D}}\}$$

where q_r is the query over the peer sources that defines the GAV local mapping for r .

Informally, $T_i(\mathcal{P}, \mathcal{D})$ denotes the extension of r that is computed by evaluating the local mapping query q_r relative to r on the extension \mathcal{D} .

From now on, for each peer $P_i \in \mathcal{P}$, we denote by $KD(P_i)$ the set of first-order sentences representing the key dependencies occurring in the schema of P_i : e.g., the KD that states that the first attribute of a relation r of arity 2 is the key of r is represented by the sentence

$$\forall x, y, z. r(x, y) \wedge r(x, z) \rightarrow y = z$$

⁷ Such an approach is to be contrasted with the one in [13], which is more indirect since it based on reducing the query answering problem into the problem of evaluating a Disjunctive Datalog program.

Definition 6. Let \mathcal{P} be a GAV_{KD} -P2PDIS and let \mathcal{D} be an extension for \mathcal{P} . For each peer $P_i \in \mathcal{P}$, let \mathcal{T}_i^u be the following set of first-order sentences:

$$\begin{aligned} \mathcal{T}_i^u = & KD(P_i) \\ & \cup T_i(\mathcal{P}, \mathcal{D}) \\ & \cup \{ \exists \mathbf{z}. \text{body}_{cqi}(\mathbf{t}, \mathbf{z}) \mid \text{there exists a P2P mapping assertion of the form (2) in } \mathcal{P} \\ & \text{and } \mathbf{t} \text{ is a tuple of constants occurring in } \mathcal{D} \} \end{aligned}$$

A first-order extension (FOE) for \mathcal{P} and \mathcal{D} is an n -tuple (T_1, \dots, T_n) where each T_i is a FOL theory such that $T_i \subseteq \mathcal{T}_i^u$.

The intuition behind a FOE is that every FOL theory T_i in a FOE represents the epistemic state of peer P_i . More specifically, we use the FOE $\mathcal{F} = (T_1, \dots, T_n)$ for \mathcal{P} and \mathcal{D} to represent a $K45_n^A$ -structure $(\sigma_1, \dots, \sigma_n, \sigma_1^a, \dots, \sigma_n^a)$ such that, for each $i \in \{1, \dots, n\}$, $\sigma_i = \sigma_i^a$ and

$$\sigma_i = \{(I, 1) \mid I \models T_i\}.$$

Moreover, as we will show in the following, in order to characterize the epistemic states of the peer P_i in \mathcal{P} for a given extension \mathcal{D} in the $K45_n^A$ -models for $\mathcal{T}_A(\mathcal{P}) \cup DB(\mathcal{D})$, it is sufficient to only consider subsets of the first-order sentences occurring in the theories \mathcal{T}_i^u . In other words, we can characterize the behaviour of the system \mathcal{P} for the extension \mathcal{D} by only looking at all the FOEs that can be built upon the set of sentences $\mathcal{T}_1^u, \dots, \mathcal{T}_n^u$.

We now formally define the correspondence between FOEs and canonical $K45_n^A$ -structures.

Definition 7. Let \mathcal{P} be a GAV_{KD} -P2PDIS, let \mathcal{D} be an extension for \mathcal{P} , and let E be a canonical $K45_n^A$ -structure. The FOE for \mathcal{P} and \mathcal{D} induced by E , denoted by \mathcal{F}_E , is the FOE (T_1, \dots, T_n) such that every T_i is defined as follows:

$$T_i = \{ \phi \mid \phi \in \mathcal{T}_i(\mathcal{P}, \mathcal{D}) \text{ and } E, w \models \mathbf{K}_i \phi \text{ for each } w \}$$

Definition 8. Let \mathcal{P} be a GAV_{KD} -P2PDIS, let \mathcal{D} be an extension for \mathcal{P} , let \mathcal{F} be a FOE for \mathcal{P} and \mathcal{D} . The $K45_n^A$ -structure associated with \mathcal{F} , denoted by $E_{\mathcal{F}}$, is the canonical $K45_n^A$ -structure $(\sigma_1, \dots, \sigma_n, \sigma_1^a, \dots, \sigma_n^a)$ in which, for each $i \in \{1, \dots, n\}$, $\sigma_i = \sigma_i^a$ and σ_i is the following set of worlds:

$$\sigma_i = \{(I, i) \mid I \models T_i\}$$

Then, we define the algorithm **verify-FOE**, which, given a FOE \mathcal{F} for \mathcal{P} and \mathcal{D} , is able to verify whether the $K45_n^A$ -structure associated with \mathcal{F} identifies a $K45_n^A$ -model for the theory $\mathcal{T}_A(\mathcal{P}) \cup DB(\mathcal{D})$.

Algorithm **verify-FOE**($\mathcal{P}, \mathcal{D}, \mathcal{F}$)

Input: P2PDIS \mathcal{P} , extension \mathcal{D} , FOE $\mathcal{F} = (T_1, \dots, T_n)$ for \mathcal{P} and \mathcal{D}

Output: *true* if for each $w \in \mathcal{W}_c$, $(E_{\mathcal{F}}, w)$ a $K45_n^A$ -model for $\mathcal{T}_A(\mathcal{P}) \cup DB(\mathcal{D})$,
false otherwise

```

begin
  for each  $i \in \{1, \dots, n\}$  do  $T'_i := T_i(\mathcal{P}, \mathcal{D});$ 
  repeat
     $\mathcal{F}' := (T'_1, \dots, T'_n);$ 
    if there exists P2P mapping assertion  $m$  (between  $P_j$  and  $P_i$ ) and tuple  $\mathbf{t}$ 
      such that  $T_j$  is satisfiable
        and  $T'_j \models \text{prec}(m, \mathbf{t})$ 
        and  $T'_i \not\models \neg \text{cons}(m, \mathbf{t})$ 
        and  $T'_i \not\models \text{cons}(m, \mathbf{t})$ 
        then  $T'_i := T'_i \cup \{\text{cons}(m, \mathbf{t})\}$ 
    until  $(T'_1, \dots, T'_n) = \mathcal{F}';$ 
    if  $\mathcal{F} = (T'_1, \dots, T'_n)$  then return true else return false
  end

```

Then, we define the algorithm `not-answer`, which is able to nondeterministically verify whether a tuple \mathbf{t} is not in the answers to a query q posed to a peer of \mathcal{P} for a given extension \mathcal{D} .

```

Algorithm not-answer( $\mathcal{P}, \mathcal{D}, i, q, \mathbf{t}$ )
Input: P2PDIS  $\mathcal{P}$ , extension  $\mathcal{D}$ , query  $q$  to peer  $P_i \in \mathcal{P}$ , tuple  $\mathbf{t}$ 
Output: true if  $\mathbf{t} \notin \text{ANS}_{K45_n^A}(q, i, \mathcal{P}, \mathcal{D})$ , false otherwise
begin
  if there exists FOE  $\mathcal{F} = (T_1, \dots, T_n)$  for  $\mathcal{P}$  and  $\mathcal{D}$ 
    such that verify-FOE( $\mathcal{P}, \mathcal{D}, \mathcal{F}$ ) returns true and  $T_i \not\models q(\mathbf{t})$ 
    then return true else return false
end

```

We now prove termination and give a computational characterization of the algorithm `not-answer`. To this aim, we start by showing two auxiliary lemmas.

Lemma 1. *Let \mathcal{P} be a GAV_{KD}-P2PDIS, let \mathcal{D} be an extension for \mathcal{P} , let $T_i \subseteq T_i^u$, and let q be a Boolean conjunctive query, i.e., a sentence of the form $\exists \mathbf{y}. \text{body}(\mathbf{t}, \mathbf{y})$. Deciding whether $T_i \models q$ can be done in time polynomial with respect to the size of \mathcal{D} .*

Proof. It is possible to check whether $T_i \models q$ by building the following database instance (set of facts) \mathcal{B} from T_i :

1. For each fact of the form $r(\mathbf{t}) \in T_i(\mathcal{P}, \mathcal{D})$, we add the fact $r(\mathbf{t})$ to \mathcal{B} .
2. For each sentence $\exists \mathbf{z}. \text{body}(\mathbf{t}, \mathbf{z})$ in T_i , with $\text{body}(\mathbf{t}, \mathbf{z}) = a_1(\mathbf{t}, \mathbf{z}) \wedge \dots \wedge a_k(\mathbf{t}, \mathbf{z})$, we add the facts $a_1(\mathbf{t}, \mathbf{s}), \dots, a_k(\mathbf{t}, \mathbf{s})$ to \mathcal{B} , where \mathbf{s} is a tuple of *soft constants*, i.e., constant symbols from an alphabet Σ disjoint from the alphabet of constants Γ . For every sentence, we use different soft constants to represent the existential variables in the sentence.
3. Then, we apply the equalities implied by the key dependencies $KD(P_i)$ to the database instance \mathcal{B} built so far. For instance, if $\text{key}(r) = 1$, for each pair of facts $r(t_1, t_2, t_3), r(t'_1, t'_2, t'_3)$ such that $t_1 = t'_1$, we derive the equalities $t_2 = t'_2$ and $t_3 = t'_3$. The derived equalities may be of two forms:

- (a) at least one of the two terms, say t_1 , is a soft constants. In this case, we apply the substitution $t_1 \leftarrow t_2$ to the whole database instance;
 - (b) both terms are syntactically different “hard” (i.e., non-soft) constants. In this case, we conclude that \mathcal{B} is inconsistent w.r.t. the key dependencies (since the key dependencies imply that two different objects are the same).
4. We iteratively apply the above step until either we conclude that \mathcal{B} is inconsistent w.r.t. the key dependencies or there are no more new derived equalities.

It is immediate to verify that the above construction of the database \mathcal{B} can be done in time polynomial in the size of \mathcal{D} . Moreover, the database \mathcal{B} thus constructed allows us to decide whether $T_i \models q$. In fact, it is easy to see that:

- T_i is unsatisfiable iff \mathcal{B} is inconsistent w.r.t. the key dependencies;
- if \mathcal{B} is consistent, then $T_i \models q$ iff the query q is true when evaluated on the database \mathcal{B} .

Consequently, $T_i \models q$ iff either \mathcal{B} is inconsistent or the query q is true when evaluated on \mathcal{B} . \square

Lemma 2. *Let \mathcal{P} be a GAV_{KD} -P2PDIS, let \mathcal{D} be an extension for \mathcal{P} , let $T_i \subseteq T_i^u$, and let q be a Boolean conjunctive query, i.e., a sentence of the form $\exists \mathbf{y}. \text{body}(\mathbf{y})$. Deciding whether $T_i \models \neg q$ can be done in time polynomial with respect to the size of \mathcal{D} .*

Proof. The proof is very similar to the proof of the above lemma. Indeed, we can decide whether $T_i \models \neg q$ by building (in time polynomial in the size of \mathcal{D}) a database instance \mathcal{B} in a way analogous to the above proof. The only difference lies in the fact that, in step 2 of the construction of \mathcal{B} , we have to also add to the database \mathcal{B} a set of facts (with soft constants) representing the Boolean conjunctive query q . Then, it is easy to verify that the database \mathcal{B} thus constructed is inconsistent w.r.t. the key dependencies iff $T_i \models \neg q(\mathbf{t})$. \square

We are now ready to prove that the algorithm $\text{verify-FOE}(\mathcal{P}, \mathcal{D}, \mathcal{F})$ terminates and can be executed in time polynomial in the size of \mathcal{D} .

Lemma 3. *Let \mathcal{P} be a GAV_{KD} -P2PDIS, \mathcal{D} an extension for \mathcal{P} , \mathcal{F} a FOE for \mathcal{P} and \mathcal{D} . The algorithm $\text{verify-FOE}(\mathcal{P}, \mathcal{D}, \mathcal{F})$ terminates and runs in polynomial time with respect to the size of \mathcal{D} .*

Proof. The proof follows from the following facts:

- every set of sentences T_i^u has size polynomial in the size of \mathcal{D} , consequently every FOE for \mathcal{P} and \mathcal{D} has size polynomial in the size of \mathcal{D} ;
- for each i , the set of facts $T_i(\mathcal{P}, \mathcal{D})$ can be computed in time polynomial with respect to the size of \mathcal{D} , since such a set can be computed by evaluating the local GAV mapping queries over the extension \mathcal{D} , which in turn corresponds to the standard evaluation of a set of conjunctive queries over a relational database;

- the number of executions of the **repeat–until** loop is bound to the number of instantiations of the P2P mapping assertions on the constants occurring in \mathcal{D} , since every iteration can be executed at most once for each instantiation of a P2P mapping assertion. Consequently, such a number is polynomial in the size of \mathcal{D} ;
 - in every iteration of the **repeat–until** loop:
 1. as explained above, the number of instantiations of the P2P mapping assertions to which the condition of the **if** statement must be checked is polynomial in the size of \mathcal{D} ;
 2. by Lemma 1, satisfiability of T_j can be verified in time polynomial in the size of \mathcal{D} ;
 3. by definition of $prec(m, \mathbf{t})$ and by Lemma 1, $T'_j \models prec(m, \mathbf{t})$ can be verified in time polynomial in the size of \mathcal{D} ;
 4. by definition of $cons(m, \mathbf{t})$ and by Lemma 2, $T_i \models \neg cons(m, \mathbf{t})$ can be verified in time polynomial in the size of \mathcal{D} ;
 5. by definition of $cons(m, \mathbf{t})$ and by Lemma 1, $T'_i \models cons(m, \mathbf{t})$ can be verified in time polynomial in the size of \mathcal{D} .
- Consequently, every iteration of the **repeat–until** loop can be executed in time polynomial in the size of \mathcal{D} . □

Based on the above property, we now show termination and complexity of the algorithm **not-answer**.

Theorem 1. *Let \mathcal{P} be a GAV_{KD} -P2PDIS, \mathcal{D} an extension for \mathcal{P} , $P_i \in \mathcal{P}$, $q \in \mathcal{L}$ a query of arity n over P_i , and \mathbf{t} a n -tuple of constants in Γ . The algorithm **not-answer**($\mathcal{P}, \mathcal{D}, i, q, \mathbf{t}$) terminates and runs in nondeterministic polynomial time with respect to the size of \mathcal{D} (i.e., in data complexity).*

Proof. The proof follows immediately from Lemma 3 and from the fact that, by Lemma 1, $T_i \not\models q(\mathbf{t})$ can be checked in polynomial time with respect to the size of \mathcal{D} . □

Then, we turn our attention to the correctness of the algorithm **not-answer** with respect to the $K45_n^A$ formalization of P2PDISs. We start with some auxiliary lemmas.

Lemma 4. *Let $\mathcal{F} = (T_1, \dots, T_n)$ be a FOE for \mathcal{P} and \mathcal{D} , let $E_{\mathcal{F}}$ be the $K45_n^A$ -structure associated with \mathcal{F} , and let q denote a Boolean conjunctive query, i.e., a sentence of the form $\exists \mathbf{y}. body(\mathbf{t}, \mathbf{y})$. Then, for each $w \in \mathcal{W}_0$, $E_{\mathcal{F}}, w \models q$ iff $T_j \models q$.*

Proof. The proof is immediate from the definition of $E_{\mathcal{F}}$. □

Lemma 5. *Let $\mathcal{F} = (T_1, \dots, T_n)$ be a FOE for \mathcal{P} and \mathcal{D} , let $E_{\mathcal{F}}$ be a canonical $K45_n^A$ -structure associated with \mathcal{F} , and let q denote a Boolean conjunctive query. Then, for each $w \in \mathcal{W}_0$, $E_{\mathcal{F}}, w \models \neg q$ iff $T_i \not\models \neg q$.*

Proof. Again, the proof follows immediately from the definition of $E_{\mathcal{F}}$. □

Lemma 6. Let \mathcal{P} be a GAV_{KD} -P2PDIS, let \mathcal{D} be an extension for \mathcal{P} , and let (E, w) be a $K45_n^A$ -interpretation such that $E, w \models \mathcal{T}_A(\mathcal{P}) \cup DB(\mathcal{D})$. Then, for each $i \in \{1, \dots, n\}$ and for each sentence $\phi \in T_i(\mathcal{P}, \mathcal{D})$, $E, w \models \mathbf{K}_i\phi$.

Proof. The proof follows from the fact that the sentences representing the local mappings of peer P_i in $\mathcal{T}_A(\mathcal{P})$ together with $DB(\mathcal{D})$ necessarily imply that the sentence $\mathbf{K}_i\phi$ is satisfied, for every fact ϕ in $T_i(\mathcal{P}, \mathcal{D})$. \square

Then, we define the set of sentences $gr_{\mathcal{D}}(\mathcal{T}_A(\mathcal{P}))$ that constitutes a partial grounding, over the constants occurring in \mathcal{D} , of the sentences in $\mathcal{T}_A(\mathcal{P})$ representing P2P mappings.

Definition 9. We define $gr_{\mathcal{D}}(\mathcal{T}_A(\mathcal{P}))$ as the $\mathcal{L}(K45_n^A)$ theory obtained from $\mathcal{T}_A(\mathcal{P})$ by substituting each sentence (encoding a P2P mapping assertion (2) in \mathcal{P}) of the form

$$\forall \mathbf{x}. \neg \mathbf{A}_j \perp_j \wedge \mathbf{K}_j(\exists \mathbf{y}. \text{body}_{cq_j}(\mathbf{x}, \mathbf{y})) \wedge \neg \mathbf{A}_i(\neg \exists \mathbf{z}. \text{body}_{cq_i}(\mathbf{x}, \mathbf{z})) \supset \mathbf{K}_i(\exists \mathbf{z}. \text{body}_{cq_i}(\mathbf{x}, \mathbf{z}))$$

with the set of sentences

$$\neg \mathbf{A}_j \perp_j \wedge \mathbf{K}_j(\exists \mathbf{y}. \text{body}_{cq_j}(\mathbf{t}, \mathbf{y})) \wedge \neg \mathbf{A}_i(\neg \exists \mathbf{z}. \text{body}_{cq_i}(\mathbf{t}, \mathbf{z})) \supset \mathbf{K}_i(\exists \mathbf{z}. \text{body}_{cq_i}(\mathbf{t}, \mathbf{z}))$$

for every tuple \mathbf{t} of constants occurring in \mathcal{D} .

We now prove that the above partial grounding $gr_{\mathcal{D}}(\mathcal{T}_A(\mathcal{P}))$ constitutes a correct representation of $\mathcal{T}_A(\mathcal{P})$.

Lemma 7. Let \mathcal{P} be a GAV_{KD} -P2PDIS, let \mathcal{D} be an extension for \mathcal{P} . A $K45_n^A$ interpretation (E, w) is a $K45_n^A$ -model for $\mathcal{T}_A(\mathcal{P}) \cup DB(\mathcal{D})$ iff (E, w) is a $K45_n^A$ -model for $gr_{\mathcal{D}}(\mathcal{T}_A(\mathcal{P})) \cup DB(\mathcal{D})$.

Proof. First, by definition of the semantics of $K45_n^A$, the theory $\mathcal{T}_A(\mathcal{P}) \cup DB(\mathcal{D})$ is equivalent to $gr_{\Gamma}(\mathcal{T}_A(\mathcal{P})) \cup DB(\mathcal{D})$, where $gr_{\Gamma}(\mathcal{T}_A(\mathcal{P}))$ is the theory in which each sentence encoding a P2P mapping assertion (2) is replaced by the set of sentences

$$\neg \mathbf{A}_j \perp_j \wedge \mathbf{K}_j(\exists \mathbf{y}. \text{body}_{cq_j}(\mathbf{t}', \mathbf{y})) \wedge \neg \mathbf{A}_i(\neg \exists \mathbf{z}. \text{body}_{cq_i}(\mathbf{t}', \mathbf{z})) \supset \mathbf{K}_i(\exists \mathbf{z}. \text{body}_{cq_i}(\mathbf{t}', \mathbf{z})) \quad (3)$$

for every tuple \mathbf{t}' of constants occurring in Γ . Thus, to prove the thesis we show that adding to $gr_{\mathcal{D}}(\mathcal{T}_A(\mathcal{P})) \cup DB(\mathcal{D})$ an instance of the above sentence (3) for \mathbf{t}' containing at least one constant not occurring in \mathcal{D} does not change the set of $K45_n^A$ -models for the theory. Let \mathbf{t}' be such a tuple and let $m(\mathbf{t}')$ denote the sentence of the above form (3). Then, let (E, w) be any $K45_n^A$ -model for $gr_{\mathcal{D}}(\mathcal{T}_A(\mathcal{P})) \cup DB(\mathcal{D})$: it is immediate to verify that the sentence $\mathbf{K}_j(\exists \mathbf{y}. \text{body}_{cq_j}(\mathbf{t}', \mathbf{y}))$ is not satisfied in (E, w) . This implies $M, w \models m(\mathbf{t}')$, which in turn implies that (E, w) is a $K45_n^A$ -model for $gr_{\mathcal{D}}(\mathcal{T}_A(\mathcal{P})) \cup DB(\mathcal{D}) \cup \{m(\mathbf{t}')\}$. Since the above holds for every $m(\mathbf{t}')$, it follows that (E, w) is a $K45_n^A$ -model for $gr_{\Gamma}(\mathcal{T}_A(\mathcal{P})) \cup DB(\mathcal{D})$, and therefore (E, w) is a $K45_n^A$ -model for $\mathcal{T}_A(\mathcal{P}) \cup DB(\mathcal{D})$.

Then, let (E, w) be a $K45_n^A$ -model for $\mathcal{T}_A(\mathcal{P}) \cup DB(\mathcal{D})$ and suppose there exists a tuple \mathbf{t}' with at least one constant not occurring in \mathcal{D} and such that $E, w \models$

$\mathbf{K}_j(\exists \mathbf{y}. \text{body}_{cq_j}(\mathbf{t}', \mathbf{y}))$. Then, there exists at least a first-order interpretation I such that $I \not\models (\exists \mathbf{y}. \text{body}_{cq_j}(\mathbf{t}', \mathbf{y}))$ and, for each tuple \mathbf{t} of constants from \mathcal{D} and for each P2P mapping assertion (2), $I \models (\exists \mathbf{y}. \text{body}_{cq_j}(\mathbf{t}, \mathbf{y}))$ iff $E, w \models \mathbf{K}_j(\exists \mathbf{y}. \text{body}_{cq_j}(\mathbf{t}, \mathbf{y}))$: therefore, $(I, 1) \notin \sigma_j$, where σ_j is the j -th cluster in E . Now it is immediate to verify that the $K45_n^A$ -structure E' obtained from E by adding the world $(I, 1)$ to σ_j , is such that $E', w \models gr_{\Gamma}(\mathcal{T}_A(\mathcal{P})) \cup DB(\mathcal{D})$, and therefore $E', w \models gr_{\Gamma}(\mathcal{T}_A(\mathcal{P})) \cup DB(\mathcal{D})$, thus by Definition 2 (E, w) is not a $K45_n^A$ -model for $\mathcal{T}_A(\mathcal{P}) \cup DB(\mathcal{D})$. Contradiction. Consequently, for each tuple \mathbf{t}' with at least one constant not occurring in \mathcal{D} , and for each P2P mapping assertion (2), $E, w \models \neg \mathbf{K}_j(\exists \mathbf{y}. \text{body}_{cq_j}(\mathbf{t}', \mathbf{y}))$. This in turn implies that (E, w) is a $K45_n^A$ -model for $gr_{\mathcal{D}}(\mathcal{T}_A(\mathcal{P})) \cup DB(\mathcal{D})$. \square

Then, we prove an important property that states that the $K45_n^A$ -structure associated with the FOE \mathcal{F}_E induced by a canonical $K45_n^A$ -structure E coincides with the $K45_n^A$ -structure E .

Lemma 8. *Let \mathcal{P} be a GAV_{KD} -P2PDIS, let \mathcal{D} be an extension for \mathcal{P} , let (E, w) be a $K45_n^A$ -model for $\mathcal{T}_A(\mathcal{P}) \cup DB(\mathcal{D})$, let \mathcal{F}_E be the FOE for \mathcal{P} and \mathcal{D} induced by E , and let $E_{\mathcal{F}_E}$ be the canonical $K45_n^A$ -structure associated with the FOE \mathcal{F}_E . Then, $E = E_{\mathcal{F}_E}$.*

Proof. Let $E = (\sigma_1, \dots, \sigma_n, \sigma_1^a, \dots, \sigma_n^a)$ with $\sigma_i = \sigma_i^a$ for each i , and let $E_{\mathcal{F}_E} = (\sigma'_1, \dots, \sigma'_n, \sigma_1^a, \dots, \sigma_n^a)$ with $\sigma'_i = \sigma_i^a$ for each i . First, since by Definition 8 each σ'_i is the set of worlds $\{(I, 1) \mid I \models T_i\}$, and since, for each $\phi \in T_i$ and for each $w \in \mathcal{W}_0$, $E, w \models \mathbf{K}_i \phi$, it follows that $\sigma_i \subseteq \sigma'_i$ for each $i \in \{1, \dots, n\}$. Now, suppose $E \neq E_{\mathcal{F}_E}$: then, there exists i such that $\sigma_i \subset \sigma'_i$, hence there exists a world w_1 such that $w_1 \in \sigma'_i - \sigma_i$. Let w be any world in \mathcal{W}_0 and let E'' be the canonical $K45_n^A$ -structure obtained from E by adding to the world w_1 to the set σ_i . Now, from Definition 7, and from Definition 8, it follows that, for each world w and for each formula $\phi \in T_i^u$, $E, w \models \mathbf{K}_i \phi$ iff $E_{\mathcal{F}_E}, w \models \mathbf{K}_i \phi$. Consequently, $E'', w \models gr_{\mathcal{D}}(\mathcal{T}_A(\mathcal{P})) \cup DB(\mathcal{D})$, therefore by Lemma 7 $E'', w \models \mathcal{T}_A(\mathcal{P}) \cup DB(\mathcal{D})$, hence by Definition 2 it follows that (E, w) is not a $K45_n^A$ -model for $\mathcal{T}_A(\mathcal{P}) \cup DB(\mathcal{D})$, thus contradicting the hypothesis. Consequently, $E = E_{\mathcal{F}_E}$. \square

We are now ready to prove correctness of the algorithm `verify-FOE`.

Lemma 9. *Let \mathcal{P} be a GAV_{KD} -P2PDIS, let \mathcal{D} be an extension for \mathcal{P} , let \mathcal{F} be a FOE for \mathcal{P} and \mathcal{D} , and let $E_{\mathcal{F}}$ be the $K45_n^A$ -structure associated with \mathcal{F} . Then, `verify-FOE`($\mathcal{P}, \mathcal{D}, \mathcal{F}$) returns true iff, for each $w \in \mathcal{W}_0$, $(E_{\mathcal{F}}, w)$ is a $K45_n^A$ -model for $\mathcal{T}_A(\mathcal{P}) \cup DB(\mathcal{D})$.*

Proof. By Lemma 7, we have to prove that `verify-FOE`($\mathcal{P}, \mathcal{D}, \mathcal{F}$) returns true iff $(E_{\mathcal{F}}, w)$ is a $K45_n^A$ -model for $gr_{\mathcal{D}}(\mathcal{T}_A(\mathcal{P})) \cup DB(\mathcal{D})$.

(\Leftarrow): Suppose `verify-FOE`($\mathcal{P}, \mathcal{D}, \mathcal{F}$) returns false. Let $\mathcal{F} = (T_1, \dots, T_n)$, and let (T'_1, \dots, T'_n) be the FOE computed by the algorithm after the execution of the **repeat-`until`** loop. Then, there exists i such that $T_i \neq T'_i$. Let w be any world in \mathcal{W}_0 . There are two possible cases:

- $T_i \supseteq T'_i$ for each i , and there exists i such that $T_i \supset T'_i$. Let E' be the $K45_n^A$ -structure $E' = (\sigma'_1, \dots, \sigma'_n, \sigma_1, \dots, \sigma_n)$. Then, $E', w \models gr_D(\mathcal{T}_A(\mathcal{P})) \cup DB(\mathcal{D})$, and since $E_{\mathcal{F}}$ is \mathbf{K} -contained in E' , from Definition 2 it follows that $(E_{\mathcal{F}}, w)$ is not a $K45_n^A$ -model for $gr_D(\mathcal{T}_A(\mathcal{P})) \cup DB(\mathcal{D})$.
- there exists i such that there exists a sentence ϕ such that $\phi \in T'_i - T_i$. But this immediately implies that $E_{\mathcal{F}}, w \not\models gr_D(\mathcal{T}_A(\mathcal{P})) \cup DB(\mathcal{D})$, since for each sentence $cons(m, \mathbf{t})$ which is added by the algorithm to T'_i , the sentence $\mathbf{K}_i cons(m, \mathbf{t})$ must necessarily be satisfied in order to satisfy the sentences in $gr_D(\mathcal{T}_A(\mathcal{P})) \cup DB(\mathcal{D})$. Consequently, $(E_{\mathcal{F}}, w)$ is not a $K45_n^A$ -model for $gr_D(\mathcal{T}_A(\mathcal{P})) \cup DB(\mathcal{D})$.

(\Rightarrow): Suppose $\text{verify-FOE}(\mathcal{P}, \mathcal{D}, \mathcal{F})$ returns *true*. Then, it is immediate to verify that, for each $w \in \mathcal{W}_0$, $E_{\mathcal{F}}, w \models gr_D(\mathcal{T}_A(\mathcal{P})) \cup DB(\mathcal{D})$. Now suppose $(E_{\mathcal{F}}, w)$ is not a $K45_n^A$ -model for $gr_D(\mathcal{T}_A(\mathcal{P})) \cup DB(\mathcal{D})$. Then, by Definition 2 there exists a $K45_n^A$ -structure E' such that $E_{\mathcal{F}}$ is \mathbf{K} -contained in E' and $E', w \models gr_D(\mathcal{T}_A(\mathcal{P})) \cup DB(\mathcal{D})$ for each world w . Now, from Definition 7, and from Definition 8, it follows that there exists i and a sentence $\phi \in T_i$ such that $E', w \not\models \mathbf{K}_i \phi$ and $E_{\mathcal{F}}, w \models \mathbf{K}_i \phi$. There are two possible cases:

1. $\phi \in T_i(\mathcal{P}, \mathcal{D})$. In this case, observe that the sentences representing the local mappings of P_i in $\mathcal{T}_A(\mathcal{P})$ together with $DB(\mathcal{D})$ necessarily imply $\mathbf{K}_i \phi'$ for every fact ϕ' in $T_i(\mathcal{P}, \mathcal{D})$. Therefore, $E', w \not\models \mathcal{T}_A(\mathcal{P}) \cup DB(\mathcal{D})$, thus contradicting the hypothesis;
2. ϕ is of the form $\exists \mathbf{z}. \text{body}_{cq_i}(\mathbf{t}, \mathbf{z})$ such that there exists a mapping assertion of the form (2) in \mathcal{P} . Now, since by hypothesis $\text{verify-FOE}(\mathcal{P}, \mathcal{D}, \mathcal{F})$ returns *true*, it follows that \mathcal{F} can be reconstructed starting from $T_i(\mathcal{P}, \mathcal{D})$ and iteratively applying the P2P mapping assertions as rules that are necessarily “fired” by the knowledge of peer P_i , which is expressed by the sentences in T'_i incrementally collected so far by the algorithm. Therefore, by induction on the structure of \mathcal{F} (the structure is derived by the construction of \mathcal{F} done by $\text{verify-FOE}(\mathcal{P}, \mathcal{D}, \mathcal{F})$), it can be proved that the hypothesis $E_{\mathcal{F}}, w \not\models \mathbf{K}_i \phi$ implies that there exists a fact ϕ' in $T_i(\mathcal{P}, \mathcal{D})$ such that $E_{\mathcal{F}}, w \not\models \mathbf{K}_i \phi'$, which, as shown in the previous point, implies that $E', w \not\models \mathcal{T}_A(\mathcal{P}) \cup DB(\mathcal{D})$, thus contradicting the hypothesis.

Consequently, the above canonical $K45_n^A$ -structure E' does not exist, which implies that $(E_{\mathcal{F}}, w)$ is a $K45_n^A$ -model for $\mathcal{T}_A(\mathcal{P}) \cup DB(\mathcal{D})$. \square

Finally, we prove correctness of the algorithm **not-answer**.

Theorem 2. *Let \mathcal{P} be a GAV_{KD} -P2PDIS, \mathcal{D} an extension for \mathcal{P} , $P_i \in \mathcal{P}$, $q \in \mathcal{L}$ a query of arity n over P_i , and \mathbf{t} an n -tuple of constants in Γ . Then, $\mathbf{t} \in \text{ANS}_{K45_n^A}(q, i, \mathcal{P}, \mathcal{D})$ iff $\text{not-answer}(\mathcal{P}, \mathcal{D}, i, q, \mathbf{t})$ returns *false*.*

Proof. (\Rightarrow): Let $E_{\mathcal{F}} = (\mathcal{W}_n, \{R_1, \dots, R_n, R_1^a, \dots, R_n^a\}, V_n)$. Suppose $\text{not-answer}(\mathcal{P}, \mathcal{D}, i, q, \mathbf{t})$ returns *false*. Then, there exists a FOE $\mathcal{F} = (T_1, \dots, T_n)$ such that $\text{verify-FOE}(\mathcal{P}, \mathcal{D}, \mathcal{F})$ returns *true* and $T_i \not\models q(\mathbf{t})$. Let $E_{\mathcal{F}}$ be the $K45_n^A$ -structure associated with \mathcal{F} . From Lemma 9, it follows that, for each world w , $(E_{\mathcal{F}}, w)$ is a $K45_n^A$ -model for $\mathcal{T}_A(\mathcal{P}) \cup DB(\mathcal{D})$. Finally, from Lemma 4, and since $T_i \not\models q(\mathbf{t})$, it follows that $\mathbf{t} \notin \text{ANS}_{K45_n^A}(q, i, \mathcal{P}, \mathcal{D})$.

(\Leftarrow): Suppose $\mathbf{t} \in \text{ANS}_{K45_n^A}(q, i, \mathcal{P}, \mathcal{D})$. Then, there exists a $K45_n^A$ -interpretation (E, w) such that (E, w) is a $K45_n^A$ -model for $\mathcal{T}_A(\mathcal{P}) \cup \text{DB}(\mathcal{D})$ and $M, w \not\models q(\mathbf{t})$. Let \mathcal{F}_E be the FOE for \mathcal{P} and \mathcal{D} induced by E . By Lemma 8, the canonical $K45_n^A$ -structure $E_{\mathcal{F}_E}$ associated with \mathcal{F}_E is equal to E , consequently, by Lemma 9, $\text{verify-FOE}(\mathcal{P}, \mathcal{D}, \mathcal{F}_E)$ returns *true*. Moreover, by Definition 8, it follows that $E, w \models \mathbf{K}_i q(\mathbf{t})$ iff $T_i \models q(\mathbf{t})$. Consequently, $\text{not-answer}(\mathcal{P}, \mathcal{D}, i, q, \mathbf{t})$ returns *true*. \square

Based on Theorem 1 and Theorem 2, we are able to characterize the computational complexity of query answering in GAV_{KD} -P2PDISs.

Theorem 3. *Let \mathcal{P} be a GAV_{KD} -P2PDIS, \mathcal{D} an extension for \mathcal{P} , $P_i \in \mathcal{P}$, $q \in \mathcal{L}$ a query of arity n over P_i , and \mathbf{t} a n -tuple of constants in Γ . The problem of establishing whether $\mathbf{t} \in \text{ANS}_{K45_n^A}(q, i, \mathcal{P}, \mathcal{D})$ is coNP-complete with respect to the size of \mathcal{D} (i.e., in data complexity).*

Proof. The hardness part can be proved by a reduction of the three-colorability problem to our problem. The proof is obtained by adapting in a straightforward way the proof showed in [11] for establishing coNP-hardness of query answering in the setting of a single inconsistent database with key dependencies.

Membership in coNP is an immediate consequence of Theorem 1 and Theorem 2. \square

In fact, the algorithm and the results above can be extended to deal with P2PDISs whose peers are more general than in GAV_{KD} -P2PDISs. For example, we may allow for both generalized equality-generating dependencies as constraints in the peer schemas (instead of key dependencies) and GLAV local mappings (instead of GAV mappings). In this case, it can be shown that both Lemma 1 and Lemma 2 still hold, and hence also Theorem 1 and Theorem 2.

8 Related work

The P2P paradigm was made popular by systems like Napster or Gnutella, that were designed to handle semantic-free, large-granularity requests for objects by identifier (essentially, sharing of video or music files) [27]. Recently, a novel line of research, P2P data integration, has focused on the problem of extending the P2P paradigm to the richer setting considered also in this paper: each peer is seen as an autonomous information system characterized by a schema that represents the domain of interest from the peer perspective; the peer is equipped with mappings providing the semantic relationship to other peers [40], and thus provides and exchanges part of the overall information available from a distributed environment [29, 6, 14, 23, 44, 15].

A first proposal in this direction, outlining the characteristic features of peer data management systems, has been the Piazza system [27, 29], in which data stored locally at each peer are described in terms of materialized views, and additionally peer mappings, interpreted under standard first-order semantics, are used to retrieve data from other peers. Due to the adoption of first-order semantics, query answering is decidable only in the case of absence of cycles in peer mappings, or when such cycles are used

only for data replication. In [30], a version of the Piazza system is presented, in which data is modeled in XML, and peers export their schemas in XML Schema. An algorithm for query reformulation in that setting is given. Several techniques for optimizing reformulations, based on pruning and minimizing navigation paths, efficient search strategies, and pre-computing semantic paths via mapping composition are presented in [44].

In the rest of the section we concentrate on work related to the management of inconsistency that is relevant to our setting. As mentioned, the problem of dealing with inconsistency has been studied extensively in Artificial Intelligence in the area of *belief revision and update* [3, 25], which addresses the issue of updating existing information with new one, with the aim of maintaining consistency by performing minimal changes (under different assumptions for minimality). In general, these studies assume that the underlying theory is an arbitrary first-order or propositional theory, and that revision or updates are done through arbitrary formulas. In the context of databases, the theory takes the form of a database schema, and the revision process focuses on data [22]. Thus, research in this setting has concentrated on algorithmic and complexity results specialized for this case. The general goal is to provide informative answers even when a database does not satisfy its integrity constraints (see, for example, [5, 11, 26, 45]). Most of these papers rely on the notion of *repair* as introduced in [5]: a repair of a database is a new database that satisfies the constraints in the schema, and minimally differs from the original one. The inference task of *consistent query answering* corresponds to determining whether a given tuple is in the answer to a query in all databases that are minimal repairs of the given inconsistent database.

The above results are not specifically tailored to the case of different consistent sources that are mutually inconsistent, which is the case of interest in data integration. More recently, some papers (see, e.g., [7, 12, 10]) have tackled data inconsistency in a data integration setting, where sources are required to be mutually consistent with respect to constraints specified in a global schema. In this setting, the basic idea is to consider repairs as applied to data retrieved from the sources, again under some minimality criteria, and the relevant task is again that of consistent query answering, performed according to such repairs.

Only few, recent papers address the problem of dealing with inconsistencies in P2P data integration. The approach in [8] makes use of trust relationships between pairs of peers that, together with P2P mappings, determine how to consider data exchanged between peers. When data retrieved from the local sources and from other peers contradict the integrity constraints of a peer P , first P tries to repair its own data according to what the dependencies to more trusted peers prescribe. Then, keeping those trusted dependencies satisfied, P tries to repair its own data or the data coming from those peers whom it trusts equally to itself. This is formalized through the notion of *solution* for a peer P , i.e., an instance for the peer database schema that stays as close as possible to the available data in the system, and that is obtained through a two-step repair process, respecting both the mappings and the trust relationships. As for repairs, solutions are not actually computed and data coming from other peers is not actually changed. Instead, the notion of solution is used to define *peer-consistent answers*, as those tuples in the answer to the query for every solution for the peer. It is worth noting that in [8]

this notion is relative to a certain peer, since the notion of solution, on which it is based, depends on the considered peer.

A further proposal for dealing with inconsistencies in a P2P setting is the one adopted in the SOMEWHERE system [2, 16], in which each local peer theory is a set of propositional clauses, and P2P mappings are propositional clauses involving variables of distinct peers that state semantic correspondences between different vocabularies. The semantics of the system is straightforward, since the global theory of the P2P system is simply the set of all propositional clauses constituting the local peer theories and the P2P mappings. One of the challenges has been in devising a totally decentralized algorithm for computing (propositional) consequences, without any peer having access to the global theory. Such an algorithm has been implemented in the SOMEWHERE platform, and its scalability up to a thousand peers has been evaluated in [1]. In SOMEWHERE, the problem of local inconsistencies is not addressed and local peer theories are assumed to be consistent. Instead, inconsistencies due to mappings are dealt with through the notion of a *nogood*, which is a set of mappings that, when added to the local peer theories, makes them inconsistent. A distributed algorithm is proposed to compute and store at the peers all the minimal nogoods. These are then used to compute *well-founded* consequences, i.e., consequences of a consistent subset of the global peer theory. Hence, the approach is similar to ours, in the sense that it does not resolve inconsistencies by repairing data, but computes answers to queries according to the semantics ignoring inconsistent information (i.e., nogoods).

We also mention the approach proposed in [15], which resembles our proposal in that it adopts a non-standard semantics for the P2P mappings. In that work, P2P mappings are formalized as logic programs with preferences, interpreted under a weak-minimal model semantics. Such an approach corresponds to importing in each peer the maximal subsets of facts that, together with the local data and the data imported from other peers, do not contradict its constraints. The approach can also cope with local inconsistencies, by first repairing the local database, using a classical notion of repair based on maximal consistent subsets.

We finally notice that in the very last years some work on repairing inconsistent databases has focused on the problem of singling out tractable cases for consistent query answering [17, 24, 28]. Tractability in these approaches is reached by posing suitable limitations on both the query language and the form of integrity constraints allowed on the database schema, which in some cases allow for solving consistent query answering via rewriting in first-order logic [24, 28]. We point out that, due to the inherently recursive nature of computation caused by the (cyclic) P2P mappings, such techniques are not directly applicable to our P2P setting.

9 Conclusions

In this paper we have proposed a multi-modal nonmonotonic formalization for P2PDISs which allowed us to properly model the modularity of a P2P system, isolate local inconsistency, and suitably handle data imported from different peers which are mutually inconsistent. Focusing on a specific P2PDIS in which peers have a simple structure that, however, already allows for inconsistencies to arise, but P2P mappings are fully general

(including the possibility of making cycles in the P2P network) we have characterized the computational complexity of query answering with respect to data complexity as coNP-complete. The upper-bound has been established by devising an algorithm for query answering that is derived directly from the basic properties of the multi-modal epistemic semantics. Such an algorithm can be extended to deal also with peers that have a richer structure than the one considered here. More generally, the technique developed in [14], which is based on the construction of a Datalog program to be evaluated over the data distributed in the peers, can be adapted to deal also with the kinds of inconsistencies considered here, by resorting to Disjunctive Datalog [13].

The setting reported here can be extended in several directions. First, we can remove the assumption that all peers share a common alphabet of constants by making use of mapping tables [34]. Then, an important issue is finding tractable subclasses of our inconsistency-tolerant framework, i.e., restrictions on the schema/mapping/query languages which allow for polynomial query answering. Also, we believe that preferences between peers can be smoothly integrated in our framework, following the lines of [8]. Finally, it would be interesting to study the case in which each peer in the system has its own strategy for resolving data inconsistency.

Acknowledgments. This research has been partially supported by the FET project TONES (Thinking ONtologiES), funded by the EU under contract number FP6-7603, by project HYPER, funded by IBM through a Shared University Research (SUR) Award grant, and by the MIUR FIRB 2005 project “Tecnologie Orientate alla Conoscenza per Aggregazioni di Imprese in Internet” (TOCAI.IT).

References

1. P. Adjiman, P. Chatalic, F. Gouasdoué, M.-C. Rousset, and L. Simon. Scalability study of peer-to-peer consequence finding. In *Proc. of the 19th Int. Joint Conf. on Artificial Intelligence (IJCAI 2005)*, pages 351–356, 2005.
2. P. Adjiman, P. Chatalic, F. Gouasdoué, M.-C. Rousset, and L. Simon. Distributed reasoning in a peer-to-peer setting: Application to the semantic web. *J. of Artificial Intelligence Research*, 25:269–314, 2006.
3. C. E. Alchourrón, P. Gärdenfors, and D. Makinson. On the logic of theory change: Partial meet contraction and revision functions. *J. of Symbolic Logic*, 50:510–530, 1985.
4. M. Arenas, P. Barcelo, R. Fagin, and L. Libkin. Locally consistent transformations and query answering in data exchange. In *Proc. of the 23rd ACM SIGACT SIGMOD SIGART Symp. on Principles of Database Systems (PODS 2004)*, pages 229–240, 2004.
5. M. Arenas, L. E. Bertossi, and J. Chomicki. Consistent query answers in inconsistent databases. In *Proc. of the 18th ACM SIGACT SIGMOD SIGART Symp. on Principles of Database Systems (PODS’99)*, pages 68–79, 1999.
6. P. A. Bernstein, F. Giunchiglia, A. Kementsietsidis, J. Mylopoulos, L. Serafini, and I. Zahrayeru. Data management for peer-to-peer computing: A vision. In *Proc. of the 5th Int. Workshop on the Web and Databases (WebDB 2002)*, 2002.
7. L. Bertossi, J. Chomicki, A. Cortes, and C. Gutierrez. Consistent answers from integrated data sources. In *Proc. of the 6th Int. Conf. on Flexible Query Answering Systems (FQAS 2002)*, pages 71–85, 2002.

8. L. E. Bertossi and L. Bravo. Query answering in peer-to-peer data exchange systems. In *Proc. of the EDBT Workshop on Peer-to-Peer Computing and Databases (P2P&DB 2004)*, pages 476–485, 2004.
9. L. Bravo and L. Bertossi. Logic programming for consistently querying data integration systems. In *Proc. of the 18th Int. Joint Conf. on Artificial Intelligence (IJCAI 2003)*, pages 10–15, 2003.
10. L. Bravo and L. Bertossi. Disjunctive deductive databases for computing certain and consistent answers to queries from mediated data integration systems. *Journal of Applied Logic – Special Issue on Logic-based Methods for Information Integration*, 3(2):329–367, 2005.
11. A. Cali, D. Lembo, and R. Rosati. On the decidability and complexity of query answering over inconsistent and incomplete databases. In *Proc. of the 22nd ACM SIGACT SIGMOD SIGART Symp. on Principles of Database Systems (PODS 2003)*, pages 260–271, 2003.
12. A. Cali, D. Lembo, and R. Rosati. Query rewriting and answering under constraints in data integration systems. In *Proc. of the 18th Int. Joint Conf. on Artificial Intelligence (IJCAI 2003)*, pages 16–21, 2003.
13. D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, and R. Rosati. Inconsistency tolerance in P2P data integration: an epistemic logic approach. In *Proc. of the 10th Int. Sym. on Database Programming Languages (DBPL 2005)*, pages 90–105, 2005.
14. D. Calvanese, G. De Giacomo, M. Lenzerini, and R. Rosati. Logical foundations of peer-to-peer data integration. In *Proc. of the 23rd ACM SIGACT SIGMOD SIGART Symp. on Principles of Database Systems (PODS 2004)*, pages 241–251, 2004.
15. L. Caroprese, S. Greco, C. Sirangelo, and E. Zumpano. A logic based approach to P2P databases. In *Proc. of the 13th Ital. Conf. on Database Systems (SEBD 2005)*, pages 67–74, 2005.
16. P. Chatalic, G. H. Nguyen, and M.-C. Rousset. Reasoning with inconsistency in propositional peer-to-peer inference systems. In *Proc. of the 17th Eur. Conf. on Artificial Intelligence (ECAI 2006)*, 2006.
17. J. Chomicki, J. Marcinkowski, and S. Staworko. Hippo: a system for computing consistent query answers to a class of SQL queries. In *Proc. of the 9th Int. Conf. on Extending Database Technology (EDBT 2004)*, pages 841–844. Springer, 2004.
18. G. De Giacomo, L. Iocchi, D. Nardi, and R. Rosati. A theory and implementation of cognitive mobile robots. *J. of Logic and Computation*, 9(5):759–785, 1999.
19. F. M. Donini, D. Nardi, and R. Rosati. Ground nonmonotonic modal logics. *J. of Logic and Computation*, 7(4):523–548, Aug. 1997.
20. R. Fagin, J. Y. Halpern, Y. Moses, and M. Y. Vardi. *Reasoning about Knowledge*. The MIT Press, 1995.
21. R. Fagin, P. G. Kolaitis, and L. Popa. Data exchange: Getting to the core. *ACM Trans. on Database Systems*, 30(1):174–210, 2005.
22. R. Fagin, J. D. Ullman, and M. Y. Vardi. On the semantics of updates in databases. In *Proc. of the 2nd ACM SIGACT SIGMOD Symp. on Principles of Database Systems (PODS'83)*, pages 352–365, 1983.
23. E. Franconi, G. Kuper, A. Lopatenko, and L. Serafini. A robust logical and computational characterisation of peer-to-peer database systems. In *Proc. of the VLDB International Workshop On Databases, Information Systems and Peer-to-Peer Computing (DBISP2P 2003)*, 2003.
24. A. Fuxman and R. J. Miller. First-order query rewriting for inconsistent databases. *J. of Computer and System Sciences*, 73(4):610–635, 2007.
25. P. Gärdenfors and H. Rott. Belief revision. In *Handbook of Logic in Artificial Intelligence and Logic Programming*, volume 4, pages 35–132. Oxford University Press, 1995.

26. G. Greco, S. Greco, and E. Zumpano. A logical framework for querying and repairing inconsistent databases. *IEEE Trans. on Knowledge and Data Engineering*, 15(6):1389–1408, 2003.
27. S. Gribble, A. Halevy, Z. Ives, M. Rodrig, and D. Suciu. What can databases do for peer-to-peer? In *Proc. of the 4th Int. Workshop on the Web and Databases (WebDB 2001)*, 2001.
28. L. Grieco, D. Lembo, M. Ruzzi, and R. Rosati. Consistent query answering under key and exclusion dependencies: Algorithms and experiments. In *Proc. of the 14th Int. Conf. on Information and Knowledge Management (CIKM 2005)*, pages 792–799, 2005.
29. A. Halevy, Z. Ives, D. Suciu, and I. Tatarinov. Schema mediation in peer data management systems. In *Proc. of the 19th IEEE Int. Conf. on Data Engineering (ICDE 2003)*, pages 505–516, 2003.
30. A. Y. Halevy, Z. G. Ives, P. Mork, and I. Tatarinov. Piazza: Data management infrastructure for Semantic Web applications. In *Proc. of the 12th Int. World Wide Web Conf. (WWW 2003)*, pages 556–567, 2003.
31. J. Y. Halpern and Y. Moses. A guide to completeness and complexity for modal logics of knowledge and belief. *Artificial Intelligence*, 54:319–379, 1992.
32. J. Hintikka. *Knowledge and belief*. Cornell University Press, Ithaca, New York, 1962.
33. G. E. Hughes and M. J. Cresswell. *A Companion to Modal Logic*. Methuen, London (United Kingdom), 1984.
34. A. Kementsietsidis, M. Arenas, and R. J. Miller. Mapping data in peer-to-peer systems: Semantics and algorithmic issues. In *Proc. of the ACM SIGMOD Int. Conf. on Management of Data*, pages 325–336, 2003.
35. M. Lenzerini. Data integration: A theoretical perspective. In *Proc. of the 21st ACM SIGACT SIGMOD SIGART Symp. on Principles of Database Systems (PODS 2002)*, pages 233–246, 2002.
36. H. J. Levesque and G. Lakemeyer. *The Logic of Knowledge Bases*. The MIT Press, 2001.
37. V. Lifschitz. Nonmonotonic databases and epistemic queries. In *Proc. of the 12th Int. Joint Conf. on Artificial Intelligence (IJCAI'91)*, pages 381–386, 1991.
38. V. Lifschitz. Minimal belief and negation as failure. *Artificial Intelligence*, 70:53–72, 1994.
39. F. Lin and Y. Shoham. A logic of knowledge and justified assumptions. *Artificial Intelligence*, 57(2–3):271–289, 1992.
40. J. Madhavan, P. A. Bernstein, P. Domingos, and A. Y. Halevy. Representing and reasoning about mappings between domain models. In *Proc. of the 18th Nat. Conf. on Artificial Intelligence (AAAI 2002)*, pages 80–86, 2002.
41. R. Reiter. A logic for default reasoning. *Artificial Intelligence*, 13:81–132, 1980.
42. R. Rosati. Reasoning about minimal belief and negation as failure. *J. of Artificial Intelligence Research*, 11:277–300, 1999.
43. R. Stalnaker. A note on non-monotonic modal logic. *Artificial Intelligence*, 64(2):183–196, 1993.
44. I. Tatarinov and A. Halevy. Efficient query reformulation in peer data management. In *Proc. of the ACM SIGMOD Int. Conf. on Management of Data*, 2004.
45. J. Wijsen. Database repairing using updates. *ACM Trans. on Database Systems*, 30(3):722–768, 2005.