A Robust Logical and Computational Characterization of Peer-to-Peer Database Systems

Enrico Franconi¹, Gabriel Kuper², Andrei Lopatenko¹,³, and Luciano Serafini⁴

¹ Free University of Bozen–Bolzano, Faculty of Computer Science, Italy, {franconi,lopatenko}@inf.unibz.it,
² University of Trento, DIT, Italy, kuper@acm.org
³ University of Manchester, Department of Computer Science, UK
⁴ ITC-irst Trento, Italy, serafini@itc.it

Abstract. In this paper we give a robust logical and computational characterisation of peer-to-peer (p2p) database systems. We first define a precise model-theoretic semantics of a p2p system, which allows for local inconsistency handling. We then characterise the general computational properties for the problem of answering queries to such a p2p system. Finally, we devise tight complexity bounds and distributed procedures for the problem of answering queries in few relevant special cases.

1 Introduction

The first question we have to answer when working on a logical characterisation of p2p database systems is the following: what is a p2p database system in the logical sense? In general, it is possible to say that a p2p database system is an integration system, composed by a set of (distributed) databases interconnected by means of some sort of logically interpreted mappings. However, we also want to distinguish p2p systems from standard classical logic-based integration systems, as for example described in [Lenzerini, 2002]. As a matter of fact, a p2p database system should be understood as a collection of independent nodes where the directed mappings between nodes have the only role to define how data migrates from a set of source nodes to a target node. This idea has been formulated in [Calvanese et al., 2003], where a framework based on epistemic logic is proposed as a possible solution.

Consider the following example. Suppose we have three distributed databases. The first one (DB₁) is the municipality’s internal database, which has a table Citizen-1. The second one (DB₂) is a public database, obtained from the municipality’s database, with two tables Male-2 and Female-2. The third database (DB₃) is the Pension Agency database, obtained from a public database, with the table Citizen-3. The three databases are interconnected by means of the following rules:

1: Citizen-1(x) ⇒ 2: (Male-2(x) ∨ Female-2(x))
   (this rule connects DB₁ with DB₂)
2: $\text{Male-2}(x) \Rightarrow 3:\text{Citizen-3}(x)$
2: $\text{Female-2}(x) \Rightarrow 3:\text{Citizen-3}(x)$
(these rules connect $DB_2$ with $DB_3$)

In the classical logical model, the Citizen-3 table in $DB_3$ should be filled with all of the individuals in the Citizen-1 table in $DB_1$, since the following rule is logically implied:

1: $\text{Citizen-1}(x) \Rightarrow 3:\text{Citizen-3}(x)$

However, in a p2p system this is not a desirable conclusion. In fact, rules should be interpreted only for fetching data, and not for logical computation. In this example, the tables Female-2 and Male-2 in $DB_2$ will be empty, since the data is fetched from $DB_1$, where the gender of any specific entry in Citizen-1 is not known. From the perspective of $DB_2$, the only thing that is known is that each citizen is in the view (Female-2 ∨ Male-2). Therefore, when $DB_3$ asks for data from $DB_2$, the result will be empty.

In other words, the rules

2: $\text{Male-2}(x) \Rightarrow 3:\text{Citizen-3}(x)$
2: $\text{Female-2}(x) \Rightarrow 3:\text{Citizen-3}(x)$

will transfer no data from $DB_2$ to $DB_3$, since no individual is known in $DB_2$ to be either definitely a male (in which case the first rule would apply) or definitely a female (in which case the second rule would apply). We only know that any citizen in $DB_1$ is either male or female in $DB_2$, and no reasoning about the rules should be allowed.

We shall give a robust logical and computational characterisation of p2p database systems, based on the principle sketched above. We say that our formalisation is robust since, unlike other formalisations, it allows for local inconsistencies in some node of the p2p network: if some database is inconsistent it will not result in the entire database being inconsistent. Furthermore, we propose a polynomial-time algorithm for query answering over realistic p2p networks, which does not have to be aware of the network structure, which can therefore change dynamically.

Our work has been influenced by the semantic definitions of [Serafini et al., 2003], which itself is based on the work of [Ghidini and Serafini, 1998]. [Serafini et al., 2003] defined the Local Relational Model (LRM) to formalise p2p systems. In LRM all nodes are assumed to be relational databases and the interaction between them is described by coordination rules and translation rules between data items. Coordination rules may have an arbitrary form and allow to express constraints between nodes. The model-theoretic semantics of coordination rules in [Ghidini and Serafini, 1998,Serafini et al., 2003] is non-classical, and it is very close to the local semantics introduced in this paper.

Various other problems of data management focusing on p2p systems have been considered in the literature with classical logic-based solutions. We mention here only few of them. In [Halevy et al., 2003b], query answering for relational database-based p2p systems under classical semantics is considered. The case