Semantics and containment of queries with internal and external conjunctions

Gösta Grahne[1]*, Nicolas Spyratos[2] and Daniel Stamate[2]

[1] Department of Computer Science, P.O. Box 26 (Teollisuuskatu 23)
FIN-00014 University of Helsinki, Finland
email: grahne@cs.helsinki.fi

[2] Université de Paris-Sud, LRI, U.R.A. 410 du CNRS, Bât. 490,
F-91405 Orsay Cedex, France
email: {spyratos,daniel}@lri.lri.fr

Abstract. We study conjunctive queries that combine information from multiple sources. The need for combining information is manifest for instance in multimedia systems. It has recently been recognized that query semantics for these systems should be based on some quantitative model, such as fuzzy logic. Further complications arise, however, since the semantics used *internally* by subsystems, and the semantics used *externally* to combine information, are not necessarily the same.

In this paper we give a solution based on general multivalued logics with lattice-based semantics. The internal and external semantics are tied to each other through the concept of a *bilattice*. Queries using both internal level and external level conjunctions have a natural semantics in bilattices. We then show that homomorphism techniques from core database theory carry over to query containment for internal/external conjunctive queries in the bilattice-setting. We also show that the computational complexity of determining containment of internal/external conjunctive queries is in general Π_2^p-complete, and NP-complete in some restricted cases.

1 Introduction

In this paper we are interested in the evaluation of conjunctive queries that combine information provided by multiple sources. That is, the data used in the evaluation of queries can reside in different database systems as well as in a variety of non-database data servers.

A typical example is a multimedia information system, where a query can access data in a number of different subsystems (sound-, image-, text-subsystem etc.). Each subsystem provides the answer to a subquery and the system must then combine these answers in order to provide the answer to the overall query.

There are, however, several questions that arise in this context. We shall illustrate these questions using an example. Consider an application of a store

* Research conducted while the first author was visiting at Université de Paris-Sud, Laboratoire de Recherche en Informatique.

that sells compact disks and assume that we want to produce a list of albums by Oscar Peterson, sorted according to their jazziness. To do this we use two sources of information:

- A relation $A(AlbumNo, Artist)$ residing on a relational database, and
- The estimates of an expert as to the type (jazzy, funky, ... etc) of the music announced on the cover of the albums. Let us think of these estimates as of a relation $E(AlbumNo, Type)$ residing on a sound-subsystem. Each tuple in relation E is associated with a number, say a real between 0 and 1, indicating the experts' opinion as to whether the album is of the type (0 if the expert thinks that the music of the album is not at all of the type). The expert could for instance be an approximate pattern matching algorithm that looks if words from the language of jazz occur in samples from the CD's.

The multimedia query Q that we want to answer is then a conjunct of two atoms as follows:

$$Q(X) \leftarrow A(X, oscarPeterson) \ \&\ E(X, jazz)$$

In order to evaluate this query, we must evaluate the body for every given album a. Now, the atom $A(a, oscarPeterson)$ can be associated with *true* or *false* depending on whether $A(a, oscarPeterson)$ is in the relation or not. The atom $E(a, jazz)$, on the other hand, can be associated with any real value between 0 and 1. So the question is: with what value should $Q(a)$ be associated? In other words, what is the definition of the conjunction & in this case?

In a recent paper [Fag96], Fagin suggests a solution based on fuzzy logic and gives algorithms for efficient evaluation of conjunctive queries assuming the properties of "strictness" and "monotonicity." Following his approach, each atom can be associated with a real number between 0 and 1. Thus $A(a, oscarPeterson)$ can be associated with either 0, meaning *false*, or 1, meaning *true*, whereas $E(a, jazz)$ can be associated with any real number between 0 and 1, according to the experts' estimate. The conjunction & can be defined to be the *min*-function, and thus $Q(a)$ is associated with the smaller of the values for $A(a, oscarPeterson)$ and $E(a, jazz)$.

A further complication arises, however, from the fact that a multimedia information system is usually "on top of" various subsystems, and may have no control over the semantics of these subsystems. For example, suppose that the multimedia system actually is "underneath" a supersystem, and that this supersystem has access to a second multimedia system. In the second multimedia system, there is the traditional relation $A(AlbumNo, Artist)$, and in a sound-subsystem there are estimates $E'(AlbumNo, Type)$ of a second approximate pattern matcher, perhaps with a different scoring matrix, or different samples, or different definitions of jazziness etc. Then the supersystem should evaluate the following query

$$\begin{aligned} &Q(X) \leftarrow \\ &\qquad [A(X, oscarPeterson) \& E(X, jazz)] \textit{ and} \\ &\qquad [A(X, oscarPeterson) \& E'(X, jazz)] \end{aligned}$$

where *and* is the conjunction used by the supersystem to combine the information from the subsystems. The conjunction *and* is not necessarily the same as the conjunction & used by the multimedia servers. As suggested by Fagin [Fag96], perhaps the most natural way to account for this is to define two conjunctions: the *internal conjunction*, to be used by the multimedia systems, and the *external conjunction*, to be used by the supersystem in combining the information from the various subsystems.

For instance, suppose each tuple in the relations is associated with *pairs* of real numbers between 0 and 1. A pair $\langle x, y\rangle$ for a tuple in the expert relations can be thought of as a weight of evidence *for* and *against*, respectively. The pattern matcher mentioned before contributes with x, and when the pattern is changed to, say, a language of "flatness," then y describes degree of match with this language. If we now have a tuple $E(a, jazz)$ with value $\langle x, y\rangle$, and a tuple $E'(a, jazz)$ with value $\langle z, w\rangle$, then how should these two values be combined? If one wants to extend Boolean logic, then one option is $\langle min(x, z), max(y, w)\rangle$, when *true* is $\langle 1, 0\rangle$, and *false* is $\langle 0, 1\rangle$. This type of conjunction is therefore suitable at the internal level (for &). However, another possibility for the external *and*, which Fitting [Fit91] calls "the consensus operator", is $\langle min(x, z), min(y, w)\rangle$. This gives rise to another logic with $\langle 1, 1\rangle$ (inconsistent) as maximal truth-value, and $\langle 0, 0\rangle$ (undefined) as minimal truth-value.

In this paper, we generalize the above example, and consider a set of autonomous systems connected in a network. Each system is capable of performing both internal conjunction and external conjunction. These conjunctions are the same for every system. We assume that a query, in a given system, is the external conjunction of subqueries. In turn, each subquery is the internal conjunction of atoms residing in one system. If we denote the external conjunction by $\otimes$, and the internal conjunction by $\wedge$, then the queries that we consider are of the form:

$$Q \leftarrow Q_1 \otimes Q_2 \otimes \cdots \otimes Q_n$$

where each subquery Q_i is of the form:

$$Q_i \leftarrow A_1^i \wedge A_2^i \wedge \cdots \wedge A_{m_i}^i,$$

where the A_j^i:s are atomic formulas.

The semantics of the queries are given by multivalued logics with two orderings, the *truth ordering* and the *knowledge ordering*, corresponding to internal and external conjunction, respectively. Our model is based on recent works by Fitting [Fit91, Fit93], using the concept of bilattice introduced by M. Ginsberg [Gin88]. In the framework of the model, we study the problem of containment between conjunctive queries, which is a central problem in query optimization. We show for instance, that in the bilattice framework the complexity of conjunctive query containment under both "internal" and "external" equivalence simultaneously is one level higher in the polynomial time hierarchy [Sto77] than in the classical case, i. e. the problem is Π_2^p-complete.

We emphasize that issues of efficient query evaluation in the framework considered here are very different from those in traditional database systems. Concrete evaluation methods are studied by Fagin [Fag96]. He assumes only internal

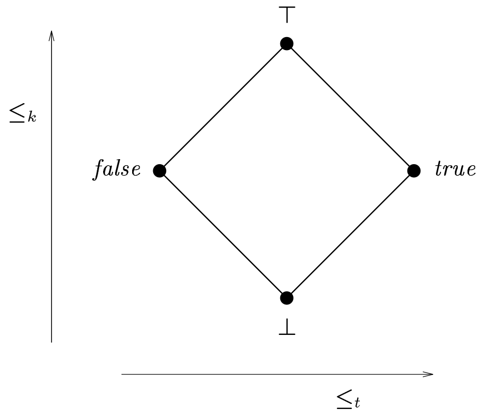

Fig. 1. The logic *FOUR*

conjunction, the semantics of which is based on fuzzy logic. Our framework generalizes the one of Fagin in the sense that we can have two different multivalued logics, one for internal conjunction, and one for external conjunction. The two logics have a combined semantics given by the bilattice. On the other hand, Fagin's approach is more general in the sense that the requirements for internal conjunction, called "strictness" and "monotonicity," follow from lattice axioms.

The rest of this paper is organized as follows: In section 2 we present the bilattice semantics and introduce our database model and queries with internal and external conjunctions. In section 3, we show that containment of internal/external conjunctive queries can be determined using homomorphism techniques. These results are related to (although incomparable with) theorems recently discovered by Ioannidis and Ramakrishnan [IR95]. In section 4 we analyze the computational complexity of determining containment of external/internal conjunctive queries. It turns out that the problem is in general Π_2^p-complete, and NP-complete when the queries use only one type of conjunction. These results are established using the homomorphism theorems and other characterizations from the previous section.

2 The Framework

In [Fit91] Fitting uses bilattices as truth-value spaces for information combined from several sources. The simplest bilattice is called *FOUR*, and it is basically Belnap's four-valued logic [Bel77]. The bilattice *FOUR* is depicted in Figure 1.

In the bilattice *FOUR* each subsystem can compute its internal conjunction (and disjunction) using classical two-valued logic (the horizontal axis). Information is combined using external conjunction (the vertical axis). Intuitively we combine conflicting truth-valued information into undefined (the bottom element in the vertical order). This can be generalized as follows.

Definition 1. A bilattice is a triple $\langle \mathcal{B}, \leq_t, \leq_k \rangle$, where $\mathcal{B}$ is a nonempty set, and $\leq_t$ and $\leq_k$ are partial orders each giving $\mathcal{B}$ the structure of a lattice with a top and bottom.

Meet and join under order $\leq_t$ are denoted $\vee$ and $\wedge$, and meet and join under $\leq_k$ are denoted $\oplus$ and $\otimes$. Top and bottom under $\leq_t$ are denoted *true* and *false*, and top and bottom under $\leq_k$ are denoted $\top$ and $\perp$. A bilattice is said to be *interlaced* if the meet and join for each partial order are monotone with respect to the other order. That is, if $a \leq_t b$ and $c \leq_t d$, then $a \otimes c \leq_t b \otimes d$, and similarly for other combinations of operations and orders.

By abuse of notation we will sometimes talk about the bilattice $\mathcal{B}$ when the orders are irrelevant or understood from the context.

A bilattice is said to be *nontrivial* if the bilattice $FOUR$ can be isomorphically embedded in it. Fitting has shown the following:

Proposition 2. *[Fit91] In any nontrivial interlaced bilattice,*

$$true \oplus false = \top, true \otimes false = \perp$$
$$\top \vee \perp = true, \top \wedge \perp = false$$

The following lemma will be useful in the sequel.

Lemma 3. *In any nontrivial interlaced bilattice* $\mathcal{B}$ *,*

$$\top \wedge a >_k \perp, \qquad true \otimes b >_t false$$

for any elements a *and* b *of* $\mathcal{B}$*.*

A bilattice is said to be *distributive* if the meet for each partial order distributes over the join of that partial order and over the meet and join of the other partial order, and similarly for the join of each partial order. In other words, $a \wedge (b \vee c) = (a \wedge b) \vee (a \vee c)$, $a \wedge (b \otimes c) = (a \wedge b) \otimes (a \wedge c)$, and ten other similar laws hold. An easy exercise yields the following

Proposition 4. *[Fit91] A distributive bilattice is interlaced.*

The bilattice $FOUR$ is distributive. The bilattice mentioned in the introduction, with pairs of reals between 0 and 1 as truth-values is formally $\langle [0,1] \times [0,1], \leq_t, \leq_k \rangle$ where $\langle x, y \rangle \leq_t \langle z, w \rangle$ iff $x \leq z$ and $w \leq y$, and $\langle x, y \rangle \leq_k \langle z, w \rangle$ iff $x \leq z$ and $y \leq w$. In this bilattice $\langle x, y \rangle \wedge \langle z, w \rangle = \langle min(x,z), max(y,w) \rangle$, and $\langle x, y \rangle \otimes \langle z, w \rangle = \langle min(x,z), min(y,w) \rangle$. This bilattice is also distributive.

In distributive bilattices the following interesting property holds.

Lemma 5. *Let* $\mathcal{B}$ *be a distributive bilattice, and* $\{a_1, \ldots, a_n, b_1, \ldots, b_m\}$ *a subset of* $\mathcal{B}$*, where* $n \geq m$*. Then*

$$(a_1 \wedge b_1) \otimes \ldots \otimes (a_m \wedge b_m) \otimes a_{m+1} \otimes \ldots \otimes a_n \leq_t$$
$$(a_1 \wedge b_1) \otimes \ldots \otimes (a_m \wedge b_m) \otimes a_{m+1} \otimes \ldots \otimes a_n \otimes (a_1 \wedge \ldots \wedge a_m)$$

The dual property, obtained by interchanging $\otimes$ *and* $\wedge$ *and replacing* $\leq_t$ *by* $\leq_k$*, also holds.*

For a proof, let us denote $(a_1 \wedge b_1) \otimes \ldots \otimes (a_m \wedge b_m) \otimes a_{m+1} \otimes \ldots \otimes a_n$ with φ. Then we have to show that $\varphi \leq_t \varphi \otimes (a_1 \wedge \ldots \wedge a_m)$. From $(a_i \wedge b_i) \leq_t a_i$ and $\varphi \leq_t \varphi$, we get by interlacing laws $\varphi \otimes (a_i \wedge b_i) \leq_t \varphi \otimes a_i$, for every $i \in \{1, \ldots, m\}$. As $\varphi \otimes (a_i \wedge b_i) = \varphi$, we get $\varphi \leq_t \varphi \otimes a_i$ for every i as above. Using now lattice laws it follows that $\varphi \leq_t (\varphi \otimes a_1) \wedge \ldots \wedge (\varphi \otimes a_m)$. As the bilattice is distributive, we have $(\varphi \otimes a_1) \wedge \ldots \wedge (\varphi \otimes a_m) = \varphi \otimes (a_1 \wedge \ldots \wedge a_m)$, and therefore $\varphi \leq_t \varphi \otimes (a_1 \wedge \ldots \wedge a_m)$.

We assume from here on that the bilattices are distributive, unless explicitly stated otherwise.

Next we consider the natural generalizations of relations and database instances. An ordinary relation can be seen as a relation where each tuple is adorned with *true* or *false*. If $\mathcal{B}$ is a bilattice, then a *$\mathcal{B}$-adorned relation* corresponding to a predicate A is a relation called also A, where each tuple t is adorned with an element $a \in \mathcal{B}$; such a tuple is denoted by $A(t) : a$, unless the relation is understood from context, in which case we may write it $t : a$. A *$\mathcal{B}$-adorned instance* is a set of $\mathcal{B}$-adorned relations. We note that Ioannidis and Ramakrishnan [IR95], and Chaudhuri and Vardi [CV93] have recently studied adorned relations. Our framework is however more general, in the sense that the adornments (labels) come from a a "two dimensional" bilattice. Furthermore, a tuple can occur in the same relation with several labels. Since there are two partial orders, there will be two notions of equivalence. We shall shortly illustrate this by an example. When the bilattice $\mathcal{B}$ is irrelevant or understood from the context, we shall sometimes omit $\mathcal{B}$ and simply talk about adorned relations and instances.

Definition 6. Let R and S be adorned relations over the same schema. Then we say that R is truth-smaller (knowledge-smaller) than S, if for each tuple $t : a$ in R, there is a tuple $u : b$ in S, such that $t = u$ and $a \leq_t b$ (respectively $a \leq_k b$).

These are clearly partial pre-orders. If we have both $R \leq_t S$ and $S \leq_t R$, then we say that R and S are *truth equivalent*, and similarly for *knowledge equivalence*.

Example 1. Let R be the relation $\langle t : \mathit{true},\ t : \mathit{false},\ u : \top,\ u : \bot \rangle$. Then R is truth-equivalent to the relation $\langle t : \mathit{true},\ u : \top,\ u : \bot \rangle$, and R is knowledge-equivalent to the relation $\langle t : \mathit{true},\ t : \mathit{false},\ u : \top \rangle$.

A *conjunctive query* Q is a formula of the form

$$\mathit{head} \leftarrow \mathit{body},$$

where *head* is a function-free atomic formula of the form $R(x_1, \cdots, x_n)$. All the variables x_i in R have to appear in *body*. A body is defined inductively as follows:

1. An atomic formula $R(x_1, \cdots, x_m)$, where R is a relation name and the x_i's are variables or constants, is a body.
2. If A and B are bodies, then so are $A \wedge B$, and $A \otimes B$.
3. Nothing else is a body

Let I be an instance, and $Q = head \leftarrow body$ a conjunctive query. Furthermore, let θ be an assignment of Q into I, that is, a mapping from the variables in Q to constants in I. Then the *result due to* θ *of* Q *over* I, denoted $Q_\theta(I)$, is the set of all adorned tuples $head\theta : c$, such that for each atom $R(x_1, \cdots, x_m)$ in *body*, the tuple $R(\theta(x_1), \cdots, \theta(x_m))$ occurs with some label in I.[3] The value of the label c is defined recursively as follows.

1. If *body* is of the form $A \circ B$, where $\circ \in \{\wedge, \otimes\}$, and $head\theta : a$ is in $(head \leftarrow A)_\theta(I)$, and $head\theta : b$ is in $(head \leftarrow B)_\theta(I)$, then $c = a \circ b$.
2. If *body* is of the form $R(x_1, \cdots, x_m)$, and the tuple $R(\theta(x_1), \cdots, \theta(x_m))$ occurs with label a in I, then $c = a$.

The result of a query Q over an instance I, denoted $Q(I)$, is then defined to be $\cup_\theta Q_\theta(I)$, where θ is any assignment of Q into I.

3 Containment of conjunctive queries

We are now ready to define containment of queries over labelled instances.

Definition 7. Let P and Q be conjunctive queries. Then we say that P is *truth*-contained in Q (P is *knowledge*-contained in Q), denoted $P \leq_t Q$ (resp. $P \leq_k Q$), if for every instance I it is true that $P(I) \leq_t Q(I)$ (resp. $P(I) \leq_k Q(I)$). If we have both $P \leq_t Q$ ($P \leq_k Q$) and $Q \leq_t P$ (resp. $Q \leq_k P$), we say that P and Q are *truth-equivalent*, denoted $P \equiv_t Q$ (we say that P and Q are *knowledge equivalent*, denoted $P \equiv_k Q$). If we have both $P \equiv_t Q$ and $P \equiv_k Q$, we write $P \equiv Q$.

Containment is of course relative to a given bilattice $\mathcal{B}$, and should really be written $P \leq_t^{\mathcal{B}} Q$ and $P \leq_k^{\mathcal{B}} Q$. We shall however omit the superscript.

The notion of homomorphism is essential in containment of conjunctive queries [AHV95]. In the context of our work, the notion of homomorphism is defined as follows.

Definition 8. Let P and Q be two conjunctive queries whose heads are compatible (*i. e.* they represent relations of the same arity). A *homomorphism* $h : Q \to P$ is a total function from the variables and constants of Q to the variables and constants of P such that if x and y are variables appearing in the same argument position in the head of Q and P respectively, then $h(x) = y$, and if $R(x_1, \cdots, x_n)$ appears in Q, then $R(h(x_1), \cdots, h(x_n))$ appears in P. Furthermore, h has to be the identity on constants.

We will assume *w. l. o. g.* that all queries use a designated predicate symbol, say H, in their heads. (The predicate symbol H has no fixed arity.) The first result concerns containment of queries that only use either external or internal conjunctions.

[3] As usual, the mapping θ is extended to be the identity on constants. Furthermore, $E\theta$ where E is a formula, a query, or a vector is obtained from the expression E by replacing every variable x in E with $\theta(x)$.

Theorem 9. *Let P and Q be conjunctive queries that only use $\wedge$ (respectively only use $\otimes$) in their bodies. Then $P \leq_t Q$ (resp. $P \leq_k Q$) if and only if there exists a homomorphism $h : Q \rightarrow P$.*

Indeed, let Q be $H(\mathbf{x}) \leftarrow A_1(\mathbf{x}_1) \wedge \cdots \wedge A_n(\mathbf{x}_n)$, and let P be $H(\mathbf{y}) \leftarrow B_1(\mathbf{y}_1) \wedge \cdots \wedge B_m(\mathbf{y}_m)$, such that $H(\mathbf{x})$ and $H(\mathbf{y})$ are compatible (here $\mathbf{x}, \mathbf{x}_1, \ldots, \mathbf{y}, \mathbf{y}_1, \ldots$, are vectors of variables). Suppose we have a homomorphism h from Q to P. That is, $h(\mathbf{x}) = \mathbf{y}$, and for each $i \in \{1, \cdots, n\}$ there is a $j_i \in \{1, \ldots, m\}$, such that $A_i(h(\mathbf{x}_i)) = B_{j_i}(\mathbf{y}_{j_i})$. Let I be an instance, and $H(t) : b$ an adorned tuple in $P(I)$. Obviously, there is an assignment θ such that $\mathbf{y}\theta = t$ and $b = b_1 \wedge \cdots \wedge b_m$, where $\{B_1(\mathbf{y}_1)\theta : b_1, \cdots, B_m(\mathbf{y}_m)\theta : b_m\} \subseteq I$. Let σ be $\theta \circ h$. From the construction of σ, it follows that $\{A_1(\mathbf{x}_1)\sigma : b_{j_1}, \cdots, A_n(\mathbf{x}_n)\sigma : b_{j_n}\} \subseteq I$. Let $a = b_{j_1} \wedge \cdots \wedge b_{j_n}$. Now we can conclude that $H(\mathbf{x})\sigma : a$ belongs to $Q(I)$. It follows from the definition of the lattice join $\wedge$ that $b \leq_t a$. Since $H(\mathbf{y})\theta = H(\mathbf{x})\sigma$, we have that $P(I) \leq_t Q(I)$.

Conversely, suppose now that $P \leq_t Q$. Let the set of variables occurring in P be $\{y_1, \ldots, y_p\}$. Suppose *w. l. o. g.* that the constants in $\{1, \cdots, p\}$ do not occur in P or Q. Let θ be the map $y_i \mapsto i$. Then θ is a bijection from $\{y_1, \ldots, y_p\}$ to $\{1, \ldots, p\}$. Let I be the instance $\{B_1(\mathbf{y}_1)\theta : \mathit{true}, \cdots, B_m(\mathbf{y}_m)\theta : \mathit{true}\}$. Obviously, $H(\mathbf{y})\theta : \mathit{true}$ is in $P(I)$, and as $P(I) \leq_t Q(I)$, it follows that $H(\mathbf{y})\theta : \mathit{true}$ is in $Q(I)$. Thus there must be an assignment σ, such that $\mathbf{x}\sigma = \mathbf{y}\theta$, or equivalently $\theta(\mathbf{y}) = \sigma(\mathbf{x})$. That is, $\mathbf{y} = \theta^{-1}(\sigma(\mathbf{x}))$. Define h as $\theta^{-1} \circ \sigma$. As $H(\mathbf{x})\sigma : \mathit{true}$ is in $Q(I)$, it must be that $\{A_1(\mathbf{x}_1)\sigma : \mathit{true}, \cdots, A_n(\mathbf{x}_n)\sigma : \mathit{true}\} \subseteq I$. From the construction of I, it follows that for each $i \in \{1, \cdots, n\}$ there is a $j_i \in \{1, \ldots, m\}$, such that $A_i(\mathbf{x}_i)\sigma = B_{j_i}(\mathbf{y}_{j_i})\theta$, so $A_i = B_{j_i}$ and $\mathbf{x}_i\sigma = \mathbf{y}_{j_i}\theta$, or equivalently $\mathbf{y}_{j_i} = h(\mathbf{x}_i)$. That is, $A_i(h(\mathbf{x}_i)) = B_{j_i}(\mathbf{y}_{j_i})$, and as $h(\mathbf{x}) = \mathbf{y}$, the mapping h is a homomorphism from Q to P.

Ioannidis and Ramakrishnan [IR95] have recently proved a similar homomorphism characterization for annotated instances, where the annotations form a structure called a "label system of type A." We note that there are lattices that are not "label systems of type A," and that there are "label systems of type A" that are not lattices. Fuzzy logic, however, is both a lattice and a "label system of type A."

Next we characterize containment of queries whose bodies contain both $\wedge$ and $\otimes$. We will need two notions of normal form. A query Q is said to be in *truth-normal form* if it is of the form

$$\mathit{head} \leftarrow Q_1 \otimes Q_2 \otimes \cdots \otimes Q_n$$

where each Q_i is of the form:

$$A_1^i \wedge A_2^i \wedge \cdots \wedge A_{m_i}^i,$$

and the A_j^i's are atomic. Likewise, a query Q is in *knowledge-normal form* if it is of the form

$$\mathit{head} \leftarrow Q_1 \wedge Q_2 \wedge \cdots \wedge Q_n$$

where each Q_i is of the form:

$$A_1^i \otimes A_2^i \otimes \cdots \otimes A_{m_i}^i,$$

with atomic A_j^i's.

If Q is a query, then its normal forms will be denoted Q^t and Q^k, respectively. Every query can be transformed into truth-normal form or knowledge-normal form recursively as follows: Let E be the body of Q. We denote by E^t the truth-normal form of E. If E is an atom, then $E^t = E$. If $E = E_1 \otimes E_2$, then $E^t = E_1^t \otimes E_2^t$. If $E = E_1 \wedge E_2$, let E_1^t be $F_1 \otimes F_2 \otimes \cdots \otimes F_n$ and let E_2^t be $G_1 \otimes G_2 \otimes \cdots \otimes G_m$. Then E^t is built as $\bigotimes_{i,j}(F_i \wedge G_j)$. The knowledge-normal form is obtained similarly. Note that in the introduction we assumed that the queries were in truth-normal form. Notice also that, as the two conjunctions are associative and commutative, there are actually several truth-normal forms for a query. Then Q^t denotes some chosen element from the set of truth-normal forms of Q, and similarly for Q^t. The normal form transformation can be implemented for instance as an ordered rewriting system. Transforming into normal form does not necessarily preserve equivalence, unless the two conjunctions are distributive with respect to each other. In general, the following relationships hold between a query and its normal forms.

Lemma 10. *$Q \leq_k Q^t$, and $Q \leq_t Q^k$ in any interlaced bilattice $\langle \mathcal{B}, \leq_t, \leq_k \rangle$. If $\mathcal{B}$ is distributive, then in addition $Q \equiv Q^t$, and $Q \equiv Q^k$.*

The proof of the lemma is an easy induction on the structure of the normal forms, using the interlacing or distribution laws.

Now Theorem 9 can be generalized to conjunctive queries using both external and internal conjunctions.

Theorem 11. *Let $C \leftarrow \bigotimes_i P_i$ and $Q = D \leftarrow \bigotimes_i Q_i$ be queries in truth-normal form (let $P = C \leftarrow \bigwedge_i P_i$ and $Q = D \leftarrow \bigwedge_i Q_i$ be queries in knowledge-normal form). Then $P \leq_t Q$ (resp. $P \leq_k Q$) in a distributive bilattice if and only if both of the following hold*

1. *For each P_i in the body of P there is a Q_j in the body of Q, such that there is a homomorphism from the query $D' \leftarrow Q_j$ to the query $C' \leftarrow P_i$, where the head C' is obtained from C by projecting on the variables appearing in P_i, and similarly for D'.*
2. *For each Q_j in the body of Q there are atoms $C_1, \ldots, C_k$ in the body of P, such that there is a homomorphism from the query $D' \leftarrow Q_j$ to the query $C' \leftarrow C_1 \wedge \ldots \wedge C_k$ (resp. to the query $C' \leftarrow C_1 \otimes \ldots \otimes C_k$), where D' and C' are obtained as above.*

The *if* part follows from applying Theorem 9 and Lemma 5.

For the *only if* part and point 1, suppose the set of variables occurring in P is $\{y_1, \cdots, y_p\}$, and let θ be the map $y_i \mapsto i$. Suppose then that $P_i = C_{i_1} \wedge \cdots \wedge C_{i_n}$. Let I be the instance containing the adorned tuples $C_{i_j}\theta$: *true*, for all $j \in$

$\{1, \ldots, n\}$, and the adorned tuples $A\theta : \mathit{false}$, for all atoms A that occur in the body of P, but not in P_i. It follows from Lemma 3 that there is an adorned tuple $C\theta : c$ in $P(I)$, such that $c >_t \mathit{false}$. Since $P(I) \leq_t Q(I)$, there must be an adorned tuple $t : d$ in $Q(I)$, such that $c \leq_t d$, and consequently $d >_t \mathit{false}$. That is, C and D are compatible and there is a conjunct $Q_j = D_{j_1} \wedge \cdots \wedge D_{j_m}$ in the body of Q, and an assignment σ, such that $(D' \leftarrow Q_j)_\sigma(I)$ contains an adorned tuple $D'\sigma : d'$, where $d' >_t \mathit{false}$ since $d' = \mathit{false}$ would entail $d = \mathit{false}$. This means that I has to contain all the adorned tuples of the form $D_{j_i}\sigma : \mathit{true}$, for all $i \in \{1, \ldots, m\}$. The desired homomorphism can now be defined as in the proof of Theorem 9.

For the *only if* part and point 2, suppose there is a Q_j in the body of Q, such that there is no set $\{C_1, \ldots, C_k\}$ of atoms from the body of P, for which we can find a homomorphism from the query $D' \leftarrow Q_j$ to the query $C' \leftarrow C_1 \wedge \cdots \wedge C_k$. Once again, let $\{y_1, \cdots, y_p\}$ be the variables occurring in P, and let θ be the map $y_i \mapsto i$. Consider the instance I, where I contains $A\theta : \mathit{true}$, for all atoms A in the body of P. Then the adorned tuple $C\theta : \mathit{true}$ is in $P(I)$. Since we assumed that $P \leq_t Q$, there must be an assignment σ such that the following requirements are satisfied: $D\sigma$ and $C\theta$ contain the same tuple t, $D\sigma : \mathit{true}$ belongs to $Q(I)$, and furthermore the result $(D' \leftarrow Q_j)_\sigma(I)$ contains the adorned tuple $D'\sigma : \mathit{true}$. Suppose $Q_j = D_{j_1} \wedge \cdots \wedge D_{j_m}$. Consider the mapping $h = \theta^{-1} \circ \sigma$. Then $h(D_{j_i})$ is in the body of P, for $i \in \{1, \ldots, m\}$; a contradiction to our counterassumption that no such homomorphism exists.

The Sagiv-Yannakakis theorem [SY80] for containment of unions of conjunctive queries comes as a special case, if we take union to be $\otimes$ in the trivial bilattice with $\mathcal{B} = \{\mathit{true}, \mathit{false}\}$, $\mathit{false} \leq_t \mathit{true}$ and $\mathit{true} \leq_k \mathit{false}$. (Strictly speaking, in the trivial bilattice we only need condition 1 of Theorem 11.) Likewise, the generalization provided by Ioannidis and Ramakrishnan [IR95] is another special case of Theorem 11 in a similar trivial structure, where $\leq_k$ is the inverse of $\leq_t$.

If the bilattice is not distributive, we need to restrict our attention to the following case. Consider a partition of the set of predicates in our language. Call each block in this partition a server. Let $P = C \leftarrow \bigotimes_i P_i$ be a query in truth-normal form. Then we say that P is *well distributed* over the server-partition, if all predicates in each P_i in the body of P are from the same server, and if two predicates come from different P_i and P_j in the body of P, then they belong to different servers. Queries combining information from multiple sources are exactly of this kind.

Now we have the following.

Theorem 12. *Let $P = C \leftarrow \bigotimes_i P_i$ and $Q = D \leftarrow \bigotimes_i Q_i$ be queries well distributed over the same server-partition. Then $P \leq_t Q$ in a possibly non-distributive interlaced bilattice if and only if both of the following hold*

1. *As in Theorem 11.*
2. *For each Q_j in the body of Q there is a P_i in the body of P, such that there is a homomorphism from the query $D' \leftarrow Q_j$ to the query $C' \leftarrow P_i$.*

Here we apply Theorem 9 and the interlacing laws. The *only if* part for point 2 is similar to the proof Theorem 11. If the queries P and Q are in truth-normal form, but not well distributed, then the conditions in Theorem 12 are only sufficient.

4 The complexity of containment

Theorem 13. *Let P and Q be queries in truth or knowledge-normal form. Then the problem of testing whether $P \leq_t Q$ or $P \leq_k Q$, respectively, is NP-complete.*

The problem is NP-hard, since existence of homomorphism between classical conjunctive queries is NP-complete [CM77]. The problem is in NP, since as a consequence of Theorem 11, it is the intersection of polynomially many problems (existence of homomorphisms) in NP.

The complexity remains the same in non-distributive bilattices, if we restrict ourselves to well distributed queries. Formally we have:

Corollary 14. *Testing for truth-containment of well distributed queries in a possibly non-distributive, interlaced bilattice is NP-complete.*

For strong containment of queries both internal and external conjunctions we have the following result.

Theorem 15. *Let P and Q be queries in truth or knowledge-normal form. Then the problem of testing whether both $P \leq_t Q$ and $P \leq_k Q$, in a distributive bilattice, is Π_2^p-complete.*

Suppose that P and Q are in truth-normal form. From Theorem 12 we know that testing for $P \leq_t Q$ is in NP. We then need to test if $P \leq_k Q$. We can check if $P^k \not\leq_k Q^k$ as follows. Guess (in NP) a conjunct P_i of P^k and consult an oracle to see if there is a conjunct Q_j of Q^k, such that $P_i \leq_k Q_j$. From Theorem 13 we know that this can be done by an NP-oracle. If the oracle answers "no" output "yes." On the other hand, if indeed $P^k \leq_k Q^k$, then the oracle will never answer "no." Since $P^k \not\leq_k Q^k$ thus is seen to be in Σ_2^p, the problem $P^k \leq_k Q^k$ is in Π_2^p. The lower bound is established reducing a $\forall\exists$-quantified Boolean formula into two queries P and Q, if $P \leq_t Q$, then the formula is true iff $P \leq_k Q$.

The case where P and Q are in knowledge-normal form is symmetric to the one above.

It follows that testing for $P \leq_t Q$ or $P \leq_k Q$, when P and Q are not in any normal form, is Π_2^p-complete. Formally:

Corollary 16. *Let P and Q be queries not necessarily in any normal form. Then the problem of testing whether $P \leq_t Q$ or $P \leq_k Q$, respectively, in a distributive bilattice, is Π_2^p-complete.*

These complexity results generalize the Sagiv-Yannakakis Π_2^p-completeness result [SY80] from the trivial bilattices mentioned after Theorem 11 to general distributive bilattices.

5 Conclusion

Queries combining information from multiple sources have a natural semantics in bilattices. Determining containment of queries under bilattice semantics can be done through homomorphism techniques. Computationally, the problem is Π_2^p-complete. If the interlaced bilattice is not distributive, and the queries are not well distributed, then only a sufficient condition for queries in normal form can currently be given. It is not known if containment of non-well distributed queries is decidable when the semantics is given by a non-distributive interlaced bilattice, let alone if the queries in this case are not in any normal form.

References

[AHV95] Abiteboul S., Hull R., and Vianu V. *Foundations of Databases.* Addison-Wesley, Reading Ma. 1995.

[Bel77] Belnap N. D., Jr. A useful four-valued logic. In: J. Michael Dunn and G. Epstein (Eds.), *Modern Uses of Multiple-Valued Logic,* Reidel, 1977, pp. 89–148.

[CM77] Chandra A. K. and Merlin P. M. Optimal implementation of conjunctive queries in relational databases. *Proc. 9th Annual ACM Symp. on the Theory of Computing*, pp. 77–90.

[CV93] Chaudhuri S. and Vardi M. Optimization of real conjunctive queries. *Proc. 12th ACM Symp. on Principles of Database Systems,* pp. 59–70.

[Fag96] Fagin R. Combining fuzzy information from multiple systems. *Proc. 15th ACM Symp. on Principles of Database Systems.*

[Fit91] Fitting M. Bilattices and the semantics of logic programming. *J. Logic Programming* **11**, 91–116.

[Fit93] Fitting M. The family of stable models. *J. Logic Programming* **17**, 197–225.

[Gin88] Ginsberg M. L. Multivalued logics: a uniform approach to reasoning in artificial intelligence. *Computational Intelligence* **4**, pp. 265–316.

[IR95] Ioannidis Y. E. and Ramakrishnan R. Containment of conjunctive queries: beyond relations as sets. *ACM Transactions on Database Systems* **20**:3, pp. 288–324.

[SY80] Sagiv Y. and Yannakakis M. Equivalences among relational expressions with the union and difference operators. *J. ACM* **27**:4, pp. 633–655.

[Sto77] Stockmeyer L. J. The polynomial-time hierarchy. *Theoret. Comput. Sci.* **3**, pp. 1–22.

This article was processed using the LaTeX macro package with LLNCS style