Accuracy and Efficiency of Fixpoint Methods for Approximate Query Answering in Locally Complete Databases

Álvaro Cortés-Calabuig[†], **Marc Denecker**[†], **Ofer Arieli**[‡], **Maurice Bruynooghe**[†]

[†] Department of Computer Science, Katholieke Universiteit Leuven, Belgium.
{alvaro.cortes,marc.denecker,maurice.bruynooghe}@cs.kuleuven.be

[‡] Department of Computer Science, The Academic College of Tel-Aviv, Israel.
oarieli@mta.ac.il

Abstract

Standard databases convey Reiter's closed-world assumption that an atom not in the database is false. This assumption is relaxed in locally closed databases that are sound but only *partially complete* about their domain. One of the consequences of the weakening of the closed-world assumption is that query answering in locally closed databases is undecidable. In this paper, we develop efficient *approximate* methods for query answering, based on fixpoint computations, and investigate conditions that assure the optimality of these methods. Our approach of approximative reasoning may be incorporated in different contexts where incompleteness plays a major role and efficient reasoning is imperative.

Introduction

The *Closed-World Assumption* (CWA) on databases (Reiter 1982) expresses that an atom not in the database is false. However, as the following example shows, in many cases database information is only *partially* complete, and so applying the CWA is not correct, and may lead to wrong conclusions.

Example 1 Consider the following database of a computer science department. This database stores information about the telephone numbers of the department's members and collaborators.

Telephone

Name	Telephone
Leen Desmet	6531421
Leen Desmet	09-23314
Bart Delvaux	5985625
Tom Demans	5845213

Department

Name	Department
Bart Delvaux	Computer Science
Leen Desmet	Philosophy
Tom Demans	Computer Science
David Finner	Biology

Assume that the database is complete with respect to the telephone number of all department members, but it is not complete regarding external collaborators. Thus, appropriate answers for the queries $\mathsf{Tel}(\text{Bart Delvaux}, 3962836)$ and $\mathsf{Tel}(\text{Leen Desmet}, 3212445)$ should be 'no' and 'unknown', respectively. If completeness of the database is taken for granted, the answer for both of these queries is 'no'. Similarly, the answer under the CWA for the query $\exists x \mathsf{Tel}(\text{David Finner}, x)$ is 'no', but as the database is complete only with respect to the computer science department, one cannot exclude the possibility that David Finner has a phone number, so the correct answer in this case should be 'unknown'.

For dealing with situations like that of Example 1, the databases considered here are *locally closed* in the sense that in addition to a standard database instance they also contain a collection of *local closed world assumptions*, expressing conditions (the *windows of expertise*) that guarantee complete knowledge about specific database predicates.

Example 2 In the context of the previous example, an expression of the form $\mathcal{LCWA}(\mathsf{Tel}(x, y), \mathsf{Dept}(x, \mathsf{CS}))$ is intuitively understood as expressing that in the database instance, the relation Tel contains all the telephone numbers of all the members of the computer science department. Here, the window of expertise is $\mathsf{Dept}(x, \mathsf{CS})$ and it is referring to the database relation $\mathsf{Tel}(x, y)$.

Not surprisingly, query answering in locally closed databases (also referred here as locally complete databases) turns out to be intractable in general. This provides the motivation for developing efficient *approximate methods* for query answering. This approach is also motivated by the fact that in many applications there is no need to have *all* the answers to a query and often it is enough to have a sufficiently large subset of them. For instance, a company searching in an (incomplete) database for a provider of some urgently required service will be satisfied by finding *some* candidate providers, so an exhaustive search for all the providers is not always needed.

The current work builds upon (Cortés-Calabuig et al. 2007), which in turn relies on an extension of the formalism of Levy (1996) for representing partially complete information in database systems. In the former, a polynomial algorithm for querying a subclass of locally closed databases, called hierarchically closed, is introduced. This algorithm is based on implicit approximations of all the models of a locally closed database using three-valued structures. In order to guarantee completeness, strong syntactical conditions are imposed on the hierarchically closed databases and on the queries. Optimality results are provided for so-called conjunctive and disjunctive queries.

The approximation techniques mentioned above are extended and improved in this paper. In particular, we introduce a novel approach, based on query rewriting and fixpoint

computations, whose main benefits are the following:

- It generalizes the class of queries and databases considered in (Cortés-Calabuig et al. 2007). More specifically, in the new setting the query answering mechanism always terminates, even in case of arbitrary circular dependencies between predicates of LCWAs and the predicate in its windows of expertise.
- Certain types of integrity constraints are integrated in the query answering process. In particular, whenever the database is consistent, primary key constraints can be incorporated into the framework. Thus, in such cases, our framework may be used for approximative query answering in standard, constraint databases.
- A thorough analysis of the optimality of approximative query answering is developed. It shows that our approach retrieves correct and complete answers for a considerably larger class of queries and locally closed databases than those that are covered in (Cortés-Calabuig et al. 2007).

The rest of this paper is organized as follows. In the next section we recall the basic concepts, properties and semantics of locally closed databases as introduced in (Levy 1996; Cortés-Calabuig et al. 2005; 2006; 2007). Then we survey some previous results regarding query answering in locally closed databases and show an undecidability result concerning the closed-world information, which motivates the use of approximative query answering techniques. The main contribution of this paper is given in the next three sections, where these techniques are defined and analyzed. Finally, we discuss some related works and conclude.

Representing Incompleteness in Databases

First, we recall the basic concepts concerning the local closed world assumption. The definitions in this section are taken from (Cortés-Calabuig et al. 2005).

We denote by Σ a finite first-order vocabulary, consisting of sets $\mathcal{R}(\Sigma)$ of predicate symbols and $\mathcal{C}(\Sigma)$ of constants. In addition, Σ contains equality $=$ and the propositional constants $\mathbf{t}$ (*true*) and $\mathbf{f}$ (*false*), interpreted in the standard way. First-order formulas over Σ are constructed as usual. $\Psi[\bar{x}]$ denotes a formula with free variables that are a subset of $\bar{x}$. An interpretation I for Σ (Σ-structures) is also defined as usual. In particular, a Herbrand interpretation has a domain $\mathcal{C}(\Sigma)$, such that each element of $\mathcal{C}(\Sigma)$ interprets itself.

For a domain element $a \in Dom(I)$, let us define $a^I = a$. For a tuple $\bar{t}$ of terms, we define $\bar{t}^I = (t_1^I, \ldots, t_n^I)$. Setting $\mathbf{f} \leqslant \mathbf{t}$ and $\mathbf{f}^{-1} = \mathbf{t}$, $\mathbf{t}^{-1} = \mathbf{f}$, the truth value of a variable free formula φ in I, denoted φ^I, is defined recursively by

$P(\bar{t})^I = \mathbf{t}$ if $\bar{t}^I \in P^I$, otherwise $P(\bar{t})^I = \mathbf{f}$;
$(\psi \wedge \phi)^I = \min_{\leqslant}(\psi^I, \phi^I)$; $(\psi \vee \phi)^I = \max_{\leqslant}(\psi^I, \phi^I)$;
$(\neg\psi)^I = (\psi^I)^{-1}$;
$(\forall x\, \psi[x])^I = \min_{\leqslant}\{(\psi[a])^I \mid a \in Dom(I)\}$;
$(\exists x\, \psi[x])^I = \max_{\leqslant}\{(\psi[a])^I \mid a \in Dom(I)\}$.

We denote by $I \models \varphi$ that $\varphi^I = \mathbf{t}$.

A *database instance* D is a Herbrand interpretation with a finite domain $Dom^D \subseteq \mathcal{C}(\Sigma)$. The set Dom^D is sometimes called the *active domain* of the database instance and contains at least all the constants in the tables of D (and often only those). For some variable-free atomic formula A, we write $A \in D$ to denote that $A = P(\bar{d})$, where $\bar{d} \in P^D$. In what follows, we will often specify a database instance D by a set of atoms. Unless the domain of D is explicitly mentioned, it consists of the set of constants that appear in these atoms. In our setting, databases represent *partial knowledge* on the domain of discourse, and as such their instances cannot be viewed as the (unique) possible state of the world.

Definition 3 *A* local closed-world assumption *(LCWA) is an expression $\mathcal{LCWA}(P(\bar{x}), \Psi[\bar{x}])$, where $P \in \mathcal{R}(\Sigma)$ is called the LCWA's* object *and $\Psi[\bar{x}]$, called the LCWA's* window of expertise*, is a first-order formula over Σ.* [1]

The intuitive reading of the expression in Definition 3 is the following: "for all objects $\bar{x}$ such that $\Psi(\bar{x})$ holds in the *real world*, if an atom of the form $P(\bar{x})$ is true in the real world, then $P(\bar{x})$ occurs *in the database*". Note that in $P(\bar{x})$ the value of the variables $\bar{x}$ are constrained by the windows of expertise, Ψ.

Definition 4 *A* locally closed database $\mathfrak{D}$ *over Σ is a pair $(D, \mathcal{L})$ of a database instance D over Σ and a finite set $\mathcal{L}$ of local closed-world assumptions over Σ.*

Given a locally closed database $\mathfrak{D} = (D, \mathcal{L})$, we denote by $dom(\mathfrak{D})$ its active domain. That is, $dom(\mathfrak{D})$ is the union of active domain Dom^D of D and the set of constants in $\mathcal{L}$. A signature where $\mathcal{R}(\Sigma_{\mathfrak{D}}) = \mathcal{R}(\Sigma)$ and $\mathcal{C}(\Sigma_{\mathfrak{D}}) = dom(\mathfrak{D})$ is denoted by $\Sigma_{\mathfrak{D}}$. The *base predicates* of $\mathfrak{D}$ are the propositional constants $\mathbf{t}, \mathbf{f}$ and all the predicates P such that the local closed-world assumption $\mathcal{LCWA}(P(\bar{x}), \mathbf{t})$ appears in $\mathcal{L}$.

Example 5 Abbreviate the database of Example 1 as follows:

$$D = \left\{ \begin{array}{l} \mathsf{Tel}(\mathsf{LD}, 6531421), \mathsf{Tel}(\mathsf{BD}, 5985625), \\ \mathsf{Tel}(\mathsf{TD}, 5845213), \mathsf{Tel}(\mathsf{LD}, \text{09-23314}), \\ \mathsf{Dept}(\mathsf{BD}, \mathsf{CS}), \mathsf{Dept}(\mathsf{LD}, \mathsf{Phil}), \\ \mathsf{Dept}(\mathsf{TD}, \mathsf{CS}), \mathsf{Dept}(\mathsf{DF}, \mathsf{Bio}) \end{array} \right\}$$

As noted in Example 2, the local closed world assumption $\mathcal{LCWA}(\mathsf{Tel}(x, y), \mathsf{Dept}(x, \mathsf{CS}))$ states that all the telephone numbers of the computer science department members are known and occur in the database instance. That is, for every x_0 in $\{x \mid \mathsf{Dept}(x, \mathsf{CS})\}$ (the window of expertise for Tel), all true atoms of the form $\mathsf{Tel}(x_0, y)$ are in the database. Similarly, $\mathcal{LCWA}(\mathsf{Tel}(x, y), x = \mathsf{LD})$ expresses that D contains all telephone numbers of Leen Desmet.

Semantics of locally closed databases

The intuitive meaning behind the LCWA expressions of Definition 3 can be formally captured using first-order formulas.

[1] Definition 3 is based on the notion of Levy's local completeness statements (Levy 1996). Apart for the (innocent) difference that we use logical notation rather than Levy's database notation, LCWAs are more expressive by allowing arbitrary first-order formulas – instead of conjunction of atoms – in the window of expertise.

Definition 6 *Let D be a database based over a first-order vocabulary Σ and let P be a predicate in D. Denote by P^D the P-tuples in D. Given a tuple $\bar{x}$ of terms, we denote by $P(\bar{x}) \in D$ the formula $\vee_{\bar{a} \in P^D}(\bar{x} = \bar{a})$.*

Definition 7 *Let D be a database over a vocabulary Σ and let $\theta = \mathcal{LCWA}(P(\bar{x}), \Psi[\bar{x}])$ be an LCWA over Σ. The* meaning *of θ in D is given by the formula*

$$\mathcal{M}_D(\theta) = \forall \bar{x}\big(\Psi[\bar{x}] \supset \big(P(\bar{x}) \supset (P(\bar{x}) \in D)\big)\big).$$

Observe that the meaning of a local closed-world assumption is induced by the database instance under consideration, as it contains the subformula $P(\bar{x}) \in D$. As such, a local closed-world assumption is a *non-monotonic* construct.

Example 8 Consider again Example 5. The meaning of $\theta = \mathcal{LCWA}(\mathsf{Tel}(x, y), \mathsf{Dept}(x, \mathsf{CS}))$ is given by

$$\mathcal{M}_D(\theta) = \forall x \forall y\big(\mathsf{Dept}(x, \mathsf{CS}) \supset \big(\mathsf{Tel}(x, y) \supset \big((x = \mathsf{LD} \wedge y = 6531421) \vee (x = \mathsf{LD} \wedge y = 09\text{-}23314) \vee (x = \mathsf{BD} \wedge y = 5985625) \vee (x = \mathsf{TD} \wedge y = 5845213)\big)\big)\big).$$

As shown in (Cortés-Calabuig et al. 2006), any collection of LCWAs on the same predicate may be combined into one (disjunctive) LCWA. We therefore assume that each predicate P in $\mathcal{R}(\Sigma)$ is the object of exactly *one* LCWA expression, whose window of expertise is denoted Ψ_P.

The meaning of a locally closed database $\mathfrak{D} = (D, \mathcal{L})$ is expressed by a first-order formula consisting of the conjunction of the database atoms, the meaning of the given local closed-world assumptions, and the following two axioms:

- *Domain Closure*: $\mathsf{DCA}(dom(\mathfrak{D})) = \forall x(\bigvee_{i=1}^{n} x = c_i)$
- *Unique Names*: $\mathsf{UNA}(dom(\mathfrak{D})) = \bigwedge_{1 \leqslant i < j \leqslant n} c_i \neq c_j$

where $dom(\mathfrak{D}) = \{c_1, \ldots, c_n\}$.

Definition 9 *Let $\mathfrak{D} = (D, \mathcal{L})$ be a locally closed database over Σ. The* meaning $\mathcal{M}(\mathfrak{D})$ *of $\mathfrak{D}$ is the first-order sentence*

$$\mathsf{UNA}(dom(\mathfrak{D})) \wedge \mathsf{DCA}(dom(\mathfrak{D})) \wedge \bigwedge_{A \in D} A \wedge \bigwedge_{\theta \in \mathcal{L}} \mathcal{M}_D(\theta).$$

The formula $\mathcal{M}(\mathfrak{D})$ expresses incomplete knowledge about the real world. Thus, in general, it has several (Herbrand) models. A $\Sigma_{\mathfrak{D}}$-model M of $\mathcal{M}(\mathfrak{D})$ is called a model of $\mathfrak{D}$, and this is denoted by $M \models \mathfrak{D}$. If every model of $\mathfrak{D}$ is also a model of a formula φ over $\Sigma_{\mathfrak{D}}$ we say that $\mathfrak{D}$ *entails* φ (or φ *follows* from $\mathfrak{D}$), and denote this by $\mathfrak{D} \models \varphi$.

Query Answering in Locally Closed Databases

In this section, we provide the basic tools for reasoning with locally closed databases. Query answering in such databases may be represented as follows:

Definition 10 *Given a locally closed database $\mathfrak{D}$ over Σ, a first-order query $\mathcal{Q}[\bar{x}]$ over Σ (whose free variables are in $\bar{x}$), and a tuple $\bar{t}$ of constants in $dom(\mathfrak{D})$, we say that*

- *$\bar{t}$ is a* certain answer *in $\mathfrak{D}$ for $\mathcal{Q}[\bar{x}]$, if $\mathfrak{D} \models \mathcal{Q}[\bar{t}/\bar{x}]$*
- *$\bar{t}$ is a* possible answer *in $\mathfrak{D}$ for $\mathcal{Q}[\bar{x}]$, if $\mathfrak{D} \cup \mathcal{Q}[\bar{t}/\bar{x}]$ is satisfiable (or, equivalently, if $\mathfrak{D} \not\models \neg\mathcal{Q}[\bar{t}/\bar{x}]$).*

We denote by $Cert_{\mathfrak{D}}(\mathcal{Q}[\bar{x}])$ the set of certain answers of $\mathcal{Q}[\bar{x}]$ in $\mathfrak{D}$ and by $Poss_{\mathfrak{D}}(\mathcal{Q}[\bar{x}])$ the set of possible answers of $\mathcal{Q}[\bar{x}]$ in $\mathfrak{D}$.

An interesting property of a query $\mathcal{Q}[\bar{x}]$ in a locally closed database $\mathfrak{D}$ is whether $\mathfrak{D}$ has complete knowledge on $\mathcal{Q}[\bar{x}]$. This has been defined as follows:

Definition 11 (Levy 1996) *A locally closed database $\mathfrak{D}$ over Σ has* complete world information (CWI) *on a query $\mathcal{Q}[\bar{x}]$, if for each tuple $\bar{t}$ of constants in $dom(\mathfrak{D})$, either $\mathfrak{D} \models \mathcal{Q}[\bar{t}]$ or $\mathfrak{D} \models \neg\mathcal{Q}[\bar{t}]$.*

Next, we focus on the computational complexity of query answering. Following the usual measure of complexity in database systems, the results below are specified in terms of data complexity, that is, in terms of the size $|D|$ of the database instance (assuming that all the rest is fixed). Accordingly, we consider the following decision problems:

$$\mathrm{Poss}_{\mathcal{L}}(\mathcal{Q}[\bar{x}]) = \{(D, \bar{t}) \mid \bar{t} \in Poss_{(D,\mathcal{L})}(\mathcal{Q}[\bar{x}])\},$$
$$\mathrm{Cert}_{\mathcal{L}}(\mathcal{Q}[\bar{x}]) = \{(D, \bar{t}) \mid \bar{t} \in Cert_{(D,\mathcal{L})}(\mathcal{Q}[\bar{x}])\},$$
$$\mathrm{CWI}_{\mathcal{L}}(\mathcal{Q}[\bar{x}]) = \{D \mid (D, \mathcal{L}) \text{ has CWI on } \mathcal{Q}[\bar{x}]\}.$$

As proven in (Cortés-Calabuig et al. 2007), the decision problem $\mathrm{Poss}_{\mathcal{L}}(\mathcal{Q}[\bar{x}])$ is in NP for all $\mathcal{L}$ and $\mathcal{Q}[\bar{x}]$, and is NP-hard for some of them. $\mathrm{Cert}_{\mathcal{L}}(\mathcal{Q}[\bar{x}])$ is in coNP for each $\mathcal{L}$ and $\mathcal{Q}[\bar{x}]$, and is coNP-hard for some of them.

When $\mathfrak{D}$ has complete information about a query, there is no uncertainty about its answers, so such queries are of practical importance. As Proposition 12 shows, queries with CWI can be answered directly in the database instance D, when D is regarded as a two-valued Herbrand structure of $\mathfrak{D}$.[2]

Proposition 12 *A locally closed database $\mathfrak{D}$ has CWI on $\mathcal{Q}[\bar{x}]$ iff $Cert_{\mathfrak{D}}(\mathcal{Q}[\bar{x}]) = Poss_{\mathfrak{D}}(\mathcal{Q}[\bar{x}]) = \{\bar{a} \mid D \models \mathcal{Q}[\bar{a}]\}$.*

Proof. See the appendix. □

Deciding whether there is CWI on a query $\mathcal{Q}[\bar{x}]$ in a *specific* database $\mathfrak{D} = (D, \mathcal{L})$ is not tractable (Cortés-Calabuig et al. 2007). The next proposition shows that the more ambitious problem, of whether there is CWI on $\mathcal{Q}[\bar{x}]$ in *all* locally closed databases containing a fixed set $\mathcal{L}$ of local closed-world assumptions, is not even decidable.

Proposition 13 *The question whether all the locally closed databases $(\cdot, \mathcal{L})$ convey CWI on a query $\mathcal{Q}[\bar{x}]$ is undecidable.*

Proof. See the appendix. □

So far, the results in this section give little reason for optimism regarding practical applicability of local closed-world assumptions. But, as it turns out, in many applications there is no need to have *all* certain answers to a query. Often, it suffices to have a sufficiently large subset of the answers. E.g., if a company searches an (incomplete) database for a provider of some urgently required service, it will be happy if it finds *some* candidate providers; this list does not need to be complete. Likewise, in many applications, it would not harm if the answers to a *possible* query contain a few extra

[2] This was Levy's motivation to study CWI.

"impossible" elements. For instance, if a company wants to advertise one of its services and queries the above database for a group of potential clients, it would not care to receive some additional companies that could not really be possibly interested. Thus, one reasonable strategy to solve the complexity problem would be to develop tractable approximate methods. This is the approach followed in the next section.

The other, more conventional approach to the complexity problem, is to restrict the expressiveness of the language so that efficient query processing is possible. As it turns out, below we obtain such results as well, though in a slightly indirect way: we will show that for certain classes of queries and local closed-world assumptions, the approximate methods are *optimal* in the sense that they compute exactly the certain and possible answers to queries. Thus, these combinations of queries and local closed-world assumptions provide tractable sub-languages.

Approximative Reasoning

Approximations by Three-Valued Structures

The basic idea of approximative reasoning is to compute a 3-valued structure that provides a 'good approximation' of all models of $\mathfrak{D}$ and then to evaluate queries with respect to this structure. The underlying semantics is, therefore, a 3-valued one, where the truth values $\mathcal{THREE} = \{\mathbf{t}, \mathbf{f}, \mathbf{u}\}$ stand for true, false, and unknown (respectively). These values are usually arranged in two orders: the *truth order*, $\leqslant$, which is a linear order given by $\mathbf{f} \leqslant \mathbf{u} \leqslant \mathbf{t}$, and the *precision order*, $\leqslant_p$, which is a partial order on $\mathcal{THREE}$ in which $\mathbf{u}$ is the least element, and $\mathbf{t}$ and $\mathbf{f}$ are incomparable maximal elements. The connectives are defined according to the truth order: Conjunction $\wedge$, disjunction $\vee$ and the negation operator $\neg$ are defined, respectively, by the $\leqslant$-glb, $\leqslant$-lub, and the $\leqslant$-involution (that is, $\neg\mathbf{t} = \mathbf{f}$, $\neg\mathbf{f} = \mathbf{t}$, and $\neg\mathbf{u} = \mathbf{u}$) on $\mathcal{THREE}$.

The notions of 3-valued (Herbrand) structures and (Herbrand) models are defined with respect to $\mathcal{THREE}$ in the standard way. The three-valued Herbrand interpretations of Σ are denoted $\mathfrak{L}^c$ and the subset of two-valued structures is denoted $\mathfrak{L}$. A truth order $\leqslant$ and a precision order $\leqslant_p$ are also definable on $\mathfrak{L}^c$ by pointwise extensions of the corresponding orders in $\mathcal{THREE}$. Clearly, $\leqslant$ is a lattice order and $\leqslant_p$ is a chain-complete order on $\mathfrak{L}^c$.

Truth assignment in the three-valued case is defined through the same recursive rules as two-valued truth assignment (but with respect to different orders). For instance,

$(\psi \wedge \phi)^{\mathcal{K}} = \min_{\leqslant}(\psi^{\mathcal{K}}, \phi^{\mathcal{K}})$;
$(\neg\psi)^{\mathcal{K}} = (\psi^{\mathcal{K}})^{-1}$;
$(\forall x\, \psi[x])^{\mathcal{K}} = \min_{\leqslant}\{(\psi[a])^{\mathcal{K}} \mid a \in Dom(\mathcal{K})\}$;

In what follows we simulate three-valued truth assignments by two-valued truth assignments as follows: given a vocabulary Σ, we introduce for each predicate $P \in \mathcal{R}(\Sigma)$ two new predicate symbols P^c and $P^{c\neg}$ (intuitively standing for 'certainly P' and 'certainly not P', respectively). Denote by Σ' the set of all constant and predicate symbols of Σ together with all the new predicate symbols.

Definition 14 *A 2-valued Σ'-structure I* simulates *a three-valued Σ-structure $\mathcal{K}$, iff $\mathcal{K}$ and I have the same domain, they assign the same interpretations to constant and function symbols, and for each predicate $P \in \mathcal{R}(\Sigma)$ it holds that $(P^c)^I = \{\bar{d} \mid P(\bar{d})^{\mathcal{K}} = \mathbf{t}\}$ and $(P^{c\neg})^I = \{\bar{d} \mid P(\bar{d})^{\mathcal{K}} = \mathbf{f}\}$.*

In the following definition, $P_i^c, P_i^p, P_i^{c\neg}$ and $P_i^{p\neg}$ are symbols representing respectively the certain and the possible tuples of P_i, and the tuples that certainly and possibly do not belong to P_i. Accordingly, Φ^c and Φ^p represent the certain instances and the possible instances of Φ when interpreted as a query. As noted in Proposition 16, these formulas can be used to compute three-valued answers for Φ.

Definition 15 *Given a database vocabulary Σ, we introduce, for each element in $\mathcal{R}(\Sigma) = \{P_1, \ldots, P_n\}$, four new predicate symbols $P_i^c, P_i^p, P_i^{c\neg}$ and $P_i^{p\neg}$ of the same arity as P_i. Now, each formula Φ with predicate symbols amongst $P_1, \ldots, P_n$ is associated with the following two formulas:*

- *Φ^c is the formula obtained by substituting $P_i^c(\bar{t})$ for each positive occurrence[3] of $P_i(\bar{t})$ in Φ, and substituting $\neg P_i^{c\neg}(\bar{t})$ for each negative occurrence of $P_i(\bar{t})$ in Φ.*
- *Φ^p is the formula obtained by substituting $P_i^p(\bar{t})$ for each positive occurrence of $P_i(\bar{t})$ in Φ, and substituting $\neg P_i^{p\neg}(\bar{t})$ for each negative occurrence of $P_i(\bar{t})$ in Φ.*

Note that $(\neg P(\bar{t}))^c = \neg\neg P^{c\neg}(\bar{t}) \equiv P^{c\neg}(\bar{t})$. Also, $P^p(\bar{t})$ and $\neg P^{c\neg}(\bar{t})$ are equivalent and so are $P^{p\neg}(\bar{t})$ and $\neg P^c(\bar{t})$. Moreover, Φ^c contains only positive occurrences of $P_i^c(\bar{t})$ and $P_i^{c\neg}(\bar{t})$. Similarly, Φ^p contains only positive occurrences of $P_i^p(\bar{t})$ and $P_i^{p\neg}(\bar{t})$.

The following proposition is well known.

Proposition 16 *If I simulates $\mathcal{K}$, then for each formula $\varphi[\bar{x}]$ and a suitable tuple of domain elements $\bar{d}$, $\varphi[\bar{d}]^{\mathcal{K}} = \mathbf{t}$ iff $(\varphi[\bar{d}]^c)^I = \mathbf{t}$ and $\varphi[\bar{d}]^{\mathcal{K}} = \mathbf{f}$ iff $((\neg\varphi[\bar{d}])^c)^I = \mathbf{f}$.*

This implies tractability of three-valued truth evaluation and query answering.

Corollary 17 *Given a finite three-valued Σ-structure $\mathcal{K}$, for each formula $\varphi[\bar{x}]$, $\{\bar{d} \mid (\varphi[\bar{d}])^{\mathcal{K}} = \mathbf{t}\}$, $\{\bar{d} \mid (\varphi[\bar{d}])^{\mathcal{K}} = \mathbf{f}\}$, and $\{\bar{d} \mid (\varphi[\bar{d}])^{\mathcal{K}} = \mathbf{u}\}$ can be computed in polynomial time in the size of $\mathcal{K}$.*

We now consider approximation theory. The definitions in the rest of this section and in the next subsection are taken from (Cortés-Calabuig et al. 2006).

Definition 18 *Let Γ be a satisfiable theory based on Σ and containing* $\mathsf{UNA}(\Sigma)$ *and* $\mathsf{DCA}(\Sigma)$. *We say that a 3-valued Herbrand Σ-interpretation $\mathcal{K}$* approximates *Γ (from below), iff for every 2-valued Herbrand model M of Γ, $\mathcal{K} \leqslant_p M$. The* optimal approximation *for Γ is the 3-valued Herbrand structure $\mathcal{O}_\Gamma = glb_{\leqslant_p}\{M \mid M \models \Gamma\}$, where M ranges over all the 2-valued Herbrand models of Γ.*

Note that $\mathcal{O}_\Gamma$ is the most precise among all 3-valued Herbrand Σ-structures approximating Γ and it is well-defined,

[3] A predicate occurs *positively* (alternatively *negatively*) in a formula, iff it appears in the scope of an even (alternatively odd) number of negation symbols.

since the set of Γ's Herbrand models is non-empty and every nonempty set $S \subseteq \mathfrak{L}^c$ has a greatest $\leqslant_p$-lower bound. Now, by the fact that all models of a theory containing $\mathsf{UNA}(\Sigma) \wedge \mathsf{DCA}(\Sigma)$ are isomorphic to Herbrand structures, we have:

Proposition 19 *Let $\mathcal{K}$ be an approximation of Γ. For any sentence φ, if $\varphi^{\mathcal{K}} = \mathbf{t}$, then $\Gamma \models \varphi$ and if $\varphi^{\mathcal{K}} = \mathbf{f}$, then $\Gamma \models \neg\varphi$.*

Definition 20 *For a 3-valued Σ-interpretation $\mathcal{K}$ and a query $\mathcal{Q}[\bar{x}]$ in Σ, define the* certain answers *and the* possible answers *of $\mathcal{Q}[\bar{x}]$ w.r.t. $\mathcal{K}$ by the following sets (respectively):*

- $Cert_{\mathcal{K}}(\mathcal{Q}[\bar{x}]) = \{\bar{a} \mid \mathcal{Q}[\bar{a}]^{\mathcal{K}} = \mathbf{t}\}$,
- $Poss_{\mathcal{K}}(\mathcal{Q}[\bar{x}]) = \{\bar{a} \mid \mathcal{Q}[\bar{a}]^{\mathcal{K}} \geqslant \mathbf{u}\}$.

As computing truth values of sentences is polynomial, we have

Proposition 21 *For each finite three-valued Σ-structure $\mathcal{K}$ and Σ-query $\mathcal{Q}[\bar{x}]$, the sets $Cert_{\mathcal{K}}(\mathcal{Q}[\bar{x}])$ and $Poss_{\mathcal{K}}(\mathcal{Q}[\bar{x}])$ are polynomially computable in the size of $\mathcal{K}$.*

Query Answering by Fixpoint Computations

From Proposition 21 it is clear that a tractable method to compute 3-valued approximations induces a tractable and sound approximative query answering method. Such a method is defined in (Cortés-Calabuig et al. 2006) as follows:

Definition 22 *Given a locally closed database $\mathfrak{D} = (D, \mathcal{L})$, the operator $\mathcal{A}pp_{\mathfrak{D}} : \mathfrak{L}^c \to \mathfrak{L}^c$ maps a three-valued structure $\mathcal{K}$ to a three-valued structure $\mathcal{K}' = \mathcal{A}pp_{\mathfrak{D}}(\mathcal{K})$ such that, for every predicate P of $\mathcal{R}(\Sigma)$ and every tuple $\bar{a}$,*

$$P(\bar{a})^{\mathcal{K}'} = \begin{cases} \mathbf{t} & \text{if } P(\bar{a}) \in D, \\ \mathbf{f} & \text{if there exists } \mathcal{LCWA}(P(\bar{x}), \Psi_P[\bar{x}]) \in \mathcal{L} \\ & \text{so that } \Psi_P[\bar{a}]^{\mathcal{K}} = \mathbf{t} \text{ and } P(\bar{a}) \notin D, \\ \mathbf{u} & \text{otherwise.} \end{cases}$$

The idea here is to start from the structure with total ignorance (i.e., a valuation that assigns $\mathbf{u}$ to every ground atom), and to iterate $\mathcal{A}pp_{\mathfrak{D}}$, thereby gradually extending the definite knowledge using the database and its LCWAs. Clearly, $\mathcal{A}pp_{\mathfrak{D}}$ is $\leqslant_p$-monotone. Thus, by (an extension of) the well-known Knaster-Tarski theorem, we have

Proposition 23 *$\mathcal{A}pp_{\mathfrak{D}}$ is a $\leqslant_p$-monotone operator on the chain complete poset $\mathfrak{L}^c$, thus it has a $\leqslant_p$-least fixpoint.*

Definition 24 *Denote by $\mathcal{C}_{\mathfrak{D}}$ the $\leqslant_p$-least fixpoint of $\mathcal{A}pp_{\mathfrak{D}}$.*

Note 25 As the number of iterations for reaching $\mathcal{C}_{\mathfrak{D}}$ is at most polynomial in the size of the database, and each iteration takes polynomial time in the size of the database, it follows that $\mathcal{C}_{\mathfrak{D}}$ can be computed in polynomial time in $|D|$.

The following proposition shows that $\mathcal{C}_{\mathfrak{D}}$ is a sound approximation of $\mathfrak{D}$.

Proposition 26 *$\mathcal{C}_{\mathfrak{D}}$ approximates $\mathfrak{D}$ and for every $\mathcal{Q}[\bar{x}]$ we have $Cert_{\mathcal{C}_{\mathfrak{D}}}(\mathcal{Q}[\bar{x}]) \subseteq Cert_{\mathfrak{D}}(\mathcal{Q}[\bar{x}]) \subseteq Poss_{\mathfrak{D}}(\mathcal{Q}[\bar{x}]) \subseteq Poss_{\mathcal{C}_{\mathfrak{D}}}(\mathcal{Q}[\bar{x}])$.*

Fixpoint Queries for LCWA

We now come to the main contribution of this paper, namely: computing fixpoint queries for locally closed databases, and analyzing the accuracy of approximative query answering in the context defined above.

A substantial drawback of query answering with $\mathcal{C}_{\mathfrak{D}}$ is the need to recompute it each time the database changes. In what follows we partially avoid this by using fixpoint formulas that symbolically describe the construction of $\mathcal{C}_{\mathfrak{D}}$. Using these expressions, certain or possible answers to queries can be computed by transforming the query into a fixpoint query or a query with respect to some datalog program. This means, in practice, that it suffices to compute the relations that are relevant for the query rather than computing all the relations in $\mathcal{C}_{\mathfrak{D}}$. Moreover, goal directed methods such as magic sets (Bancilhon et al. 1986) or tabling (Swift 1999), will often need only fractions of those relations.

Definition 27 *Let $\mathfrak{D} = (D, \mathcal{L})$ be a locally closed database. For a query $\mathcal{Q}[\bar{x}]$ we introduce two new variables Q^c and $Q^{c\neg}$, the arity of which is the number of free variables of $\mathcal{Q}[\bar{x}]$, and define the set $\Delta_{\mathcal{Q},\mathcal{L}}$ by*

$$\left\{ \begin{array}{l} Q^c(\bar{x}) \leftarrow \mathcal{Q}[\bar{x}]^c \\ Q^{c\neg}(\bar{x}) \leftarrow (\neg\mathcal{Q}[\bar{x}])^c \end{array} \right\} \cup \bigcup \left\{ \begin{array}{l} P_i^c(\bar{x}_i) \leftarrow P_i(\bar{x}_i) \\ P_i^{c\neg}(\bar{x}_i) \leftarrow \neg P_i(\bar{x}_i) \wedge (\Psi_{P_i}[\bar{x}_i])^c \end{array} \right\}$$

where the right union is over the database predicates P_i, and Ψ_{P_i} is the window of expertise of P_i.

Intuitively, Q^c is meant to represent the collection of certain instances of $\mathcal{Q}[\bar{x}]$, and $Q^{c\neg}$ represents the certain instances of $\neg\mathcal{Q}[\bar{x}]$. This is captured by the fixpoint computations on $\Delta_{\mathcal{Q},\mathcal{L}}$ in the logic LFP^{simult}, described below. The fixpoint expressions that are simultaneously computed in this logic are of the form

$$[\mathbf{lfp}_{R_i,\Delta}](\bar{t})$$

for some predicates $\{R_1, \ldots, R_n\}$, where $\bar{t}$ is a tuple of terms whose arity is the same as that of R_i, and Δ is a collection of rules of the form

$$\{R_j(\bar{x}_j) \leftarrow \varphi_j[\bar{x}_j] \mid 1 \leq j \leq n\}.^4$$

Definition 28 *Let Γ_Δ be the standard immediate consequence fixpoint operator on Δ. A fixpoint expression $[\mathbf{lfp}_{R_i,\Delta}](\bar{t})$ is true in a structure $\mathfrak{A}$ and variable assignment ν, if $\bar{t}^{\mathfrak{A},\nu} \in \mathcal{R}_i$, where $\mathcal{R}_i$ is the i'th argument in the least fixpoint $(\mathcal{R}_1, \ldots, \mathcal{R}_n)$ of Γ_Δ, associated to Δ and $\mathfrak{A}$. Now, given a locally closed database $\mathfrak{D} = (D, \mathcal{L})$, define the certain query answer for $\mathcal{Q}[\bar{x}]$ as $[\mathbf{lfp}_{Q^c,\Delta_{\mathcal{Q},\mathcal{L}}}](\bar{x})$ and the possible query answer for $\mathcal{Q}[\bar{x}]$ as $\neg[\mathbf{lfp}_{Q^{c\neg},\Delta_{\mathcal{Q},\mathcal{L}}}](\bar{x})$, where both of these expressions are evaluated in D.*

It is worth noting that Δ is an extended datalog program as defined in (Van Gelder 1993) or a positive definition as defined in FO[ID] (Denecker and Ternovska 2007) and that its semantics, i.e., its least fixpoint, coincides with the well-founded model of Δ. It follows that Q^c is the collection of

[4] Here, each φ_j – the definition of R_j – is a formula over Σ and $\{R_1 \ldots, R_n\}$, where $R_1, \ldots, R_n$ may occur possitively in φ_j, and the length of $\bar{x}_j$ is the same as the arity of R_j.

certain instances of $\mathcal{Q}[\bar{x}]$ and $Q^{c\neg}$ is the collection of certain instances of $\neg\mathcal{Q}[\bar{x}]$. Those instances are represented by $[\mathbf{lfp}_{Q^c,\Delta_{\mathcal{Q},\mathcal{L}}}](\bar{x})$ and $[\mathbf{lfp}_{Q^{c\neg},\Delta_{\mathcal{Q},\mathcal{L}}}](\bar{x})$, respectively.

Example 29 Consider again the local closed-world assumption in Example 5: $\mathcal{LCWA}(\mathsf{Tel}(x,y),\mathsf{Dept}(x,\mathsf{CS}))$. Assume that there is no closure for the relation Dept, that is, $\mathcal{LCWA}(\mathsf{Dept}(x,y),\mathbf{f})$. Let $\mathcal{Q} = \mathsf{Tel}(\mathsf{BD},3962836)$ (as in Example 1). Then:

$$\Delta_{\mathcal{Q},\mathcal{L}} = \left\{ \begin{array}{l} Q^c \leftarrow \mathsf{Tel}^c(\mathsf{BD},3962836). \\ Q^{c\neg} \leftarrow \mathsf{Tel}^{c\neg}(\mathsf{BD},3962836). \\ \mathsf{Tel}^c(x,y) \leftarrow \mathsf{Tel}(x,y). \\ \mathsf{Tel}^{c\neg}(x,y) \leftarrow \neg\mathsf{Tel}(x,y) \wedge \mathsf{Dept}^c(x,\mathsf{CS}). \\ \mathsf{Dept}^c(x,y) \leftarrow \mathsf{Dept}(x,y). \\ \mathsf{Dept}^{c\neg}(x,y) \leftarrow \neg\mathsf{Dept}(x,y) \wedge \mathbf{f}. \end{array} \right\}$$

As $\mathbf{lfp}_{Q^{c\neg},\Delta_{\mathcal{Q},\mathcal{L}}}$ is true in D, $\mathsf{Tel}(\mathsf{BD},3962836)$ is certainly false.

Proposition 30 *Given a locally closed database $(D,\mathcal{L})$ and a query $\mathcal{Q}[\bar{x}]$. Let $(\mathcal{R}^c_{\mathcal{Q}},\mathcal{R}^{c\neg}_{\mathcal{Q}},\mathcal{R}^c_1,\mathcal{R}^{c\neg}_1,\ldots,\mathcal{R}^c_n,\mathcal{R}^{c\neg}_n)$ be the relations defined by $\Delta_{\mathcal{Q},\mathcal{L}}$ in D. Then, for all $1 \leq i \leq n$,*

$$\mathcal{R}^c_i = \{\bar{d} \mid P_i(\bar{d})^{\mathcal{C}_{\mathfrak{D}}} = \mathbf{t}\},\ \ \mathcal{R}^{c\neg}_i = \{\bar{d} \mid P_i(\bar{d})^{\mathcal{C}_{\mathfrak{D}}} = \mathbf{f}\},$$
$$\mathcal{R}^c_{\mathcal{Q}} = \{\bar{d} \mid \mathcal{Q}(\bar{d})^{\mathcal{C}_{\mathfrak{D}}} = \mathbf{t}\},\ \ \mathcal{R}^{c\neg}_{\mathcal{Q}} = \{\bar{d} \mid \mathcal{Q}(\bar{d})^{\mathcal{C}_{\mathfrak{D}}} = \mathbf{f}\}.$$

Proof. See the appendix. □

Note 31 Dually, one may define the set $\Delta'_{\mathcal{Q},\mathcal{L}}$ by

$$\left\{ \begin{array}{l} Q^p(\bar{x}) \leftarrow \mathcal{Q}[\bar{x}]^p \\ Q^{p\neg}(\bar{x}) \leftarrow (\neg\mathcal{Q}[\bar{x}])^p \end{array} \right\} \cup \bigcup \left\{ \begin{array}{l} P^p_i(\bar{x}_i) \leftarrow P_i(\bar{x}_i) \vee (\neg\Psi_{P_i}[\bar{x}_i])^p \\ P^{p\neg}_i(\bar{x}_i) \leftarrow \neg P_i(\bar{x}_i) \end{array} \right\}$$

and then consider the expressions of the greatest fixpoints $\neg[\mathbf{gfp}_{Q^{p\neg},\Delta'_{\mathcal{Q},\mathcal{L}}}](\bar{x})$ and $[\mathbf{gfp}_{Q^p,\Delta'_{\mathcal{Q},\mathcal{L}}}](\bar{x})$ for representing the certain and the possible answers of $\mathcal{Q}[\bar{x}]$, respectively. This follows from the fact that if $(\mathcal{R}_1,\ldots,\mathcal{R}_n)$ is the least fixpoint of Γ_Δ, and $(\mathcal{R}'_1,\ldots,\mathcal{R}'_n)$ is the greatest fixpoint of $\Gamma_{\Delta'}$, the relations $\mathcal{R}_i,\mathcal{R}'_i$ are complements, and so $[\mathbf{lfp}_{R_i,\Delta}](\bar{t})$ and $\neg[\mathbf{gfp}_{R'_i,\Delta'}](\bar{t})$ are logically equivalent.

Note 32 It is interesting to note that for consistent databases, the approximate propagation mechanism, as implemented by the rules above, may be used for imitating common types of integrity constraints in database systems. For instance, in Examples 5 and 29, a primary key constraint on telephone-numbers, that is:

$$\forall xyz(\mathsf{Tel}(x,z) \wedge \mathsf{Tel}(y,z)) \supset x = y,$$

can be expressed as the rewriting rule

$$\mathsf{Tel}^{c\neg}(x,z) \leftarrow \exists y\mathsf{Tel}^c(y,z) \wedge x \neq y.$$

Similarly, a foreign key constraint of the form

$$\forall xz(\mathsf{Tel}(x,z) \supset \exists y\mathsf{Dept}(x,y))$$

can be expressed as the rewriting rule

$$\mathsf{Tel}^{c\neg}(x,y) \leftarrow \forall z\mathsf{Dept}^{c\neg}(x,z).$$

Observe that these extra rules allow us to derive more negative information, e.g., $\mathsf{Tel}^{c\neg}(\mathsf{LD},5845213)$.

The Accuracy of Approximate Query Answering

The results above give us a tractable method for computing possible and certain answers to queries by first computing $\mathcal{C}_{\mathfrak{D}}$ and then evaluating queries against it, using standard database techniques. Tractability, however, has a price. As the next example shows, in certain cases we lose accuracy.

Example 33 Below, we abbreviate the optimal approximation of $\mathfrak{D} = (D,\mathcal{L})$ by $\mathcal{O}_{\mathfrak{D}}$ (instead of $\mathcal{O}_{\mathcal{M}(\mathfrak{D})}$).

1. Let $D = \emptyset$ and $\mathcal{L} = \{\mathcal{LCWA}(P,P)\}$. As the meaning of $\mathfrak{D}$ is equivalent to $\neg P$ in this case, we have that $P^{\mathcal{O}_{\mathfrak{D}}} = \mathbf{f}$. Yet, $P^{\mathcal{C}_{\mathfrak{D}}} = \mathbf{u}$, as P is unknown in the initial step of the constructed approximation of $\mathfrak{D}$, and remains unknown when applying $\mathcal{A}pp_{\mathfrak{D}}$ (see Definition 22).
2. Let $D = \emptyset$ and $\mathcal{L} = \{\mathcal{LCWA}(Q,P\vee\neg P)\}$. This database has models in which P is true and others in which P is false but, because of its LCWA, Q is false in all of them. Thus, $P^{\mathcal{O}_{\mathfrak{D}}} = \mathbf{u}$ and $Q^{\mathcal{O}_{\mathfrak{D}}} = \mathbf{f}$. However, since $P\vee\neg P$ evaluates to $\mathbf{u}$ in each structure $\mathcal{K}$ for which $P^{\mathcal{K}} = \mathbf{u}$, we have that $Q^{\mathcal{C}_{\mathfrak{D}}} = \mathbf{u}$. The answer for the query $\neg Q$ in $\mathcal{C}_{\mathfrak{D}}$ is therefore $\mathbf{u}$, while it is $\mathbf{t}$ with respect to $\mathfrak{D}$ or $\mathcal{O}_{\mathfrak{D}}$.
3. Let $D = \emptyset$ and $\mathcal{L} = \{\mathcal{LCWA}(P,R),\mathcal{LCWA}(Q,R\supset\neg P)\}$. Here, $\mathcal{M}(\mathfrak{D}) = (R \supset \neg P) \wedge ((R \supset \neg P) \supset \neg Q)$, which obviously entails $\neg Q$, and so $Q^{\mathcal{O}_{\mathfrak{D}}} = \mathbf{f}$. The fact that in this case the window of expertise of the second LCWA is exactly the meaning of the first LCWA is not captured by $\mathcal{C}_{\mathfrak{D}}$, and so $Q^{\mathcal{C}_{\mathfrak{D}}} = \mathbf{u}$.

In what follows, we consider some important cases in which accuracy of the approximation method is guaranteed.

As the first item of Example 33 shows, the precision of $\mathcal{C}_{\mathfrak{D}}$ may be lost when cycles exist in the local closed world assumptions. This leads to the next definition.

Definition 34 *The* LCWA dependency graph *of $\mathcal{L}$ is the directed graph on $\mathcal{R}(\Sigma)$, containing a directed edge from predicate P to Q iff there exists $\mathcal{LCWA}(Q(\bar{x}),\Psi[\bar{x}]) \in \mathcal{L}$ such that P occurs in Ψ. A* hierarchically closed database *is a locally closed database $\mathfrak{D} = (D,\mathcal{L})$ in which the LCWA dependency graph of $\mathcal{L}$ is cycle-free.*

We may extend the LCWA-dependency graph with a query $\mathcal{Q}[\bar{x}]$ by adding it as a node and adding edges to it from each predicate P that occurs in $\mathcal{Q}[\bar{x}]$. The transitive closure of the LCWA-dependency graph $\mathcal{G}$ of $\mathfrak{D}$ is denoted $\prec$, and consists of all pairs (P,Q) such that there is a path from P to Q in $\mathcal{G}$. Thus, a hierarchically closed database is one such that $\prec$ is cycle free. Let us consider also the smaller graph containing a directed edge from predicate P to Q iff there exists $\mathcal{LCWA}(Q(\bar{x}),\Psi[\bar{x}]) \in \mathcal{L}$ such that P occurs *negatively* in Ψ, and let us denote its transitive closure by $\prec^-$. As before, we may extend $\prec^-$ for a given query by adding an edge to the query from all predicates that occur negatively in it and by closing the graph under the transitivity rule. We observe that $\prec^-$ is cycle free iff for any query $\mathcal{Q}[\bar{x}]$, the set $\Delta_{\mathcal{Q},\mathcal{L}}$ (and also $\Delta'_{\mathcal{Q},\mathcal{L}}$) is non-recursive. The reflexive closures of $\prec$ and $\prec^-$ are denoted $\preceq$ and $\preceq^-$, respectively.

To study the accuracy of approximate query answering in locally closed databases, we need the following two definitions:

Definition 35 *Let Γ be a consistent theory over Σ containing $\mathsf{DCA}(\Sigma) \wedge \mathsf{UNA}(\Sigma)$, and let HU be the Herbrand universe of Σ. A query $\mathcal{Q}[\bar{x}]$ is* squared *in Γ if for every $\overline{d} \in \mathsf{HU}^n$, if $\Gamma \models \mathcal{Q}[\overline{d}]$ then $\mathcal{Q}[\overline{d}]^{\mathcal{O}_\Gamma} = \mathbf{t}$.*

It is easy to verify that each one of the following conditions is equivalent to the fact that $\mathcal{Q}[\bar{x}]$ is squared in Γ:

- $\forall \bar{d} \in \mathsf{HU}^n$ if $\mathcal{Q}[\overline{d}]^{\mathcal{O}_\Gamma} = \mathbf{u}$ then $\Gamma \cup \{\neg \mathcal{Q}[\overline{d}]\}$ is satisfiable,
- $Cert_{\mathcal{O}_\Gamma}(\mathcal{Q}[\bar{x}]) = Cert_\Gamma(\mathcal{Q}[\bar{x}])$,
- $Poss_{\mathcal{O}_\Gamma}(\neg\mathcal{Q}[\bar{x}]) = Poss_\Gamma(\neg\mathcal{Q}[\bar{x}])$.

It follows that the construction of the optimal approximation $\mathcal{O}_\Gamma$ of Γ allows to compute all the certain answers of queries that are squared in Γ.

In what follows, when $\mathcal{Q}[\bar{x}]$ is squared in a theory $\Gamma = \mathcal{M}(\mathfrak{D})$ (see Definition 9), we shall say that $\mathcal{Q}[\bar{x}]$ is squared in $\mathfrak{D}$.

Definition 36 *For a query $\mathcal{Q}[\bar{x}]$, let σ be the downward closed set $\{R \mid \exists S : S \prec^- \mathcal{Q}[\bar{x}]$ such that $R \preceq S\}$.*

- *A predicate P is* positive free *in $\mathcal{Q}[\bar{x}]$ if $P \notin \sigma$.*
- *We denote by $\mathfrak{D}_\mathcal{Q}$ be the extension of $\mathfrak{D} = (D, \mathcal{L})$, obtained by adding $\mathcal{LCWA}(P(\bar{x}), \mathbf{t})$ to $\mathcal{L}$, for each predicate P that is positive free in $\mathcal{Q}[\bar{x}]$.*

The set σ in the last definition consists of all the predicates on which negatively occurring predicates in $\mathcal{Q}[\bar{x}]$ depend. In practice, this is often a small set in which case most predicates are positive free and $\mathfrak{D}_\mathcal{Q}$ is almost a complete database. This is useful because databases with more base predicates have more squared queries.

Example 37 In Example 5, all the predicates are positive free for $\mathsf{Tel}(x, y)$ and none of them is positive free for $\neg\mathsf{Tel}(x, y)$, as Tel depends on Dept.

Theorem 38 (Completeness) *Let $\mathcal{Q}[\bar{x}]$ be a query that is squared in $\mathfrak{D}_\mathcal{Q}$ and suppose that for every predicate P that occurs negatively in $\mathcal{Q}$, we have that $P^{\mathcal{C}_\mathfrak{D}} = P^{\mathcal{O}_\mathfrak{D}}$. Then $Cert_{\mathcal{C}_\mathfrak{D}}(\mathcal{Q}[\bar{x}]) = Cert_\mathfrak{D}(\mathcal{Q}[\bar{x}])$.*

Proofs of this theorem and of most of the other propositions in this section, are given in the appendix.

Note that if $\mathcal{Q}[\bar{x}]$ is a positive query, i.e., does not contain negative occurrences of predicates, then all predicates are positive free, $\mathfrak{D}_\mathcal{Q}$ is a complete database and has a two-valued $\mathcal{O}_{\mathfrak{D}_\mathcal{Q}}$. Hence, $\mathcal{Q}[\bar{x}]$ is squared in $\mathfrak{D}_\mathcal{Q}$. By Theorem 38, $Cert_{\mathcal{C}_\mathfrak{D}}(\mathcal{Q}[\bar{x}])$ is optimal.

Example 39 Consider again Examples 5 and 29. The query $\mathcal{Q} = \mathsf{Tel}(\mathsf{BD}, 3962836)$ is positive, thus it is squared in $\mathfrak{D}_\mathcal{Q}$, and so Theorem 38 implies that $Cert_{\mathcal{C}_\mathfrak{D}}(\mathcal{Q})$ is optimal in this case.

To get other concrete results from Theorem 38, we need sufficient conditions for the two assumptions under which it can be applied, namely:

1. squaredness of queries, and
2. optimality of predicate approximations.

A) proving squaredness of queries: The propositions below provide syntactic conditions for squared queries.

Proposition 40 *$\mathcal{Q}[\bar{x}]$ is squared in $\mathfrak{D}_\mathcal{Q}$ if it is of the form*

$$\forall \bar{y}(C_1 \vee \cdots \vee C_n),$$

where each C_i is a conjunction in which

1. *each non-literal conjunct C_{ij} is an arbitrary subformula containing only base predicates of $\mathfrak{D}$ and positive free predicates in $\mathcal{Q}[\bar{x}]$,*
2. *each pair C_i, C_j $(i \neq j)$ is mutually exclusive w.r.t. $\mathcal{C}_\mathfrak{D}$.*[5]

This proposition shows that the class of squared queries includes, among others, formulas of the following forms:

1. literals and conjunctions of literals,
2. positive formulas,
3. decision tree-like formulas, in which the test formulas contain only base predicates of $\mathfrak{D}$ and the leaves consist of conjunctions of database literals and formulas containing base predicates and (positive occurrences of) positive free predicates of $\mathcal{Q}[\bar{x}]$.

Example 41 According to Proposition 40, both of the formulas in Example 37 and the query of Example 39 are squared in the database of Example 5.

To define an alternative semantical condition for squaredness, we need the following definition.

Definition 42 *A theory Γ is atomical iff for each 2-valued Herbrand structure I such that $\mathcal{O}_\Gamma \leqslant_p I$, it holds that $I \models \Gamma$.*

Note that every consistent atomical theory Γ that includes $\mathsf{DCA}(\Sigma) \wedge \mathsf{UNA}(\Sigma)$ is equivalent with the first-order theory consisting of $\mathsf{DCA}(\Sigma) \wedge \mathsf{UNA}(\Sigma)$ and the set of all ground literals entailed by Γ.

Proposition 43 *$\mathcal{Q}[\bar{x}]$ is squared in $\mathfrak{D}_\mathcal{Q}$ if $\mathfrak{D}_\mathcal{Q}$ is atomical and $\mathcal{Q}[\bar{x}]$ is a formula such that all the predicates with positive and negative occurrences in $\mathcal{Q}[\bar{x}]$ are two-valued in $\mathcal{C}_\mathfrak{D}$.*

The class of queries covered by this proposition allows arbitrary quantification. Since $\mathfrak{D}_\mathcal{Q}$ is atomical if $\mathcal{Q}[\bar{x}]$ is a positive query, positive queries are covered by this proposition as well.

It can be shown that $\mathfrak{D}_\mathcal{Q}$ is atomical if the database has CWI on the windows of expertise. It appears quite natural that locally closed databases will often meet this condition.

B) proving partial optimality of $\mathcal{C}_\mathfrak{D}$: The second assumption of Theorem 38 requires the optimality of predicate approximations. The next result reduces the problem of proving optimality of $\mathcal{C}_\mathfrak{D}$ in a predicate P to the problem of proving optimality of $\mathcal{C}_\mathfrak{D}$ in predicates Q that occur negatively in Ψ_P.

Proposition 44 *Let $\mathfrak{D}$ be a locally closed database. Then $P^{\mathcal{C}_\mathfrak{D}} = P^{\mathcal{O}_\mathfrak{D}}$ if the following conditions are satisfied:*

[5] Two conjunctions C_1 and C_2 are called *mutually exclusive* with respect to a three-valued structure $\mathcal{K}$, if C_1 has a conjunct whose negation is a conjunct of C_2, and that contains only predicates that are two-valued in $\mathcal{K}$.

- $\Psi_P[\bar{x}]$ *is squared in* $\mathfrak{D}$,
- *for every predicate* Q *such that* $Q \preceq^- P$, P *does not occur positively in* Ψ_Q,
- *for every predicate* Q *that has negative occurrence in* Ψ_P, $Q^{\mathcal{C}_\mathfrak{D}} = Q^{\mathcal{O}_\mathfrak{D}}$.

By induction on the depth of the strict (well-founded) order $\preceq^-$ on $\{Q \mid Q \preceq^- P\}$, using Proposition 44, the following way of showing optimality on P is obtained:

Proposition 45 *Let* $\mathfrak{D}$ *be a locally closed database. Then* $P^{\mathcal{C}_\mathfrak{D}} = P^{\mathcal{O}_\mathfrak{D}}$ *if for every predicate* Q *such that* $Q \preceq^- P$, *it holds that* $Q \not\prec^- Q$, $\Psi_Q[\bar{x}]$ *is squared in* $\mathfrak{D}$, *and* Q *does not occur positively in* Ψ_S *for any* $S \preceq^- Q$.

Note that the conditions of the proposition are satisfied if P's window of expertise is a positive formula. It is also satisfied if for every $Q \preceq^- P$, $\Psi_Q[\bar{x}]$ is squared in $\mathfrak{D}$ and $Q \not\prec Q$, i.e., the LCWA dependency graph does not cycle through Q.

C) summary: By Theorem 38 and Proposition 45 we get the following theorem for the accuracy of approximative query answering.

Theorem 46 (Completeness) *Let* $\mathcal{Q}[\bar{x}]$ *be a query that is squared in* $\mathfrak{D}$ *and suppose that for every predicate* P *that occurs negatively in* $\mathcal{Q}$ *it holds that* $P \not\prec^- P$, $\Psi_P[\bar{x}]$ *is squared in* $\mathfrak{D}$, *and* P *does not occur positively in* Ψ_Q *for any* $Q \preceq^- P$. *Then* $Cert_{\mathcal{C}_\mathfrak{D}}(\mathcal{Q}[\bar{x}]) = Cert_\mathfrak{D}(\mathcal{Q}[\bar{x}])$.

Example 47 Consider again the database instance of Examples 1 and 5, together with the following LCWAs:

- $\mathcal{LCWA}(\mathsf{Tel}(x,y), \mathsf{Dept}(x,\mathsf{CS}))$,
- $\mathcal{LCWA}(\mathsf{Dep}(x,y),\ y = \mathsf{CS} \wedge \exists z \mathsf{Tel}(x,z))$.

The first assumption is considered in Examples 2 and 5; the second assumption expresses complete knowledge of the database about all the people of the computer science department that have a telephone.

This locally closed database is circular, as the window of expertise in each assumption mentions an object predicate of the other assumption. Yet, Theorem 46 is applicable here:

- The windows of expertise are positive formulas, hence the contained predicates are positive free. It follows that both are squared in $\mathfrak{D}$. Also, $\prec^-$ is the empty graph, hence it is acyclic. It follows that $\mathcal{C}_\mathfrak{D}$ is optimal.
- The queries $\mathsf{Tel}(x,y)$ and $\neg\mathsf{Tel}(x,y)$ are literals, hence they are squared. It follows that the approximate methods compute the optimal answers for certain and possible answers of this query.

So let us compute the answers for $\mathsf{Tel}(x,y)$ by our approximate methods, using the following set $\Delta_{\mathcal{Q},\mathcal{L}}$ of certain rules (implicit universal quantification is assumed here):

$$\left\{\begin{array}{l} Q^c(x,y) \leftarrow \mathsf{Tel}^c(x,y). \\ Q^{c\neg}(x,y) \leftarrow \mathsf{Tel}^{c\neg}(x,y). \\ \mathsf{Tel}^c(x,y) \leftarrow \mathsf{Tel}(x,y). \\ \mathsf{Tel}^{c\neg}(x,y) \leftarrow \neg\mathsf{Tel}(x,y) \wedge \mathsf{Dept}^c(x,\mathsf{CS}). \\ \mathsf{Dept}^c(x,y) \leftarrow \mathsf{Dept}(x,y). \\ \mathsf{Dept}^{c\neg}(x,y) \leftarrow \neg\mathsf{Dept}(x,y) \wedge y = \mathsf{CS} \wedge \exists z \mathsf{Tel}^c(x,z). \end{array}\right\}$$

Interestingly, this set of rules is non-recursive. It follows that its least and greatest fixpoint coincide. As expected, the expression $[\mathbf{lfp}_{Q^c,\Delta_{\mathcal{Q},\mathcal{L}}}](x,y)$ for $\mathcal{Q} = \mathsf{Tel}(x,y)$, gives the certain answers

$$\mathsf{Tel}(\mathsf{LD}, 6531421),\ \mathsf{Tel}(\mathsf{LD}, \text{09-23314}),$$
$$\mathsf{Tel}(\mathsf{BD}, 5985625),\ \mathsf{Tel}(\mathsf{TD}, 5845213),$$

which are the database tuples of Tel.

For the certainly false tuples of the query, we unfold the expression $[\mathbf{lfp}_{Q^{c\neg},\Delta_{\mathcal{Q},\mathcal{L}}}](x,y)$, yielding the database query $\neg\mathsf{Tel}(x,y) \wedge \mathsf{Dept}(x,\mathsf{CS})$. Its answer is the following set:

$$\{\mathsf{Tel}(\mathsf{BD}, y) \mid y \neq 5985625\} \cup \{\mathsf{Tel}(\mathsf{TD}, y) \mid y \neq 5845213\}.$$

The set of possible but uncertain answers is the complement of the union of the certain answers and the certainly false answers. This set is specified by the database query $\neg\mathsf{Tel}(x,y) \wedge (\mathsf{Tel}(x,y) \vee \neg\mathsf{Dept}(x,\mathsf{CS}))$, which can be simplified to $\neg\mathsf{Tel}(x,y) \wedge \neg\mathsf{Dept}(x,\mathsf{CS})$. Its answer is the set:

$$\{\mathsf{Tel}(\mathsf{LD}, y) \mid y \neq 6531421 \wedge y \neq \text{09-23314}\} \cup \{\mathsf{Tel}(\mathsf{DF}, y)\}.$$

As noted above, Theorem 46 guarantees that these query answers are precise.

To summarize, the results in this section allow to prove the optimality of the approximate certain answers in the context of queries $\mathcal{Q}[\bar{x}]$ and (a subset of the) windows of expertise in the form of Proposition 40: e.g., positive formulas, or conjunctions of literals, or decision-tree like formulas with base predicates in the tests, etc. Even larger classes of queries can be optimally answered with respect to atomical locally closed databases (Proposition 43). Clearly, this is a rich class of queries and a rich class of databases, for example allowing many forms of cycles in the LCWA-dependency graph. In contrast, the optimality theorem in (Cortés-Calabuig et al. 2007) is for queries in the form of conjunctions of literals and on hierarchically closed databases with conjunctions of literals in the windows of expertise.

We currently lack experience to evaluate the precision of the approximate query answering methods beyond the conditions in the above optimality propositions. On one hand, we have constructed examples showing a drastic loss of precision of the approximate answers. On the other hand, an analysis of the optimality proofs suggests that in many other applications where the conditions do not hold, the approximate methods should still be quite precise.

Related Works

Incompleteness in relational databases has been investigated almost since their inception back in the seventies. This issue is continuously arising whenever a new database paradigm is introduced. The general problem of dealing with incompleteness in relational databases has already been discussed in (Imielinski and Lipski 1981) and in (Grahne 1984). Reiter (1986) provides an early semantic characterization of a database containing null values, and defines a sound algorithm for querying such databases. At the representation level, Motro (1989) is perhaps the first to introduce a language – similar in spirit to ours – to represent partial completeness using logical views. This is followed by

the works of Levy (1996), Etzioni et al. (1997), and Doherty et al. (2000). Incompleteness of databases is also addressed in (Grahne 2002) from a data integration perspective. In (Grahne and Mendelson 1999) a framework for dealing with incompleteness in mediator-based systems is introduced, based on tableaux techniques for query answering. Incompleteness in data exchange has recently been considered in (Libkin 2006).

Evidently, the concept of a locally closed database has strong ties also to non-monotonic reasoning, so there is no wonder that it can be expressed by a variety of non-monotonic formalisms. A detailed analysis on the relationship between the LCWA and circumscription is provided in (Cortés-Calabuig et al. 2005) and a discussion on how to express LCWAs using (general) logic programs as presented in (Gelfond et al. 1991) is given in (Cortés-Calabuig et al. 2006). The idea of using logic programs for querying incomplete databases is also investigated in (Baral et al. 1998) and (Bonatti et al. 1996). In the former, incompleteness is expressed by extending a relational database to a set of literals, and it is shown how queries are expanded from complete databases to be applicable to incomplete databases, using general logic programs. In (Bonatti et al. 1996) incomplete information is represented by disjunctive databases, and query answering is defined by providing semantics to such databases using an extension of logic programs that is based on autoepistemic logic.

Conclusions

The ability to correctly and efficiently reason with partially complete databases is a major goal whose importance is obvious. However, as we have shown, the corresponding decisions problems are not tractable and sometimes are even not decidable. To handle this, we introduced a rewriting technique to compute certain or possible answers in polynomial time from such databases. Our approach is based on approximating all models of the database's theory by means of three-valued structures, which are implicitly represented by fixpoint queries. We have shown that this approach is applicable on hierarchically closed databases, an extended notion of the one introduced in (Cortés-Calabuig et al. 2007). Moreover, in this new setting, certain types of important integrity constraints can be incorporated in the framework.

We have also provided a 'toolbox' to prove optimality of the approximate query answering methods for quite a broad class of queries and databases. Our results suggest that the approximate reasoning methods may often be quite precise, and frequently optimal. We believe that, in practice, the approximate methods may offer very often a high degree of accuracy, even when applied beyond the conditions that guarantee optimality.

Finally, we view the current work as a further step towards the more ambitious goal of providing a unifying framework for efficient approximative query answering in incomplete databases in the presence of arbitrary integrity constraints. Towards this goal, efficient reasoning methods developed for first order logic theories as the one recently investigated in (Wittocx et al. 2008), seem a promising path to explore. In the other direction, we believe the results concerning approximative reasoning presented in this work and the way they were obtained can be adapted to other contexts in which efficient query answering is imperative, such as OWL ontologies or databases of a less structured nature (XML).

Acknowledgments

This work was in part funded by the projects GOA/2003/08 "Inductive Knowledge Bases" and G.0357.06 "Design, implementation and application of model generation techniques for ID-logic", Research Foundation - Flanders (FWO-Vlaanderen).

References

Bancilhon, F.; Maier, D.; Sagiv, Y.; and Ullman, J. 1986. Magic sets and other strange ways to implement logic programs. In *Proc. 5th PODS*, 1–15. ACM Press.

Baral, C.; Gelfond, M.; and Kosheleva, O. 1998. Expanding queries to incomplete databases by interpolating general logic programs. *J. Log. Program.* 35(3):195–230.

Bonatti, P.; and Eiter, T. 1996. Querying disjunctive databases through nonmonotonic logics. *Theor. Comput. Sci.* 160(1–2):321–363.

Cortés-Calabuig, A.; Denecker, M.; Arieli, O.; Van Nuffelen, B.; and Bruynooghe, M. 2005. On the local closed-world assumption of data-sources. In *Proc. 8th LPNMR*, LNCS 3662, 145–157. Springer.

Cortés-Calabuig, A.; Denecker, M.; Arieli, O.; and Bruynooghe, M. 2006. Representation of partial knowledge and query answering in locally complete databases. In *Proc. 13th LPAR*, LNCS 4246, 407–421. Springer.

Cortés-Calabuig, A.; Denecker, M.; Arieli, O.; and Bruynooghe, M. 2007. Approximate query answering in locally closed databases. In *Proc. 22nd AAAI*, 397–402. AAAI Press.

Denecker, M.; and Ternovska, E. 2007. Inductive situation calculus. *Artificial Intelligence* 171(5-6):332–360.

Doherty, P.; Łukaszewicz, W.; and Szalas, A. 2000. Efficient reasoning using the local closed-world assumption. In *Proc. 9th AIMSA*, LNCS 2407, 49–58. Springer.

Etzioni, O.; Golden, K.; and Weld, D. 1997. Sound and efficient closed-world reasoning for planning. *Artificial Intelligence* 89(1–2):113–148.

Gelfond, M.; Lifschitz, V. 1991. Classical negation in logic programs and disjunctive databases. *New Generation Computing* 9:367–387.

Grahne, G. 1984. Dependency satisfaction in databases with incomplete information. In *Proc. 10th VLDB*, 37–45. Morgan Kaufmann.

Grahne, G. 2002. Information integration and incomplete information. *IEEE Data Engineering Bulletin* 25(3):46–52.

Grahne, G., Mendelzon A. 1999. Tableau techniques for querying information sources through global schemas. In *Proc. 7th ICDT*, LNCS 1540, 332–347. Springer.

Imielinski, T.; and Lipski Jr, W. 1981. On representing incomplete information in a relational data base. In *Proc. 7th VLDB*, 388–397. IEEE Press.

Levy, A. 1996. Obtaining complete answers from incomplete databases. In *Proc. 22nd VLDB*, 402–412. Morgan Kaufmann.

Libkin, L. 2006. Data exchange and incomplete information. In *Proc. 25th PODS*, 60–69. ACM Press.

Motro, A. 1989. Integrity = Validity + Completeness. *ACM Trans. Database Syst.* 14(4):480–502.

Reiter, R. 1982. Towards a logical reconstruction of relational database theory. In *On Conceptual Modelling (Intervale Workshop)*, 191–233.

Reiter, R. 1986. A sound and sometimes complete query evaluation algorithm for relational databases with null values. *J. ACM* 33(2):349–370.

Swift, T. 1999. Tabling for non-monotonic programming. *Annals of Mathematics and Artificial Intelligence* 25(3-4):201–240.

Trakhtenbrot, B. 1963. Impossibility of an algorithm for the decision problem in finite classes. *American Mathematical Society Transaction* 3(2):1–5.

Van Gelder, A. 1993. The alternating fixpoint of logic programs with negation. *Journal of Computer and System Sciences* 47(1):185–221.

Wittocx, J.; Mariën, M.; and Denecker, M. 2008. Grounding with bounds. In *Proc. 23rd AAAI*, 572–577. AAAI Press.

Appendix: Proofs

Proof of Proposition 12. Obviously, when $\mathfrak{D}$ has complete information about $\mathcal{Q}[\bar{x}]$, certain and possible answers coincide, i.e., $Cert_{\mathfrak{D}}(\mathcal{Q}[\bar{x}]) = Poss_{\mathfrak{D}}(\mathcal{Q}[\bar{x}])$. Thus, since D is a model of $\mathfrak{D}$, we have: $\{\bar{a} \mid D \models \mathcal{Q}[\bar{a}]\} \subseteq Poss_{\mathfrak{D}}(\mathcal{Q}[\bar{x}]) = Cert_{\mathfrak{D}}(\mathcal{Q}[\bar{x}]) \subseteq \{\bar{a} \mid D \models \mathcal{Q}[\bar{a}]\}$. Conversely, if $Cert_{\mathfrak{D}}(\mathcal{Q}[\bar{x}]) = Poss_{\mathfrak{D}}(\mathcal{Q}[\bar{x}])$, then obviously either $\mathfrak{D} \models \mathcal{Q}[\bar{x}]$ or $\mathfrak{D} \models \neg\mathcal{Q}[\bar{x}]$, i.e. there is CWI on $\mathcal{Q}[\bar{x}]$. □

Proof of Proposition 13. Consider the query $\mathcal{Q}[\bar{x}] = P(c)$ and $\mathcal{L} = \{\mathcal{LCWA}(P(c), \varphi)\}$, where φ is a sentence not containing P. We observe that a database $(D, \mathcal{L})$ has no CWI on $P(c)$ iff $\neg\varphi$ has a finite model. It follows that there is CWI on $P(c)$ in all databases $(D, \mathcal{L})$ iff φ is satisfied in all finite structures. This is a validity checking problem of a first-order formula with respect to the class of finite structures. By Trakhtenbrot's theorem (Trakhtenbrot 1963), this problem is undecidable. □

Proof outline of Proposition 30. By induction on the number of iterations in the computations of the operators $\mathcal{A}pp_{\mathfrak{D}}$ (Definition 22) and $\Gamma_{\Delta_{\mathcal{Q},\mathcal{L}}}$ (Definition 28), one shows that the structure that is obtained by the latter in a certain iteration simulates (in the sense of Definition 14) the structure that is obtained by the former in the same iteration. Suppose now that $\mathcal{A}pp_{\mathfrak{D}}$ reaches a fixpoint after α iterations. This fixpoint is $\mathcal{C}_{\mathfrak{D}}$. The structure I_α obtained by $\Gamma_{\Delta_{\mathcal{Q},\mathcal{L}}}$ at this iteration simulates $\mathcal{C}_{\mathfrak{D}}$, and it is a fixpoint on all the predicates P_i^c and $P_i^{c\neg}$. After one more iteration $\Gamma_{\Delta_{\mathcal{Q},\mathcal{L}}}$ reaches a fixpoint also on the predicates Q^c and $Q^{c\neg}$. By Proposition 16, it holds that $\bar{d} \in \mathcal{R}^c_{\mathcal{Q}}$ iff $(\mathcal{Q}[\bar{d}]^c)^{I_\alpha} = \mathbf{t}$ iff $(\mathcal{Q}[\bar{d}])^{\mathcal{C}_{\mathfrak{D}}} = \mathbf{t}$. Likewise, $\bar{d} \in \mathcal{R}^{c\neg}_{\mathcal{Q}}$ iff $(\mathcal{Q}[\bar{d}])^{\mathcal{C}_{\mathfrak{D}}} = \mathbf{f}$. □

Proof of Theorem 38. First, we show the following lemmas:

Lemma 48 *Let $\mathcal{K}$ be an approximation of $\mathfrak{D}$ such $P(\bar{a})^{\mathcal{K}} = \mathbf{t}$ iff $P(\bar{a}) \in D$. Let $\mathcal{Q}[\bar{x}]$ be a query squared in $\mathfrak{D}$ such that for every database predicate P that occurs negatively in $\mathcal{Q}$, $P^{\mathcal{K}} = P^{\mathcal{O}_{\mathfrak{D}}}$. It holds that $Cert_{\mathfrak{D}}(\mathcal{Q}[\bar{x}]) = Cert_{\mathcal{K}}(\mathcal{Q}[\bar{x}])$.*

Proof. We have that $Cert_{\mathcal{K}}(\mathcal{Q}[\bar{x}]) \subseteq Cert_{\mathfrak{D}}(\mathcal{Q}[\bar{x}])$. To show the other direction, assume that $\mathfrak{D} \models \mathcal{Q}[\bar{d}]$. Since $\mathcal{Q}[\bar{x}]$ is squared in $\mathfrak{D}$, it follows that $\mathcal{Q}[\bar{d}]^{\mathcal{O}_{\mathfrak{D}}} = \mathbf{t}$. For every atom $P(\bar{a})$ of a predicate that occurs in $\mathcal{Q}[\bar{x}]$, if $P(\bar{a})^{\mathcal{K}} \neq P(\bar{a})^{\mathcal{O}_{\mathfrak{D}}}$, then P occurs only positively in $\mathcal{Q}[\bar{x}]$ and, since $\mathcal{K} \leqslant_p \mathcal{O}_{\mathfrak{D}}$, it holds that $P(\bar{a})^{\mathcal{K}} = \mathbf{u}$ and $P(\bar{a})^{\mathcal{O}_{\mathfrak{D}}} \neq \mathbf{u}$. By the assumption we made about $\mathcal{K}$, it holds that $P(\bar{a})^{\mathcal{O}_{\mathfrak{D}}} = \mathbf{t}$ iff $P(\bar{a}) \in D$ iff $P(\bar{a})^{\mathcal{K}} = \mathbf{t}$. Hence, if $P(\bar{a})^{\mathcal{K}} \neq P(\bar{a})^{\mathcal{O}_{\mathfrak{D}}}$ then $P(\bar{a})^{\mathcal{K}} = \mathbf{u}$ and $P(\bar{a})^{\mathcal{O}_{\mathfrak{D}}} = \mathbf{f}$. It follows from a well-known monotonicity property of the standard three-valued truth assignment, that $\mathcal{Q}[\bar{d}]^{\mathcal{K}} \geq \mathcal{Q}[\bar{d}]^{\mathcal{O}_{\mathfrak{D}}} = \mathbf{t}$.

Lemma 49 *If $\mathcal{Q}[\bar{x}]$ is squared in $\mathfrak{D}_{\mathcal{Q}}$, then $\mathcal{Q}[\bar{x}]$ is squared in $\mathfrak{D}$.*

Proof. Note that the models of $\mathfrak{D}_{\mathcal{Q}}$ correspond exactly to the models M' of $\mathfrak{D}$ such that $P^{M'} = P^D$, for every positive free predicate P of $\mathcal{Q}[\bar{x}]$. Also, for every model M of $\mathfrak{D}$, the structure M' obtained by setting $P^{M'} = P^D$ if P is positive free and $P^{M'} = P^M$ otherwise, is a model of $\mathfrak{D}_{\mathcal{Q}}$. It follows that $\mathcal{Q}[\bar{d}]^{M'} \leq \mathcal{Q}[\bar{d}]^M$, as for all predicates P that occur negatively in $\mathcal{Q}[\bar{x}]$, it holds that $P^{M'} = P^M$ while predicates P with only positive occurrences have $P^{M'} \subseteq P^M$. Thus, $\mathfrak{D} \models \mathcal{Q}[\bar{d}]$ iff $\mathfrak{D}_{\mathcal{Q}} \models \mathcal{Q}[\bar{d}]$, and so $Cert_{\mathfrak{D}_{\mathcal{Q}}}(\mathcal{Q}[\bar{x}]) = Cert_{\mathfrak{D}}(\mathcal{Q}[\bar{x}])$.

Observe, also, that $\mathcal{O}_{\mathfrak{D}}$ is an approximation of $\mathfrak{D}_{\mathcal{Q}}$ that satisfies the condition on $\mathcal{K}$ in Lemma 48. It follows that $Cert_{\mathcal{O}_{\mathfrak{D}}}(\mathcal{Q}[\bar{x}]) = Cert_{\mathcal{O}_{\mathfrak{D}_{\mathcal{Q}}}}(\mathcal{Q}[\bar{x}])$ and so, since $\mathcal{Q}[\bar{x}]$ is squared in $\mathfrak{D}_{\mathcal{Q}}$, $Cert_{\mathcal{O}_{\mathfrak{D}}}(\mathcal{Q}[\bar{x}]) = Cert_{\mathfrak{D}_{\mathcal{Q}}}(\mathcal{Q}[\bar{x}])$. By what we have obtained above, it follows that $Cert_{\mathcal{O}_{\mathfrak{D}}}(\mathcal{Q}[\bar{x}]) = Cert_{\mathfrak{D}}(\mathcal{Q}[\bar{x}])$, which implies that $\mathcal{Q}[\bar{x}]$ is squared in $\mathfrak{D}$.

To compete the proof of Theorem 38, let $\mathcal{Q}[\bar{x}]$ be a query that is squared in $\mathfrak{D}_{\mathcal{Q}}$. By Lemma 49, $\mathcal{Q}[\bar{x}]$ is squared in $\mathfrak{D}$, and so by Lemma 48 for $\mathcal{K} = \mathcal{C}_{\mathfrak{D}}$ the theorem is obtained. □

We now turn to Propositions 40 and 43 that identify conditions for squardeness. For this, we need the following definitions:

Definition 50 *The* supervaluation $sv_{\mathcal{K}}(\varphi)$ *of a sentence φ with respect to a three-valued structure $\mathcal{K}$ is defined as*

$$sv_{\mathcal{K}}(\varphi) = lub_{\leqslant_p}\{\varphi^M \mid \mathcal{K} \leqslant_p M\}.$$

where M ranges over two-valued structures.

Definition 51 *A query $\mathcal{Q}[\bar{x}]$ is* literal-based *in a satisfiable theory Γ containing $DCA \wedge UNA$ iff*

$$Cert_\Gamma(\mathcal{Q}[\bar{x}]) = \{\bar{d} \in \mathsf{HU}^n \mid sv_{\mathcal{O}_\Gamma}(\mathcal{Q}[\bar{d}]) = \mathbf{t}\}.$$

$\mathcal{Q}[\bar{x}]$ is Kleene-precise *in a three-valued structure $\mathcal{K}$ with domain Dom iff*

$$Cert_\mathcal{K}(\mathcal{Q}[\bar{x}]) = \{\bar{d} \in Dom^n \mid sv_\mathcal{K}(\mathcal{Q}[\bar{d}]) = \mathbf{t}\}.$$

Again, in the sequel, if $\mathcal{Q}[\bar{x}]$ is literal-based in a theory $\Gamma = \mathcal{M}(\mathfrak{D})$ (see Definition 9), we shall sometimes say that $\mathcal{Q}[\bar{x}]$ is literal-based in $\mathfrak{D}$.

Note that since a supervaluation is more precise than standard Kleene truth assignment, it follows that the certain answers of a Kleene-precise query under standard Kleene truth assignment coincides with the supervaluation.

Lemma 52 *A query $\mathcal{Q}[\bar{x}]$ is squared in Γ iff $\mathcal{Q}[\bar{x}]$ is literal-based in Γ and Kleene-precise in $\mathcal{O}_\Gamma$.*

Proof. For every Γ and $\mathcal{Q}[\bar{x}]$, the following inequalities hold:

$$\begin{array}{rcl} Cert_{\mathcal{O}_\Gamma}(\mathcal{Q}[\bar{x}]) & \subseteq & \{\bar{d} \in \mathsf{HU}^n \mid sv_{\mathcal{O}_\Gamma}(\mathcal{Q}[\bar{d}]) = \mathbf{t}\} \\ & \subseteq & Cert_\Gamma(\mathcal{Q}[\bar{x}]). \end{array}$$

If $\mathcal{Q}[\bar{x}]$ is literal-based in Γ and Kleene-precise in $\mathcal{O}_\Gamma$, the inequalities turn into equalities. Vice versa, if $\mathcal{Q}[\bar{x}]$ is squared, the three terms are equal, and it follows that the query is literal-based and Kleene-precise. □

By Lemma 52, then, for showing Propositions 40 and 43, it is enough to show that $\mathcal{Q}[\bar{x}]$ is literal-based in $\mathfrak{D}_\mathcal{Q}$ and is Kleene-precise in $\mathcal{O}_{\mathfrak{D}_\mathcal{Q}}$.

Proof of Proposition 40. Let $\Gamma = \mathcal{M}(\mathfrak{D}_\mathcal{Q})$. Since Γ contains $\mathsf{DCA}(\Sigma) \wedge \mathsf{UNA}(\Sigma)$, the formula $\forall x \varphi[\bar{x}]$ will be literal-based in Γ if $\varphi[\bar{x}]$ is literal-based in Γ. Hence, it suffices to prove the proposition for the case that $\mathcal{Q}[\bar{x}]$ is quantifier free.

Now, given that all predicates of non-literal conjuncts C_{ij} are two-valued in $\mathcal{O}_\Gamma$, it is obvious that $\Gamma \models C_{ij}[\bar{d}]$ iff $sv_{\mathcal{O}_\Gamma}(C_{ij}[\bar{d}]) = \mathbf{t}$, iff $C_{ij}[\bar{d}]^{\mathcal{O}_\Gamma} = \mathbf{t}$ for each $\bar{d} \in \mathsf{HU}^n$.

Since each pair $C_i[\bar{x}], C_j[\bar{x}]$ of $\mathcal{Q}[\bar{x}]$ is mutually exclusive in $\mathcal{O}_\Gamma$, it follows that for all $\bar{d}$, there is at most one conjunct C_i such that $\Gamma \not\models \neg C_{ij}[\bar{d}]$. It is then easy to verify that

$$\begin{array}{ll} \Gamma \models C_i[\bar{d}] & \text{iff } \Gamma \models C_{ij}[\bar{d}], \text{ for each conjunct } C_{ij} \text{ of } C_i, \\ & \text{iff } sv_{\mathcal{O}_\Gamma}(C_{ij}[\bar{d}]) = \mathbf{t} \text{ for each conjunct } C_{ij} \\ & \text{iff } (C_{ij}[\bar{d}])^{\mathcal{O}_\Gamma} = \mathbf{t} \text{ for each conjunct } C_{ij} \\ & \text{iff } sv_{\mathcal{O}_\Gamma}(C_i[\bar{d}]) = \mathbf{t} \\ & \text{iff } (C_i[\bar{d}])^{\mathcal{O}_\Gamma} = \mathbf{t}. \end{array}$$

Thus $\mathcal{Q}[\bar{x}]$ is literal-based in Γ and Kleene-precise in $\mathcal{O}_\Gamma$. □

Proof outline of Proposition 43. As $\mathfrak{D}_\mathcal{Q}$ is atomical, every $\mathcal{Q}[\bar{x}]$ is literal-based in it. By Lemma 52 it remains to show that $\mathcal{Q}[\bar{x}]$ is Kleene-precise in $\mathcal{O}_{\mathfrak{D}_\mathcal{Q}}$. All we need to do is to show that if $\mathcal{Q}[\bar{d}]^{\mathcal{O}_{\mathfrak{D}_\mathcal{Q}}} = \mathbf{u}$ then $sv_{\mathcal{O}_{\mathfrak{D}_\mathcal{Q}}}(\mathcal{Q}[\bar{d}]) = \mathbf{u}$. We construct two-valued structures K and K' from $\mathcal{O}_{\mathfrak{D}_\mathcal{Q}}$ where K maps each unknown atom $P(\bar{a})$ of $\mathcal{O}_{\mathfrak{D}_\mathcal{Q}}$ to $\mathbf{t}$ if P occurs positively in φ and to $\mathbf{f}$ otherwise; likewise, K' follows the inverse strategy and maps unknown atoms $P(\bar{a})$ to $\mathbf{f}$ if P occurs positively in φ and to $\mathbf{t}$ otherwise. By a routine induction, one shows that if $\mathcal{Q}[\bar{d}]^{\mathcal{O}_{\mathfrak{D}_\mathcal{Q}}} = \mathbf{u}$, then $\mathcal{Q}[\bar{d}]^K = \mathbf{t}$ and $\mathcal{Q}[\bar{d}]^{K'} = \mathbf{f}$, and hence, $sv_{\mathcal{O}_{\mathfrak{D}_\mathcal{Q}}}(\mathcal{Q}[\bar{d}]) = \mathbf{u}$. □

Proof of Proposition 44. By the soundness of $\mathcal{C}_\mathfrak{D}$, it holds for every $\bar{a} \in Dom^n$, that if $P(\bar{a})^{\mathcal{C}_\mathfrak{D}} \neq \mathbf{u}$, then $P(\bar{a})^{\mathcal{C}_\mathfrak{D}} = P(\bar{a})^{\mathcal{O}_\mathfrak{D}}$. So, let us assume that $P(\bar{a})^{\mathcal{C}_\mathfrak{D}} = \mathbf{u}$. Observe that, in this case, $P(\bar{a}) \notin D$.

To show that $P(\bar{a})^{\mathcal{O}_\mathfrak{D}} = \mathbf{u}$, we need to construct two models M, M' of $\mathfrak{D}$ such that $M \models P(\bar{a})$ and $M' \models \neg P(\bar{a})$. Since $P(\bar{a}) \notin D$, we can take $M' = D$ which is indeed a model of $\mathfrak{D}$. Let us now construct M. By construction of $\mathcal{C}_\mathfrak{D}$, it holds that $P(\bar{a}) \notin D$ and $\Psi_P[\bar{a}]^{\mathcal{C}_\mathfrak{D}} \neq \mathbf{t}$. The sentence $\Psi_P[\bar{a}]$ is squared in $\mathfrak{D}$ and by assumption, it holds that $Q^{\mathcal{C}_\mathfrak{D}} = Q^{\mathcal{O}_\mathfrak{D}}$ for each negatively occurring predicate Q in this formula. Therefore, the conditions of Proposition 48 are satisfied, so there is a model N of $\mathfrak{D}$ such that $N \models \neg\Psi_P[\bar{a}]$. If $N \models \neg P(\bar{a})$, then we can take $M = N$. Otherwise, we transform N into a model M in which $P(\bar{a})$ is false.

Consider the set $S_P = \{Q \in \mathcal{R}(\Sigma) |$ for some $Q' \preceq^- Q$, $P \in^+ \Psi_{Q'}\}$. It consists of all predicates in whose window of expertise P has a positive occurrence, and all predicates that negatively depend on these. By the condition of the proposition, it holds that $P \notin S_P$.

Define M as the structure obtained by modifying N as follows:

- $P^M = P^N \cup \{\bar{a}\}$, i.e., $P(\bar{a})$ is made true;
- $Q^M = Q^D$ for $Q \in S_P$.

This modification increases P and decreases all predicates of S_P; i.e., $P^N \leq P^M$, $Q^M \leq Q^N$ for $Q \in S_P$, and $Q^N = Q^M$ otherwise. Thus, formulas with only positive occurrences of P and only negative occurrences of predicates $Q \in S_P$ have a larger truth value in M than in N.

To verify that M is a model of $\mathfrak{D}$, it suffices to check that M satisfies all local closed world assumptions. Consider any instance of a local closed world assumption:

$$\varphi \equiv \neg\Psi_Q[\bar{d}] \vee \neg Q(\bar{d}) \vee (Q(\bar{d}) \in D)$$

Each of these formulas is satisfied in N. Let us verify that it is satisfied in M as well. There are four cases:

- $Q = P$ and $\bar{d} = \bar{a}$: in this case, $N \models \neg\Psi_P[\bar{a}]$. The formula $\neg\Psi_P[\bar{a}]$ contains only positive occurrences of P and only negative occurrences of predicates $Q \in S_P$, hence $\mathbf{t} = (\neg\Psi_P[\bar{a}])^N \leq (\neg\Psi_P[\bar{a}])^M$.
- $Q = P$ and $\bar{d} \neq \bar{a}$: we have $(\neg P(\bar{d}) \vee (P(\bar{d}) \in D))^N = (\neg P(\bar{d}) \vee (P(\bar{d}) \in D))^M$ and $\neg\Psi_P[\bar{a}]$ contains only positive occurrences of P and only negative occurrences of predicates $Q \in S_P$. It follows that $\mathbf{t} = \varphi^N \leq \varphi^M$.
- $Q \in S_P$: M satisfies $\neg Q(\bar{d}) \vee (Q(\bar{d}) \in D)$.
- $Q \notin S_P$ and $Q \neq P$: φ contains only positive occurrences of P and only negative occurrences of predicates of S_Q; hence $\mathbf{t} = \varphi^N \leq \varphi^M$. □