Efficient Fixpoint Methods for Approximate Query Answering in Locally Complete Databases

Álvaro Cortés-Calabuig[1], Marc Denecker[1], Ofer Arieli[2], Maurice Bruynooghe[1]

[1] Department of Computer Science, Katholieke Universiteit Leuven, Belgium
{alvaro,marc.denecker,maurice.bruynooghe}@cs.kuleuven.be
[2] 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 complete 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 not tractable. In this paper we develop efficient *approximate* methods for query answering, based on fixpoint computations. We present preliminary results for a broad class of locally closed databases in which this method produces complete answers to queries.

1 Introduction

The *Closed-World Assumption* (CWA) on databases [9] 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

Deparment

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

Assume that in this case the database is complete with respect to all department members, but it is not complete regarding external collaborators. Thus, appropriate answers for $\mathsf{Tel}(\text{Bart Delvaux}, 3962836)$ and $\mathsf{Tel}(\text{Leen Desmet}, 3212445)$ are 'no' and 'unknown', respectively. If completeness of the database is taken for granted, then the answer for 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 should be 'unknown'.

Not surprisingly, however, query answering in locally closed databases turns out to be intractable in general. This provides the motivation for developing efficient *approximate methods* for query answering based on fixpoint semantics in the context of three-valued logic. 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 is not always needed.

The current work builds upon [4], which in turn relies on a extension of the formalism of Levy [8] for representing partially complete information in database systems. In [4], a polynomial algorithm for querying a subclass of locally closed databases – called hierarchically closed – was introduced. This algorithm was based on implicit approximations of all the models of a locally closed database using three-valued structures. In this paper, we use similar approximation techniques but we generalize the work in [4] by describing a polynomial fixpoint procedure that computes answers from a more general – and at the same time useful – set of locally closed databases. For a large class of queries and locally closed databases the new algorithm retrieves correct and complete answers. We show, moreover, that in the general case deciding complete world information, i.e. whether possible and certain answers coincide, is undecidable.

This paper is organized as follows. In Section 2 we recall the basic concepts, properties and semantics of locally closed databases as introduced in [8, 3–5]. In Section 3 we survey some previous results regarding intractability of query answering in locally closed databases and show a new undecidability result concerning the closed-world information. This motivates the approximative approach, for query answering, presented in Section 4. In Section 5 we conclude.

2 Representing Incompleteness in Database Systems

In this section, we recall the concepts of the LCWA as introduced in [3–5]. We denote by Σ a finite first-order vocabulary, consisting of sets $\mathcal{R}(\Sigma)$ of predicate symbols and $\mathcal{C}(\Sigma)$ of constants. First-order formulas over Σ are constructed as usual. $\Psi[\bar{x}]$ denotes a formula with free variables that are a subset of $\bar{x}$. Interpretations for Σ (Σ-structures) are 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.

Definition 1. *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 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 2 ([5]). *A* local closed-world assumption *(LCWA) is an expression of the form $\mathcal{LCWA}(P(\bar{x}), \Psi[\bar{x}])$, 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.*

The intuitive reading of the expression in Definition 2 is "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 Ψ. For this reason we call Ψ a *window of expertise* of the predicate P.

Note 1. This definition is based on the notion of Levy's local completeness statements [8]. 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 3 ([5]). *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.*

We denote by $dom(\mathfrak{D})$ the active domain of a locally closed database $\mathfrak{D} = (D, \mathcal{L})$. That is, $dom(\mathfrak{D})$ is the finite set consisting of all the constants in D and $\mathcal{L}$. We define $\Sigma_{\mathfrak{D}}$ as the extension of Σ such that $\mathcal{R}(\Sigma_{\mathfrak{D}}) = \mathcal{R}(\Sigma)$ and $\mathcal{C}(\Sigma_{\mathfrak{D}}) = dom(\mathfrak{D})$.

Example 2. Abbreviate the database of Example 1 as follows:

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

Some examples of local closed-world assumptions for this database are the following:

1. $\mathcal{LCWA}(\mathsf{Tel}(x, y), \mathsf{Dept}(x, \mathsf{CS}))$ states that all the telephone numbers of the computer science department members are known and occur in the database. 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.
2. $\mathcal{LCWA}(\mathsf{Dept}(x, y), y = \mathsf{CS})$ expresses that all the members of the computer science department are known and are mentioned in the database.

2.1 The Meaning of Local Closed-World Assumptions

The intuitive meaning behind the LCWA expressions of Definition 2 can be formally captured using first-order formulas. For this, we first introduce the following notation.

Definition 4 ([5]). *Let D be a database and P a predicate in D. Denote by P^D the P-tuples in D. Given a tuple $\bar{t}$ of terms, $P(\bar{t}) \in D$ denotes the formula $\vee_{\bar{a} \in P^D}(\bar{t} = \bar{a})$.*

Definition 5 ([5]). *Let D be a database over 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*

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

Example 3. The meaning of $\theta = \mathcal{LCWA}(\mathsf{Tel}(x,y), \mathsf{Dept}(x,\mathsf{CS}))$ in item 1 of Example 2 is given by $\mathcal{M}_D(\theta) = \forall x \forall y \big(\mathsf{Dept}(x,\mathsf{CS}) \supset \big(\mathsf{Tel}(x,y) \supset$

$$\big((x = \mathsf{LD} \wedge y = 6531421) \vee (x = \mathsf{LD} \wedge y = \text{09-23314}) \vee$$
$$(x = \mathsf{BD} \wedge y = 5985625) \vee (x = \mathsf{TD} \wedge y = 5845213)\big)\big).$$

The two extreme cases of local closed-world assumptions are the following:

- $\theta = \mathcal{LCWA}(P(\bar{x}), \mathbf{t})$: this expresses that D has complete knowledge on P.
- $\theta = \mathcal{LCWA}(P(\bar{x}), \mathbf{f})$: this LCWA does not expresses any closure. In fact, $\mathcal{M}_D(\theta)$ is a tautology for every D.

A useful property of the local closed-world assumption is that any collection of LCWAs on the same predicate may be combined into one (disjunctive) LCWA, that is, the set of LCWA $\theta_i = \mathcal{LCWA}(P(\bar{x}), \Psi_i[\bar{x}_i])$, $i = 1, \ldots, n$, is equivalent to $\theta = \mathcal{LCWA}(P(\bar{x}), \bigvee_{i=1}^{n} \Psi_i[\bar{x}_i])$. We therefore assume, without a loss of generality, that each predicate symbol 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 Axiom*: $\mathsf{DCA}(dom(\mathfrak{D})) = \forall x (\bigvee_{i=1}^{n} x = C_i)$
- *Unique Name Axioms*: $\mathsf{UNA}(dom(\mathfrak{D})) = \bigwedge_{1 \leqslant i < j \leqslant n} C_i \neq C_j$

where $dom(\mathfrak{D}) = \{C_1, \ldots, C_n\}$.

Definition 6 ([5]). *Let $\mathfrak{D} = (D, \mathcal{L})$ be a locally closed database over Σ. The* meaning *of $\mathfrak{D}$ is the first-order sentence*

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

The formula $\mathcal{M}(\mathfrak{D})$ expresses incomplete knowledge about the real world. Thus, in general, it has several models. A $\Sigma_{\mathfrak{D}}$-model M of $\mathcal{M}(\mathfrak{D})$ is also 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 $\mathfrak{D} \models \varphi$.

Note that locally closed databases generalize the concepts of closed-world assumption and, at the other extreme, open-world assumption (OWA) [1, 7] (i.e, no closure at all) by setting all windows of expertise to respectively $\mathbf{t}$ and $\mathbf{f}$.

3 Query Answering in Locally Closed Databases

In this section, we provide the basic tools for reasoning with locally closed databases. Our main result is stated in Proposition 4, where we show that deciding closed-world information is undecidable. This result provides the motivation for the introduction of the approximate methods of Section 4.

Query answering in locally closed databases may be represented as follows: 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 (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 $\mathfrak{D}$.

Another interesting question for 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 can be defined as follows:

Definition 7 ([8]). *A locally closed database $\mathfrak{D}$ over Σ has* complete information *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 $\mathfrak{D} \models \neg\mathcal{Q}[\bar{t}]$.*

The idea of complete information on queries is sometimes called *Closed-World Information* (CWI) on a query, and it was investigated by Levy [8] in the context of incomplete databases. Observe that the LCWA and CWI are related concepts that capture different phenomena. The LCWA expresses completeness of a part of a *database relation*, while the CWI identifies completeness of a *query* posed to the database.

Next, we investigate the computational complexity of query answering. Following the usual measure of complexity in databases, 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:

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

Proposition 1 ([4]). *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 the next proposition shows, queries with CWI can be answered directly in the database instance D, when D is regarded as a two-valued Herbrand structure of $\mathfrak{D}$.[3]

Proposition 2. *If $\mathfrak{D}$ has CWI on query $\mathcal{Q}[\bar{x}]$, then $Cert_{\mathfrak{D}}(\mathcal{Q}[\bar{x}]) = Poss_{\mathfrak{D}}(\mathcal{Q}[\bar{x}]) = \{\bar{a} \mid D \models \mathcal{Q}[\bar{a}]\}$.*

Proof. 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: $\{\bar{a} \mid D \models \mathcal{Q}[\bar{a}]\} \subseteq Poss_{\mathfrak{D}}(\mathcal{Q}[\bar{x}]) = Cert_{\mathfrak{D}}(\mathcal{Q}[\bar{x}]) \subseteq \{\bar{a} \mid D \models \mathcal{Q}[\bar{a}]\}$. □

Proposition 3 ([4]). *The decision problem $\mathrm{CWI}_{\mathcal{L}}(\mathcal{Q}[\bar{x}])$ is in* coNP *for each $\mathcal{L}$ and $\mathcal{Q}[\bar{x}]$, and is* coNP*-hard for some of them.*

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

Proposition 3, shows that deciding whether there is CWI on a query $\mathcal{Q}[\bar{x}]$ in a *specific* database $\mathfrak{D} = (D, \mathcal{L})$ is not tractable. 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 4. *The question whether all locally closed databases $(\cdot, \mathcal{L})$ convey CWI on a query $\mathcal{Q}[\bar{x}]$ is undecidable.*

Proof. Consider $\mathcal{Q}[\bar{x}] = P(c)$ 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 [11], this problem is undecidable. □

Note 2. In [8], Levy shows that for a particular case in which the windows of expertise are (positive) special cases of conjunctive queries (called by Levy variable-interval queries) and $\mathcal{Q}[\bar{x}]$ is the union of (positive) conjunctive queries, the decision problem considered in Proposition 4 can be solved in polynomial time.

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. 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 contained a few extra "impossible" elements. E.g., 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. So, 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 expressivity 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 sublanguages.

4 Approximative Reasoning

4.1 Approximations by Three-Valued Structures

The basic idea of the 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 $\mathfrak{L}^c$.

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 function symbols of Σ together with all the new predicate symbols.

Definition 8. *We say that a 2-valued Σ'-structure I* simulates *a three-valued Σ-structure $\mathcal{K}$, iff $\mathcal{K}$ and I have the same domain and assign the same interpretations to constant and function symbols, and for each predicate $P \in \mathcal{R}(\Sigma)$, $(P^c)^I = \{\bar{d} \mid P(\bar{d})^{\mathcal{K}} = \mathbf{t}\}$ 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 5 below, these formulas can be used to compute three-valued answers for Φ.

Definition 9. *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 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, inversely, 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 $P_i^{c\neg}(\bar{t})$, and 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 5. *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-value truth evaluation and query answering.

Corollary 1. *Given a finite three-valued Σ-structure $\mathcal{K}$, for each formula $\varphi[\bar{x}]$, the sets $\{\bar{d} \mid (\varphi[\bar{d}])^{\mathcal{K}} = \mathbf{t}\}$, $\{\bar{d} \mid (\varphi[\bar{d}])^{\mathcal{K}} = \mathbf{f}\}$, 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:

Definition 10 ([3]). *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 of all 3-valued Herbrand Σ-structures approximating Γ and is well-defined 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.

Proposition 6. *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 $\Gamma \models \neg\varphi$.*

Proof. By the fact that all models of a theory containing $\mathsf{UNA}(\Sigma) \wedge \mathsf{DCA}(\Sigma)$ are isomorphic to Herbrand structures. □

The converse of Proposition 6 does not hold, of course, not even when $\mathcal{K} = \mathcal{O}_\Gamma$. For example, take $\mathcal{R}(\Sigma) = \{P\}$ and $\Gamma = \emptyset$. The optimal approximation of Γ is $\{P : \mathbf{u}\}$. It holds that $\emptyset \models P \vee \neg P$ while $(P \vee \neg P)^{\mathcal{O}_\emptyset} = \mathbf{u}$.

Definition 11 ([3]). *For a 3-valued Σ-interpretation $\mathcal{K}$ and a query $\mathcal{Q}[\bar{x}]$ in Σ, define:*

- *the* certain answers *of $\mathcal{Q}[\bar{x}]$ w.r.t. $\mathcal{K}$: $Cert_{\mathcal{K}}(\mathcal{Q}[\bar{x}]) = \{\bar{a} \mid \mathcal{Q}[\bar{a}]^{\mathcal{K}} = \mathbf{t}\}$.*
- *the* possible answers *of $\mathcal{Q}[\bar{x}]$ w.r.t. $\mathcal{K}$: $Poss_{\mathcal{K}}(\mathcal{Q}[\bar{x}]) = \{\bar{a} \mid \mathcal{Q}[\bar{a}]^{\mathcal{K}} \leqslant \mathbf{u}\}$.*

As computing truth values of sentences is polynomial, we have:

Proposition 7 ([3]). *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 $\mathcal{K}$.*

4.2 Query Answering by Fixpoint Computations

From Proposition 7 it is clear that a tractable method to compute 3-valued approximations induces a tractable sound approximative query answering method. Next, we consider such a method.

Definition 12 ([3]). *Given a locally closed database $\mathfrak{D} = (D, \mathcal{L})$, the operator $App_{\mathfrak{D}} : \mathfrak{L}^c \to \mathfrak{L}^c$ maps a three-valued structure $\mathcal{K}$ to a three-valued structure $\mathcal{K}' = App_{\mathfrak{D}}(\mathcal{K})$ such that, for every predicate P of $\mathcal{R}(\Sigma)$ and every tuple $\bar{a}$,*

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

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 8 ([3]). *$\mathcal{A}pp_{\mathfrak{D}}$ is a $\leqslant_p$-monotone operator on the chain complete poset $\mathfrak{L}^c$, thus it has a $\leqslant_p$-least fixpoint.*

Definition 13. *Denote by $\mathcal{C}_{\mathfrak{D}}$ the $\leqslant_p$-least fixpoint of $\mathcal{A}pp_{\mathfrak{D}}$.*

Note 3. 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 $|D|$.

Example 4. Consider Example 1 and the assumption $\mathcal{LCWA}(\mathsf{Dept}(x,y), y = \mathsf{CS})$ of Example 2 (second item). In this case, $\mathsf{Dept}(x,y)^{\mathcal{C}_{\mathfrak{D}}} = \mathbf{t}$ for all the tuples (x,y) s.t. $\mathsf{Dept}(x,y) \in D$, $\mathsf{Dept}(x,\mathsf{CS})^{\mathcal{C}_{\mathfrak{D}}} = \mathbf{f}$ for every $x \notin \{\mathsf{BD}, \mathsf{TD}\}$, and $\mathsf{Dept}(x,y)^{\mathcal{C}_{\mathfrak{D}}} = \mathbf{u}$ in all the other cases.

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

Proposition 9 ([3]). *$\mathcal{C}_{\mathfrak{D}}$ approximates $\mathfrak{D}$ and for every $\mathcal{Q}[\overline{x}]$ it holds that $Cert_{\mathcal{C}_{\mathfrak{D}}}(\mathcal{Q}[\overline{x}]) \subseteq Cert_{\mathfrak{D}}(\mathcal{Q}[\overline{x}]) \subseteq Poss_{\mathfrak{D}}(\mathcal{Q}[\overline{x}]) \subseteq Poss_{\mathcal{C}_{\mathfrak{D}}}(\mathcal{Q}[\overline{x}])$.*

4.3 Fixpoint Queries for the LCWA

A substantial drawback of query answering with $\mathcal{C}_{\mathfrak{D}}$ is the need to recompute it each time that 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 [2] or tabling [10], will often need only fractions of those relations.

Definition 14. *Let $\mathfrak{D} = (D, \mathcal{L})$ be a locally close 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:*

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

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 following fixpoint computations on $\Delta_{\mathcal{Q},\mathcal{L}}$.

Definition 15. *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 [12] or a positive definition as defined in FO[ID] [6] 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 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 $\mathbf{lfp}_{Q^c,\Delta_{\mathcal{Q},\mathcal{L}}}$ and $\mathbf{lfp}_{Q^{c\neg},\Delta_{\mathcal{Q},\mathcal{L}}}$, respectively

Example 5. Consider gain Example 2 and the local closed-world assumption of item 1 $\mathcal{LCWA}(\mathsf{Tel}(x,y), \mathsf{Dept}(x,\mathsf{CS}))$. Assume that no closure exists for the Dept relation, i.e. $\mathcal{LCWA}(\mathsf{Dept}(x,y), \mathbf{f})$. Let $\mathcal{Q} = \mathsf{Tel}(\mathsf{BD}, 3962836)$ (see 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, $\mathsf{Tel}(\text{Bart Delvaux}, 3962836)$ is certainly false.

Proposition 10. *Given a locally closed database $\mathfrak{D} = (D, \mathcal{L})$ and a query $\mathcal{Q}[\bar{x}]$. Let $(\mathcal{R}^c_\mathcal{Q}, \mathcal{R}^{c\neg}_\mathcal{Q}, \mathcal{R}^c_1, \mathcal{R}^{\neg c}_1, \ldots, \mathcal{R}^c_n, \mathcal{R}^{\neg c}_n)$ be the relations defined by $\Delta_{\mathcal{Q},\mathcal{L}}$ in D. Then:*

$$\mathcal{R}^c_i = \{\bar{d} \mid P_i(\bar{d})^{\mathcal{C}_\mathfrak{D}} = \mathbf{t}\},\ \ \mathcal{R}^{c\neg}_i = \{\bar{d} \mid P_i(\bar{d})^{\mathcal{C}_\mathfrak{D}} = \mathbf{f}\}\ \ (i = 1, \ldots, n),$$
$$\mathcal{R}^c_\mathcal{Q} = \{\bar{d} \mid \mathcal{Q}(\bar{d})^{\mathcal{C}_\mathfrak{D}} = \mathbf{t}\},\ \ \mathcal{R}^{c\neg}_\mathcal{Q} = \{\bar{d} \mid \mathcal{Q}(\bar{d})^{\mathcal{C}_\mathfrak{D}} = \mathbf{f}\}.$$

Proof (outline). By induction on the number of iterations in the computations of the operators $\mathcal{A}pp_\mathfrak{D}$ (Definition 12) and $\Gamma_{\Delta_{\mathcal{Q},\mathcal{L}}}$ (Definition 15), one shows that the structure that is obtained by the latter in a certain iteration simulates (in the sense of Definition 8) 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^c_i and $P^{c\neg}_i$. 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 5, 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 $(\mathcal{Q}[\bar{d}])^{\mathcal{C}_\mathfrak{D}} = \mathbf{f}$. □

Note 4. Dually, one may define the set

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

and consider the greatest fixpoint expressions $\neg[\mathbf{gfp}_{Q^{p\neg},\Delta'_{\mathcal{Q},\mathcal{L}}}](\bar{x})$ 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.

4.4 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 following example shows, in certain cases we lose accuracy.

Example 6. Below, we abbreviate the optimal approximation of $\mathfrak{D}$ by $\mathcal{O}_\mathfrak{D}$ (instead of $\mathcal{O}_{\mathcal{M}(\mathfrak{D})}$).

1. 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 $\mathbf{t}$ when posed with respect to $\mathfrak{D}$ or $\mathcal{O}_\mathfrak{D}$.
2. Let $D=\emptyset$ and $\mathcal{L}=\{\mathcal{LCWA}(P,R),\ \mathcal{LCWA}(Q,R\supset\neg P)\}$. Note that in this case $\mathcal{M}(D,\mathcal{L})=(R\supset\neg P)\wedge((R\supset\neg P)\supset\neg Q)$, which obviously entails $\neg Q$, and so $Q^{\mathcal{O}_\mathfrak{D}}=\mathbf{t}$. 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 one case in which accuracy of the approximation method is guaranteed.

Definition 16. *The* LCWA dependency graph *of $\mathcal{L}$ is the directed graph on $\mathcal{R}(\Sigma)$, containing a directed edge from predicate Q to P iff there exists $\mathcal{LCWA}(P(\overline{x}),\Psi[\overline{x}])\in\mathcal{L}$ such that Q occurs negatively in Ψ. A* hierarchically closed database *$\mathfrak{D}$ is a locally closed database in which the LCWA dependency graph is cycle-free.*

Note 5. The notion of hierarchically closed databases introduced in Definition 16, is a variant of a stronger condition defined in [3, 4], which excludes *any* circular relation between object predicates and predicates in the window of expertise of the LCWAs. The purpose of such restriction in [3, 4] was to avoid infinite loops and ensure termination of the query answering mechanism. Since the evaluation of the fixpoint query always reaches a fixpoint (see proof of Proposition 10), the strong condition of acyclicity can be lifted for ensuring termination. In order to guarantee completeness, however, in the current approach loops through negation are to be avoided – and hence the need of the previous definition. We observe, also, that a locally closed database $\mathfrak{D}$ is hierarchically closed in the sense of Definition 16, 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.

Proposition 11. *Let $\mathfrak{D} = (D, \mathcal{L})$ be a hierarchically closed database such that every window of expertise in $\mathcal{L}$ is a conjunction of literals. If a query $\mathcal{Q}[\bar{x}]$ is a conjunction of literals, then $Cert_{\mathcal{C}_\mathfrak{D}}(\mathcal{Q}[\bar{x}]) = Cert_\mathfrak{D}(\mathcal{Q}[\bar{x}])$. If $\mathcal{Q}[\bar{x}]$ is a disjunction of literals, then $Poss_{\mathcal{C}_\mathfrak{D}}(\mathcal{Q}[\bar{x}]) = Poss_\mathfrak{D}(\mathcal{Q}[\bar{x}])$.*

Proposition 11 can be generalized in several ways for showing the optimality of our approach with respect to broader classes of query–database pairs. This, however, involves the introduction of several technical notions, the definitions of which are beyond the space limits of this paper. Further results concerning optimal approximations will be reported in a future work.

5 Conclusions

We have presented a rewriting technique to compute certain or possible answers in polynomial time from locally closed databases. Our algorithm is based on approximating all models of the database's theory by means of three-valued structures, which are implicitly represented by fixpoint queries. The present work builds upon [4], but generalizes it by lifting the constraint that the database has to be hierarchically closed in a stricter sense. For a large class of queries and databases, our method is complete.

References

1. S. Abiteboul and O.M. Duschka. Complexity of answering queries using materialized views. In *Proc. 17th PODS*, pages 254–263, 1998.
2. F. Bancilhon, D. Maier, Y. Sagiv, and Ullman J.D. Magic sets and other strange ways to implement logic programs. In *Proc. 5th PODS*, pages 1–15, 1986.
3. A. Cortés-Calabuig, M. Denecker, O. Arieli, and M. Bruynooghe. Representation of partial knowledge and query answering in locally complete databases. In *Proc. 13th LPAR*, LNCS 4246, pages 407–421. Springer, 2006.
4. A. Cortés-Calabuig, M. Denecker, O. Arieli, and M. Bruynooghe. Approximate query answering in locally closed databases. In *Proc. 22nd AAAI*, pages 397–402. AAAI Press, 2007.
5. A. Cortés-Calabuig, M. Denecker, O. Arieli, B. Van Nuffelen, and M. Bruynooghe. On the local closed-world assumption of data-sources. In *Proc. 8th LPNMR*, LNCS 3662, pages 145–157. Springer, 2005.
6. M. Denecker and E. Ternovska. Inductive situation calculus. *Artif. Intell.*, 171(5-6):332–360, 2007.
7. G. Grahne. Information integration and incomplete information. *IEEE Data Engineering Bulletin*, 25(3):46–52, 2002.
8. A. Levy. Obtaining complete answers from incomplete databases. In *Proc. 22nd VLDB*, pages 402–412, 1996.
9. R. Reiter. Towards a logical reconstruction of relational database theory. In *On Conceptual Modelling (Intervale Workshop)*, pages 191–233, 1982.
10. T. Swift. Tabling for non-monotonic programming. *Annals of Mathematics and Artificial Intelligence*, 25(3-4):201–240, 1999.
11. B. Trakhtenbrot. Impossibility of an algorithm for the decision problem in finite classes. *American Mathematical Society Transaction*, 3(2):1–5, 1963.
12. A. Van Gelder. The alternating fixpoint of logic programs with negation. *Journal of Computer and System Sciences*, 47(1):185–221, 1993.