

Representation of Partial Knowledge and Query Answering in Locally Complete Databases

Álvaro Cortés-Calabuig¹, Marc Denecker¹, Ofer Arieli², and Maurice Bruynooghe¹

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. The *Local Closed-World Assumption* (LCWA) is a generalization of Reiter's Closed-World Assumption (CWA) for relational databases that may be incomplete. Two basic questions that are related to this assumption are: (1) how to *represent* the fact that only part of the information is known to be complete, and (2) how to properly *reason* with this information, that is: how to determine whether an answer to a database query is complete even though the database information is incomplete. In this paper we concentrate on the second issue based on a treatment of the first issue developed in earlier work of the authors. For this we consider a fixpoint semantics for declarative theories that represent locally complete databases. This semantics is based on 3-valued interpretations that allow to distinguish between the certain and possible consequences of the database's theory.

1 Introduction

In database theory it is common to falsify any atomic fact that does not appear in the database instance. This approach follows Reiter's Closed-World Assumption (CWA) [13], that presupposes a complete knowledge about the database's domain of discourse.

Databases, however, are not always complete¹. There are many reasons for this fact, including ignorance about the domain, lack of proper maintenance, incomplete migration, accidental deletion of tuples, the intrinsic nature of database mediator-based systems (see [10]), and so forth. Unless properly handled, partial information in database systems might lead to erroneous conclusions, as illustrated in the following example:

Example 1. Consider a database of a computer science (CS) department which stores information about the telephone numbers of the department's members and collaborators. A fragment of the database is represented in Figure 1. A reasonable assumption in this case is that this database is complete with respect to all CS department members, but possibly incomplete regarding its external collaborators. Thus, appropriate answers for the queries `Telephone(Bart Delvaux,3962836)` and `Telephone(Leen Desmet,3212445)` are "no" and "unknown", respectively.

¹ Nor they are always correct, but we do not address this problem here.

Telephone		Department	
Name	Telephone	Name	Department
Leen Desmet	6531421	Bart Delvaux	Computer Science
Bart Delvaux	5985625	Leen Desmet	Philosophy
Tom Demans	5845213	Tom Demans	Computer Science

Fig. 1. A database of contact phone numbers for a CS department

Example 1 illustrates a situation in which database information is *locally* complete, and so applying the CWA is not realistic, and might even lead to wrong conclusions. The other extreme approach, known as *Open-World Assumption* (OWA) [1, 9], is often used for maintaining distributed knowledge, e.g. for mediator-based systems. In this approach a relational database is considered as a correct but possibly incomplete representation of the domain of discourse. The main weakness of the OWA is that it does not allow to express locally complete information, and so in the example above, for instance, one cannot state a full knowledge regarding the phone numbers of the CS department members.

In order to overcome the drawbacks of the CWA and the OWA in representing partial knowledge in reality, Etzioni [6] and Motro [12] introduced the notion of *Local Closed-World Assumption* (LCWA) that, intuitively, is “a specification of the areas in the real world in which a database contains all true tuples” [3]. In Example 1, for instance, such an assumption would state a meta-knowledge that the information in the Telephone relation is complete for the members of the CS department.

At the practical level, the LCWA poses some important challenges. First, a proper way of *representing* the fact that only part of the information is known to be complete is required. In the literature there are several proposals for this, using e.g. theories in a logic programming style [7] or second-order circumscriptive formulae [5]. Here we follow the first-order representation considered in [3].² Another challenge, which is addressed in this paper, is the problem of query answering in the presence of LCWA. This involves not only the query computation itself, but also a determination whether the query answer is complete even though the database information is incomplete. Our approach is based on a 3-valued fixpoint semantics and corresponding algorithms for constructing a 3-valued interpretation that evaluates queries under certain and possible semantics. More specifically, the following issues are addressed:

- **Fixpoint theory for the LCWA.** A sound fixpoint operator for the LCWA is introduced, and conditions for assuring its completeness are defined. This yields a mechanism for computing a 3-valued interpretation that approximates all the 2-valued interpretations of the database’s theory and so allows informative query answering.

² It is shown there that this representation and its second-order derivative capture and generalize both Reiter’s CWA and the OWA.

- **Query answering algorithm.** A simple yet general algorithm for query answering under the LCWA is presented. It distinguishes between certain and possible answers and can be easily implemented by standard relational databases engines.
- **Reconciliation of paradigms.** Alternative approaches to the LCWA are considered. In particular, LCWA handling in the context of database systems [3, 12, 11] is related to LCWA formalizations for modelling logic-based agents [5, 6].

The rest of this paper is organized as follows. In Section 2 we recall some preliminary definitions and facts about the LCWA. In Section 3 we introduce a fixpoint semantics for locally complete databases, and in Section 4 we define corresponding query answering formalisms. Some related works are discussed in Section 5 and future research is sketched in Section 6. Full proofs of all the propositions in this paper appear in [2].

2 Preliminaries

In what follows we denote by Σ a first-order vocabulary consisting of predicate symbols $\mathcal{R}(\Sigma)$ (a relational schema in database terminology) and a finite set $\mathcal{C}(\Sigma)$ of constants representing the elements of the domain of discourse. For a formula Ψ in Σ we denote by $\Psi[\bar{x}]$ that the free variables of Ψ are a subset of \bar{x} . The Herbrand base of Σ is the set $HB(\Sigma)$ of atomic formulas formed using $\mathcal{C}(\Sigma)$ and the predicate symbols in $\mathcal{R}(\Sigma)$. A *database* is a finite set of ground atoms in Σ .

Definition 1. [3] A *local closed-world assumption* (LCWA), is an expression of the form

$$\mathcal{LCWA}(P(\bar{x}), \Psi[\bar{x}]),$$

where $P \in \Sigma$ is a predicate symbol, called the LCWA's *object*; and $\Psi[\bar{x}]$, called a *window of expertise* for P , is a first-order formula over Σ .

Example 2. The expression $\mathcal{LCWA}(\text{Tel}(x, y), \text{Dept}(x, \text{CS}))$ is a local closed-world assumption stating that the telephone numbers of all the members of the computer science department are known. That is, for every x_0 in $\{x \mid \text{Dept}(x, \text{CS})\}$ (the window of expertise for Tel), all atoms of the form $\text{Tel}(x_0, y)$ are in the database.

Definition 2. [3] Let $\theta = \mathcal{LCWA}(P(\bar{x}), \Psi[\bar{x}])$ be a local closed-world assumption and D a database under vocabulary Σ . Denote by P^D the set of tuples corresponding to the set of atoms of P in D . We abbreviate the formula $\bigvee_{\bar{a} \in P^D} (\bar{t} = \bar{a})$ by $P(\bar{t}) \in P^D$, where \bar{t} is a tuple of terms. The *meaning* of θ under D is the formula

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

The intuition behind this formula is simple: for all tuples \bar{x} of domain elements such that $\Psi(\bar{x})$ holds in the *real world*, if $P(\bar{x})$ is true (again, in *reality*), then \bar{x} must be a tuple in the table of P in D .

Example 3. The meaning of $\theta = \mathcal{LCWA}(\text{Tel}(x, y), \text{Dept}(x, \text{CS}))$ in the database D of Example 1 is given by

$$\begin{aligned} \mathcal{M}_D(\theta) = & \forall x \forall y (\text{Dept}(x, \text{CS}) \supset \text{Tel}(x, y) \supset \\ & ((x = \text{Leen Desmet} \wedge y = 6531421) \vee (x = \text{Bart Delvaux} \wedge y = 5985625) \\ & \vee (x = \text{Tom Demans} \wedge y = 5845213))) \end{aligned}$$

In some cases, we may want to express within the same expression local closed-world assumption on different predicates. In doing so, we need to extend the basic notion of an LCWA expression to allow for set of objects:

$$\theta = \mathcal{LCWA}(\{P_1(\bar{x}_1), \dots, P_n(\bar{x}_n)\}, \Psi[\bar{x}]),$$

where the $P_i \in \Sigma$ are predicate symbols (the LCWA's objects) and $\Psi[\bar{x}]$ is a first-order formula over Σ with free variables \bar{y} s.t. $\bar{y} \subseteq \bigcup_{i=1}^n \bar{x}_i = \bar{x}$. When an LCWA expression takes this form, the meaning of θ under a database D is extended as follows:

$$\mathcal{M}_D(\theta) = \forall \bar{x} (\Psi[\bar{x}] \supset (\bigwedge_{i=1}^n (P_i(\bar{x}_i) \supset (P_i(\bar{x}_i) \in P_i^D)))$$

In this paper we assume only one predicate object for each LCWA expression. As the following proposition shows, this assumption does not harm generality:

Proposition 1. [3] *Given a formula Ψ , denote by $\exists|\bar{x}\Psi$ the existential quantification of all free variables in Ψ , except those in \bar{x} . Let $\theta = \mathcal{LCWA}(\{P_1(\bar{x}_1), \dots, P_n(\bar{x}_n)\}, \Psi)$ and $\theta_i = \mathcal{LCWA}(P_i(\bar{x}_i), \exists|\bar{x}_i\Psi)$, $i = 1, \dots, n$. Then $\mathcal{M}_D(\theta) \equiv \bigwedge_{i=1}^n \mathcal{M}_D(\theta_i)$.*

Similarly, one may split a disjunctive window of expertise to its disjuncts and still preserve the original LCWA, and any collection of LCWAs on the same predicate may be combined to one (disjunctive) LCWA.

Proposition 2. *Let $\theta = \mathcal{LCWA}(P(\bar{x}), \bigvee_{i=1}^n \Psi_i[\bar{x}_i])$ and $\theta_i = \mathcal{LCWA}(P(\bar{x}), \Psi_i[\bar{x}_i])$, $i = 1, \dots, n$. Then $\mathcal{M}_D(\theta) \equiv \bigwedge_{i=1}^n \mathcal{M}_D(\theta_i)$.*

In the sequel we assume, without loss of generality, one LCWA expression per predicate of $\mathcal{R}(\Sigma)$. The predicates in $\mathcal{R}(\Sigma)$ that do not appear as objects in any LCWA expression are considered as objects of LCWAs in which the windows of expertise is false. In other words, there is no context in which those predicates are complete.

The meaning of a database is now defined by the conjunction of its atoms augmented with the meaning of the given local closed-world assumptions, and the following two general assumptions:

- *Domain Closure Axiom:* $\text{DCA}(\Sigma) = \forall x (\bigvee_{i=1}^n x = C_i)$,
- *Unique Name Axiom:* $\text{UNA}(\Sigma) = \bigwedge_{1 \leq i < j \leq n} C_i \neq C_j$,

where C_1, \dots, C_n are the constant symbols of Σ (i.e., in $\mathcal{C}(\Sigma)$).

Definition 3. Let D be a database and let \mathcal{L} be a set of LCWA expressions $\theta_j = \text{LCWA}(P_j, \Psi_j)$, $j=1, \dots, n$ applied on D . The *meaning* of D and \mathcal{L} is given by:

$$\mathcal{M}(D, \mathcal{L}) = \bigwedge_{A \in D} A \wedge \bigwedge_{j=1}^n \mathcal{M}_D(\theta_j) \wedge \text{UNA}(\Sigma) \wedge \text{DCA}(\Sigma).$$

The theory consisting of the meaning of D and \mathcal{L} is consistent and decidable³. Consistency is shown in [3] and decidability follows from the fact that the language does not contain function symbols and that UNA and DCA are imposed, ensuring a fixed and finite domain. Note also that each model of this theory is isomorphic to a Herbrand model.

Our goal is to evaluate queries with respect to the meaning $\mathcal{M}(D, \mathcal{L})$ of a database D and a set \mathcal{L} of LCWA expressions. In this context, particularly interesting queries are those formulas that are either entailed by $\mathcal{M}(D, \mathcal{L})$ or are necessarily falsified by it. Such queries induce definitive answers. This idea is formalized in different ways in [3, 6, 11]. In what follows we adopt the definition of [6].

Definition 4. A first-order theory Γ determines *complete world information* (CWI) on a query $Q[\bar{x}]$ iff for every ground tuple \bar{d} , either $\Gamma \models Q[\bar{d}]$ or $\Gamma \models \neg Q[\bar{d}]$ holds.

Observe that the LCWA and CWI are related concepts that capture different phenomena. The LCWA expresses completeness of a set of atoms in a relational database while the CWI identifies completeness of queries posed to the databases. Frequently, one or more LCWAs determine CWI on a query with respect to a given database. In Section 3 we consider sufficient conditions for assuring this.

As the meaning $\mathcal{M}(D, \mathcal{L})$ of a database is a first-order formula, one way of evaluating queries is by using off-the-shelf theorem provers. This requires a new derivation for every ground instance of the query, which makes the whole process time consuming. An alternative approach is to generate a 3-valued Herbrand interpretation approximating all models of $\mathcal{M}(D, \mathcal{L})$ and then evaluate different queries with respect to this interpretation. The advantage of this approach is twofold. From a theoretical point of view, it is a good tool to distinguish the complete consequences of the theory from the incomplete ones, and in particular CWI can be easily determined. From a more practical perspective, the 3-valued Herbrand interpretation can be used to compute approximating answers to queries. In the next sections we consider this approach.

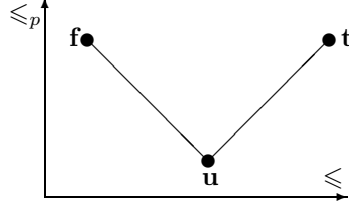
3 3-valued fixpoint theory for LCWA

3.1 3-valued semantics

The truth values $\text{THREE} = \{\mathbf{t}, \mathbf{u}, \mathbf{f}\}$, standing for true, unknown and false, of 3-valued semantics are usually arranged in two orders: the truth order, \leq , which is a linear order given by $\mathbf{f} \leq \mathbf{u} \leq \mathbf{t}$, and the precision order \leq_p , which is a partial order

³ There is an effective way of deciding whether any given sentence is a theorem of the theory.

on \mathcal{THRE} in which \mathbf{u} is the least element, and \mathbf{t} and \mathbf{f} are incomparable maximal elements. The structure of \mathcal{THRE} is drawn in the following diagram.



The conjunction \wedge in \mathcal{THRE} is defined by the \leq -glb of this structure, the disjunction \vee is defined by the \leq -lub, and the negation operator \neg is associated with the \leq -involution, that is: $\neg\mathbf{t} = \mathbf{f}$, $\neg\mathbf{f} = \mathbf{t}$, and $\neg\mathbf{u} = \mathbf{u}$.

In this paper we focus on Herbrand interpretations. Two-valued (respectively, 3-valued) (Herbrand) interpretations of Σ are total functions from the set $HB(\Sigma)$ of all ground atoms of Σ to the set of truth values $\{\mathbf{t}, \mathbf{f}\}$ (respectively, to the elements in \mathcal{THRE}). Equivalently, two-valued interpretations are sometimes represented as sets of (true) atoms. Extensions to complex formulas are as usual. An interpretation I satisfies a formula ψ if $\psi^I = \mathbf{t}$; I satisfies a set Γ of formulas if it satisfies every formula in Γ . In this case I is called a *model* of Γ . The orders \leq and \leq_p between truth values are extended to 3-valued Herbrand interpretations by pointwise comparisons. For a fixed Σ , the set \mathcal{L} of 2-valued Herbrand Σ -interpretations forms a complete lattice under \leq . The set \mathcal{L}^c of 3-valued Herbrand Σ -interpretations is a chain complete poset⁴ under \leq_p .

There is an interesting lattice theoretic way to construct the 3-valued interpretations from \mathcal{L} . Given a lattice \mathcal{L} , we can define \mathcal{L}_c as the set of *consistent* pairs $(x, y) \in \mathcal{L}$, i.e., pairs such that $x \leq y$. On \mathcal{L}_c , we can define two orders:

1. $(x, y) \leq (x', y')$ if $x \leq x'$ and $y \leq y'$
2. $(x, y) \leq_p (x', y')$ if $x \leq x'$ and $y' \leq y$

In general, \leq is a lattice order and \leq_p is a chain-complete order. The following mapping from 3-valued interpretations \mathcal{K} to consistent pairs (I, J) of two-valued interpretations is a one-to-one correspondence from \mathcal{L}^c to \mathcal{L}_c , preserving both \leq and \leq_p :

$$\begin{aligned} I &= \{P(\bar{a}) \in HB(\Sigma) \mid P(\bar{a})^{\mathcal{K}} = \mathbf{t}\} \\ J &= \{P(\bar{a}) \in HB(\Sigma) \mid P(\bar{a})^{\mathcal{K}} = \mathbf{t} \text{ or } P(\bar{a})^{\mathcal{K}} = \mathbf{u}\} \end{aligned}$$

Conversely, \mathcal{K} can be constructed from (I, J) by defining for every atom $P(\bar{a})$:

$$P(\bar{a})^{\mathcal{K}} = \begin{cases} \mathbf{t} & \text{if } P(\bar{a})^I = \mathbf{t} \\ \mathbf{f} & \text{if } P(\bar{a})^J = \mathbf{f} \\ \mathbf{u} & \text{otherwise} \end{cases}$$

⁴ A poset P is chain-complete if every totally ordered subset $C \subseteq P$ has a least upper bound.

3.2 Fixpoint operators for LCWAs

Below, we focus on theories Γ that include UNA and DCA, i.e., every model is isomorphic with a Herbrand model.

Definition 5. Let Γ be a consistent theory based on Σ . 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 , $\mathcal{K} \leq_p M$. The *optimal approximation* for Γ is a 3-valued Σ -interpretation, defined by $\mathcal{K}_{opt}(\Gamma) = glb_{\leq_p}(\{M \mid M \models \Gamma\})$, where M ranges over 2-valued Herbrand models of Γ .

$\mathcal{K}_{opt}(\Gamma)$ is the most precise of all 3-valued Herbrand Σ -interpretations approximating Γ and is well-defined since every nonempty set $S \subseteq \mathcal{L}^c$ has a greatest lower bound and Γ is consistent. Back in the LCWA context, note that $\mathcal{M}(D, \mathcal{L})$ satisfies the consistency condition, hence $\mathcal{K}_{opt}(\mathcal{M}(D, \mathcal{L}))$ is well-defined.

In order to construct a 3-valued approximation for $\mathcal{M}(D, \mathcal{L})$, we first introduce a fixpoint operator on the chain complete poset of 3-valued Herbrand Σ -interpretations.

Definition 6. Define an operator $\mathcal{O}_{LCWA} : \mathcal{L}^c \rightarrow \mathcal{L}^c$ as follows: for every $\mathcal{K} \in \mathcal{L}^c$, the interpretation $\mathcal{K}' = \mathcal{O}_{LCWA}(\mathcal{K})$ is defined for each ground atom $P(\bar{a})$ by:

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

The idea is to iterate \mathcal{O}_{LCWA} starting with total ignorance (a valuation that assigns \mathbf{u} to every ground atom), and gradually extend the definite knowledge according to the database and its LCWAs.

Proposition 3. \mathcal{O}_{LCWA} is a \leq_p -monotone operator on the chain complete poset of 3-valued Σ -interpretations, thus it has a least fixpoint. Moreover, the least fixpoint can be computed in polynomial time in the size of the database.

Proof. Monotonicity follows from \leq_p -monotonicity of the truth assignment. By an extension of the well-known Knaster-Tarski theorem, every monotone operator in a chain complete poset has a fixpoint. Polynomial complexity follows from the fact that per application of the operator, the number of queries to be solved is polynomial in the size of the database and each query can be solved in polynomial time, while the number of iterations is at most polynomial in the size of the database. \square

Definition 7. Denote by $\mathcal{O}_{LCWA}^\dagger$ the \leq_p -least fixpoint of \mathcal{O}_{LCWA} .

Example 4.

1. If $D = \emptyset$ and $\theta_1 = \mathcal{LCWA}(P, R)$ then $R^{\mathcal{O}_{LCWA}^\dagger} = \mathbf{u}$ and $P^{\mathcal{O}_{LCWA}^\dagger} = \mathbf{u}$.
2. Suppose that $D = \emptyset$, $\theta_1 = \mathcal{LCWA}(Q, \mathbf{t})$, and $\theta_2 = \mathcal{LCWA}(P, \neg Q)$. In this case $Q^{\mathcal{O}_{LCWA}^\dagger} = \mathbf{f}$ and $P^{\mathcal{O}_{LCWA}^\dagger} = \mathbf{f}$.

3. Consider the database D of Example 1 and \mathcal{LCWA} ($\text{Tel}(x, y), \text{Dept}(x, \text{CS})$), that is discussed in Example 2. Here $\text{Tel}(\text{Bart Delvaux}, 1234567)^{\mathcal{O}_{\mathcal{LCWA}}^\dagger} = \mathbf{f}$, while $\text{Tel}(\text{Leen Desmet}, 1234567)^{\mathcal{O}_{\mathcal{LCWA}}^\dagger} = \mathbf{u}$.

The following theorem shows that $\mathcal{O}_{\mathcal{LCWA}}^\dagger$ is a sound approximation of $\mathcal{M}(D, \mathcal{L})$.

Theorem 1. (Soundness) $\mathcal{O}_{\mathcal{LCWA}}^\dagger$ approximates $\mathcal{M}(D, \mathcal{L})$.

Proof (Outline). Denote by \perp the interpretation that assigns \mathbf{u} to every ground atom, and by $\mathcal{O}_{\mathcal{LCWA}}(\perp)^i$ the i -th iteration of $\mathcal{O}_{\mathcal{LCWA}}$ starting from \perp . Let M be a model of $\mathcal{M}(D, \mathcal{L})$. By induction on i , one shows that $\mathcal{O}_{\mathcal{LCWA}}(\perp)^i \leq_p M$. \square

Example 5. In the following two cases $\mathcal{O}_{\mathcal{LCWA}}^\dagger$ is strictly less than $\mathcal{K}_{opt}(\mathcal{M}(D, \mathcal{L}))$:

1. $D = \emptyset$, $\theta = \mathcal{LCWA}(Q, P \vee \neg P)$.
All the models of $\mathcal{M}(D, \mathcal{L})$ assign \mathbf{f} to Q , thus $Q^{\mathcal{K}_{opt}(\mathcal{M}(D, \mathcal{L}))} = \mathbf{f}$. However, as in the standard 3-valued Kripke-Kleene semantics $P \vee \neg P$ is unknown whenever P is unknown, we have that $Q^{\mathcal{O}_{\mathcal{LCWA}}^\dagger} = \mathbf{u}$.
2. $D = \emptyset$, $\theta_1 = \mathcal{LCWA}(P, R)$, $\theta_2 = \mathcal{LCWA}(Q, R \supset \neg P)$.
Here again $Q^{\mathcal{O}_{\mathcal{LCWA}}^\dagger} = \mathbf{u}$, while $Q^{\mathcal{K}_{opt}(\mathcal{M}(D, \mathcal{L}))} = \mathbf{f}$.

One way to address the phenomenon in item (1) is to extend the Kripke-Kleene semantics to supervaluations [4]. Under this semantics, two-valued and 3-valued tautologies/contradictions coincide. In what follows we avoid this problem by representing tautologies and contradictions only with the standard \mathbf{t} and \mathbf{f} symbols (respectively).

The difference between $\mathcal{K}_{opt}(\mathcal{M}(D, \mathcal{L}))$ and $\mathcal{O}_{\mathcal{LCWA}}^\dagger$ in item 2 is more subtle. In this case $\mathcal{M}(D, \mathcal{L})$ is the formula $(R \supset \neg P) \wedge ((R \supset \neg P) \supset \neg Q)$, which obviously entails $\neg Q$. The intuitive reason for the difference is that the window of expertise in θ_2 is exactly the meaning of θ_1 , and this link is not captured by $\mathcal{O}_{\mathcal{LCWA}}$. To gain completeness, some restrictions need to be imposed to the windows of expertise. In the following section we study such conditions.

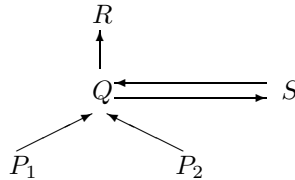
3.3 A hierarchy of LCWAs

Definition 8. An *LCWA dependency graph* that is determined by a set of LCWAs \mathcal{L} , is a directed graph whose nodes correspond to $R(\Sigma)$, such that there is a directed edge from Q to P iff there exists $\mathcal{LCWA}(P(\bar{x}), \Psi[\bar{x}]) \in \mathcal{L}$ such that Q occurs in Ψ .

Example 6. Consider the following set of local-closed world assumptions:

$$\mathcal{L} = \left\{ \begin{array}{lll} \mathcal{LCWA}(P_1(x), \mathbf{t}) & \mathcal{LCWA}(P_2(x), \mathbf{t}) & \mathcal{LCWA}(Q(x), P_1(x) \wedge P_2(x)) \\ \mathcal{LCWA}(Q(x), S(x)) & \mathcal{LCWA}(S(x), Q(x)) & \mathcal{LCWA}(R(x), Q(x)) \end{array} \right\}$$

The corresponding (cyclic) dependency graph is shown below:



Definition 9. A local closed-world assumption expression $\mathcal{LCWA}(P(\bar{x}), \Psi[\bar{x}])$ is *primitive* iff $\Psi[\bar{x}]$ is either **t**, **f**, contains only the equality predicate or any Boolean combination of those. Likewise, a predicate P is *primitive* iff P appears as object of a primitive LCWA expression.

Primitive LCWAs induce CWI on appropriate subsets of their object predicates. Consider for instance $\mathcal{LCWA}(P(x), x = a)$. This is a primitive LCWA and it specifies that CWI on P is obtained for the domain constant a . Thus $\mathcal{M}(D, \mathcal{L}) \models P(a)$ or $\mathcal{M}(D, \mathcal{L}) \models \neg P(a)$, for any database D . The following proposition formalizes this property and establishes the relationship between primitive LCWA and CWI.

Proposition 4. *Let D be a database and let $\theta = \mathcal{LCWA}(P(\bar{x}), \Psi) \in \mathcal{L}$ be a primitive LCWA on D . Then for any \bar{c} s.t. $\Psi[\bar{c}]$ is true, $\mathcal{M}(D, \mathcal{L}) \models P(\bar{c})$ iff $P(\bar{c}) \in D$ and $\mathcal{M}(D, \mathcal{L}) \models \neg P(\bar{c})$ iff $P(\bar{c}) \notin D$. In this case, then, every model of $\mathcal{M}(D, \mathcal{L})$ satisfies $\neg P(\bar{c})$.*

Proof. Let HU be the Herbrand universe of Σ . As θ is a primitive LCWA, for every $\bar{c} \in HU^n$ either $\Psi[\bar{c}]$ holds in every Herbrand interpretation of $\mathcal{M}(D, \mathcal{L})$ or else $\Psi[\bar{c}]$ is falsified by every Herbrand interpretation. Assume that $\Psi[\bar{c}]$ is true. If $P(\bar{c}) \in D$, then $\mathcal{M}(D, \mathcal{L})$ contains the conjunct $P(\bar{c})$, hence every model of $\mathcal{M}(D, \mathcal{L})$ satisfies $P(\bar{c})$. If $P(\bar{c}) \notin D$, then every model M of $\mathcal{M}(D, \mathcal{L})$ satisfies:

$$\Psi[\bar{c}] \supset (P(\bar{c}) \supset P(\bar{c}) \in D)$$

It follows that $P(\bar{c})$ is false in M . □

Corollary 1. *Let D be a database and let $\theta = \mathcal{LCWA}(P(\bar{x}), \Psi) \in \mathcal{L}$ be a primitive LCWA on D . For any \bar{c} such that $\Psi[\bar{c}]$ holds, $\mathcal{M}(D, \mathcal{L})$ determines CWI on $P(\bar{c})$.*

Another interesting relation between LCWA and CWI is the following:

Proposition 5. *Let $\theta = \mathcal{LCWA}(P(\bar{x}), \Psi[\bar{x}]) \in \mathcal{L}$. If $\mathcal{M}(D, \mathcal{L})$ determines CWI on $\Psi[\bar{x}]$, then it also determines CWI on $P(\bar{x}) \wedge \Psi[\bar{x}]$ and on $\neg P(\bar{x}) \wedge \Psi[\bar{x}]$.*

Proof. Again, denote by HU the Herbrand universe of Σ . By assumption, for every $\bar{c} \in HU^n$ either $\mathcal{M}(D, \mathcal{L}) \models \Psi[\bar{c}]$ or $\mathcal{M}(D, \mathcal{L}) \models \neg \Psi[\bar{c}]$. In the second case, $\mathcal{M}(D, \mathcal{L}) \models \neg(P(\bar{c}) \wedge \Psi[\bar{c}])$ and $\mathcal{M}(D, \mathcal{L}) \models \neg(\neg P(\bar{c}) \wedge \Psi[\bar{c}])$. In the first case, the values of $P(\bar{c}) \wedge \Psi[\bar{c}]$ and $\neg P(\bar{c}) \wedge \Psi[\bar{c}]$ are determined by the question whether $P(\bar{c})$ and $\neg P(\bar{c})$ are true. As in the proof of Proposition 4, this reduces to the validity of $P(\bar{c}) \in P^D$, which is determined by the content of D . □

Example 7. Consider again Example 6. By the last proposition, some of the formulas to which $\mathcal{M}(D, \mathcal{L})$ determines CWI can be inductively defined by the following stages:

1. $P_1(x), \neg P_1(x), P_2(x), \neg P_2(x)$.
2. $P_1(x) \wedge P_2(x), \neg P_1(x) \wedge P_2(x), P_1(x) \wedge \neg P_2(x), \neg P_1(x) \wedge \neg P_2(x)$.
3. $Q(x) \wedge P_1(x) \wedge P_2(x), \neg Q(x) \wedge P_1(x) \wedge P_2(x)$, and so forth.

Definition 10. A *hierarchically closed* database \mathfrak{D} based on vocabulary Σ is a pair (D, \mathcal{L}) , where D is a database and \mathcal{L} is a set of LCWAs inducing a cycle-free dependency graph.

The transitive closure of a cycle-free LCWA dependency graph is a well-founded strict order on $\mathcal{R}(\Sigma)$, denoted by $<_{\mathcal{L}}$. The minimal predicates in this order are those that are the object of a primitive LCWA (recall that every predicate is the object of exactly one LCWA in \mathcal{L}). This property together with the definition of $\mathcal{O}_{\mathcal{L}CW\mathcal{A}}$ are the corner stones for the following constructive definition of an approximation of $\mathcal{M}(D, \mathcal{L})$.

Definition 11. Let $\mathfrak{D} = (D, \mathcal{L})$ be a hierarchically closed database. The interpretation $\mathcal{K}_{\mathcal{L}}$ that is *induced* by \mathfrak{D} is defined by induction on $<_{\mathcal{L}}$ as follows: for each predicate P of Σ and every tuple \bar{a} ,

$$P(\bar{a})^{\mathcal{K}_{\mathcal{L}}} = \begin{cases} \mathbf{t} & \text{if } P(\bar{a}) \in D \\ \mathbf{f} & \text{if there exists } \mathcal{L}CW\mathcal{A} (P(\bar{x}), \Psi[\bar{x}]) \in \mathcal{L} \text{ s.t.} \\ & \Psi[\bar{a}]^{\mathcal{K}_{\mathcal{L}}} = \mathbf{t} \text{ and } P(\bar{a}) \notin D \\ \mathbf{u} & \text{otherwise} \end{cases}$$

Note 1. In spite of their similar forms, Definitions 6 and 11 define 3-valued interpretations in different ways. In Definition 6 a given operator is iterated so that several 3-valued interpretations are constructed until a fixpoint is reached. Definition 11, on the other hand, induces a gradual construction of a *single* interpretation, starting from bottom elements of the underlying LCWA dependency graph. This construction only works for cycle-free graphs.

It is easy to see how an algorithm to compute $\mathcal{K}_{\mathcal{L}}$ could look like: first, a primitive predicate P from Σ is (non-deterministically) selected. For every tuple \bar{a} of the domain, if $P(\bar{a})$ is in the database, then $P(\bar{a})^{\mathcal{K}_{\mathcal{L}}} = \mathbf{t}$. Otherwise, if the corresponding window of expertise $\Psi_P(\bar{a})$ holds in $\mathcal{K}_{\mathcal{L}}$, then $P(\bar{a})^{\mathcal{K}_{\mathcal{L}}} = \mathbf{f}$. Otherwise, $P(\bar{a})^{\mathcal{K}_{\mathcal{L}}}$ is \mathbf{u} . The same steps can be repeated for any predicate Q once all the predicates on which Q depends have been evaluated.

Proposition 6. $\mathcal{K}_{\mathcal{L}}$ can be computed in polynomial time in the size of the database D .

Proposition 7. $\mathcal{K}_{\mathcal{L}} \equiv \mathcal{O}_{\mathcal{L}CW\mathcal{A}}^{\dagger}$ for every hierarchically closed database $\mathfrak{D} = (D, \mathcal{L})$.

Proof. Let $rank(P) = \max(\{1 + rank(Q) \mid Q <_{\mathcal{L}} P\})$ and $rank(Q) = 1$ iff Q is primitive. The proof is by induction on these ranks: If $rank(P) = 1$ then as P is a primitive predicate, by the definitions of $\mathcal{O}_{\mathcal{L}CW\mathcal{A}}^{\dagger}$ and $\mathcal{K}_{\mathcal{L}}$, $P^{\mathcal{O}_{\mathcal{L}CW\mathcal{A}}^{\dagger}} = P^{\mathcal{K}_{\mathcal{L}}}$. If $rank(P) = n$ for some $n > 1$, then $P^{\mathcal{O}_{\mathcal{L}CW\mathcal{A}}^{\dagger}}$ is computed using the elements in $\{Q \mid Q <_{\mathcal{L}} P\}$. As the rank of each element Q in this set is strictly smaller than n , by induction hypothesis $Q^{\mathcal{O}_{\mathcal{L}CW\mathcal{A}}^{\dagger}} = Q^{\mathcal{K}_{\mathcal{L}}}$. It follows then that $\mathcal{O}_{\mathcal{L}CW\mathcal{A}}^{\dagger}$ and $\mathcal{K}_{\mathcal{L}}$ must assign the same truth value for P , so again $P^{\mathcal{O}_{\mathcal{L}CW\mathcal{A}}^{\dagger}} = P^{\mathcal{K}_{\mathcal{L}}}$. \square

Given Proposition 7, the soundness of $\mathcal{K}_{\mathcal{L}}$ is obtained as a corollary of Theorem 1.

Corollary 2. (*Soundness*) Let $\mathfrak{D} = (D, \mathcal{L})$ be a hierarchically closed database. Then $\mathcal{K}_{\mathcal{L}}$ approximates $\mathcal{M}(D, \mathcal{L})$.

The next theorem states conditions under which *optimal* approximation for $\mathcal{M}(D, \mathcal{L})$ can be effectively constructed. Below, $\leq_{\mathcal{L}}$ denotes the reflexive closure of $<_{\mathcal{L}}$.

Theorem 2. (Completeness) Let $\mathfrak{D} = (D, \mathcal{L})$ be a hierarchically closed database. If Ψ_θ (the windows of expertise of an LCWA expression θ) is a conjunction of literals for every LCWA θ in \mathcal{L} , then $\mathcal{K}_\mathcal{L} = \mathcal{K}_{opt}(\mathcal{M}(D, \mathcal{L}))$.

Proof (Outline). The proof is based on the fact that for every predicate P , $\mathcal{K}_\mathcal{L}|_{\{Q|Q \leq_\mathcal{L} P\}} = glb(\{I|_{\{Q|Q \leq_\mathcal{L} P\}} | I \models \mathcal{M}(D, \mathcal{L})\})$. This is proven by induction on the dependency order on predicates. The crucial step is when $P(\bar{a})^{\mathcal{K}_\mathcal{L}} = \mathbf{u}$, which is when $P(\bar{a}) \notin D$ and $\Psi_P(\bar{a})^{\mathcal{K}_\mathcal{L}} \neq \mathbf{t}$. In that case, we show the existence of models I and I' of $\mathcal{M}(D, \mathcal{L})$ making $P(\bar{a})$ \mathbf{t} , resp. \mathbf{f} . The database D itself represents a model of $\mathcal{M}(D, \mathcal{L})$ in which $P(\bar{a})$ is false. In constructing a model in which $P(\bar{a})$ is true, we exploit the fact that $\Psi_P(\bar{a})^{\mathcal{K}_\mathcal{L}} \neq \mathbf{t}$. Since Ψ_P is a conjunction of literals, there is a literal l s.t. $l^{\mathcal{K}_\mathcal{L}} \neq \mathbf{t}$ and l is less than P in the dependency order. Using the induction hypothesis we construct a model I' of $\mathcal{M}(D, \mathcal{L})$ s.t. $P(\bar{a})^{I'} = \mathbf{t}$ but $l^{I'} = \mathbf{f}$ and hence $\Psi_P(\bar{a})^{I'} = \mathbf{f}$. \square

Note 2. As Example 5 shows, the requirement in Theorem 2 that Ψ_θ should be a conjunction of literals is indeed necessary.

Example 8. Consider a database in which $D = \{P_1(a), P_1(b), P_2(a), Q(c)\}$ and \mathcal{L} is the set of LCWA of Example 6 without LCWA $(Q(x), S(x))$. That is,

$$\mathcal{L} = \left\{ \begin{array}{lll} \text{LCWA}(P_1(x), \mathbf{t}) & \text{LCWA}(P_2(x), \mathbf{t}) & \text{LCWA}(Q(x), P_1(x) \wedge P_2(x)) \\ \text{LCWA}(S(x), Q(x)) & \text{LCWA}(R(x), Q(x)) & \end{array} \right\}$$

This database is hierarchically closed as the dependency graph induced by \mathcal{L} is acyclic. Also, as each window of expertise is a conjunction of literals, the conditions of Theorem 2 are satisfied in this case. Clearly, $Q(a)^{\mathcal{K}_\mathcal{L}} = \mathbf{f}$, $Q(b)^{\mathcal{K}_\mathcal{L}} = \mathbf{u}$ and $Q(c)^{\mathcal{K}_\mathcal{L}} = \mathbf{t}$. By Corollary 2, then, $\mathcal{M}(D, \mathcal{L}) \models \neg Q(a)$ and $\mathcal{M}(D, \mathcal{L}) \models Q(c)$. Moreover, by Theorem 2, $\mathcal{K}_\mathcal{L}$ is an optimal approximation of $\mathcal{M}(D, \mathcal{L})$.

4 Query Answering

In the previous section we presented techniques to compute an (optimal) 3-valued interpretation for the meaning of a database and its set of LCWAs. In this section we show how these interpretations can be used for query answering in incomplete databases.

Definition 12. Given a 3-valued Σ -interpretation \mathcal{K} and a query $Q[\bar{x}]$ in Σ , define:

- *Certain answers:* $Cert(Q[\bar{x}]) = \{\bar{x} \mid Q[\bar{x}]^{\mathcal{K}} = \mathbf{t}\}$.
- *Possible answers:* $Poss(Q[\bar{x}]) = \{\bar{x} \mid Q[\bar{x}]^{\mathcal{K}} \geq \mathbf{u}\}$.

Note 3. The notions of certain and possible answers proposed here differ from definitions considered in some domains of incomplete databases (see for instance [1]), where the certain and possible answers depend on whether an open or closed-world is assumed. Our definition is based on 3-valued semantics and it does not rely on the assumption that is adopted for the database.

Proposition 8. Let $\mathfrak{D} = (D, \mathcal{L})$ be a hierarchically closed database. If a ground tuple \bar{d} is a certain answer for Q in $\mathcal{K}_\mathcal{L}$, then $\mathcal{M}(D, \mathcal{L}) \models Q[\bar{d}]$.

We can not prove in general that for any possible answer \bar{d} for a query Q in $\mathcal{K}_{\mathcal{L}}$, that $\mathcal{M}(D, \mathcal{L}) \cup \{Q[\bar{d}]\}$ is consistent. All we can assure is that the set of possible answers for Q constitutes a safe (and usually quite precise) over-approximation of this set.

Proposition 9. *Let $\mathcal{D} = (D, \mathcal{L})$ be a hierarchically closed database and Q a disjunction of literals. If a ground tuple \bar{d} is a possible answer for Q in $\mathcal{K}_{\mathcal{L}}$ and $\mathcal{K}_{\mathcal{L}} = \mathcal{K}_{opt}(\mathcal{M}(D, \mathcal{L}))$, then $\mathcal{M}(D, \mathcal{L}) \cup \{Q[\bar{d}]\}$ is satisfiable.*

Given a 3-valued Herbrand Σ -interpretation \mathcal{K} and a query $Q[\bar{x}]$ in Σ , we compute certain/possible answers by Algorithm 1 below. In this algorithm, steps 1 and 2 are computed once (unless \mathcal{K} is changed), while step 3 is executed for each new query. Observe that in step 3.a we under-approximate positive occurrences and over-approximate negative occurrences of predicates, while in 3.b we do the converse.

Algorithm 1 : Computing Certain/Possible Answers

- 1: For each predicate P , define $\Sigma' = \{R \in \Sigma : R^{\mathcal{K}} \text{ is 2-valued}\} \cup \{P_u, P_t : P^{\mathcal{K}} \text{ is 3-valued}\}$.
- 2: Define $I^{\mathcal{K}}$ as the 2-valued Herbrand Σ' -interpretation, such that:

- $R^{I^{\mathcal{K}}} = R^{\mathcal{K}}$ if $R^{\mathcal{K}}$ is 2-valued in \mathcal{K} .
- $P_t^{I^{\mathcal{K}}} = \{\bar{d} \mid P(\bar{d})^{\mathcal{K}} = \mathbf{t}\}$.
- $P_u^{I^{\mathcal{K}}} = \{\bar{d} \mid P(\bar{d})^{\mathcal{K}} \geq \mathbf{u}\}$.

- 3: Consider the query $Q[\bar{x}]$.

3.a To obtain **certain** answers for $Q[\bar{x}]$ compute $Q^c[\bar{x}]$ as follows:

- Replace positive occurrences of 3-valued predicates P by P_t .
- Replace negative occurrences of 3-valued predicates P by P_u .

3.b To obtain **possible** answers for $Q[\bar{x}]$ compute $Q^p[\bar{x}]$ as follows:

- Replace positive occurrences of 3-valued predicates P by P_u .
 - Replace negative occurrences of 3-valued predicates P by P_t .
-

Theorem 3. *A ground tuple \bar{d} is a certain (alternatively possible) answer for $Q[\bar{x}]$ in \mathcal{K} iff \bar{d} is an answer to Q^c (alternatively Q^p) in $I^{\mathcal{K}}$.*

Proof. Consider the (non-standard representation of) truth assignment interpreting a formula φ in a pair of structures (I, J) with the same domain, such that positively occurring atoms of φ are interpreted by I , and negatively occurring ones by J . A satisfaction relation \models between these valuations and formulas in Σ is inductively defined as follows:

- $(I, J) \models P(\bar{d})$ iff $I \models P(\bar{d})$, i.e., $\bar{d}^I \in P^I$;
- $(I, J) \models \neg\varphi$ iff $(J, I) \not\models \varphi$;
- $(I, J) \models \varphi \vee \psi$ iff $(I, J) \models \varphi$ or $(I, J) \models \psi$;

- $(I, J) \models \exists \bar{x} \varphi(\bar{x})$ iff there is a $\bar{d} \in \text{dom}(I)$, such that $(I[\bar{x}/\bar{d}], J[\bar{x}/\bar{d}]) \models \varphi(\bar{x})$.

There is a strong link with 3-valued logic. Indeed, when (I, J) is the result of splitting a 3-valued interpretation \mathcal{K} in a 2-valued underestimation I and 2-valued overestimation J (see Section 3.1), then it is well-known that :

- $\varphi^{\mathcal{K}} = \mathbf{f}$ iff $(J, I) \not\models \varphi$;
- $\varphi^{\mathcal{K}} = \mathbf{u}$ iff $(J, I) \models \varphi$ and $(I, J) \not\models \varphi$;
- $\varphi^{\mathcal{K}} = \mathbf{t}$ iff $(I, J) \models \varphi$.

A straightforward consequence of these equalities is $\varphi^{\mathcal{K}} \geq \mathbf{u}$ iff $(J, I) \models \varphi$. Let (I, J) be the pair of 2-valued Herbrand Σ -interpretations associated with \mathcal{K} . By construction of $I^{\mathcal{K}}$ (see step 2 in Algorithm 1), for every 3-valued predicate P , $(P_t)^{I^{\mathcal{K}}} = P^I$ and $(P_u)^{I^{\mathcal{K}}} = P^J$. It is easy to see that for any tuple of domain elements \bar{d} , $(I, J) \models Q[\bar{d}]$ iff $I^{\mathcal{K}} \models Q^c[\bar{d}]$ and $(J, I) \models Q[\bar{d}]$ iff $I^{\mathcal{K}} \models Q^p[\bar{d}]$. It follows that if $I^{\mathcal{K}} \models Q^c[\bar{d}]$, then $Q[\bar{d}]^{\mathcal{K}} = \mathbf{t}$, i.e., \bar{d} is a certain answer for Q , and if $I^{\mathcal{K}} \models Q^p[\bar{d}]$ then $Q[\bar{d}]^{\mathcal{K}} \geq \mathbf{u}$, i.e., \bar{d} is a possible answer for Q . \square

Example 9. Consider the database \mathfrak{D} of Example 8. Let us check the query $Q[x] = P_1(x) \wedge R(x)$ for $x = b$. Let $\mathcal{K}_{\mathcal{L}}$ be the 3-valued interpretation that is induced by \mathfrak{D} (obtained by the inductive construction of Definition 11), and let I be the 2-valued interpretation derived from $\mathcal{K}_{\mathcal{L}}$ by step 2 of Algorithm 1. Since P_1 is 2-valued in $\mathcal{K}_{\mathcal{L}}$, for every constant a , $P_1(a)^{\mathcal{K}_{\mathcal{L}}} = P_1(a)^I$. The predicate R is 3-valued in $\mathcal{K}_{\mathcal{L}}$, and following step 2 we replace it by R_t and R_u in I . It follows that $R_t(b)^I = \mathbf{f}$ and $R_u(b)^I = \mathbf{t}$. Rewriting $Q[x]$ using now step 3, we obtain $Q^c[x] = P_1(x) \wedge R_t(x)$ and $Q^p[x] = P_1(x) \wedge R_u(x)$. It follows that $x = b$ is a possible but not a certain answer for $Q[x]$ (that is, $b \in \text{Poss}(Q) - \text{Cert}(Q)$).

The last example demonstrates query answering in hierarchically closed databases. The whole process can be summarized as follows:

- 1) Given a database $\mathfrak{D} = (D, \mathcal{L})$, check whether it is hierarchically closed (this is simple, as both the construction of the dependency graph and cycle checking for it are polynomial in the size of \mathcal{L}).
- 2) Apply the procedure that is sketched in the paragraph above the proof of Proposition 6 for constructing the 3-valued interpretation $\mathcal{K}_{\mathcal{L}}$ that is induced by \mathfrak{D} .
- 3) Given a query Q , apply Algorithm 1 with $\mathcal{K}_{\mathcal{L}}$ and Q .

5 Related Works

In [6], the authors introduced the concept of closed-world information on a formula Q in the context of logical agents. The idea amounts to the specification of which parts of a logical theory are complete. More formally, an agent has CWI relative to a formula $Q(\bar{x})$ if every ground instance $Q(\bar{a})$ is either entailed by the first-order knowledge base or necessarily false. As argued in this paper (see the discussion after Definition 4), the notions of CWI and LCWA (as presented here) are closely related, but they capture different properties of different entities.

Doherty et al. [5] propose a semantics for the CWI of Etzioni et al. in terms of (second-order) circumscription. This approach generalizes the one in [6] by allowing limited use of negation and disjunction in the representation of CWI, while retaining tractability of reasoning. The query processing in [5] is based on fixpoint semantics. One of the robust characteristics of this approach is the possibility to express inductive definitions by means of CWI formulas. The correspondence between this approach and our notion of LCWA is considered in detail in [3].

The database community has made important contributions to the study of locally complete databases. In [12], Motro formalizes the concept of partial completeness in relational databases by means of *completeness constraints*, which are concerned with “true information that must be part of a complete database”. Such constraints are represented in the conjunctive relational calculus. The role of Motro’s completeness constraints is taken here by the windows of expertise, which are represented by a more expressive language (function-free first-order). In [11], Levy investigates how to determine whether the answer to a given query is complete even if the database is incomplete. The notion of completeness is semantically characterised in terms of *virtual relations* (those that are true in the real world) and *available relations* (the actual database). Neither Motro nor Levy address the problem of obtaining possible answers to queries, as we do in this paper.

In answer-set programming, Gelfond and Lifschitz [7] introduce the possibility to *partially* define a predicate by combining in one rule classical negation and negation as failure. Consider, for instance, the following program⁵: $\{\neg p \leftarrow \text{not } p, \varphi\}$. Under stable-models semantics [8], the truth of $\neg p$ can be established if no evidence is found about p and φ is provable. The relationship with the LCWA in this context is clear: φ is a window of expertise of p .

6 Conclusions and Future Work

We have presented a general algebraic fixpoint theory for the local closed-world assumption of relational databases. The framework proposed here re-conciliates the notions of CWI and LCWA used in the contexts of knowledge-base agents and relational databases. More specifically, (i) important cases in which LCWA induces CWI are identified, and (ii) a simple query answering mechanism that allows to distinguish between possible and certain answers is introduced. Future lines of research include the following topics:

- **Extension from relational to deductive databases.** In deductive databases (sometimes referred as Datalog), so called intensional predicates are defined in terms of extensional relations. Given a relational database and set of LCWAs, it would be informative to identify where locally complete extensional relations induce complete knowledge on intensional ones. The immaterial nature of intensional predicates presumably requires some extension of the notion of LCWA considered in this paper.

⁵ Following standard conventions, the symbols \neg and not represent classical negation and negation as failure, respectively.

- **Integration with mediator-based systems [10].** In [3] it was shown how to represent LCWA information over a number of different data-sources. Informally, the idea amounts to represent that a set of data-sources, taken together, may store complete knowledge about certain predicates. An open question is how to explore this additional knowledge to retrieve more informative answers from queries to the mediator. An approach based on the semantic considerations presented in this paper could provide a well-founded solution to the elusive problem of answering negative queries from mediator systems.
- **Efficient query answering techniques for the LCWA.** The methods we presented here for query answering require the computation of a 3-valued interpretation. This approach allows a straightforward identification of the exact knowledge endorsed by a database, but, if the database or the set of the LCWA expressions is updated, the 3-valued interpretation must be re-computed. Incremental methods for updating 3-valued interpretations may be incorporated for reducing the computational complexity of the revision process.

References

1. S. Abiteboul and O.M. Duschka. Complexity of answering queries using materialized views. In *Proc. 17th PODS*, pages 254–263, 1998.
2. A. Cortés-Calabuig, M. Denecker, O. Arieli, and M. Bruynooghe. Representation of partial knowledge and query answering in locally complete databases. *Technical Report 457, Department of Computer Science, K.U. Leuven, August 2006*, 2006.
3. 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, 2005.
4. M. Denecker, V.W. Marek, and M. Truszczyński. Uniform semantic treatment of default and autoepistemic logics. *Artificial Intelligence*, 143(1):79–122, 2003.
5. P. Doherty, W. Łukaszewicz, and A. Szalas. Efficient reasoning using the local closed-world assumption. In *Proc. 9th AIMSA, LNCS 2407*, pages 49–58, 2000.
6. O. Etzioni, K. Golden, and D. Weld. Sound and efficient closed-world reasoning for planning. *Artificial Intelligence*, 89(1-2):113–148, 1997.
7. M. Gelfond and V. Lifschitz. Logic programs with classical negation. In *Proc. 7th ICLP*, pages 579–597, 1990.
8. M. Gelfond and V. Lifschitz. Classical negation in logic programs and disjunctive databases. *New Generation Comput.*, 9(3/4):365–386, 1991.
9. G. Grahne. Information integration and incomplete information. *IEEE Data Engineering Bulletin*, 25(3):46–52, 2002.
10. M. Lenzerini. Data integration: A theoretical perspective. In *Proc. 21st PODS*, pages 233–246, 2002.
11. A.Y. Levy. Obtaining complete answers from incomplete databases. In *Proc. 22nd VLDB*, pages 402–412, 1996.
12. A. Motro. Integrity = validity + completeness. *ACM Trans. Database Syst.*, 14(4):480–502, 1989.
13. R. Reiter. Towards a logical reconstruction of relational database theory. In *Conceptual Modelling*, pages 191–233, 1982.