Approximate Reasoning in First-Order Logic Theories

Johan Wittocx and **Maarten Mariën** and **Marc Denecker**
Department of Computer Science, K.U. Leuven, Belgium
{johan.wittocx,maarten.marien,marc.denecker}@cs.kuleuven.be

Abstract

Many computational settings are concerned with finding (all) models of a first-order logic theory for a fixed, finite domain. In this paper, we present a method to compute from a given theory and finite domain an *approximate structure*: a structure that approximates all models. We show confluence of this method and investigate its complexity. We discuss some applications, including 3-valued query answering in integrated and partially incomplete databases, and improved grounding in the context of model expansion for first-order logic.

Introduction

In this paper, a technique is developed to approximate all models of a given first-order logic (FO) theory that share a fixed finite domain. The method is an any-time algorithm that takes as input a FO theory and a finite three-valued structure specifying the domain and (partial) knowledge about the predicate and function symbols, and that produces a refined three-valued structure that approximates all models of the theory that are compatible with the input structure. Such a technique is useful for many reasoning tasks in the context of incomplete knowledge.

As an example, consider a database application allowing university students to make up their teaching program by selecting certain didactic modules and courses. Below, we illustrate some integrity constraints that might be imposed on the selections:

$$\neg(Selected(c_1) \land Selected(c_2)) \qquad (1)$$

$$\exists m\ Module(m) \land Selected(m) \qquad (2)$$

$$\forall m\ (Module(m) \land Selected(m) \supset \forall c\ (Course(c) \land In(m,c) \supset Selected(c))) \qquad (3)$$

The first constraint states that courses c_1 and c_2 are mutually exclusive, the second one expresses that at least one module should be taken and the third one ensures that all courses of a selected module are selected. Assume that, at some point in the application, an *incomplete* database is obtained, specifying that some student has selected some modules or courses, that the student has rejected some others, and that he or she is still undecided about the rest. It is possible now to use the integrity constraints to derive more complete information about what will be the final selection of the student. E.g., if course c_1 has been selected, we can derive that c_2 is not going to be in the selection. If c_2 is rejected from the selection and belongs to module m_2, then m_2 cannot be selected. If m_1 is the only module that still can be selected, m_1 will be in the students selection. Etc. Clearly, this form of inference could be useful for integrity checking and for query answering in incomplete databases under integrity constraints (Greco and Molinaro 2007; Cortés-Calabuig et al. 2006).

In general, a method to perform the above sort of derivations is by computing the set $\mathcal{S}$ of all models of a theory of integrity constraints that extend the given incomplete database. In the above example, if a course or module is selected in all models in $\mathcal{S}$, it will occur in the ultimate selection of the student. Vice versa, if it is not selected in any model in $\mathcal{S}$, it cannot be selected by the student without causing an integrity violation. The problem with the sketched method is that, in general, the size of $\mathcal{S}$ is exponential in the size of the database, making this method impractical in realistic applications. Therefore, it is useful to develop techniques that compute an approximation of the desired results without having to compute $\mathcal{S}$. The method presented in this paper accomplishes this by starting from a partial structure and gradually refining it by propagating the individual integrity constraints, thus producing a converging sequence of gradually more precise partial structures, each of which approximates all elements of $\mathcal{S}$. We show that if the FO formulas are in a specific form, called Equivalence Normal Form, the algorithm is tractable, and that each FO theory can be brought in this form in linear time. To further reduce the complexity of the method, we investigate a variant of the algorithm that uses symbolic representations of the partial structures that approximate $\mathcal{S}$. We also briefly study the accuracy of the computed approximation. Finally, we discuss applications, not only in the context of partially incomplete or integrated databases (Grahne and Mendelzon 1999; Cortés-Calabuig et al. 2007), but also in the context of grounding for FO model expansion problems (Mitchell and Ternovska 2005).

Preliminaries

First-Order Logic (FO)

We assume the reader is familiar with FO. We introduce the conventions and notations used throughout this paper.

A vocabulary Σ consists of $\bot$, $\top$, variables, function and predicate symbols . Variables are denoted by lowercase letters, predicate and function symbols by uppercase letters. Tuples of variables are denoted by $\overline{x}$, $\overline{y}$, etc. Abusing notation, we also use $\overline{x}$ to denote the set of variables occurring in the tuple $\overline{x}$. For a formula φ, we often write $\varphi[\overline{x}]$, to indicate that $\overline{x}$ are precisely the free variables of φ.

A formula is in *term normal form* (TNF) when all its atomic subformulas are of the form $P(x_1, \dots, x_n)$, $F(x_1, \dots, x_n) = x_{n+1}$ or $x_1 = x_2$, where $x_1, \dots, x_{n+1}$ are $n + 1$ different variables. Each formula is equivalent to a formula in TNF.

The truth values *true* and *false* are denoted by **t** and **f**. A Σ-interpretation I consists of a domain D and an assignment of appropriate values to each of its symbols, i.e.:

- **t** to $\top$ and **f** to $\bot$;
- an element $x^I \in D$ to every variable $x \in \Sigma$;
- a relation $P^I \subseteq D^n$ to every n-ary predicate symbol $P \in \Sigma$;
- a function $F^I : D^n \to D$ to every n-ary function symbol $F \in \Sigma$.

An interpretation for only the predicate and function symbols of Σ is called a Σ-*structure*. The restriction of a Σ-interpretation I to a subvocabulary $\sigma \subseteq \Sigma$ is denoted by $I|_\sigma$. For a variable x and an element $d \in D$, $I[x/d]$ is the interpretation that assigns d to x and corresponds to I on all other symbols. This notation is extended to tuples of variables and domain elements of the same length. The truth value $I(\varphi)$ of a formula φ in I, and the satisfaction relation $\models$ are defined as usual (see, e.g., (Enderton 1972)). Abusing notation, we write $I(\varphi[\overline{d}])$ and $I \models \varphi[\overline{d}]$ instead of $I[\overline{x}/\overline{d}](\varphi[\overline{x}])$, respectively $I[\overline{x}/\overline{d}] \models \varphi[\overline{x}]$.

Approximate Structures

In this section, the notion of an *approximate structure* for a vocabulary Σ is introduced. To ease the presentation, we assume from now on that a vocabulary contains no function symbols. This assumption can be made without loss of generality, because there exists a standard transformation[1] from any theory T, to a theory T' containing no function symbols and such that there is a one-to-one correspondence between the models of T and T'. A relation R can be approximated by providing a lower and upper bound for R. The lower bound expresses which tuples certainly belong to R. Vice versa, the upper bound states which tuples possibly belong to R.

Definition 1. An n-ary *approximate relation* $\tilde{R}$ is a tuple $\langle \tilde{R}^l, \tilde{R}^u \rangle$ of two n-ary relations. $\tilde{R}$ *approximates* an n-ary relation R if $\tilde{R}^l \subseteq R \subseteq \tilde{R}^u$.

An approximate relation $\tilde{R}$ is called *consistent* if $\tilde{R}^l \subseteq \tilde{R}^u$, i.e., it approximates at least one relation. It is called *exact* if $\tilde{R}^l = \tilde{R}^u$, i.e., it approximates precisely one relation. If $\tilde{R}$ is exact, we identify $\tilde{R}$ with the unique relation R it approximates and write R instead of $\langle R, R \rangle$. An approximate relation $\tilde{R}_1$ is less precise than an approximate relation $\tilde{R}_2$, denoted $\tilde{R}_1 \leq_p \tilde{R}_2$, if $\tilde{R}_1^l \subseteq \tilde{R}_2^l$ and $\tilde{R}_1^u \supseteq \tilde{R}_2^u$. If $\tilde{R}_1 \leq_p \tilde{R}_2$, then $\tilde{R}_1$ approximates all relations approximated by $\tilde{R}_2$.

Approximate structures are defined similarly to approximate relations.

Definition 2. An *approximate* Σ-*structure* $\tilde{I}$ with domain D is a tuple $\langle \tilde{I}^l, \tilde{I}^u \rangle$ of two Σ-structures with domain D. The approximate relation $P^{\tilde{I}}$ assigned by $\tilde{I}$ to a predicate symbol $P \in \Sigma$ is the approximate relation $\langle P^{\tilde{I}^l}, P^{\tilde{I}^u} \rangle$. $\tilde{I}$ *approximates* a Σ-structure I with domain D if for each predicate symbol $P \in \Sigma$, P^I is approximated by $P^{\tilde{I}}$.

The notions of consistency, exactness and precision pointwise extend to approximate structures. I.e., $\tilde{I}$ is consistent (exact) if for all predicate symbols P, $P^{\tilde{I}}$ is consistent (exact) and $\tilde{I}_1 \leq_p \tilde{I}_2$ if for all predicate symbols P, $P^{\tilde{I}_1} \leq_p P^{\tilde{I}_2}$. Note that $\tilde{I}$ approximates a structure I iff $\tilde{I}$ is less precise than the exact approximate structure $\langle I, I \rangle$. The least precise approximate structure with domain D is denoted $\bot^D_{\leq_p}$. Vice versa, the most precise one is denoted $\top^D_{\leq_p}$. $\bot^D_{\leq_p}$ approximates all structures with domain D, $\top^D_{\leq_p}$ is inconsistent. The size $|\tilde{I}|$ of $\tilde{I}$ is defined as the size of its domain D.

The truth value $\tilde{I}(\varphi)$ of a formula φ in an approximate structure $\tilde{I}$ with domain D is inductively defined as follows:

- $\tilde{I}(\varphi) = \tilde{I}^l(\varphi)$ if φ is atomic;
- $\tilde{I}(\neg\varphi) = \neg(\langle \tilde{I}^u, \tilde{I}^l \rangle(\varphi))$;
- $\tilde{I}(\varphi \land \psi) = \tilde{I}(\varphi) \land \tilde{I}(\psi)$;
- $\tilde{I}(\exists x\ \varphi[x, \overline{y}]) =$ **t** iff for some $d \in D$, $\tilde{I}(\varphi[d, \overline{y}]) =$ **t**. Otherwise, $\tilde{I}(\exists x\ \varphi[x, \overline{y}]) =$ **f**.

Observe that if a structure I is approximated by $\tilde{I}$, then $\tilde{I}(\varphi)$ underestimates $I(\varphi)$. I.e., if $\tilde{I}(\varphi) =$ **t**, then $I(\varphi) =$ **t**. As such, we say that φ is *certainly true* according to $\tilde{I}$ if $\tilde{I}(\varphi) =$ **t**. Vice versa, $\langle \tilde{I}^u, \tilde{I}^l \rangle(\varphi)$ is an overestimation of $I(\varphi)$. I.e., if $I(\varphi) =$ **t**, then $\langle \tilde{I}^u, \tilde{I}^l \rangle(\varphi) =$ **t** and we say that φ is *possibly true* in $\tilde{I}$. Similarly, φ is *certainly false* in $\tilde{I}$ if $\tilde{I}(\neg\varphi) =$ **t**, or, equivalently, $\langle \tilde{I}^u, \tilde{I}^l \rangle(\varphi) =$ **f**. Finally, φ is *possibly false* in $\tilde{I}$ if $\tilde{I}(\varphi) =$ **f**.

Definition 3. An approximate structure $\tilde{I}$ with domain D is an *approximation of a theory* T, if it approximates all models of T with domain D. An approximate structure $\tilde{J}$ with domain D is an *approximation of* T *above* $\tilde{I}$ if $\tilde{I} \leq_p \tilde{J}$ and all models of T approximated by $\tilde{I}$ are approximated by $\tilde{J}$.

[1] First transform T to an equivalent theory in TNF. Then replace each occurrence of an atom $F(\overline{x}) = y$ in the resulting theory by $G_F(\overline{x}, y)$, where G_F is a new predicate symbol, and add the sentences $\forall \overline{x}\ \exists y\ G_F(\overline{x}, y)$ and $\forall \overline{x}, y_1, y_2\ (G_F(\overline{x}, y_1) \land G_F(\overline{x}, y_2) \supset y_1 = y_2)$.

Approximate Reasoning

As mentioned in the introduction, we are interested in approximations for a given theory T (e.g., the integrity constraints of a database) above a given approximate structure $\tilde{I}$ (e.g., the tables of an incomplete database) with finite domain D. The most precise approximation that can be obtained is given by $\mathrm{glb}_{\leq_p}(\{\langle M, M\rangle \mid \langle M, M\rangle \geq_p \tilde{I}$ and $M \models T\})$. We call this the *optimal approximation* and denote it by $\mathcal{O}^T(\tilde{I})$. If T has no models approximated by $\tilde{I}$, then $\mathcal{O}^T(\tilde{I}) = \top^D_{\leq_p}$. In particular, this is the case when $\tilde{I}$ is inconsistent.

Example 1. Consider the theory $T_1 = \{(1); (2); (3)\}$ of the sentences of the introduction and let $\tilde{I}$ be the approximate structure with domain $D = \{m_1, m_2, c_1, c_2, c_3, c_4\}$ given by $Module^{\tilde{I}} = \{m_1, m_2\}$, $Course^{\tilde{I}} = \{c_1, c_2, c_3, c_4\}$, $In^{\tilde{I}} = \{(m_1, c_1); (m_1, c_3), (m_2, c_2)\}$ and $Selected^{\tilde{I}} = \langle\{c_1\}, D\rangle$. Then $\mathcal{O}^{T_1}(\tilde{I})$ assigns $\langle\{m_1, c_1, c_3\}, \{m_1, c_1, c_3, c_4\}\rangle$ to $Selected$.

The problem of constructing $\mathcal{O}^T(\tilde{I})$ is at least as hard as deciding whether T has a model approximated by $\tilde{I}$, which is intractable. Even for a fixed T and varying $\tilde{I}$, this decision problem is NP-complete (Fagin 1974).

The usual complexity measure for algorithms in the context of databases with integrity constraints is called *data complexity*, and is defined as the complexity in the size $|\tilde{I}|$ of $\tilde{I}$. I.e., for this measure, the theory T is considered to be fixed. All complexity results in this paper concern data complexity. In this section we develop a method to compute an approximation for T above $\tilde{I}$ in polynomial time in $|\tilde{I}|$.

The method we present consists of computing and combining the optimal approximations for individual sentences of T, instead of computing the optimal approximation for T as a whole. The resulting approximate structure $\tilde{J}$ is not guaranteed to be as precise as $\mathcal{O}^T(\tilde{I})$. Since an individual sentence can be as complex as a theory, this computation can be as hard as that of $\mathcal{O}^T(\tilde{I})$. We therefore proceed by introducing a normal form for which this computation is guaranteed to be polynomial, and provide a linear reduction of arbitrary theories to this normal form.

Refinement Sequences

For the rest of this paper, let T be a theory and $\tilde{I}$ an approximate structure with finite domain D.

Similarly as for theories, we associate to each sentence φ the operator $\mathcal{O}^\varphi$ which maps an approximate structure $\tilde{I}$ to the optimal approximation of φ above $\tilde{I}$, i.e., $\mathcal{O}^\varphi(\tilde{I}) = \mathrm{glb}_{\leq_p}(\{\langle M, M\rangle \mid \langle M, M\rangle \geq_p \tilde{I}$ and $M \models \varphi\})$. Observe that for a given $\varphi \in T$, $\{\langle M, M\rangle \geq_p \tilde{I} \mid M \models \varphi\} \supseteq \{\langle M, M\rangle \geq_p \tilde{I} \mid M \models T\}$, hence $\mathcal{O}^T(\tilde{I})$ is more precise than $\mathcal{O}^\varphi(\tilde{I})$. Also note that for any sentence φ, $\mathcal{O}^\varphi$ is a $\leq_p$-monotone operator on the lattice of approximate structures with domain D.

Lemma 4. *If φ is a sentence of T, then $\mathcal{O}^\varphi(\tilde{I})$ is more precise than $\tilde{I}$ and approximates T above $\tilde{I}$.*

Lemma 4 shows the soundness of the following procedure for computing approximations $\tilde{J}$ of T above $\tilde{I}$. Start by setting $\tilde{J} = \tilde{I}$. Then choose a sentence φ of T and replace $\tilde{J}$ by $\mathcal{O}^\varphi(\tilde{J})$. Repeat.
Interesting properties of this procedure are termination and confluence. To state these formally, we introduce the concept of a *refinement sequence*.

Definition 5. A *refinement sequence for T above $\tilde{I}$* is a sequence $\langle \tilde{J}_i\rangle_{0\leq i\leq n}$ of approximate structures such that $\tilde{J}_0 = \tilde{I}$, $\tilde{J}_i <_p \tilde{J}_{i+1}$ and $\tilde{J}_{i+1} = \mathcal{O}^\varphi(\tilde{J}_i)$ for some sentence φ of T. A refinement sequence is called *terminal* if it cannot be extended anymore. For a terminal refinement sequence $\langle \tilde{J}_i\rangle_{0\leq i\leq n}$, the structure $\tilde{J}_n$ is called its *limit*.

Theorem 6. *Every refinement sequence for T above $\tilde{I}$ is finite and every terminal refinement sequence has the same limit.*

We denote the unique limit of a terminal refinement sequence for T above $\tilde{I}$ by $\mathfrak{D}^T(\tilde{I})$.

Example 1 (continued). Let $\langle \tilde{I}_i\rangle_{0\leq i\leq 4}$ be the refinement sequence for T_1 above $\tilde{I}$ given by $\tilde{I}_1 = \mathcal{O}^{(1)}(\tilde{I}_0)$, $\tilde{I}_2 = \mathcal{O}^{(3)}(\tilde{I}_1)$, $\tilde{I}_3 = \mathcal{O}^{(2)}(\tilde{I}_2)$ and $\tilde{I}_4 = \mathcal{O}^{(3)}(\tilde{I}_3)$. One can verify that $c_2 \notin Selected^{\tilde{I}^u_1}$, i.e., in $\tilde{I}_1$ it is already derived that c_2 cannot be selected. Also, $m_2 \notin Selected^{\tilde{I}^u_2}$, $m_1 \in Selected^{\tilde{I}^l_3}$ and $c_3 \in Selected^{\tilde{I}^l_4}$. Hence, $\tilde{I}_4 = \mathcal{O}^{T_1}(\tilde{I})$, the refinement sequence is terminal and $\mathfrak{D}^{T_1}(\tilde{I}) = \mathcal{O}^{T_1}(\tilde{I})$.

It is not necessarily the case that $\mathfrak{D}^T(\tilde{I})$ is equal to $\mathcal{O}^T(\tilde{I})$. E.g., if $T = \{P \equiv Q; P \equiv \neg Q\}$ and $\tilde{I} = \bot^D_{\leq_p}$, then $\mathfrak{D}^T(\tilde{I}) = \bot^D_{\leq_p}$. However, because T has no model, the optimal approximation $\mathcal{O}^T(\tilde{I})$ is equal to $\top^D_{\leq_p}$.

Also, for two logically equivalent theories T and T', $\mathfrak{D}^T(\tilde{I})$ and $\mathfrak{D}^{T'}(\tilde{I})$ are not necessarily the same. E.g., this is not the case for $\tilde{I} = \bot^D_{\leq_p}$, $T = \{P \equiv Q; P \equiv \neg Q\}$ and $T' = \{(P \equiv Q) \wedge (P \equiv \neg Q)\}$.

Equivalence Normal Form

The length of a terminal refinement sequence for T above $\tilde{I}$ is polynomial[2] in $|\tilde{I}|$. Consequently, if for every sentence φ of T, $\mathcal{O}^\varphi$ is computable in polynomial time, a terminal refinement sequence for a fixed T can be computed in polynomial time.

We now define a normal form for which $\mathcal{O}^\varphi$ is computable in polynomial time.

Definition 7. A TNF sentence φ is in *equivalence normal form* (ENF) if φ is of the form $\forall \overline{x}\ (P(\overline{x}) \equiv \psi[\overline{y}])$, such that P is a predicate symbol or the symbol $\top$, $\overline{y} \subseteq \overline{x}$ and $\psi[\overline{y}]$ is of the form $Q(\overline{y})$, $\neg Q(\overline{y})$, $Q(\overline{y}) \wedge R(\overline{y})$, $Q(\overline{y}) \vee R(\overline{y})$, $\forall v\ Q(\overline{y}, v)$, $\exists v\ Q(\overline{y}, v)$, $y_1 = y_2$, where Q and R are

[2] Note that the length of a refinement sequence can be exponential in the arity of the symbols occurring in T.

predicate symbols, different from P. A theory is in ENF if all its sentences are.

Note that for ENF sentences of the form $\forall\overline{x}\ (P(\overline{x}) \equiv Q(\overline{y}) \wedge R(\overline{y}))$ and $\forall\overline{x}\ (P(\overline{x}) \equiv Q(\overline{y}) \vee R(\overline{y}))$, the variables occurring in Q and R are exactly the same.

The following transformation (akin to the Tseitin transformation for propositional logic (Tseitin 1968)) reduces arbitrary theories T to ENF theories T', such that the models of T' restricted to T's vocabulary are the models of T. Start by transforming T to TNF. Then replace every sentence φ of the resulting theory by $\top \equiv \varphi$. Finally, as long as the theory contains sentences $\forall\overline{x}\ (P(\overline{x}) \equiv \psi[\overline{y}])$ that are not in ENF, replace a subformula $\chi[\overline{z}]$ of ψ by $Q(\overline{z})$, where Q is a new predicate symbol, and add the sentence $\forall\overline{z}\ (Q(\overline{z}) \equiv \chi[\overline{z}])$ to the theory. Note that at most one new predicate symbol is introduced for each node in the parse tree of φ. Hence, the size of T' is linear in the size of T.

Example 2. Applying the transformation outlined above on sentence (2) from the introduction yields the two sentences $\top \equiv \exists m\ Q(m)$ and $\forall m\ (Q(m) \equiv Module(m) \wedge Selected(m))$, where Q is a new predicate symbol.

We now show that for any ENF sentence φ, $\mathcal{O}^{\varphi}$ is computable in polynomial time. The method we present is based on the simple observation that if M is a model of the ENF sentence $\forall\overline{y},\overline{z}\ (P(\overline{y},\overline{z}) \equiv \psi[\overline{y}])$ and $\overline{d}_y, \overline{d}_z$ are tuples of domain elements, then $M(P(\overline{d}_y,\overline{d}_z)) = M(\psi[\overline{d}_y])$. This has the following consequences:

1. If $\tilde{I}(\psi[\overline{d}_y]) = \mathbf{t}$, i.e., $\psi[\overline{d}]$ is certainly true in $\tilde{I}$, then for any model M of φ approximated by $\tilde{I}$, it holds that $M(P(\overline{d}_y,\overline{d}_z)) = M(\psi[\overline{d}_y]) = \mathbf{t}$. Therefore, $\tilde{I}(\psi[\overline{d}_y]) = \mathbf{t}$ implies that $\mathcal{O}^{\varphi}(\tilde{I})(P(\overline{d}_y,\overline{d}_z)) = \mathbf{t}$. Similarly, if $\psi[\overline{d}_y]$ is certainly false in $\tilde{I}$, then so is $P(\overline{d}_y,\overline{d}_z)$ in $\mathcal{O}^{\varphi}(\tilde{I})$.

2. Vice versa, if $\tilde{I}(P(\overline{d}_y,\overline{d}_z)) = \mathbf{t}$, then $M(\psi[\overline{d}_y]) = \mathbf{t}$ in any model M of φ approximated by $\tilde{I}$. Therefore, if $\tilde{I}(P(\overline{d}_y,\overline{d}_z)) = \mathbf{t}$ and there is only one way to ensure that $\tilde{J}(\psi[\overline{d}_y]) = \mathbf{t}$ in an approximate structure $\tilde{J} \geq_p \tilde{I}$, this is reflected in $\mathcal{O}^{\varphi}(\tilde{I})$. We illustrate this on two examples:
   - If $\psi[\overline{y}]$ is the formula $Q(\overline{y}) \wedge R(\overline{y})$, then the only way to ensure that $\mathcal{O}^{\varphi}(\tilde{I})(\psi[\overline{d}_y]) = \mathbf{t}$, is by making both $Q(\overline{d}_y)$ and $R(\overline{d}_y)$ certainly true in $\mathcal{O}^{\varphi}(\tilde{I})$. As such, $\tilde{I}(P(\overline{d}_y,\overline{d}_z)) = \mathbf{t}$, then $\mathcal{O}^{\varphi}(\tilde{I})(Q(\overline{d}_y)) = \mathbf{t}$ and $\mathcal{O}^{\varphi}(\tilde{I})(R(\overline{d}_y)) = \mathbf{t}$.
   - If $\psi[\overline{y}]$ is the formula $Q(\overline{y}) \vee R(\overline{y})$, $\tilde{I}(P(\overline{d}_y,\overline{d}_z)) = \mathbf{t}$ and $\tilde{I}(\neg R(\overline{d}_y)) = \mathbf{t}$, then $\mathcal{O}^{\varphi}(\tilde{I})(Q(\overline{d}_y)) = \mathbf{t}$.

   A similar observation can be made if $P(\overline{d}_y,\overline{d}_z)$ is certainly false in $\tilde{I}$.

3. For tuples $\overline{d}_y$, $\overline{d}_z$ and $\overline{d}_z'$ of domain elements and any model M of φ, $M(P(\overline{d}_y,\overline{d}_z)) = M(\psi[\overline{d}_y]) = M(P(\overline{d}_y,\overline{d}_z'))$. Therefore, if there exists a tuple $\overline{d}_z$ such that $\tilde{I}(P(\overline{d}_y,\overline{d}_z)) = \mathbf{t}$, then for all tuples $\overline{d}_z'$, $\mathcal{O}^{\varphi}(\tilde{I})(P(\overline{d}_y,\overline{d}_z')) = \mathbf{t}$. Similarly if $\tilde{I}(P(\overline{d}_y,\overline{d}_z)) = \mathbf{f}$.

We can now show how to compute $\mathcal{O}^{\varphi}$. Lemma 8, based on items 1 and 3 above, shows how to compute $P^{\mathcal{O}^{\varphi}(\tilde{I})}$; Lemma 9, based on item 2 above, does the same for the predicates that occur in ψ.

Lemma 8. *Let φ be the ENF sentence $\forall\overline{y},\overline{z}\ (P(\overline{y},\overline{z}) \equiv \psi[\overline{y}])$. Denote by $\xi[\overline{y}]$ the formula $\exists\overline{z}\ P(\overline{y},\overline{z})$ and by $\xi^{\neg}[\overline{y}]$ the formula $\exists\overline{z}\ \neg P(\overline{y},\overline{z})$. If $\mathcal{O}^{\varphi}(\tilde{I})$ is consistent, then for any tuple of domain elements $\overline{d}_y$ and $\overline{d}_z$, $\mathcal{O}^{\varphi}(\tilde{I})^l(P(\overline{d}_y,\overline{d}_z)) = \tilde{I}(\psi[\overline{d}_y] \vee \xi[\overline{d}_y])$ and $\mathcal{O}^{\varphi}(\tilde{I})^u(P(\overline{d}_y,\overline{d}_z)) = \neg\tilde{I}(\neg\psi[\overline{d}_y] \vee \xi^{\neg}[\overline{d}_y])$. If $\mathcal{O}^{\varphi}(\tilde{I})$ is inconsistent, then either $\tilde{I}$ is inconsistent or there exist a tuple $\overline{d}_y$ such that $\tilde{I}(\psi[\overline{d}_y] \vee \xi[\overline{d}_y]) = \tilde{I}(\neg\psi[\overline{d}_y] \vee \xi^{\neg}[\overline{d}_y])$.*

Lemma 9. *Let φ, $\xi[\overline{y}]$ and $\xi^{\neg}[\overline{y}]$ be as in the previous lemma and let Q be a predicate that occurs in ψ. If $\mathcal{O}^{\varphi}(\tilde{I})$ is consistent, then $\mathcal{O}^{\varphi}(\tilde{I})^l(Q(\overline{d})) = \tilde{I}(Q(\overline{d}) \vee \chi_{ct}[\overline{d}])$ and $\mathcal{O}^{\varphi}(\tilde{I})^u(Q(\overline{d})) = \neg\tilde{I}(\neg Q(\overline{d}) \vee \chi_{cf}[\overline{d}])$, where χ_{ct} and χ_{cf} are the formulas that are defined as follows, depending on ψ.*

ψ	χ_{ct}	χ_{cf}
$Q(\overline{y})$	ξ	$\xi^{\neg}$
$Q(\overline{y}) \wedge R(\overline{y})$	ξ	$\xi^{\neg} \wedge R(\overline{y})$
$\forall v\ Q(\overline{y}, v)$	ξ	$\xi^{\neg} \wedge \forall v'\ (v \neq v' \supset Q(\overline{y}, v'))$

ψ	χ_{ct}	χ_{cf}
$\neg Q(\overline{y})$	$\xi^{\neg}$	ξ
$Q(\overline{y}) \vee R(\overline{y})$	$\xi \wedge \neg R(\overline{y})$	$\xi^{\neg}$
$\exists v\ Q(\overline{y}, v)$	$\xi \wedge \forall v'\ (v \neq v' \supset \neg Q(\overline{y}, v'))$	$\xi^{\neg}$

Observe that equality atoms in an ENF theory occur only in sentences of the form $\forall P(y_1, y_2, \overline{z}) \equiv y_1 = y_2$. Since the interpretation of $=$ is given and fixed (by the identity relation), such sentences can only be used to refine the interpretation of P, which is described by Lemma 8.

Because the evaluation of a formula φ in an approximate structure $\tilde{I}$ can be done in polynomial time in the size of $\tilde{I}$, lemmas 8 and 9 provide a method to compute $\mathcal{O}^{\varphi}(\tilde{I})$ in polynomial time for any ENF sentence φ. Indeed, first assume that $\mathcal{O}^{\varphi}(\tilde{I})$ is consistent and compute it according to the characterization provided by the lemmas. Then check whether the result is consistent. If so, then it is indeed $\mathcal{O}^{\varphi}(\tilde{I})$. Else, by lemma 8, $\mathcal{O}^{\varphi}(\tilde{I}) = \top^{D}_{\leq_p}$.

Theorem 10. *Let T be a theory in ENF and $\tilde{I}$ an approximate structure with finite domain D. Then a terminal refinement sequence for T above $\tilde{I}$ is computable in time polynomial in $|\tilde{I}|$.*

Hence, the following algorithm can be used to compute an approximation for an arbitrary theory T over a vocabulary Σ above $\tilde{I}$.

1. Transform T to a theory T' over Σ' in ENF and extend $\tilde{I}$ to Σ' by assigning $(\emptyset, D^n)$ to every n-ary predicate symbol $P \in \Sigma' \setminus \Sigma$.
2. Construct a (terminal) refinement sequence $\langle \tilde{I}_i \rangle_{0 \leq i \leq n}$ for T' above $\tilde{I}$ and return $\tilde{I}_n|_{\Sigma}$.

Note that this is an any-time algorithm: the refinement sequence constructed in step 2 can be terminal, but this is not necessary. The following examples illustrate what the algorithm can achieve.

Example 1 (continued). If T_1 is transformed to an ENF theory T' as described below definition 7, then a terminal refinement sequence for T' above $\tilde{I}$ is equal to $\mathcal{O}^T(\tilde{I})$. I.e., the algorithm above on T_1 and $\tilde{I}$ computes the optimal result.

Example 3. Consider the theory T_2, taken from some planning domain, consisting of the sentence $\forall a_0, a_p, t_0\ Prec(a_p, a_0) \wedge Do(a_0, t_0) \supset \exists t_p\ (t_p < t_0 \wedge Do(a_p, t_p))$. This sentence describes that some action a_0 with precondition a_p can only be performed at timepoint t_0 if a_p is performed at an earlier timepoint t_p. Let $\tilde{I}$ be an approximate structure. Then the algorithm above derives that if t_0 is smaller than the ith timepoint and there exists a chain of actions $a_1, \dots, a_i$ such that $\tilde{I}(Prec(a_1, a_0) \wedge \dots \wedge Prec(a_i, a_{i-1}))$ holds, then a_0 can certainly not be performed at timepoint t_0.

In an actual implementation of the algorithm, it is important to construct a good refinement sequence. I.e., a short sequence with a sufficiently precise last element. To find such a sequence, a good heuristic is needed to decide at each step, which operator $\mathcal{O}^\varphi$ to apply to extend the sequence.

A simple proposal consists of keeping track of a set C of predicates whose approximation recently changed. To decide which $\mathcal{O}^\varphi$ to apply, one could search for the predicate $P \in C$ with the greatest difference between its current and previous approximation. Then, apply $\mathcal{O}^\varphi$ for each φ containing an occurrence of P. Finally, if P's approximation did not change by applying these operators, remove P from C.

Also, to efficiently compute the next approximate structure in a refinement sequence, one should try to avoid recomputing the information that is already present in the previous approximate structures. This can be done in a similar way as the semi-naive evaluation technique (Ullman 1988) avoids to recompute already known answers to a datalog query.

Optimality

As mentioned above, it is not necessarily the case that $\mathfrak{O}^T(\tilde{I}) = \mathcal{O}^T(\tilde{I})$. This raises the question how well $\mathfrak{O}^T(\tilde{I})$ approximates $\mathcal{O}^T(\tilde{I})$. Generally spoken, the less axioms T contains, the closer $\mathfrak{O}^T(\tilde{I})$ is to $\mathcal{O}^T(\tilde{I})$, with $\mathfrak{O}^T(\tilde{I}) = \mathcal{O}^T(\tilde{I})$ if T is a singleton. But since translation to ENF tends to split up a theory in many small sentences, the question of the accuracy of the computed approximation is certainly an important one.

Answering this question is far from trivial. Below, we present two simple lemmas that can be helpful in proving $\mathfrak{O}^T(\tilde{I}) = \mathcal{O}^T(\tilde{I})$ for a given T and $\tilde{I}$. Finding more elaborate results is part of future work. Other preliminary results, in the more restricted setting of locally closed databases, are presented in (Cortés-Calabuig et al. 2007).

Lemma 11. *Let $\tilde{I}$ be an approximate structure and let T be the theory $\{\varphi \wedge \chi, \varphi_1, \dots, \varphi_n\}$ and T' the theory $\{\varphi, \chi, \varphi_1, \dots, \varphi_n\}$. If one of the following conditions is satisfied, then $\mathfrak{O}^T(\tilde{I}) = \mathfrak{O}^{T'}(\tilde{I})$:*

- *there is no predicate symbol that occurs in both φ and χ;*
- *$\tilde{I}(\neg\varphi) =$ **t** or $\tilde{I}(\varphi) =$ **t**.*

Lemma 12. *Let T and T' be as in lemma 11 and let $\tilde{I}$ be an approximate structure. If $\langle \tilde{I}_i \rangle_{0 \leq i \leq n}$ is a refinement sequence for T' above $\tilde{I}$ and $\mathfrak{O}^T(\tilde{I}_n) = \mathfrak{O}^{T'}(\tilde{I}_n)$, then $\mathfrak{O}^T(\tilde{I}) = \mathfrak{O}^{T'}(\tilde{I})$.*

The following example illustrates the use of lemmas 11 and 12 to prove that for a certain T and $\tilde{I}$, the transformation to ENF does not result in a less optimal approximation of T above $\tilde{I}$.

Example 4. Consider the theory T_3 over Σ consisting of the sentence $\forall x\ UGCourse(x) \vee (GCourse(x) \wedge Difficult(x))$, which states that a course is either for undergraduates, or for graduates and difficult. An ENF transformation T_4 of T_3 consists of, e.g., the three sentences

$$\top \equiv \forall x\ S_1(x) \tag{4}$$

$$S_1(x) \equiv UGCourse(x) \vee S_2(x) \tag{5}$$

$$S_2(x) \equiv GCourse(x) \wedge Difficult(x) \tag{6}$$

We show that $\mathfrak{O}^{T_4}(\tilde{I})|_\Sigma = \mathcal{O}^{T_3}(\tilde{I})|_\Sigma$ for any approximate structure $\tilde{I}$ for which $UGCourse^{\tilde{I}}$ is exact, i.e., $\tilde{I}$ contains complete information about the undergraduate courses.

Denote by T_5 the theory $\{(4); (5) \wedge (6)\}$ and by T_6 the theory $\{(4) \wedge (5) \wedge (6)\}$. Because clearly, $\mathfrak{O}^{T_6}(\tilde{I}) = \mathcal{O}^{T_6}(\tilde{I})$ and $\mathcal{O}^{T_6}(\tilde{I})|_\Sigma = \mathcal{O}^{T_3}(\tilde{I})|_\Sigma$, it is sufficient to prove that $\mathfrak{O}^{T_6}(\tilde{I}) = \mathfrak{O}^{T_4}(\tilde{I})$.

First observe that $\mathcal{O}^{(4)}(\tilde{I})$ assigns the approximate relation $\langle D, D \rangle$ to S_1, i.e. $\mathcal{O}^{(4)}(\tilde{I})(S_1(d)) =$ **t** for each $d \in D$. Hence, $\mathcal{O}^{(4)}(\tilde{I})((4)) =$ **t** and therefore, by lemma 11, $\mathfrak{O}^{T_5}(\mathcal{O}^{(4)}(\tilde{I})) = \mathfrak{O}^{T_6}(\mathcal{O}^{(4)}(\tilde{I}))$. By lemma 12, also $\mathfrak{O}^{T_5}(\tilde{I}) = \mathfrak{O}^{T_6}(\tilde{I})$.

Because $UGCourse^{\tilde{I}}$ is exact, also $\mathcal{O}^{(5)}(\mathcal{O}^{(4)}(\tilde{I}))((5)) =$ **t**. As before, we conclude by lemma 11 and 12 that $\mathfrak{O}^{T_4}(\tilde{I}) = \mathfrak{O}^{T_5}(\tilde{I})$.

Symbolic Approximate Reasoning

The presented algorithm to compute an approximation for T above some approximate structure $\tilde{I}$ with domain D will often be too expensive for practical purposes. If the vocabulary of T contains a predicate P with arity n, then the approximate relation assigned to P can contain up to $2 \cdot |D|^n$ tuples. For a large D, storing and manipulating such an approximative relation can become infeasible in practice. Note that n is not necessarily small. Even if the original theory T contains no predicates with a large arity, transforming T to an ENF theory creates predicates with arity equal to the number of free variables of the subformula in T it represents.

To obtain a more practical algorithm, the computed refinement sequence can be represented in a compact, symbolic way, independent of D. In this section, we show how such a symbolic representation can be obtained.

Symbolic Approximate Structures

Let T be an ENF theory over a vocabulary Σ and let σ be a vocabulary, not necessarily related to Σ.

Definition 13. A *symbolic approximate* Σ*-structure* $\tilde{\Phi}$ over σ is an assignment of a tuple $\langle \tilde{\Phi}^l_P[\overline{x}], \tilde{\Phi}^u_P[\overline{x}] \rangle$ to each n-ary predicate $P \in \Sigma$, where $\tilde{\Phi}^l_P[\overline{x}]$ and $\tilde{\Phi}^u_P[\overline{x}]$ are two formulas over σ with n free variables.

Given a fixed σ-structure I, to each symbolic approximate Σ-structure $\tilde{\Phi}$ over σ, an approximate Σ-structure is associated, by evaluating it in I. As such, $\tilde{\Phi}$ can be seen as a symbolic representation of this associated approximate Σ-structure.

Definition 14. The *evaluation of a symbolic approximate* Σ*-structure* $\tilde{\Phi}$ *over* σ *in a* σ*-structure* I is the approximate Σ-structure $I(\tilde{\Phi})$, defined by $P^{I(\tilde{\Phi})} = \langle \{\overline{d} \mid I \models \tilde{\Phi}^l_P[\overline{d}]\}, \{\overline{d} \mid I \models \tilde{\Phi}^u_P[\overline{d}]\} \rangle$.

Example 5. Let σ be the vocabulary containing two n-ary relation symbols P^l and P^u for every n-ary relation symbol $P \in \Sigma$. Let $\tilde{I}$ be an approximate Σ-structure, $\tilde{\Phi}$ the symbolic approximate structure over σ assigning $\langle P^l(\overline{x}), P^u(\overline{x}) \rangle$ to every $P \in \Sigma$ and I the σ-structure assigning $P^{\tilde{I}^l}$ to P^l and $P^{\tilde{I}^u}$ to P^u for every $P \in \Sigma$. Then the evaluation of $\tilde{\Phi}$ in I is precisely $\tilde{I}$. Hence, given I, $\tilde{\Phi}$ can be seen as a symbolic representation of $\tilde{I}$.

For a formula φ over Σ, its *evaluation in a symbolic approximative structure* $\tilde{\Phi}$ is the sentence $\tilde{\Phi}(\varphi)$, obtained by, for every predicate symbol $P \in \Sigma$, substituting each positive occurrence of an atom $P(\overline{x})$ by $\Phi^l_P[\overline{x}]$ and each negative occurrence by $\Phi^u_P[\overline{x}]$. First evaluating φ in $\tilde{\Phi}$ and then evaluating the result in a σ-structure I, corresponds to evaluating φ in $I(\tilde{\Phi})$:

Lemma 15. *Let* φ *be a formula over* Σ, $\tilde{\Phi}$ *a symbolic approximative structure over* σ *and* I *a* σ*-structure. Then* $I(\tilde{\Phi}(\varphi)) = I(\tilde{\Phi})(\varphi)$.

Let φ be an ENF sentence. Based on lemmas 8 and 9, we define the operator $\tilde{\mathcal{O}}^\varphi$ on the set of symbolic approximate structures. This operator is the symbolic variant of $\mathcal{O}^\varphi$, in the sense that for any symbolic approximate structure $\tilde{\Phi}$, it holds that $I(\tilde{\mathcal{O}}^\varphi(\tilde{\Phi})) = \mathcal{O}^\varphi(I(\tilde{\Phi}))$.

Definition 16. Let φ be the ENF sentence $\forall \overline{y}, \overline{z}\ (P(\overline{y}, \overline{z}) \equiv \psi[\overline{y}])$. $\tilde{\mathcal{O}}^\varphi$ is the operator on the set of symbolic approximate structures defined by

- $(\tilde{\mathcal{O}}^\varphi(\tilde{\Phi}))^l_P = \tilde{\Phi}(\psi[\overline{x}] \vee \exists \overline{z}\ P(\overline{x}))$ and $(\tilde{\mathcal{O}}^\varphi(\tilde{\Phi}))^u_P = \neg(\tilde{\Phi}(\neg\psi[\overline{x}] \vee \exists \overline{z}\ \neg P(\overline{x})))$;
- for an atom $Q(\overline{y})$ that occurs in ψ, $(\tilde{\mathcal{O}}^\varphi(\tilde{\Phi}))^l_Q = \tilde{\Phi}(\chi_{ct} \vee Q(\overline{y}))$ and $(\tilde{\mathcal{O}}^\varphi(\tilde{\Phi}))^u_Q = \neg(\tilde{\Phi}(\chi_{cf} \vee \neg Q(\overline{y})))$, where χ_{ct} and χ_{cf} are defined as in lemma 9.
- $\tilde{\mathcal{O}}^\varphi(\tilde{\Phi})$ corresponds to $\tilde{\Phi}$ on all predicates that do not occur in φ.

Because of lemmas 8, 9, and 15, we have the following result, as desired.

Proposition 17. *For every ENF sentence* φ*, symbolic approximate structure* $\tilde{\Phi}$ *over* σ *and* σ*-structure* I*, it holds that* $I(\tilde{\mathcal{O}}^\varphi(\tilde{\Phi})) = \mathcal{O}^\varphi(I(\tilde{\Phi}))$ *if* $\mathcal{O}^\varphi(I(\tilde{\Phi}))$ *is consistent. If* $\mathcal{O}^\varphi(I(\tilde{\Phi}))$ *is inconsistent, then so is* $I(\tilde{\mathcal{O}}^\varphi(\tilde{\Phi}))$.

Symbolic Refinement Sequences

Let $\tilde{\Phi}$ be a symbolic approximate structure. Similarly to a refinement sequence, a *symbolic refinement sequence for* T *above* $\tilde{\Phi}$ is a sequence $\langle \tilde{\Phi}_i \rangle_{0 \leq i \leq n}$ of symbolic approximate structures such that $\tilde{\Phi}_0 = \tilde{\Phi}$ and for each $i < n$, there is a sentence φ in T such that $\tilde{\Phi}_{i+1} = \tilde{\mathcal{O}}^\varphi(\tilde{\Phi}_i)$.

Based on proposition 17, an approximate structure for T above $\tilde{I}$ can be obtained by first defining an approximate structure $\tilde{\Phi}$ over a vocabulary σ and a σ-structure such that $I(\tilde{\Phi}) = \tilde{I}$. Example 5 shows a possible definition of these. Then construct a symbolic refinement sequence $\langle \tilde{\Phi}_i \rangle_{0 \leq i \leq n}$ for T above $\tilde{\Phi}$. Finally, return $I(\tilde{\Phi}_n)$.

However, using symbolic refinement sequences has its costs. Note that for any symbolic approximate structure $\tilde{\Phi}$ and ENF sentence φ, the formulas assigned by $\tilde{\mathcal{O}}^\varphi(\tilde{\Phi})$ are strictly larger than the ones assigned by $\tilde{\Phi}$. Also, a symbolic refinement sequence is not guaranteed to be finite. These problems can sometimes be avoided by replacing the formulas assigned by $\tilde{\mathcal{O}}^\varphi(\tilde{\Phi})$ by logically equivalent, but smaller ones. Such smaller, equivalent formulas can be computed by simplification algorithms for FO (Goubault 1995).

Applications

In this section, we discuss some applications of computing an approximation of an FO theory.

Model Expansion

Searching approximations for a theory T above an approximate structure $\tilde{I}$ is strongly related to the setting of model expansion for FO (MX(FO)). This setting has been proposed as a framework to solve NP problems declaratively (Mitchell and Ternovska 2005).

Definition 18. Given a theory T over a vocabulary Σ, a vocabulary $\sigma \in \Sigma$ and a finite σ-structure I_σ, the *model expansion (MX(FO)) search problem for input* $\langle \Sigma, T, \sigma, I_\sigma \rangle$ is the problem of finding models M of T that expand I_σ, i.e., $M|_\sigma = I_\sigma$. The *MX(FO) decision problem* for the same input is the problem of deciding whether such a model exists.

The techniques developed in this paper can be used to give approximate answers for MX(FO) problems with input $\langle \Sigma, T, \sigma, I_\sigma \rangle$. Let D be the domain of I_σ and let $\tilde{I}_\sigma$ be the approximate structure with domain D that assigns $\langle P^{I_\sigma}, P^{I_\sigma} \rangle$ to every predicate symbol $P \in \sigma$ and $\langle \emptyset, D^n \rangle$ to all other n-ary predicate symbols $P \in \Sigma$. Then an approximation $\tilde{J}$ of T above $\tilde{I}_\sigma$ is an approximation to the answers for the MX(FO) search problem. If $\tilde{J}$ is inconsistent, then certainly the answer to the MX decision problem is negative.

The existing solvers for MX(FO) (Mariën, Wittocx, and Denecker 2006; Mitchell et al. 2006) work by first *grounding* T, i.e., creating a propositional theory T_G equivalent to

T, and then applying a SAT solver. To obtain a compact grounding T_G, subformulas in T_G that are certainly true or false according to $\tilde{I}_\sigma$ are replaced by respectively $\top$ and $\bot$. In (Wittocx, Mariën, and Denecker 2008), it was observed that an even more compact grounding can be obtained by first computing an approximation $\tilde{J}$ for T above $\tilde{I}_\sigma$ and then substituting subformulas that are certainly true or false according to $\tilde{J}$. Experiments in that paper showed that computing and using $\tilde{J}$ almost never incurs overhead, while the size of T_G decreases. Moreover, often the time to create the grounding decreases drastically.

Incomplete Databases

A recent trend in databases is the development of approximate methods to reason about databases with incomplete knowledge. The incompleteness of the database may stem from the use of null values, or of a restricted form of closed world assumption as in (Cortés-Calabuig et al. 2007), or it arises from integrating a collection of local databases each based on its own *local schema* into one virtual database over a *global schema* (Grahne and Mendelzon 1999). In all these cases, the data complexity of certain and possible query answering is computationally hard (co-NP, respectively NP). For this reason fast (and often very precise) polynomial approximate query answering methods are developed, which compute an underestimation of the certain, and an overestimation of the possible answers.

The tables of an incomplete database are naturally represented as an approximate structure $\tilde{I}$. The integrity constraints, local closed world assumption or mediator scheme corresponds to a logic theory T. Approximate answering a query $\varphi[\overline{x}]$ boils down to computing two relations R^l and R^u consisting of tuples $\overline{d}$ which are true in all models, respectively some model, of T above $\tilde{I}$. Hence, a possible technique to compute R^l and R^u consists of computing a refinement sequence $\langle \tilde{I}_i \rangle_{0 \leq i \leq n}$ for T above $\tilde{I}$ and then computing $\tilde{I}_n(\varphi[\overline{x}])$, respectively the complement of $\tilde{I}_n(\neg\varphi[\overline{x}])$.

It turns out that this method of approximate query answering generalizes the algorithm of (Cortés-Calabuig et al. 2006). Computing a symbolic refinement sequence $\langle \tilde{\Phi}_i \rangle_{0 \leq i \leq n}$ and then computing $\tilde{\Phi}_n(\varphi[\overline{x}])$, respectively $\neg\tilde{\Phi}_n(\neg\varphi[\overline{x}])$, generalizes the query rewriting technique presented in (Cortés-Calabuig et al. 2007).

Related Work

The notion of *approximation* in our work is quite different from the one in most other works about approximate reasoning in logic theories. E.g., the approach started by Levesque (Levesque 1988) and further explored by, a.o., Schaerf and Cadoli (Schaerf and Cadoli 1995) performs approximate query answering for one query, while our method computes an approximate structure approximating all models of a theory. Their methods could hardly be used for, e.g., grounding. The same holds for the *knowledge compilation* approach, started by Selman and Kautz (Selman and Kautz 1991), which is often applied in description logics.

In the literature, we find mathematical constructs similar to the ones used in this paper in two other research areas: *rough set theory* (Pawlak 1992) and four-valued logic (Ackermann 1967). In rough set theory, attributes of elements of a universe U are used to classify each $x \in U$ as certainly in, possibly in, or certainly not in a rough set $X \subseteq U$ of interest. In our work, the computed sets $P^{\mathfrak{O}^T(\tilde{I})}$ can be viewed as non-exact sets similar to rough sets but, in contrast, they are are not given explicitly but computed from T and $\tilde{I}$.

Also, (consistent) approximate relations and structures can be viewed as (three-valued) four-valued relations and structures, but in multi-valued logics, relation symbols denote three-valued relations, whereas we use standard two-valued FO-logic and use the approximate relations only to approximate collections of possible relations.

As pointed out in the previous section, the work presented here is strongly related to certain approximate methods in databases with incomplete knowledge, for instance in the context of integration of distributed databases (Lenzerini 2002; Grahne and Mendelzon 1999), or of *locally closed databases* (Cortés-Calabuig et al. 2006; 2007). Our methods could be useful in generalizations of locally closed databases, e.g., for databases with integrity constraints.

In (Wittocx, Mariën, and Denecker 2008) we presented in more detail how symbolic approximations can be used for grounding in the context of MX(FO). The non-symbolic algorithm, with its results on termination, confluence, complexity and optimality is not described in that paper. Also, the symbolic method presented there is less precise and less general than the one in this paper.

Conclusions

We presented a method to compute an approximation of all models with a given finite domain of given FO theory. For an important class of theories, the data-complexity of the method is polynomial. Some preliminary results about precision were stated. We also presented how to obtain symbolic representations of approximations, which can improve the efficiency of the method. Finally, we discussed some applications in the context of databases and model expansion and mentioned related work.

Acknowledgments

Johan Wittocx is research assistant of the *Fonds voor Wetenschappelijk Onderzoek - Vlaanderen* (FWO Vlaanderen).

Appendix

Proof of lemma 4 First note that for every model M of T approximated by $\tilde{I}$, it holds that $\langle M, M \rangle \geq_p \tilde{I}$ and $M \models \varphi$. Hence, by the definition of $\mathcal{O}^\varphi$, $\mathcal{O}^\varphi(\tilde{I})$ approximates T above $\tilde{I}$.

Denote by $\mathcal{M}$ the set $\{\langle M, M \rangle \geq_p \tilde{I} \mid M \models \varphi\}$. If $\mathcal{M} = \emptyset$, then $\mathcal{O}^\varphi(\tilde{I}) = \top^D_{\leq_p}$, which is clearly more precise than $\tilde{I}$. On the other hand, if $\mathcal{M} \neq \emptyset$, then $\tilde{I}$ is a $\leq_p$-lower bound of $\mathcal{M}$ and hence, also in this case $\mathrm{glb}_{leq_p}(\mathcal{M}) \geq_p \tilde{I}$.

Proof of theorem 6 Because the domain D of $\tilde{I}$ is finite, there are only a finite number of approximate structures $\tilde{J} \geq_p \tilde{I}$. As such, every refinement sequence above $\tilde{I}$ is finite.

To prove confluence, let $\langle \tilde{J}_i \rangle_{0 \leq i \leq n}$ and $\langle \tilde{K}_j \rangle_{0 \leq j \leq m}$ be two terminal refinement sequences and denote by φ_i and ψ_j the sentences of T such that respectively $\mathcal{O}^{\varphi_i}(\tilde{J}_i) = \tilde{J}_{i+1}$ and $\mathcal{O}^{\psi_j}(\tilde{K}_j) = \tilde{K}_{j+1}$. Then, because for each sentence χ, $\mathcal{O}^{\chi}$ is a $\leq_p$-monotone operator, $\tilde{K}_m = \mathcal{O}^{\psi_{m-1}}(\mathcal{O}^{\psi_{m-2}}(\dots(\mathcal{O}^{\psi_0}(\tilde{I}_n)))) \leq_p \mathcal{O}^{\psi_{m-1}}(\mathcal{O}^{\psi_{m-2}}(\dots(\mathcal{O}^{\psi_0}(\tilde{J}_n)))) = \tilde{J}_n$. Similarly, $\tilde{J}_n \leq_p \tilde{K}_m$. Hence, $\tilde{J}_n = \tilde{K}_m$.

Proof of lemmas 8 and 9 The proofs of lemmas 8 and 9 merely consist of a simple, but long and tedious case-by-case analysis. As an example, we prove the case where φ is the ENF sentence $\forall y, z\ (P(y,z) \equiv \forall v\ Q(y,v))$.

Denote by $\tilde{J}$ the approximate structure suggested by the lemmas, i.e., the approximate structure such that for any $d_y, d_z, d_v \in D$:

– $\tilde{J}(P(d_y,d_z)) = \tilde{I}(\exists z\ P(d_y,z) \vee \forall v\ Q(d_y,v))$;

– $\tilde{J}(\neg P(d_y,d_z)) = \tilde{I}(\exists z\ \neg P(d_y,z) \vee \exists v\ \neg Q(d_y,v))$;

– $\tilde{J}(Q(d_y,d_v)) = \tilde{I}(Q(d_y,d_v) \vee \exists z\ P(d_y,z))$;

– $\tilde{J}(\neg Q(d_y,d_v)) = \tilde{I}(\neg Q(d_y,d_v) \vee (\exists z\ \neg P(d_y,z) \wedge \forall v'\ (v' \neq d_v \supset Q(d_y,v'))))$;

– $\tilde{J}$ corresponds to $\tilde{I}$ on all symbols that do not occur in φ.

One can easily check that $\tilde{J} \geq_p \tilde{I}$. From the discussion above lemma 8, it follows that $\tilde{J} \leq_p \mathcal{O}^{\varphi}(\tilde{I})$.

Assume $\mathcal{O}^{\varphi}(\tilde{I})$ is inconsistent. Then either $\tilde{I}$ is inconsistent or one of the following is true:

– for some $d_y, d_z, d_v \in D$, $\tilde{I}(P(d_y,d_z)) =$ **t** and $\tilde{I}(\neg Q(d_y,d_v)) =$ **t**;

– for some $d_y, d_z \in D$, $\tilde{I}(\neg P(d_y,d_z)) =$ **t** and $\tilde{I}(\forall v\ Q(d_y,v)) =$ **t**.

In all cases, also $\tilde{J}$ is inconsistent.

Now assume $\mathcal{O}^{\varphi}(\tilde{I})$ is consistent. It is sufficient to prove that for any atom $P(d_y,d_z)$, respectively $Q(d_y,d_v)$ such that $\tilde{J}(P(d_y,d_z)) = \tilde{J}(\neg P(d_y,d_z)) =$ **f**, respectively $\tilde{J}(Q(d_y,d_v)) = \tilde{J}(\neg Q(d_y,d_v)) =$ **f**, also $\mathcal{O}^{\varphi}(\tilde{I})(P(d_y,d_z)) = \mathcal{O}^{\varphi}(\tilde{I})(\neg P(d_y,d_z)) =$ **f**, respectively $\mathcal{O}^{\varphi}(\tilde{I})(Q(d_y,d_v)) = \mathcal{O}^{\varphi}(\tilde{I})(\neg Q(d_y,d_v)) =$ **f**.

– Assume that $\tilde{J}(P(d_y,d_z)) = \tilde{J}(\neg P(d_y,d_z)) =$ **f**. Let M be a model of φ approximated by $\tilde{I}$. We will construct a model M' of φ approximated by $\tilde{I}$ such that $M(P(d_y,d_z)) \neq M(P(d_y,d_z))$. Assume $M(P(d_y,d_z)) =$ **t**. Because $\tilde{J}(P(d_y,d_z)) =$ **f**, both $\tilde{I}(\exists z\ P(d_y,z)) =$ **f** and $\tilde{I}(\forall v\ Q(d_y,v)) =$ **f**. Hence, there exists some d_v such that $\tilde{I}(Q(d_y,d_v)) =$ **f**. Let M' be the structure that corresponds to M, except that it assigns $M'(P(d_y,d'_z)) =$ **f** for every $d'_z \in D$ and $M'(Q(d_y,d_v)) =$ **f**. Clearly, M' is approximated by $\tilde{I}$ and M' is a model of φ. Since we now have that both M and M' belong to $\{I \mid \langle I, I \rangle \geq_p \tilde{I}$ and $I \models \varphi\}$, it holds that $\mathcal{O}^{\varphi}(\tilde{I})(P(d_y,d_z)) = \mathcal{O}^{\varphi}(\tilde{I})(P(d_y,d_z)) =$ **f**, as desired.
  The case where $M(P(d_y,d_z)) =$ **f** is similar.

– The case where $\tilde{J}(Q(d_y,d_v)) = \tilde{J}(\neg Q(d_y,d_v)) =$ **f** is similar to the previous case.

Proof of lemma 11 The case where φ and χ have no predicate symbols in common is trivial, as is the case where $\tilde{I}(\neg\varphi) =$ **t**.

To prove the case where $\tilde{I}(\varphi) =$ **t**, assume $\tilde{I}(\varphi) =$ **t** and let $\tilde{J}$ be an arbitrary approximate structure such that $\tilde{J} \geq_p \tilde{I}$. It is sufficient to prove that $\mathcal{O}^{\chi}(\mathcal{O}^{\varphi}(\tilde{J})) = \mathcal{O}^{\varphi \wedge \chi}(\tilde{J})$. Indeed, then every refinement for T above $\tilde{I}$ can be transformed to a refinement sequence for T' above $\tilde{I}$ by replacing each application of $\mathcal{O}^{\varphi \wedge \chi}$ by an application of $\mathcal{O}^{\varphi}$ followed by $\mathcal{O}^{\chi}$. To obtain a contradiction, assume $\mathcal{O}^{\chi}(\mathcal{O}^{\varphi}(\tilde{J})) <_p \mathcal{O}^{\varphi \wedge \chi}(\tilde{J})$. Then there exists a structure M such that $\langle M, M \rangle \geq_p \tilde{J}$, $M \models \chi$ and $M \not\models \varphi \wedge \chi$. Hence $M \not\models \varphi$, which is a contradiction to $\tilde{J}(\varphi) =$ **t**.

Proof of lemma 12 Observe that it is sufficient to prove that $\mathfrak{O}^{T}(\tilde{I}) \leq_p \mathfrak{O}^{T'}(\tilde{I})$, as it is obvious that $\mathfrak{O}^{T}(\tilde{I}) \geq_p \mathfrak{O}^{T'}(\tilde{I})$ holds. Because $\tilde{I}_n \geq_p \tilde{I}$, we have $\mathfrak{O}^{T}(\tilde{I}) \leq_p \mathfrak{O}^{T}(\tilde{I}_n) = \mathfrak{O}^{T'}(\tilde{I}_n)$. Since $\langle \tilde{I}_n \rangle_{0 \leq i \leq n}$ is a refinement sequence for T' above $\tilde{I}$, also $\mathfrak{O}^{T'}(\tilde{I}_n) = \mathfrak{O}^{T'}(\tilde{I})$.

References

Ackermann, R. 1967. *An Introduction to Many-Valued Logics*. London: Routledge and Kegan Paul.

Cortés-Calabuig, A.; Denecker, M.; Arieli, O.; and Bruynooghe, M. 2006. Representation of partial knowledge and query answering in locally complete databases. In Hermann, M., and Voronkov, A., eds., *LPAR*, volume 4246 of *Lecture Notes in Computer Science*, 407–421. Springer.

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

Enderton, H. B. 1972. *A Mathematical Introduction To Logic*. Academic Press.

Fagin, R. 1974. Generalized first-order spectra and polynomial-time recognizable sets. *Complexity of Computation* 7:43–74.

Goubault, J. 1995. A bdd-based simplification and skolemization procedure. *Logic Journal of IGPL* 3(6):827–855.

Grahne, G., and Mendelzon, A. 1999. Tableau techniques for querying information sources through global schemas. In *Proceedings of 7th International Conference on Database Theory ICDT*, volume 1540, 332–347. LNCS.

Greco, S., and Molinaro, C. 2007. Querying and repairing inconsistent databases under three-valued semantics. In

Dahl, V., and Niemelä, I., eds., *ICLP*, volume 4670 of *Lecture Notes in Computer Science*, 149–164. Springer.

Lenzerini, M. 2002. Data integration: A theoretical perspective. In *Proceedings of the Twenty-first ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, June 3-5, Madison, Wisconsin, USA*, 233–246.

Levesque, H. J. 1988. Logic and the complexity of reasoning. *Journal of Philosophical Logic* 17(4):355–389.

Mariën, M.; Wittocx, J.; and Denecker, M. 2006. The IDP framework for declarative problem solving. In *Search and Logic: Answer Set Programming and SAT*, 19–34.

Mitchell, D., and Ternovska, E. 2005. A framework for representing and solving NP search problems. In *AAAI'05*, 430–435. AAAI Press/MIT Press.

Mitchell, D.; Ternovska, E.; Hach, F.; and Mohebali, R. 2006. Model expansion as a framework for modelling and solving search problems. Technical Report TR2006-24, Simon Fraser University.

Pawlak, Z. 1992. *Rough Sets: Theoretical Aspects of Reasoning about Data*. Norwell, MA, USA: Kluwer Academic Publishers.

Schaerf, M., and Cadoli, M. 1995. Tractable reasoning via approximation. *Artificial Intelligence* 74(2):249–310.

Selman, B., and Kautz, H. A. 1991. Knowledge compilation using horn approximations. In *National Conference on Artificial Intelligence*, 904–909.

Tseitin, G. S. 1968. On the complexity of derivation in propositional calculus. In Slisenko, A. O., ed., *Studies in Constructive Mathematics and Mathematical Logic II*, volume 8 of *Seminars in Mathematics: Steklov Mathem. Inst.* New York: Consultants Bureau. 115–125.

Ullman, J. D. 1988. *Principles of Database and Knowledge-Base Systems, Volume I*. Computer Science Press.

Wittocx, J.; Mariën, M.; and Denecker, M. 2008. Grounding with bounds. In *AAAI'08*. accepted.