Maximal Expansions of Database Updates ★

Michael Dekhtyar ‡, **Alexander Dikovsky** §, **Sergey Dudakov** ‡, and **Nicolas Spyratos** ¶

Abstract. Databases with integrity constraints (IC) are considered. For each DB update, i.e. a set of facts to add and of facts to delete, the IC implies its *correct expansion*: new facts to add and new facts to delete. Simultaneously, each expanded update induces a *correct simplification* of the IC. In the limit this sequence of expansions and simplifications converges to the maximal correct update expansion independent from the initial DB state. We show that such maximal expansion is computed in square time for partial databases, and that its computation is a co-NP-complete problem in classical databases. However, it is also square time computable in classical DBs under ICs with some restrictions on the use of negation.
Computing the real change of the initial DB state after accomplishing an update is a hard problem. The use of maximal update expansion in the place of initial update can substantially simplify computation of a new correct DB state.

1 Introduction

Conventional databases roll back after updates violating integrity constraints (IC). Meanwhile, some developed contemporary databases (for instance, Oracle), enable its administrators or application developers to present IC in the form of systems of rules (e.g., business rules), and use automatic means, such as triggers, which enforce these rules after updates. The intensive recent work on active

★ This work was sponsored by the Russian Fundamental Studies Foundation (Grants 97-01-00973, 98-01-00204, 99-01-00374).

‡ Dept. of CS, Tver State Univ. 3 Zheljabova str. Tver, Russia, 170013 Email: Michael.Dekhtyar@tversu.ru .

§ Université de Nantes. IRIN, UPREF, EA No 2157. 2, rue de la Houssinière BP 92208 F 44322 Nantes Cedex 3 France, Alexandre.Dikovsky@irin.univ-nantes.fr
and
Keldysh Institute for Applied Math. 4 Miusskaya sq. Moscow, Russia, 125047 .

¶ Université de Paris-Sud, LRI, U.R.A. 410 du CNRS, Bât. 490 F-91405 Orsay Cedex, France Email: Nicolas.Spyratos@lri.fr .

databases (cf. [14]) shows that in the future this ability of intelligent update enforcement will be developed.

The effect of extensional updates on data can be specified declaratively by expressions of algorithmic algebras generalizing SQL (cf. [1]), or by formulas of a logical language (see e.g. [4]). However, when IC are presented in the form of rule systems, the effect of an update manifests itself indirectly: either following to some intentional semantics of rules, or operationally, in the form of a derivation. Let us consider the following simplified example.

Example 1 *The IC Φ below expresses a typical case of an exception from a general rule. It consists of two clauses. The first one expresses the general rule: "children (proposition* children*) can bathe (*bathe*) when with parents (*parents*)". The other one expresses an exception from this rule: "children cannot bathe while the ebb tide (proposition* ebb*)":*

$$\mathsf{bathe} \leftarrow \mathsf{children}, \mathsf{parents}$$
$$\neg\mathsf{bathe} \leftarrow \mathsf{children}, \mathsf{ebb}.$$

Let us consider a DB state where children cannot bathe because of the ebb. This state is materialized differently in classical and partial databases. In classical databases the absence of a fact means that its negation holds. In a partial database S a holds if $a \in S$, $\neg a$ holds if $\neg a \in S$ explicitely, otherwise a is unknown.

Let us consider first the classical databases. This means that we have the DB state $I = \{\mathsf{children}, \mathsf{ebb}\}$. Suppose that the parents arrive, which is expressed as the addition of the fact parents *to I. This addition causes the conflict with the first rule. The possible solutions are simple but nontrivial. The first solution is to replace* ebb *by* bathe*. The result is the DB state where children's bathing is allowed: $I_1 = \{\mathsf{children}, \mathsf{parents}, \mathsf{bathe}\}$. The other is just to eliminate* children*. The resulting DB state is that where no children's bathing is needed: $I_2 = \{\mathsf{parents}, \mathsf{ebb}\}$.*

Now, let us consider the case of partial databases. The initial DB is in this case $I = \{\mathsf{children}, \mathsf{ebb}, \neg\mathsf{bathe}\}$. The first solution is then to replace ebb *and* $\neg$bathe *by* bathe*, the resulting DB state being: $I_1 = \{\mathsf{children}, \mathsf{parents}, \mathsf{bathe}\}$. The other solution is again to eliminate* children*, the resulting DB state being this time: $I_2 = \{\mathsf{parents}, \mathsf{ebb}, \neg\mathsf{bathe}\}$.*

In logical terms, the result of such an intensional update is defined as a model of a logical theory (*knowledge base (KB) update*), or as a set of consequences of an extensional DB state (*DB view update*), or else, its effect is defined through logical inference, e.g. abduction (see e.g. [20, 12, 15, 10]), sometimes enforced by bottom-up hypothesis generation for integrity checking (e.g. [5]). Sometimes updates are specified indirectly as a minimal (in a sense) change of the given state sufficient to attain an intended KB state (see e.g. [18, 19]), sometimes they are specified by an update program, so the change is defined by both programs: KB and update ([2]). In the case where negative knowledge is treated, these approaches work exclusively with some sort of intended models: for instance, stable [11] or well founded [21].

In this paper we develop the conflict resolution approach to updates, proposed in [6, 7, 13]. It departs from the premise that IC have nothing to do with data or knowledge definition. They only specify the conflicts to avoid after the update. So the use of exclusively "intended" models of IC may lead to the loss of information or to unjustified conflict resolution failures. The above example shows such a loss [1]. So in our approach one should consider all classical models of IC, but among these models one should find a model minimally deviating from the model before update. A bit more formally, the problem we tackle is formulated as follows. Given a logic program Φ which formalizes the IC, a correct initial DB state $I \models \Phi$, and an external update Δ which specifies the facts D^+ to be added to I and the facts D^- to be deleted from it, one should find the minimal real change $\Psi(I)$ of I, sufficient to accomplish Δ and to restore Φ if and when it is violated (i.e. to guarantee that $D^+ \subseteq \Psi(I), \Psi(I) \cap D^- = \emptyset$, and $\Psi(I) \models \Phi$).

Unfortunately, the problem of minimal real change with respect to IC is still harder from the complexity point of view than the problem of intended knowledge update. The latter is of the type "guess and check" (which corresponds to NP), whereas, the former is proven in [8] to be Σ_2^p-complete for classical databases. In this paper we prove that its solution is co-NP-complete for partial databases.

In our earlier papers [6, 7] we introduce a broad class of update operators $\Psi(\Phi, \Delta, I)$ (we call them *conservative*), which apply to generalized logic programs Φ with explicit negation (possible in the bodies as well as in the heads of clauses) and are based on a mixed minimal change criterion which is a combination of the maximal intersection, and of the minimal symmetric difference of states. We describe various nondeterministic and deterministic conservative update algorithms based on a conflict resolution techniques. In the recent paper [8] we propose a practical method of speeding up the conservative update algorithms. The method is based on the idea that the initial update $\Delta = (D^+, D^-)$ can be incrementally and correctly expanded to a broader update after being iteratively propagated into the IC Φ. *Correctness* of an expansion means that it preserves the set $Acc(\Phi, \Delta)$ of DB states in which Δ is accomplished and which satisfy Φ. Φ being fixed, each compatible update Δ implies correct expansions: $\Delta_1 = (M^+, M^-) \sqsupseteq \Delta$ and simultaneously correct IC simplifications $\Phi_1 \preceq \Phi$. Using Φ_1, Δ_1 in the place of Φ, Δ can substantially simplify the choice of a new correct DB state after accomplishing update. Indeed, this new state separates M^+ from M^-, so the expansion narrows the choice space.

In this paper we obtain a result which gives a definitive theoretical foundation to our uproach, and has a clear practical sense. We describe the optimal update

[1] It seems that the "intended model" methods fail to find the solutions it demonstrates. Indeed, since there are no rules with parents in the head, there is no direct refutation or abduction proof of the added fact. I, I_1, I_2 are not stable models of Φ. One could propose to add the clauses children $\leftarrow$ and ebb $\leftarrow$ to Φ and then to update the resulting program Φ' by the update program {parents $\leftarrow$ }. In this case the states would become stable models of Φ', but again the inertia rules of [2, 19] would prevent to infer $\neg ebb$. The answer set semantics of [3, 16] does not help because there is no negation as failure in Φ.

expansion and IC simplification as those which imply all others and preserve the models in $Acc(\Phi, \Delta)$. The optimal expansion is the maximal real change independent from the initial DB state. We show that both, the optimal update expansion and the optimal IC simplification are incrementally computed from Φ and Δ in square time for partial databases. Unfortunately, their computation is a co-NP-complete problem in classical databases. Nevertheless, we find an important class of ICs where their computation is easily reduced to that in partial databases, so they are again computable in square time.

The paper is organized as follows. All preliminary notions and notation are given in the next section. The conservative update problem is formulated in Section 3, and its complexity is established in Section 4. Update expansion operators are defined in Section 5. The maximal update expansion is explored in Section 6 for partial DBs, and in Section 7 for classical DBs.

2 Preliminaries

We assume that the reader is familiar with the basic concepts and terminology of logic programming (see [17]).

Language. We fix a 1st order signature $\mathbf{S}$ with an infinite set of constants $\mathbf{C}$ and no other function symbols. A *domain* is a finite subset $\mathbf{D}$ of $\mathbf{C}$. For each domain $\mathbf{D}$ by $\mathbf{A}(\mathbf{D})$, $\mathbf{L}(\mathbf{D})$, $\mathbf{B}(\mathbf{D})$ and $\mathbf{LB}(\mathbf{D})$ we denote respectively the sets of all atoms, all literals, all ground atoms, and all ground literals in the signature $\mathbf{S}$ with constants in $\mathbf{D}$. A literal contrary to a literal l is denoted by $\neg.l$. We set $\neg.M = \{\neg.l \mid l \in M\}$.

Logic programs. We consider generalized logic programs in $\mathbf{S}$ with explicit negation, i.e. finite sets of clauses of the form $r = (l \leftarrow l_1, ..., l_n)$ where $n \geq 0$ and $l, l_i \in \mathbf{L}(\mathbf{D})$, (note that negative literals are possible in the bodies and in the heads of the clauses). For a clause r $head(r)$ denotes its *head*, and $body(r)$ its *body*. We will treat $body(r)$ as a set of literals. Integrity constraints (IC) are expressed by a logic program Φ of this kind. $\mathbf{IC}(\mathbf{D})$ denotes the set of all *ground* integrity constraints in the signature $\mathbf{S}$ with constants in $\mathbf{D}$. We consider the following *simplification order* on $\mathbf{IC}(\mathbf{D})$: $\Phi_1 \preceq \Phi_2$ if $\forall r \in \Phi_1\ \exists r' \in \Phi_2\ (head(r) = head(r')\ \&\ body(r) \subseteq body(r'))$. This relationship between Φ_1 and Φ_2 means that Φ_1 consists of stronger versions of some clauses of Φ_2.

Correct DB states. In this paper we consider both kinds of interpretations of ICs, total and partial, over *closed domains.* This means that a certain domain $\mathbf{D}$ is fixed for each problem. A *partial* interpretation (*DB state*) over $\mathbf{D}$ is a finite subset of $\mathbf{LB}(\mathbf{D})$. For such an interpretation $I \subseteq \mathbf{LB}(\mathbf{D})$ we set $I^+ = I \cap \mathbf{B}(\mathbf{D})$ and $I^- = I \cap \neg.\mathbf{B}(\mathbf{D})$. I is *consistent* if it contains no contrary pair of literals $l, \neg.l$. Intuitively, in a consistent partial DB state I the atoms in I^+ are regarded as true, the atoms in $\neg.I^-$ are regarded as false, and all others are regarded as unknown. A partial interpretation I is *total* if $I^+ \cup \neg.I^- = \emptyset$ and $I^+ \cup \neg.I^- = \mathbf{B}(\mathbf{D})$. So the total interpretations are consistent by definition. They are completely defined by their positive parts, therefore, we will identify total interpretations with subsets of $\mathbf{B}(\mathbf{D})$. $\mathbf{D}$ being fixed we consider ICs with constants in $\mathbf{D}$ and groundisations over $\mathbf{D}$. Given an IC $\Phi \in \mathbf{IC}(\mathbf{D})$ and a DB state I over $\mathbf{D}$, a (ground) clause $r = (l \leftarrow l_1, ..., l_n)$ of Φ is *valid* in I (denoted $I \models r$) if $I \models l$ whenever $I \models l_i$ for each $1 \leq i \leq n$. For a partial DB state I and a ground literal l $I \models l$ means $l \in I$. For a total DB state I and a ground atom a $I \models a$ means $a \in I$, and $I \models \neg.a$ means $a \notin I$. I is *a correct DB state* or a *model* of Φ (denoted $I \models \Phi$) if it is consistent (which is always true for total DB states) and every clause in Φ is valid in I.

Consequence closure. Let $\Phi \in \mathbf{IC}(\mathbf{D})$. For a partial interpretation I we set $cl_\Phi(I) = \{l | \exists r = (l \leftarrow l_1, ..., l_n) \in \Phi \ (\bigwedge_{i=1}^{n} I \models l_i \ )\}$. A strong immediate consequence operator is the total operator

$$T_\Phi^\in(I) = \begin{cases} cl_\Phi(I) & : \ cl_\Phi(I) \ \textit{is consistent} \\ \mathbf{LB}(\mathbf{D}) & : \ cl_\Phi(I) \ \textit{is inconsistent.} \end{cases}$$

Being continuous, $T_\Phi^\in$ has the least fixed point $lfp(T_\Phi^\in) = \bigcup_{i=0}^{\infty} (T_\Phi^\in(\emptyset))^i$. We denote this set by M_Φ^{min}. It is clear that if M_Φ^{min} is consistent, then it is the least (partial) model of Φ. For any partial DB state I we set $M_\Phi^{min}(I) = M_{\Phi \cup I}^{min}$.

Updates. When partial interpretations over $\mathbf{D}$ are considered, an *update* is a pair $\Delta = (D^+, D^-)$ where D^+, D^- are subsets of $\mathbf{LB}(\mathbf{D})$. In the case of total interpretations D^+, D^- are subsets of $\mathbf{B}(\mathbf{D})$. In both cases $D^+ \cap D^- = \emptyset$. Intuitively, the literals of D^+ are to be added to DB state I, and those of D^- are to be removed from I. For both kinds of interpretations $\mathbf{UP}(\mathbf{D})$ will denote the set of all updates in the signature $\mathbf{S}$ and with constants in $\mathbf{D}$. We say that $\Delta = (D^+, D^-)$ is *accomplished* in I if $D^+ \subseteq I$ and $D^- \cap I = \emptyset$. A partial DB state I *agrees* with (partial or total) Δ if $I \cap (D^- \cup \neg.D^+) = \emptyset$. The updates in $\mathbf{UP}(\mathbf{D})$ will be partially ordered by the componentwise inclusion relation: $\Delta_1 \sqsubseteq \Delta_2$ iff $D_1^+ \subseteq D_2^+$ and $D_1^- \subseteq D_2^-$.

Update operators. Let Γ be an operator of the type $\Gamma : \mathbf{IC}(\mathbf{D}) \times \mathbf{UP}(\mathbf{D}) \rightarrow \mathbf{IC}(\mathbf{D}) \times \mathbf{UP}(\mathbf{D})$, $\Gamma(\Phi, \Delta) = (\Phi', \Delta')$, and $\Delta' = (D'^+, D'^-)$. In the definitions to follow Φ', Δ', D'^+, and D'^- are denoted respectively by $\Gamma(\Phi, \Delta)^{ic}$, $\Gamma(\Phi, \Delta)^{up}$, $\Gamma(\Phi, \Delta)^+$, and $\Gamma(\Phi, \Delta)^-$. We denote by Γ^n the n-fold composition of Γ and by Γ^ω the operator $\Gamma^\omega(\Phi, \Delta) = \lim_{n \to \infty} \Gamma^n(\Phi, \Delta)$.

In the sequel we will omit $\mathbf{D}$ when it causes no ambiguity. So when $\mathbf{D}$ is subsumed, in the place of $\mathbf{A}(\mathbf{D})$, $\mathbf{L}(\mathbf{D})$, $\mathbf{B}(\mathbf{D})$, $\mathbf{LB}(\mathbf{D})$ we will use the notation $\mathbf{A}$, $\mathbf{L}$, $\mathbf{B}$, $\mathbf{LB}$.

3 Conservative update operators

In general an update may contradict a constraint. So a reasonable definition of an update operator should either contain a requirement of "compatibility" of an update and a constraint, or specify a part of the update "compatible" with the constraint. The requirement of compatibility is easy to formalize.

Definition 1 *For $\Phi \in \mathbf{IC}$ and $\Delta \in \mathbf{UP}$ let us denote by $Acc(\Phi, \Delta)$ the set of all models $I \models \Phi$ where Δ is accomplished. An update Δ is* compatible *with an IC Φ if $Acc(\Phi, \Delta) \neq \emptyset$.*

Compatibility of Φ with $\Delta = (D^+, D^-)$ means that there is a model $I \in Acc(\Phi, \Delta)$. For partial interpretations this guarantees the existence of the *least model* $M_\Phi^{min}(D^+)$ which is in fact the set of all ground consequences of the facts in D^+ with respect to Φ. We denote this model by M_Δ^Φ. M_Δ^Φ is constructed in time linear with respect to the summary size of Φ and Δ. This means that for partial interpretations compatibility of Φ and Δ is recognized in linear time. For total interpretations the consistency of M_Δ^Φ does not guarantee compatibility of Φ and Δ, as the following example shows.

Example 2 *Let us consider IC $\Phi = \{r_1 : \neg c \leftarrow a, b;\ r_2 : \neg b \leftarrow \neg a, c\}$ and update $\Delta = (\{b, c\}, \emptyset)$. Then it is easy to see that $M_\Delta^\Phi = \{b, c\}$. But of course, there is no total DB state satisfying Φ, where Δ is accomplished.*

As we show in [8], the compatibility problem for total interpretations is NP-complete.

Conservative update operators provide in a sense a minimal real change of the initial DB state after its update. In [6] we propose the following minimal change criterion intended to keep as much initial facts as possible, and then to add possibly fewer new facts.

Definition 2 *Let I, I_1 be two DB states, and $\mathbf{K}$ be a class of DB states.
We say that I_1 is* minimally deviating from *I* with respect to *$\mathbf{K}$ if*
$\forall I_2 \in \mathbf{K}\ (\neg(I \cap I_1 \subsetneq I \cap I_2)\ \&\ ((I \cap I_1 = I \cap I_2) \rightarrow \neg(I_2 \setminus I \subsetneq I_1 \setminus I)))$.

Conservative update operators [7] have the following definition (for interpretations of both kinds).

Definition 3 *Let Δ be an update compatible with IC Φ. An operator Ψ on the set of DB states $\mathbf{UP}$ is a* conservative update operator *if for each DB state I :*

- *$\Psi(I)$ is a model of Φ,*
- *Δ is accomplished in $\Psi(I)$,*
- *$\Psi(I)$ is minimally deviating from I with respect to $Acc(\Phi, \Delta)$.*

Clearly, the conservative update operators are nondeterministic. The "children-ebb-tide" example in the Introduction describes two different DB states minimally deviating from the initial one for the interpretations of both kinds. The problem of finding one of the conservative update operator values for given IC, update and initial DB state is called in [6] the *enforced update problem (EUP).*

As it is shown in [7], these operators have the following model completeness property: for any two DB states I_1, I_2 satisfying IC Φ there exists an update Δ such that $I_2 = \Psi(I_1)$.

4 Complexity of conservative updates

In order to measure the complexity of conservative updates we consider two standard algorithmic problems: *Optimistic and Pessimistic Fall-Into-Problem* (**OFIP**, **PFIP**) (cf. [9]).

OFIP: Given some $\Delta \in$ **UP** compatible with $\Phi \in$ **IC**, an initial DB state I, and a literal $l \in$ **LB**, one should check whether there exists a DB state [2] I_1 such that:
(a) $I_1 \in Acc(\Phi, \Delta)$,
(b) I_1 is minimally deviating from I with respect to $Acc(\Phi, \Delta)$, and
(c) $I_1 \models l$.

PFIP: requires (c) be true *for all* DB states I_1 satisfying (a) and (b).

We consider the combined complexity of these problems with respect to the problem size evaluated as $N = |\mathbf{D}| + |I| + |\Delta| + |\Phi| + |l|$ ($|\ |$ is the size of constant or literal sets and of programs in some standard encoding). We denote respectively by **OFIP** and **PFIP** the sets of all solutions (I, Δ, Φ, l) of these problems. These problems are "co-problems" in total interpretations in the sense that $(I, \Delta, \Phi, l) \in$ **PFIP** iff $(I, \Delta, \Phi, \neg.l) \notin$ **OFIP**. In [8] we show that for total DBs **OFIP** is Σ_2^p-complete, so **PFIP** is Π_2^p-complete. For partial DBs these problems are simpler.

Theorem 1 *(Case of partial DBs)*
(1) **OFIP** *and* **PFIP** *belong to P in the case where:*
a) Φ is normal (i.e. there are no negations in the heads of clauses),
b) there are no deletions and negations in Δ, i.e. $D^+ \subseteq \mathbf{B}$ and $D^- = \emptyset$, and
c) there are no negations in I, i.e. $I \subseteq \mathbf{B}$.
(2) If any of conditions a), b), c) is violated, then **OFIP** *is NP-complete and* **PFIP** *is co-NP-complete.*

The following standard algorithm *Dp_search* resolves *EUP* in partial interpretations [3].

[2] We remind that some finite domain **D** is fixed.

[3] Since DB states and updates are finite, and the domain is closed, the space of DB states resulting from updates is finite as well. For each subset X of this space we fix the topological order with respect to set inclusion on subsets of X, with the successor function $next_X$. We set $next_X(X) = \perp$ for some constant $\perp$.

Algorithm $Dp\_search(I, \Phi, \Delta)$

Input: a DB state I, and some compatible update $\Delta = (D^+, D^-)$ and $\Phi \in \mathbf{IC}$.
Local variables: $\tilde{I}, H^-, H_{del}$: sets of literals; b : boolean.
Output: I_1.

$\tilde{I} := (I \cup D^+) \setminus D^-$;
$H^- := \tilde{I} \setminus D^+$;
$H_{del} := \emptyset$; $b := false$;
WHILE $\neg b$ **AND** $H_{del} \neq \bot$ **DO**
  $I_1 := M^{min}_{\Phi}(\tilde{I} \setminus H_{del})$;
  IF I_1 is inconsistent **OR** I_1 does not agree with Δ
  THEN $H_{del} := next_{H^-}(H_{del})$
  ELSE $b := true$
  END_IF
END_DO;
Output I_1.

Theorem 2 *Algorithm Dp_search implements conservative update operators for partial interpretations in linear space and in time* $O(2^d N)$, *where* $N = |\Phi| + |\Delta|$ *and* d *is the size (the number of literals) of* H^-.

In [8] we describe a similar algorithm *D_search* which implements a conservative update operators for total interpretations in linear space and in time $O(2^{d+a} N)$, where a is the size of the choice space for facts to add.

This complexity analysis leaves no hope to optimize substantially the standard algorithms by some general theoretical method. However, there may exist some practical speed-up methods. Below we propose one such efficient method.

5 Update expansion operators

If we look at the *EUP* solutions, we see that Φ, Δ, and I being fixed, each solution I_1 is represented as $I_1 = (I \cup M^+) \setminus M^-$, where $M^+ \supseteq D^+$, $M^- \supseteq D^-$, and $M^+ \cap M^- = \emptyset$. In [8] we have proposed an operator Γ_{lim} which gives an approximation (D_0^+, D_0^-) to (M^+, M^-) in the case of total interpretations: $M^+ \supseteq D_0^+ \supseteq D^+$, $M^- \supseteq D_0^- \supseteq D^-$. The expansion (D_0^+, D_0^-) of (D^+, D^-) does not depend on I. It is computed from Φ and Δ incrementally in deterministic square time. Moreover, Γ_{lim} equivalently simplifies the IC Φ itself with respect to the expanded update. So to find (M^+, M^-) we use this simplified IC in the place of Φ. Our new idea is to provide a more powerful operator Γ_{max} which gives the maximal update expansion and IC simplification implied by Φ and Δ. To arrive at its definition we modify the concepts of [8].

We start by the following factorization of the space $\mathbf{IC} \times \mathbf{UP}$.

Definition 4 *The pairs* $(\Phi, \Delta), (\Phi', \Delta') \in \mathbf{IC} \times \mathbf{UP}$ *are* update-equivalent *(notation:* $(\Phi, \Delta) \equiv_u (\Phi', \Delta')$*) if* $Acc(\Phi, \Delta) = Acc(\Phi', \Delta')$. *We set* $Equ(\Phi, \Delta) = \{(\Phi', \Delta') \mid (\Phi', \Delta') \equiv_u (\Phi, \Delta)\}$.

The orders $\sqsubseteq$ on **UP** and $\preceq$ on **IC** induce the following natural partial order on $\mathbf{IC} \times \mathbf{UP}$:

$$(\Phi_1, \Delta_1) \preceq (\Phi_2, \Delta_2) \text{ iff } \Delta_1 \sqsubseteq \Delta_2 \text{ and } \Phi_2 \preceq \Phi_1.$$

This order has evident computational sense: expansions of updates narrow the search space of standard algorithms *D_search* and *Dp_search*, and simplifications of IC at least simplify model checking.

The general definition of expansion operators is as follows.

Definition 5 *An operator* $\Gamma : \mathbf{IC} \times \mathbf{UP} \to \mathbf{IC} \times \mathbf{UP}$ *is an* update expansion operator *if*

- $(\Phi, \Delta) \equiv_u \Gamma(\Phi, \Delta)$ *and*
- $(\Phi, \Delta) \preceq \Gamma(\Phi, \Delta)$

for all compatible $\Phi \in \mathbf{IC}$ *and* $\Delta \in \mathbf{UP}$.

Quite evidently, the set of update expansion operators is closed under composition.

We are interested in expansion operators which provide the best expansion independent of the initial DB state. As it concerns updates, such a best expansion always exists. Indeed, let us remark that $(\Phi, (D_1^+ \cup D_2^+, D_1^- \cup D_2^-)) \in Equ(\Phi, \Delta)$ for any $(\Phi_1, (D_1^+, D_1^-)), (\Phi_2, (D_2^+, D_2^-)) \in Equ(\Phi, \Delta)$. So the following proposition is true.

Lemma 1 *(For interpretations of both kinds)*
1) $Equ(\Phi, \Delta)$ *contains pairs with the greatest update* $\Delta^{\Phi}_{max} = (D^+_{max}, D^-_{max})$, *where* $D^+_{max} = \bigcup\{D^+ \mid (\Phi', (D^+, D'^-)) \in Equ(\Phi, \Delta)\}$ *and* $D^-_{max} = \bigcup\{D^- \mid (\Phi', (D'^+, D^-)) \in Equ(\Phi, \Delta)\}$.
2) $D^+_{max} \subseteq \bigcap \{I \mid I \in Acc(\Phi, \Delta)\}$.
3) $D^-_{max} \cap \bigcup \{I \mid I \in Acc(\Phi, \Delta)\} = \emptyset$.

As it concerns IC simplification, it is possible that two or more minimal equivalent programs are $\preceq$-incomparable.

Example 3 *Let* $\Phi_1 = \{r_1 : a \leftarrow b;\ r_2 : b \leftarrow a; r_3 : c \leftarrow a\}$ $\Phi_2 = \{r_1, r_2\} \cup \{r_3' : c \leftarrow b\}$ *and* $\Delta = (\{d\}, \emptyset)$. *Then* $Equ(\Phi_1, \Delta)\}$ *includes two incomparable maximal pairs:* $((\{d\}, \emptyset), \Phi_1)$ *and* $((\{d\}, \emptyset), \Phi_2)$.

So we propose the following definition.

Definition 6 *An expansion operator* Γ *is* optimal *if for all compatible* Φ *and* Δ

- $\Gamma(\Phi, \Delta)^{up} = \Delta^{\Phi}_{max}$ *and*
- $\Phi' \prec \Gamma(\Phi, \Delta)^{ic}$ *for no* Φ' *such that* $(\Phi', \Delta') \in Equ(\Phi, \Delta)$ *for some* Δ'.

The update expansion operators in [8] propagate into Φ the initial update $\Delta = (D^+, D^-)$ and so define the set of literals l which it requires ($\Delta \models l$), and those to which it contradicts ($\Delta \not\models l$). Table 1 below describes the primary relations

$\models$ and $\not\models$ between Δ and ground literals $a, \neg a \in D^+$ and $a, \neg a \in D^-$ in the case of partial interpretations. Table 2 describes these relations for total DB states and for ground atoms $a \in D^+$ and $a \in D^-$. These relations can be extended to conjunctions of ground literals $l_1, ..., l_k \in \mathbf{BL}$: $\Delta \models l_1, ..., l_k$ if $\forall j\ (\Delta \models l_j)$, and $\Delta \not\models l_1, ..., l_k$ if $\exists j\ (\Delta \not\models l_j)$. In particular, $\Delta \models \emptyset$, and $\Delta \not\models \emptyset$ is not true.

Table 1. Relations $\Delta \models l$ and $\Delta \not\models l$ for partial DB states.

$\in$	D^+	D^-
a	$\Delta \models a,\ \Delta \not\models \neg a$	$\Delta \not\models a$
$\neg a$	$\Delta \models \neg a,\ \Delta \not\models a$	$\Delta \not\models \neg a$

Table 2. Relations $\Delta \models l$ and $\Delta \not\models l$ for total DB states.

$\in$	D^+	D^-
a	$\Delta \models a,\ \Delta \not\models \neg a$	$\Delta \models \neg a,\ \Delta \not\models a$

These definitions allow to carry over the validity of literals from updates to the DB states in which the updates are accomplished. Namely, an update Δ being accomplished in an DB state I,

(1) if $\Delta \models l_1, ..., l_k$ then $I \models l_1, ...l_k$ and if $\Delta \not\models l_1, ..., l_k$ then $\neg(I \models l_1, ...l_k)$,
(2) if $I \models l_1, ...l_k$ then it is not the case that $\Delta \not\models l_1, ..., l_k$.

Both relations $\models$ and $\not\models$ are monotone with respect to updates.

The simplification order on programs we use conforms with the following residue operator simplifying logic programs via updates.

Definition 7 *The residue of* $\Phi \in \mathbf{IC}$ *with respect to* Δ *is defined as:*
$res(\Phi, \Delta) = \{l \leftarrow \alpha \mid \exists r \in \Phi\ (head(r) = l\ \&\ \neg(\Delta \models l)\ \&\ \neg(\Delta \not\models body(r))\ \&\ \alpha \subseteq body(r)\ \&\ body(r) \setminus \alpha = max\{\beta \subseteq body(r) \mid \Delta \models \beta\})\}$.

The effect of the residue operator is different in total and partial interpretations. However, in both cases it is correct with respect to $Acc(\Phi, \Delta)$, confluent with respect to updates, and computable in linear time.

The particular update expansion operators we propose in this paper generalize those in [8]. They also propagate the relations $\Delta \models l$ and $\Delta \not\models l$ into the clauses of Φ in opposite directions: from bodies to heads and back.

The case of partial DBs

Forward operator F on $\Phi \in \mathbf{IC}$ and $\Delta = (D^+, D^-) \in \mathbf{UP}$:
$F(\Phi, \Delta)^+ = D^+ \cup \{l \mid \exists r \in \Phi\ (l = head(r)\ \&\ \Delta \models body(r))\}$
$F(\Phi, \Delta)^- = D^- \cup \{\neg.l \mid l \in F(\Phi, \Delta)^+\}$.

Backward operator B on $\Phi \in \mathbf{IC}$ and $\Delta = (D^+, D^-) \in \mathbf{UP}$:
$B(\Phi, \Delta)^+ = D^+$
$B(\Phi, \Delta)^- = D^- \cup \{l \mid M_\Phi^{min}(D^+ \cup \{l\})$ *does not agree with* $\Delta\}$.

The case of total DBs

Forward operator F on $\Phi \in \mathbf{IC}$ and $\Delta = (D^+, D^-) \in \mathbf{UP}$:
$F(\Phi,\Delta)^+ \;=\; D^+ \;\cup\; \{a \in \mathbf{B} \mid \exists r \in \Phi\; (a = head(r) \;\&\; \Delta \models body(r))\}$
$F(\Phi,\Delta)^- \;=\; D^- \;\cup\; \{a \in \mathbf{B} \mid \exists r \in \Phi\; (\neg a = head(r) \;\&\; \Delta \models body(r))\}$.

Backward operator B on $\Phi \in \mathbf{IC}$ and $\Delta = (D^+, D^-) \in \mathbf{UP}$:
$B(\Phi,\Delta)^+ \;=\; D^+ \;\cup\; \{a \in \mathbf{B} \mid M^{min}_{\Phi}(D^+ \cup \neg.D^- \cup \{\neg a\})$ does not agree with $\Delta\}$ [4].
$B(\Phi,\Delta)^- \;=\; D^- \;\cup\; \{a \in \mathbf{B} \mid M^{min}_{\Phi}(D^+ \cup \neg.D^- \cup \{a\})$ does not agree with $\Delta\}$.

It is clear that both operators F and B are monotone with respect to the order on **UP**. They are also invariant with respect to the residue operator res and do not change models in $Acc(\Phi,\Delta)$.

We now use these operators to define the forward and backward update expansions.

Forward update expansion Γ_f :

$$\begin{aligned}
&\gamma_f^0(\Phi,\Delta) = (\Phi,\Delta)\\
&\gamma_f(\Phi,\Delta) \;=\; (res(\Phi,\Delta),\\
&\qquad\qquad F(res(\Phi,\Delta),\Delta))\\
&\gamma_f^{n+1}(\Phi,\Delta) \;=\gamma_f(\gamma_f^n(\Phi,\Delta))\\
&\Gamma_f(\Phi,\Delta) = \lim_{n\to\infty} \gamma_f^n(\Phi,\Delta).
\end{aligned}$$

Backward update expansion Γ_b :

$$\begin{aligned}
&\gamma_b^0(\Phi,\Delta) = (\Phi,\Delta)\\
&\gamma_b(\Phi,\Delta) \;=\; (res(\Phi,\Delta),\\
&\qquad\qquad B(res(\Phi,\Delta),\Delta))\\
&\gamma_b^{n+1}(\Phi,\Delta) \;=\gamma_b(\gamma_b^n(\Phi,\Delta))\\
&\Gamma_b(\Phi,\Delta) = \lim_{n\to\infty} \gamma_b^n(\Phi,\Delta).
\end{aligned}$$

The operators F, B, and res we have introduced, have good properties which guarantee the existence of limits for the directed sets of their iterations, and that the operators Γ_f and Γ_b are update expansion operators (see [8]). Meanwhile, their properties in partial and in total interpretations are very different.

6 Optimal update expansion in partial databases

The composition of operators $(\Gamma_f \circ \Gamma_b)$ is again an expansion operator. One can consider the powers of this composition of the form $(\Gamma_f \circ \Gamma_b)^n,\;\; n > 1$. It is surprising that in partial and in total DBs these operators have different properties. In [8] we show that for total DBs these powers can form a proper hierarchy. The expansion operator Γ_{lim} mentioned in Section 5 is the limit of this power hierarchy: $\Gamma_{lim} = (\Gamma_f \circ \Gamma_b)^{\omega}$. In partial DBs the powers hierarchy degenerates.

Theorem 3 *In the case of partial DBs* $(\Gamma_f \circ \Gamma_b)^n = \Gamma_f \circ \Gamma_b$ *for all* $n \geq 1$.

In order to reach the optimal expansion we introduce the following nondeterministic refinement operator on ICs.

[4] We remind that M^{min}_{Φ} is syntactic, i.e. is understood as in partial interpretations.

Definition 8 *Let* $\Phi \in \mathbf{IC}$ *and* $\Delta = (D^+, D^-) \in \mathbf{UP}$. *The* refinement *of* Φ *by* Δ *is a maximal program* $\Phi' = ref(\Phi, \Delta) \preceq \Phi$ *in which every clause* r *meets the conditions:*

(a) *if* $head(r)$ *agrees with* Δ, *then* $M^{min}_{\Phi'}(body(r))$ *agrees with* Δ;

(b) $r \in \Phi'$ *is independent of* $\Phi' \setminus \{r\}$ *(i.e.* $head(r) \notin M^{min}_{\Phi' \setminus \{r\}}(body(r))$*);*

(c) $body(r) \setminus M^{min}_{\Phi'}(\alpha) \neq \emptyset$ *for any* $\alpha \subsetneq body(r)$.

The refinement operator is correct, i.e. $Acc(ref(\Phi, \Delta), \Delta) = Acc(\Phi, \Delta)$, and is computable in square time. Together with the limit expansion operator it gives the needed optimal expansion.

Theorem 4 *Let* Γ_{max} *be the operator on* $\mathbf{IC} \times \mathbf{UP}$ *defined by the equalities:*

(i) $\Gamma_{max}(\Phi, \Delta)^{up} = \Gamma_f \circ \Gamma_b(\Phi, \Delta)^{up}$,

(ii) $\Gamma_{max}(\Phi, \Delta)^{ic} = ref(\Gamma_f \circ \Gamma_b(\Phi, \Delta)^{ic}, \Gamma_f \circ \Gamma_b(\Phi, \Delta)^{up})$.

Then:

1) Γ_{max} *is an optimal update expansion operator.*

2) Γ_{max} *is computable in square time.*

7 Optimal update expansion in total databases

Unfortunately, in the case of total DBs the update Δ^{Φ}_{max} is rather complex.

Theorem 5 *For total DBs the set* $\{(\Phi, \Delta, a) \mid a \in (\Delta^{\Phi}_{max})^+\}$ *is co-NP-complete.*

The composition of the limit operator $(\Gamma_f \circ \Gamma_b)^{\omega}$ and of the refinement operator ref is computed in square time, but it does not give the optimal expansion in general. It can serve as a more or less good approximation to the optimal expansion operator. Nevertheless, there is an interesting subclass of total DBs for which the optimal expansion operator is computable in polynomial time. It is the class of DBs with ICs whose clauses have positive bodies: $\mathbf{IC^{pb}} = \{\Phi \in \mathbf{IC} \mid (\forall r \in \Phi)(\ body(r) \subseteq \mathbf{B})\}$. For this class there exists a close relationship between maximal elements of $Equ(\Phi, \Delta)$ in total interpretations and those in partial ones [5].

Theorem 6 *Let a total update* $\Delta = (D^+, D^-)$ *be compatible with an IC* $\Phi \in \mathbf{IC^{pb}}$. *Then*

(1) *if* Δ^p_{max} *is the maximal update in* $Equ^p(\Phi, \Delta)$, *then* $\Delta^t_{max} = (\neg.D^-_p \cap \mathbf{B}, D^-_p \cap \mathbf{B})$ *is the maximal update in* $Equ^t(\Phi, \Delta)$;

(2) *if* Δ^t_{max} *is the maximal update in* $Equ^t(\Phi, \Delta)$, *then* $\Delta_p = (D^+_t \cup \{\neg a | \exists r \in \Phi\ (head(r) = \neg a\ \&\ body(r) \subseteq D^+_t)\}, \neg.D^+_t \cup D^-_t)$ *is the maximal update in* $Equ^p(\Phi, \Delta)$.

We should amend the definition of partial refinement because it does not give the t-simplest IC, as the following example shows.

[5] In order to distinguish notation in partial and total interpretations we will use in this section upper indices p and t respectively. Considering updates $\Delta = (D^+, D^-)$ with $D^+ \cup D^- \subseteq \mathbf{B}$ in the context of partial DBs we will call them *total*.

Example 4 *Let* $\Phi_1 = \{\neg a \leftarrow b, c;\ c \leftarrow a\}$ *and* $\Phi_2 = \{\neg a \leftarrow b;\ c \leftarrow a\}$. *Clearly,* $\Phi_2 \prec \Phi_1$. *For* $\Delta = (\emptyset, \emptyset)$ *the partial DB state* $\{b\}$ *belongs to* $Acc^p(\Phi_1, \Delta) \setminus Acc^p(\Phi_2, \Delta)$. *However,* $Acc^t(\Phi_1, \Delta) = Acc^t(\Phi_2, \Delta) = \{\emptyset, \{b\}, \{c\}, \{a, c\}, \{b, c\}\}$.

Definition 9 *Let* $\Phi \in \mathbf{IC^{Pb}}$ *and* $\Delta = (D^+, D^-) \in \mathbf{UP}$. *The* t-refinement *of* Φ *by* Δ *is a maximal program* $\Phi' = ref^t(\Phi, \Delta) \preceq \Phi$ *in which every clause r meets the conditions (a)-(c) of definition 8 and also the conditions:*

(d) if $head(r)$ *is a negative literal, then* $M^{min}_{\Phi'}(body(r) \cup \neg.head(r))$ *agrees with* Δ;

(e) if $head(r)$ *is a negative literal, then* $body(r) \setminus M^{min}_{\Phi'}(\alpha \cup \neg.head(r)) \neq \emptyset$ *for any* $\alpha \subsetneq body(r)$.

This nondeterministic operator is correct with respect to $Acc^t(\Phi, \Delta)$ for ICs in $\mathbf{IC^{Pb}}$ and computable in square time. Being combined with partial limit expansion operator it gives the optimal expansion.

Theorem 7 *Let* Γ^t_{max} *be the operator on* $\mathbf{IC^{Pb}} \times \mathbf{UP}$ *defined by the equalities:*

(i) $\Gamma^t_{max}(\Phi, \Delta)^+ = \neg.(\Gamma_f \circ \Gamma_b(\Phi, \Delta)^-) \cap \mathbf{B}$,

(ii) $\Gamma^t_{max}(\Phi, \Delta)^- = \Gamma_f \circ \Gamma_b(\Phi, \Delta)^- \cap \mathbf{B}$,

(iii) $\Gamma^t_{max}(\Phi, \Delta)^{ic} = ref^t(\Gamma_f \circ \Gamma_b(\Phi, \Delta)^{ic}, \Gamma_f \circ \Gamma_b(\Phi, \Delta)^{up})$.

Then:

1) Γ^t_{max} *is an optimal update expansion operator for total DBs with ICs in* $\mathbf{IC^{Pb}}$.

2) Γ^t_{max} *is computable in square time.*

Restricting ourself to ICs in $\mathbf{IC^{Pb}}$ we find a simple relation between the solutions of the *EUP* in partial and total DBs.

Theorem 8 *Let a total update* $\Delta = (D^+, D^-)$ *be compatible with IC* $\Phi \in \mathbf{IC^{Pb}}$ *and* $I \subseteq \mathbf{B}$ *be a DB state. Then:*

(1) if a partial DB state $I_1 \in Acc^p(\Phi, \Delta)$ *is minimally deviating from I with respect to* $Acc^p(\Phi, \Delta)$, *then the total DB state* I_1^+ *belongs to* $Acc^t(\Phi, \Delta)$ *and is minimally deviating from I with respect to* $Acc^t(\Phi, \Delta)$;

(2) if a total DB state $I_1 \in Acc^t(\Phi, \Delta)$ *is minimally deviating from I with respect to* $Acc^t(\Phi, \Delta)$, *then the partial DB state* $I'_1 = I_1 \cup \{\neg a \mid \exists r \in \Phi\ (head(r) = \neg a\ \&\ body(r) \subseteq I_1\}$ *belongs to* $Acc^p(\Phi, \Delta)$ *and is minimally deviating from I with respect to* $Acc^p(\Phi, \Delta)$.

Example 5 *Consider the following IC* $\Phi \in \mathbf{IC^{Pb}}$:

$r_1 : salary(100) \leftarrow dept(cs), pos(programmer);$

$r_2 : \neg pos(programmer) \leftarrow dept(cs), salary(30), edu(high);$

$r_3 : \neg salary(30) \leftarrow salary(100);$

$r_4 : \neg dept(cs) \leftarrow edu(low);$

$r_5 : edu(high) \leftarrow pos(programmer).$

Clearly, $I = \{salary(30), pos(programmer), edu(high)\}$ *is a partial as well as a total model of* Φ. *Suppose we add a new fact to this DB state:* $D^+ = \{dept(cs)\}$. *There are two ways to treat this update. We can consider it partial. Then the operator* Γ_{max} *returns* Φ_1 :

$r'_1 : salary(100) \leftarrow pos(programmer);$
$r'_2 : \neg pos(programmer) \leftarrow salary(30), edu(high);$
$r_3 : \neg salary(30) \leftarrow salary(100);$
$r_5 : edu(high) \leftarrow pos(programmer).$

and $D_p^+ = \{dept(cs)\}$, $D_p^- = \{\neg dept(cs), edu(low)\}$. *Applied to this expansion, Dp_search gives the DB state* $I_1 = \{salary(100), pos(programmer), edu(high), dept(cs), \neg salary(30)\}$ *minimally deviating from* I *with respect to* $Acc^p(\Phi, \Delta)$. *By theorem 8 it is transformed into the total DB state:* $I_1^+ = \{salary(100), pos(programmer), edu(high), dept(cs)\}$ *minimally deviating from* I *with respect to* $Acc^t(\Phi, \Delta)$.

We can also consider this update total. Then the operator Γ^t_{max} *returns a slightly simpler IC* Φ''_1 :

$r'_1 : salary(100) \leftarrow pos(programmer);$
$r''_2 : \neg pos(programmer) \leftarrow salary(30);$
$r_3 : \neg salary(30) \leftarrow salary(100);$
$r_5 : edu(high) \leftarrow pos(programmer).$

and $D_t^+ = \{dept(cs)\}$, $D_t^- = \{edu(low)\}$. *The resulting total DB state* $I_t = \{dept(cs), edu(high), salary(100), pos(programmer)\}$ *minimally deviating from* I *is the same as above.* r''_2 *results from* r_2 *by point (e) of the definition of* ref^t *because* $edu(high) \in M^{min}(\neg.head(r'_2)) = M^{min}(pos(programmer))$.

8 Conclusion

The problem of computing the minimal real change restoring the correctness of an updated DB state, is proven to be hard. We subdivide this problem into two parts:

- first part is to maximally expand the update, taking into account all its consequences with respect to IC, and simultaneously to maximally simplify the IC itself with respect to this expansion;
- second part is to find the minimal real change of a given DB state.

The first part does not depend on the initial DB state, so it should be included in any DB update procedure. We show that in partial DBs this part is computable in square time. In classical DBs this problem is itself quite hard. So we propose its reasonable approximation computed in square time. Moreover, we show that for ICs with positive clause bodies the maximal expansion is computed in square time for classical DBs. As it concerns the second part, even standard complete choice procedures solving this part of the update problem can be substantially simplified, being applied to the expanded update and the simplified IC in the place of the initial ones (cf. the dynamic optimization method in [8]). Moreover, this techniques may lead to efficient interactive update algorithms, where the maximal expanded update is accomplished in current DB state, then conflicts are proposed to resolve, which leads to a new update, etc.

9 Acknowledgements

We are grateful to the anonymous referees for their helpful comments.

References

1. Abiteboul, S.:Updates a new Frontier. In: *Proc. of the Second International Conference on the Theory of Databases, ICDT'88.* LNCS **326** (1988) 1-18.
2. Alferes, J.J.,Pereira, L.M.: Update-Programs Can Update Programs. In: J.Dix, L.M. Pereira, T.C. Przymusinski, editors: *Second International Workshop, NMELP'96. Selected Papers.* LNCS **1216** (1997) 110-131.
3. Chitta Baral and Michael Gelfond: Logic programming and knowledge representation. *Journal of Logic Programming*, 12:1–80, 1994.
4. Bonner, A.J., Kifer, M.: An Overview of Transaction Logic. *Theoretical Computer Science,* **133**(2) (1994), 2-5-265.
5. Decker H.: An extension of SLD by abduction and integrity maintenance for view updating in deductive databases. In: *Proc. of the 1996 International Conference on Logic Programming.* MIT Press, (1996), 157-169.
6. Dekhtyar, M., Dikovsky, A., Spyratos, N.: On Conservative Enforced Updates. In: Dix, J., Furbach, U., Nerode, A., editors: *Proceedings of 4th International Conference, LPNMR'97.* Dagstuhl Castle, Germany, LNCS **1265** (1997) 244-257.
7. Dekhtyar, M., Dikovsky, A., Spyratos, N.: On Logically Justified Updates. In: J. Jaffar, editor: *Proc. of the 1998 Joint International Conference and Symposium on Logic Programming.* MIT Press, (1998), 250-264.
8. Dekhtyar, M., Dikovsky, A., Dudakov S., Spyratos, N.: Monotone expansion of updates in logical databases. To be published in the Proc. of LPNMR'99.
9. Eiter, T., Gottlob, G.: On the complexity of propositional knowledge base revision, updates, and counterfactuals. *Artificial Intelligence.* **57** (1992) 227-270.
10. Eshghi, K., Kowalski, R. A.: Abduction Compared with Negation by Failure. In: *Proc. of the 1989 International Conference on Logic Programming.* (1989)
11. Gelfond, M., Lfschitz, V.: The stable semantics for logic programs. In: R.Kovalsky and K.Bowen, editors, *Proc. of the 5th Intern. Symp. on Logic Programming.* Cambridge, MA, MIT Press (1988) 1070-1080.
12. Guessoum A., Lloyd J.W.: Updating knowledge bases. *New Generation Computing.* **8** (1990), 71-89.
13. Halfeld Ferrari Alves, M., Laurent, D., Spyratos, N., Stamate, D.: Update rules and revision programs. Rapport de Recherche Université de Paris-Sud, Centre d'Orsay, LRI **1010** (12 / 1995).
14. Hanson, E.N., Widom, J., An Overview of Production Rules in Database Systems. *The Knowledge Engineering Review.* **8** (1993), N. 2, 121-143.
15. Kakas A.C., Mancarella P.: Database updates through abduction. In: *Proc. 16th VLBD Conference.* (1990) 650-661.
16. Vladimir Lifschitz: Foundations of logic programming. In: Gerhard Brewka, editor, *Principles of Knowledge Representation*, pages 69–128. CSLI Publications, 1996.
17. Lloyd, J.W., Foundations of Logic Programming. Second, Extended Edition. Springer-Verlag. (1993)
18. Marek, V.W., Truszcińsky, M.: Revision programming, database updates and integrity constraints. In: *International Conference on Data Base theory, ICDT.* LNCS **893** (1995) 368-382.
19. Przymusinski, T.C., Turner, H.: Update by Means of Inference Rules. In: V.W.Marek, A.Nerode, M.Truszczyński, editors, *Logic Programming and Nonmonotonic Reasoning.* Proc. of the Third Int. Conf. LPNMR'95, Lexington, KY, USA (1995) 166-174.

20. Tomasic A.: View update translation via deduction and annotation. In: *Proc. of the Second International Conference on the Theory of Databases, ICDT'88*. LNCS **326** (1988) 338-352.
21. Van Gelder, A., Ross, K.A., and Schlipf, J.S.: The Well-Founded Semantics for General Logic Programs. *Journal of the ACM.* **38** (1991) 620-650.