On Complexity of Updates Through Integrity Constraints*

Michael Dekhtyar[1] , Alexander Dikovsky[2] and Sergey Dudakov[1]

[1] Dept. of CS, Tver State Univ. 33 Zheljabova str. Tver, Russia, 170000
Michael.Dekhtyar@tversu.ru, p000104@tversu.ru
[2] 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

Abstract. The computational complexity is explored of finding the minimal real change of a database after an update constrained by a logic program. A polynomial time algorithm is discovered which solves this problem for ground IC in partial interpretations. Formulated in a "property" form, even under the premise of fixed database scheme, this problem turns out to be complete in the first three classes of Σ and Π polynomial hierarchies, depending on many factors: type of interpretation (total or partial), presence of variables, use of negation, arity of predicates, etc. Meanwhile, we show that under strong restrictions to negative constraints the problem is solvable in polynomial time. If the database scheme may vary, the complexity grows exponentially.

1 Introduction

Database updates whose impact on database states is specified by systems of IF-THEN rules or by logic programs are in the focus of research till late 80ies. The interest in such updates has quickened in the past few years by the emergence of databases with intelligent update enforcement features (such as triggers), in particular, of active databases ([8, 5, 15, 27, 22, 25, 6]). Initially, the interest in rule based updates was aroused by the need in generalizations of SQL-like declarative update definitions (cf. [1, 2]). Subsequently, this field was influenced by investigation of knowledge base updates initiated by [3]. One approach, influenced by [26], regards the result of a theory update as the theory of its updated (revised) models (see [21, 23]). Another approach follows the line of [19], where a propositional update formula transforms an initial formula into a new formula. In the first order case both the updated theory, and the update itself are represented by logic programs (see e.g. [4]). Model based updates provide new models (DB states) minimally deviating (in some sense) from the initial ones, sometimes explicitly, sometimes not. An alternative operational approach to updates is based on derivations in logic programs. For instance, abduction, sometimes combined with SLD or SLDNF, is used for view updates (cf. [18, 9, 16]).

Still another approach to database updates was proposed in [10, 11, 17], and

* This work was sponsored by the Russian Fundamental Studies Foundation (Grants 98-01-00204, 99-01-00374, 00-01-00254).

developed in [12, 13]. It applies to databases with integrity constraints (IC), expressed in the form of a logic program. Accomplishing updates in presence of such IC needs subsequent conflict resolution. This approach departs from the premise that IC is not intended for data or knowledge definition. Rather, they specify the conflicts to avoid after updates. So in this approach the use of exclusively "intended" models of IC may lead to the loss of information or to unjustified conflict resolution failures, which is illustrated by the following simple 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*)":*

$$\begin{aligned} \mathsf{bathe} &\leftarrow \mathsf{children}, \mathsf{parents} \\ \neg\mathsf{bathe} &\leftarrow \mathsf{children}, \mathsf{ebb}. \end{aligned}$$

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 fact a *holds if* $\mathsf{a} \in S$, $\neg\mathsf{a}$ *holds if* $\neg\mathsf{a} \in S$ *explicitly, 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 positive update 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 same update in 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:* $I_2 = \{\mathsf{parents}, \mathsf{ebb}, \neg\mathsf{bathe}\}$*.*

This is why after an update all models of IC are considered, where the update is accomplished. However, among these models one should find one minimally deviating from the initial model. A bit more formally, this "enforced update problem (EUP)" 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 $\Psi(I) \models \Phi$). So we see that the EUP is a "function" and not a "property" problem. The closest "property" problems are those of existence of EUP-solutions marked by presence/absence of a given fact (**OFIP**), and of presence/absence of a given fact in all EUP-solutions (**PFIP**). In papers [12, 13] these problems were investigated for databases with ground IC and were shown to be untractable in worst case: namely, complete in first two classes of Σ and Π polynomial hierarchies. In contrast with this, in this paper we find a polynomial time algorithm for the EUP itself in the same class of ground IC in partial interpretations. We suspect that there is no such algorithm in total interpretations.

In this paper we investigate the impact of interpretation type and the use of variables in clauses of IC on the complexity of **OFIP** and **PFIP**. We show that for definite ground IC, and positive updates and initial states these problems are solvable in polynomial time in both interpretations. The use of variables in this positive case makes the problems complete respectively in *NP* and *co-NP*. Possibility of deletions makes them complete respectively in *NP* and *co-NP* for ground definite IC, and complete respectively in Σ_2^p and Π_2^p for definite IC with variables. In general they are respectively Σ_2^p-complete and Π_2^p-complete in partial interpretations, and Σ_3^p-complete and Π_3^p-complete in total interpretations. In the case, where the DB signature may vary, the complexity of these problems grows exponentially.

The paper is organized as follows. The next section contains preliminary notions and notation. In Section 3 we formulate the EUP. Subsection 4.1 contains complexity results for partial and total interpretations under the premise of fixed signature. In subsection 4.2 the case of varying signature is explored.

2 Preliminaries

We assume that the reader is familiar with the basic concepts and terminology of logic programming (see [20]).

Language. Let $\mathbf{S}$ be a 1st order signature with a set of constants $\mathbf{C}$ and no other function symbols. Sometimes in this paper $\mathbf{S}$ will be fixed with infinite $\mathbf{C}$, sometimes it will be finite and its size will be considered as a parameter of complexity. A *domain* is a finite subset $\mathbf{D}$ of $\mathbf{C}$. For each domain $\mathbf{D}$ by $\mathbf{A}(\mathbf{S},\mathbf{D})$, $\mathbf{L}(\mathbf{S},\mathbf{D})$, $\mathbf{B}(\mathbf{S},\mathbf{D})$ and $\mathbf{LB}(\mathbf{S},\mathbf{D})$ 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. Integrity constraints (IC) will be expressed by generalized logic programs in $\mathbf{S}$ and $\mathbf{D}$ with explicit negation, i.e. finite sets of clauses of the form $r = (l \leftarrow l_1, ..., l_n)$ where $n \geq 0$ and $l, l_i \in \mathbf{L}(\mathbf{S},\mathbf{D})$, (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. $\mathbf{D}$ being fixed, we consider groundisations of clauses only over $\mathbf{D}$. $gr(\Phi)$ will denote the set of all ground instances of clauses in Φ. $\mathbf{IC}(\mathbf{S},\mathbf{D})$ will denote the set of all integrity constraints in the signature $\mathbf{S}$ with constants in $\mathbf{D}$.

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 $\mathbf{LB}(\mathbf{S},\mathbf{D})$. For such an interpretation $I \subseteq \mathbf{LB}(\mathbf{S},\mathbf{D})$ we set $I^+ = I \cap \mathbf{B}(\mathbf{S},\mathbf{D})$ and $I^- = I \cap \neg.\mathbf{B}(\mathbf{S},\mathbf{D})$. 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

$I^+ \cup \neg.I^- = \mathbf{B}(\mathbf{S},\mathbf{D})$ and $I^+ \cap \neg.I^- = \emptyset$. Note that total interpretations are completely defined by their positive parts, so we will identify total interpretations with subsets of $\mathbf{B}(\mathbf{S},\mathbf{D})$. Given an IC $\Phi \in \mathbf{IC}(\mathbf{S},\mathbf{D})$ and a DB state I over $\mathbf{D}$, a ground clause $r = (l \leftarrow l_1, ..., l_n)$ in $gr(\Phi)$ 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 $gr(\Phi)$ is valid in I.

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

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

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

Updates. When partial interpretations over $\mathbf{D}$ are considered, an *update* is a pair $\Delta = (D^+, D^-)$ where D^+, D^- are subsets of $\mathbf{LB}(\mathbf{S},\mathbf{D})$. In the case of total interpretations D^+, D^- are subsets of $\mathbf{B}(\mathbf{S},\mathbf{D})$. 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. We will denote the components D^+ and D^- respectively by Δ^+ and Δ^-. For both kinds of interpretations $\mathbf{UP}(\mathbf{S},\mathbf{D})$ will denote the set of all updates in the signature $\mathbf{S}$ and with constants in $\mathbf{D}$. We say that Δ is *accomplished* in I if $\Delta^+ \subseteq I$ and $\Delta^- \cap I = \emptyset$.

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

3 Conservative rule based updates

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

Definition 1 *For $\Phi \in$ **IC** and $\Delta \in$ **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 $Acc(\Phi, \Delta) \neq \emptyset$.*

In [10] we propose the following minimal deviation criterion implementing the intention 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* K *be a class of DB states. We say that* I_1 *is* minimally deviating from I with respect to K *if*
$\forall I_2 \in K\ (\neg(I \cap I_1 \subsetneq I \cap I_2)\ \&\ ((I \cap I_1 = I \cap I_2) \ \rightarrow\ \neg(I_2 \setminus I \subsetneq I_1 \setminus I)))$.

In terms of this criterion the *conservative update operators* we consider have been defined in [11] as follows.

Definition 3 *Let* Δ *be a given update which is compatible with IC* Φ*. An operator* Ψ *on the set of DB states 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* $Acc(\Phi, \Delta)$.

4 Computational complexity of conservative updates

The Enforced Update Problem (EUP) we discuss in the Introduction is the problem of calculation of a conservative update operator Ψ for some given IC, update and input state. So it is a "function" type problem. In order to measure the complexity of conservative updates in a "property" form we use two standard algorithmic problems: *Optimistic* and *Pessimistic Fall-Into-Problem* (**OFIP** and respectively **PFIP**) (cf. [14]).
OFIP: Given some $\Delta \in \mathbf{UP}$ compatible with $\Phi \in \mathbf{IC}$, an initial state I, and a literal $l \in \mathbf{LB}$, one should check whether there exists a DB state 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** models I_1 satisfying (a),(b).
We denote respectively by **OFIP** and **PFIP** the sets of all solutions (I, Δ, Φ, l) of these problems.

Typical database updates do not change database scheme. In logical terms this corresponds to the situation, where predicate signature **S** is fixed. Under this premise we consider the combined complexity of **OFIP** and **PFIP** with respect to the problem size evaluated as $N = |\mathbf{D}| + |I| + |\Delta| + |\Phi| + |l|$ ($|\ |$ being the size of constant or literal sets, and of programs in some standard encoding). Signature **S** being fixed, for a given domain **D** the maximal size of a DB state is bounded by a polynomial of the order $\boldsymbol{O}(|\mathbf{D}|^a)$, where a is the maximal arity of predicates in **S**. When **S** is not fixed, its size is included into the problem size.

We use in this paper the following multiparameter reduction scheme which serves for most lower bounds in the case of a fixed one-predicate signature.

Let $\mathbf{S} = \{s^{(2)}\}$, $\alpha = d_1 \wedge ... \wedge d_n$ be a 3-CNF, where in each clause $d_i = (\neg)u_{i1} \vee (\neg)u_{i2} \vee (\neg)u_{i3}$ $(1 \leq i \leq n)$ u_{ij} are propositional variables. Let V be the set of all these variables. Given a subset $R \subseteq V$, we form the set of constants $C(\alpha, R) = \{t, f, t_1, t_2, t_3, f_1, f_2, f_3, 1, 2, 3, p_{000}, p_{001}, \ldots, p_{111}, d_1, \ldots, d_n\} \cup \{c_x | x \in R\}$. Informally, d_i encodes the ith clause, t or f fixes its value, 1, 2, or

3 fixes j in u_{ij}, t_j or f_j fixes the value of u_{ij}, $p_{b_1 b_2 b_3}$ fixes values of u_{i1},u_{i2},u_{i3}, and c_x is the DB constant for the boolean variable $x \in R$. We construct the following α, R-dependent part $J(\alpha, R)$ of initial DB states.
$J(\alpha, R) = J \cup J_\alpha \cup J_R$, where :

$$J = \{s(t,t_j), s(f,f_j) | 1 \leq j \leq 3\} \cup \{s(t_j,j), s(f_j,j) | 1 \leq j \leq 3\} \cup \bigcup_{b_1 b_2 b_3} \{\{s(t_j, p_{b_1 b_2 b_3}) \mid b_j = 1\} \cup \{s(f_j, p_{b_1 b_2 b_3}) \mid b_j = 0\}\};$$

$$J_\alpha = \bigcup_{b_1 b_2 b_3} \{s(d_i, p_{b_1 b_2 b_3}) | \text{ if } i\text{-th clause is true on } b_1 b_2 b_3, 1 \leq i \leq 3\}, \text{ and}$$

$$J_R = \{s(t, c_x), s(f, c_x) \mid x \in R\}.$$

Given α and two sets of variables $R_1, R_2 \subseteq V$, $R_1 \cap R_2 = \emptyset$, we construct the set of atoms

$$\varphi(\alpha, R_1, R_2) = \bigcup_{i=1}^{n} \{s(d_i, P_i)\} \cup \beta_1^i(u_{i1}) \cup \beta_2^i(u_{i2}) \cup \beta_3^i(u_{i3}),$$

where

$$\beta_j^i(x) = \begin{cases} \{s(V_x, c_x), s(V_x, W_{ij}), s(W_{ij}, j), s(W_{ij}, P_i)\} & , \ for\ x \in R_1 \\ \{s(V_x, W_{ij}), s(W_{ij}, j), s(W_{ij}, P_i)\} & , \ for\ x \in R_2. \end{cases}$$

P_i, W_{ij}, V_x being object variables. Intuitively, P_i fixes a triple $p_{b_1 b_2 b_3}$ which makes d_i true, the value of W_{ij} fixes the value b_j of u_{ij} in this triple, and finally, V_x fixes the value (t or f) of x. Therefore, $\beta_j^i(x)$ describes the value of the propositional variable x in j-th literal of i-th clause. Notice that both, $J(\alpha, R)$ and $\varphi(\alpha, R_1, R_2)$ contain only positive literals.

The following lemma relates the satisfiability of α to the validity of φ on J.

Lemma 1 *Let $\bar{x}, \bar{y}$ be a partition of the set of variables of a 3-CNF $\alpha(\bar{x}, \bar{y})$. In our construction let $C = C(\alpha, \bar{x})$ and*

$$I = J(\alpha, \emptyset) \cup \bigcup_{x \in \bar{x}} \{s(t, c_x) \mid \sigma(x) = 1\} \cup \{s(f, c_x) \mid \sigma(x) = 0\},$$

for some boolean substitution $\sigma : \bar{x} \to \{0,1\}$. Then $\alpha(\sigma\bar{x}, \bar{y})$ is satisfiable iff $I \models \varphi(\alpha, \bar{x}, \bar{y}) \circ \tau$, for some object variables substitution τ.

Proof. If α is satisfiable, then there is an extension σ' of σ to $\bar{y}$ such that $\alpha(\sigma'\bar{x}, \sigma'\bar{y})$ is true. Then we set $\tau P_i = p_{\sigma'(u_{i1} u_{i2} u_{i3})}$,

$$\tau V_u = \begin{cases} t & , \ if\ \sigma'(u) = 1, \\ f & , \ if\ \sigma'(u) = 0, \end{cases} \text{ for all } u \in \bar{x} \cup \bar{y},$$

$$\tau W_{ij} = \begin{cases} t_j & , \ if\ \sigma'(u_{ij}) = 1, \\ f_j & , \ if\ \sigma'(u_{ij}) = 0, \end{cases} \text{ for } 1 \leq i \leq n, 1 \leq j \leq 3.$$

It easy to check that $I \models \varphi(\alpha, \bar{x}, \bar{y}) \circ \tau$.

Let $\varphi(\alpha, \bar{x}, \bar{y})$ be valid in I under some object variables substitution τ. Then $\tau P_i \in \{p_{000}, p_{001}, ..., p_{111}\}$ since $I \models s(d_i, P_i) \circ \tau$, $\tau W_{ij} \in \{t_j, f_j\}$ since $I \models s(W_{ij}, j) \circ \tau$, and $\tau V_u \in \{t, f\}$ since $I \models s(V_u, W_{ij}) \circ \tau$. We define σ' from τ as follows : $\sigma'(u) = 1$ if $\tau V_u = t$ and $\sigma'(u) = 0$ otherwise. Let us observe that σ' coincides with σ on $\bar{x}$. Indeed, $\varphi(\alpha, \bar{x}, \bar{y})$ contains a fact $s(V_x, c_x)$ valid in I only if $\sigma(x) = \sigma'(x)$, for all $x \in \bar{x}$. As $\varphi(\alpha, \bar{x}, \bar{y})$ contains the facts $s(V_{u_{ij}}, W_{ij})$, $s(W_{ij}, j)$, and $s(W_{ij}, P_i)$, then $\tau P_i = p_{\sigma'(u_{i1} u_{i2} u_{i3})}$. Now, from $I \models s(d_i, P_i) \circ \tau$ it follows that the clause d_i is true under σ' so $\alpha(\sigma\bar{x}, \sigma'\bar{y})$ is true. $\Box$.

The complexity of the problems we consider depends on many factors: presence of variables in clauses of IC, use of negation, arity of predicates, etc. The main factor is the interpretation type. It turns out that the same problems are simpler in partial interpretations than in classical total interpretations. For example, in partial interpretations the compatibility problem $Acc(\Phi, \Delta) \neq \emptyset$ for ground IC Φ is resolved in linear time, whereas, it is NP-complete in total interpretations [12] for the same class of IC. As will be shown, in partial interpretations both problems **OFIP** and **PFIP** have a wide spectrum of complexity depending on specific factors, such as presence of negative literals in DB states or in update. The other important factor is groundness of IC. Such basic problem as model checking is resolved in linear time for ground IC in both interpretations. Meanwhile, even for definite IC Φ the problem $MC = \{< I, \Phi > \mid I \models \Phi\}$ is *co-NP*-complete in both interpretations, which follows for example from the respective complexity bounds for conjunctive queries [7].

4.1 The case of fixed signature

Complexity in partial interpretations. We begin with several simple observations.

Proposition 1 *For any IC Φ:*
(1) if $M_\Phi^{min}(S)$ is consistent for some set of literals S, then $M_\Phi^{min}(S) \models \Phi$;
(2) if $I \models \Phi$ and $S \subseteq I$, then $M_\Phi^{min}(S) \models \Phi$;
(3) if Φ is compatible with some Δ, then $M_\Phi^{min}(\Delta^+)$ is the least model in $Acc(\Phi, \Delta)$;
(4) if Φ is a definite IC compatible with some positive Δ (i.e., $\Delta^+ \subseteq \mathbf{B}$ and $\Delta^- = \emptyset$), and $I \subseteq \mathbf{B}$, then $M_\Phi^{min}(\Delta^+ \cup I)$ is the only DB state minimally deviating from I with respect to $Acc(\Phi, \Delta)$.

The premise of fixed predicate signature provides important polynomial time algorithms.

Proposition 2 *Let Φ be an IC and S be a set of literals.*
(1) There is an algorithm that constructs $M_\Phi^{min}(S)$ in polynomial time, if Φ is ground.
(2) There is a nondeterministic uniformly (i.e. in all computations) polynomial time algorithm, which constructs $M_\Phi^{min}(S)$ in some its computation, and a subset of $M_\Phi^{min}(S)$ in any its computation.

These propositions lead to the following interesting characterization of the EUP solutions in partial interpretations.

Theorem 1 *Let Φ and Δ be compatible. Then I_1 is minimally deviating from I with respect to $Acc(\Phi, \Delta)$ iff there is a maximal subset $S \subseteq I$ such that*

$$I_1 = M_\Phi^{min}(S \cup \Delta^+) \text{ is consistent and } I_1 \cap \Delta^- = \emptyset.$$

The proof of this theorem uses the following lemma, which is interesting for itself and provides important consequences for ground IC.

Lemma 2 *There is a polynomial time algorithm constructing some DB state $I_1 \in Acc(\Phi, \Delta)$* minimally deviating *from I with respect to $Acc(\Phi, \Delta)$, from an initial DB state I, any DB state $I_0 \in Acc(\Phi, \Delta)$, and ground IC Φ compatible with Δ.*

Proof scheme: Let $I = \{l_1, l_2, \ldots, l_n\}$, Δ, Φ and some DB state $I_0 \in Acc(\Phi, \Delta)$ be given. We define the following sequence of sets S_i, $0 \leq i \leq n$.

$$S_0 = (I_0 \cap I) \cup D^+ \text{ and } S_{i+1} = \begin{cases} S_i, & \text{if } M_\Phi^{min}(S_i \cup \{l_{i+1}\}) \text{ is inconsistent or} \\ & M_\Phi^{min}(S_i \cup \{l_{i+1}\}) \cap D^- \neq \emptyset \\ S_i \cup \{l_{i+1}\}, \text{ otherwise.} & \end{cases}$$

Let $I_1 = M_\Phi^{min}(S_n)$. By construction and by Proposition 1, $I_1 \in Acc(\Phi, \Delta)$. No literal $l \in I \setminus I_1$ can be added to I_1, because $\{l\} \cup I_1$ is inconsistent or contradicts Δ^-. No literal $l \in I_1 \setminus I$ can be removed from I_1, because it is inferred from S_n, which contains only literals in I or in D^+. Hence, I_1 minimally deviates from I with respect to $Acc(\Phi, \Delta)$. Note that if I_0 minimally deviates from I with respect to $Acc(\Phi, \Delta)$, then $I_1 = I_0$. Clearly, I_1 is constructed in polynomial time. □

Now we can prove Theorem 1:

Proof scheme: ($\Rightarrow$) Let $S = I_1 \cap I$ and $I_0 = M_\Phi^{min}(S \cup \Delta^+)$. Being a subset of I_1, I_0 is consistent. So $I_0 \models \Phi$. By monotonicity of $M_\Phi^{min}(X)$, $\Delta^+ \subseteq I_0$ and $\Delta^- \cap I_0 = \emptyset$, so $I_0 \in Acc(\Phi, \Delta)$. For the same reason, $I_0 \cap I = I_1 \cap I = S$. So if there is some $l \in I_1 \setminus I_0$, then I_1 is not minimally deviating from $Acc(\Phi, \Delta)$.

($\Leftarrow$) Being consistent, $I_1 = M_\Phi^{min}(S \cup \Delta^+)$ is a model of Φ. So $I_1 \in Acc(\Phi, \Delta)$. Apply Lemma 1 to so defined I_1 in the role of I_0. The result will coincide with I_1 by construction. Therefore, I_1 is minimally deviating from $Acc(\Phi, \Delta)$. □

If Φ and Δ are compatible, then by Proposition 1, the model $I_0 = M_\Phi^{min}(\Delta^+)$ is in $Acc(\Phi, \Delta)$. So together with Proposition 2, this lemma gives a surprising consequence: for ground IC one can find **some** solution of the EUP in polynomial time.

Corollary 1 *There is a polynomial time algorithm constructing* some *DB state $I_1 \in Acc(\Phi, \Delta)$ minimally deviating from I with respect to $Acc(\Phi, \Delta)$, from an initial DB state I and ground IC Φ compatible with Δ.*

Sure, this doesn't work for **OFIP**, because the given literal l may not fall into this particular solution. Indeed, as it is shown in [13], **OFIP** is a hard problem even for ground IC.

Theorem 2 *[13] Let IC be ground. Then:*

(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) Δ is positive, i.e. $\Delta^+ \subseteq \mathbf{B}$ and $\Delta^- = \emptyset$, 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.*

Let us analyze the complexity of **OFIP** and **PFIP** for general IC with variables. There is one very special case, where these problems are solvable in polynomial time: that of positive DB states and updates and definite monadic IC. In this case one can construct the consequence closure of polynomial size. Following to Proposition 1.4, **OFIP** and **PFIP** are equivalent in this case.

Proposition 3 *There is a polynomial time algorithm, which decides whether* $(\Delta, \Phi,\ I,\ l) \in$ **OFIP** *(same for* **PFIP***) for definite* Φ *containing only unary predicates, and for positive* $\Delta, I,$ *and* l.

Even the use of a single binary predicate can increase the complexity of both problems, when Δ is positive and Φ is a definite IC with variables. Interestingly, the complexity depends on positivity of l.

Theorem 3 *In the case, where IC are definite, and updates are positive:*
(1) **OFIP** *is NP-complete if* l *is positive;*
(2) **OFIP** *is co-NP-complete if* l *is negative;*
both lower bounds are valid even for a one binary predicate signature.

Proof scheme: *Lower bound.* In our reduction scheme for a 3-CNF α with variables V we take some new variable a and set $C = C(\alpha, \{a\})$, $\Delta = (J(\alpha, \emptyset), \emptyset)$, $\Phi = \{s(T, A) \leftarrow \varphi(\alpha, \emptyset, V)\}$, and $I = \{\neg s(T, A)\}$. Then α is satisfiable iff $(I, \Delta, \Phi, s(T, A)) \in$ **OFIP**, and iff $(I, \Delta, \Phi, \neg s(T, A)) \notin$ **OFIP**. □

Corollary 2 *In the case, where IC are definite, and* Δ *and* I *are positive,* **PFIP** *is NP-complete (co-NP-complete), if* l *is positive (negative).*

For the same class of IC and of updates, emergence of negative literals in initial DB states increases the complexity of **PFIP**.

Theorem 4 *In the case, where IC are definite, and* Δ *are positive,* **PFIP** *is* Π_2^p*-complete.*

Proof scheme: *Lower bound.* Let us consider a sentence $\beta = \forall \overline{x} \exists \overline{y} \alpha(\overline{x}, \overline{y})$, where α is 3-CNF and V_X and V_Y are the sets of x and of y respectively. Let a and b be new different variables. Then we set $C = C(\alpha, V_X \cup \{a, b\})$, $I = J(\alpha, V_X) \cup \{\neg s(T, A)\}$, and define Φ by:
$s(T, A) \leftarrow s(F, A), s(T, X), s(F, X)$, for $x \in V_X$;
$s(T, B) \leftarrow s(T, A)$;
$s(T, B) \leftarrow \{s(F, A)\} \cup \varphi(\alpha, V_X, V_Y)$;
$s(x, y) \leftarrow$ for all (x, y) such that $s(x, y) \in J(\alpha, \emptyset)$.
Let $\Delta = (\{s(F, A)\}, \emptyset)$ and $l = s(T, B)$. Then one can prove that $(I, \Delta, \Phi, l) \in$ **PFIP** iff β is true. □

Discarding the constraint of positivity of updates, we still increase the complexity of both problems. As it concerns monadic definite IC, their complexity is as that of ground IC.

Proposition 4 *In the case, where IC are definite and use only unary predicates,* **OFIP** *is NP-complete, and* **PFIP** *is co-NP-complete.*

For definite IC with arbitrary predicates the problems become complete on the second level of polynomial Σ and Π hierarchies.

Theorem 5 *In the case, where IC are definite:*
(1) **OFIP** *is* Σ_2^p*-complete;*
(2) **PFIP** *is* Π_2^p*-complete;*
both lower bounds are valid even for a one binary predicate signature.

Proof scheme: (1) *Lower bound.* Let $\beta = \exists\overline{x}\forall\overline{y}\neg\alpha(\overline{x},\overline{y})$ for a 3-CNF α. Let V_X and V_Y be the sets of all variables x and y respectively. We set $C = C(\alpha, V_X \cup \{a,b,c,d\} \cup \{b_x : x \in V_X\})$, $I = J(\alpha, V_X) \cup \{s(T,D), s(T,A)\}$, and define Φ by:
$s(T,A) \leftarrow s(T,X), s(F,X)$, for $x \in V_X$;
$s(T,B_x) \leftarrow s(T,C), s(T,X)$, and $s(T,B_x) \leftarrow s(T,C), s(F,X)$, for $x \in V_X$;
$s(T,A) \leftarrow \{s(T,D)\} \cup \varphi(\alpha, V_X, V_Y)$;
$s(T,B) \leftarrow s(T,D), s(T,B_{x_1}), \ldots, s(T,B_{x_n})$;
$s(x,y) \leftarrow$ for all (x,y) such that $s(x,y) \in J(\alpha,\emptyset)$.
Finally, let $\Delta = (\{s(T,C)\}, \{s(T,A)\})$, and $l = s(T,B)$.

Note that for any $I_1 \in Acc(\Phi,\Delta)$ at most one of facts $s(T,X)$, $s(F,X)$ can belong to I_1. If for some x neither $s(T,X)$ nor $s(F,X)$ is in I_1 then I_1 doesn't contain $s(T,B)$.

Now, if formula β is true, then there is a substitution σ such that $\alpha(\sigma\overline{x},\overline{y})$ is false for all $\overline{y}$. Let us add $s(T,X)$ to I_1 if $\sigma x = true$ and $s(F,X)$ otherwise. Then we have to add to I_1 also all $s(T,B_x)$. As $\alpha(\sigma\overline{x},\overline{y})$ is false for all $\overline{y}$, $\varphi(\alpha, V_x, V_Y)$ cannot be true. Hence we add $s(T,D)$ and therefore, also $s(T,B)$ to I_1.

If β is false, then for every combination of $s(T,X)$ and $s(F,X)$ there is a substitution such that $\varphi(\alpha, V_X, V_Y)$ is true. Hence, we should delete $s(T,D)$ from I_1. If we don't delete $s(T,D)$, then we must delete some other fact. But we can delete only literals of the form $s(z,X)$ where $z \in \{T,F\}$. So we remove $s(T,X)$ and $s(F,X)$ for some x, and we cannot obtain $s(T,B_x)$ for this X. In any case, $s(T,B)$ cannot be proven, and does not belong to I_1.

(2) *Lower bound.* Let us consider a sentence $\beta = \forall\overline{x}\exists\overline{y}\alpha(\overline{x},\overline{y})$, where α is a 3-CNF, and V_X and V_Y are the sets of all variables x and y respectively. Let a and b be some new variables. We construct $C = C(\alpha, V_X \cup \{a,b\})$, $I = J(\alpha,\emptyset)$, and IC Φ with clauses:
$s(T,A) \leftarrow s(F,A), s(T,X), s(F,X)$, for $x \in V_X$;
$s(T,B) \leftarrow \{s(F,A)\} \cup \varphi(\alpha, V_X, V_Y)$;
$s(x,y) \leftarrow$ for all (x,y) such that $s(x,y) \in J(\alpha,\emptyset)$.
We set $\Delta = (\{s(F,A)\}, \{s(T,A)\})$ and $l = s(T,B)$. Then $(I,\Delta,\Phi,l) \in$ **PFIP** iff β is true. □

Interestingly enough, the general case in partial interpretations is polynomially reduced to that of definite IC.

Corollary 3 *In general case:*
(1) **OFIP** *is* Σ_2^p*-complete;*
(2) **PFIP** *is* Π_2^p*-complete.*

Complexity in total interpretations. As it was shown in [12], in the case of ground IC the complexity of **OFIP** and **PFIP** is greater for total interpretations than that in partial ones. In fact, **OFIP** and **PFIP** are "co-problems" in total interpretations in the sense that $(I,\Delta,\Phi,l) \in$ **PFIP** iff $(I,\Delta,\Phi,\neg l) \notin$ **OFIP** . So it is enough to establish complexity bounds for one of the problems, e.g. for **OFIP**, if no constraints are imposed on literal l.

Theorem 6 *[12] In the case, where IC are ground:*
(1) both **OFIP** *and* **PFIP** *belong to P, when IC are definite and updates are positive;*
(2) **OFIP** *is NP-complete, when IC are definite;*
(3) in the general case **OFIP** *is Σ_2^p-complete.*

Corollary 4 *In the case, where IC are ground:*
(1) **PFIP** *is co-NP-complete, when IC are definite;*
(2) **PFIP** *is Π_2^p-complete in the general case.*
(3) **OFIP** *(***PFIP***) is Σ_2^p-complete (Π_2^p-complete) when IC are monadic programs with variables.*

As in the case of partial interpretations, for definite IC and positive updates[1] the problems are equivalent and solvable in polynomial time, when IC are monadic, and are hard otherwise.

Proposition 5 *In the case, where IC are definite:*
(1) **OFIP** *and* **PFIP** *are in P, if IC use only unary predicates, and updates are positive ;*
(2) **OFIP** *and* **PFIP** *are NP-complete (co-NP-complete), if updates are positive and l is positive (negative);*
(3) **OFIP** *is Σ_2^p-complete in general.*

Corollary 5 **PFIP** *is Π_2^p-complete, when IC are definite.*

In the case of general IC **OFIP** and **PFIP** become complete on the third level of polynomial Σ and Π hierarchies.

Theorem 7
(1) **OFIP** *(***PFIP***) is Σ_3^p-complete (Π_3^p-complete);*
(2) the lower bounds are valid even for a signature consisting of one binary predicate.

Proof scheme: *Lower bound.* Let $\beta = \exists\overline{x}\forall\overline{y}\exists\overline{z}\alpha(\overline{x},\overline{y},\overline{z})$, where α is a 3-CNF. Let V_X, V_Y and V_Z be the sets of all variables x, y, and z respectively. We construct $C = C(V_X \cup V_Y \cup \{a,b\})$, $I = J(\alpha, V_X)$, and IC Φ with clauses:
$\neg s(T,A) \leftarrow s(T,X), s(F,X)$, for $x \in V_X$;
$\neg s(T,A) \leftarrow \neg s(T,X), \neg s(F,X)$, for $x \in V_X$;
$s(T,B) \leftarrow s(T,A), s(T,Y), s(F,Y)$, for $y \in V_Y$;
$s(T,B) \leftarrow s(T,A), \neg s(T,Y), \neg s(F,Y)$, for $y \in V_Y$;
$s(T,Y) \leftarrow s(T,B)$, and $s(F,Y) \leftarrow s(T,B)$, for $y \in V_Y$;
$s(T,B) \leftarrow \{s(T,A)\} \cup \varphi(\alpha, V_X \cup V_Y, V_Z)$.
We set $\Delta = (\{s(T,A)\}, \emptyset)$, and $l = s(T,B)$.
Then in every I_1 minimally deviating from $Acc(\Phi,\Delta)$ there is exactly one of the facts $s(T,X)$, $s(F,X)$ for each $x \in V_X$.

Let β be true. Then there is a substitution σ such that $\forall\overline{y}\exists\overline{z}\alpha(\sigma\overline{x},\overline{y},\overline{z})$ is true. Let

[1] in total interpretations this means $\Delta^- = \emptyset$.

us include $s(T,X)$ in I_1 if $\sigma x = true$ and $s(F,X)$ otherwise. Then we also must add at least one of $s(T,Y)$ and $s(F,Y)$ for all $y \in V_Y$. If for some $y \in V_Y$ both $s(T,Y)$ and $s(F,Y)$ are present, then $s(T,B)$ and all $s(T,Y)$, $s(F,Y)$ must be present too. Let us denote this DB state by I_1. In order to obtain some I' which is closer to I than I_1 in our deviation order, we should add to I' exactly one of $s(T,Y)$, $s(F,Y)$ for each $y \in V_Y$, and we should not add $s(T,B)$. This, however, is impossible because $\forall \overline{y} \exists \overline{z} \alpha(\sigma \overline{x}, \overline{y}, \overline{z})$ is true.

Let $l \in I_1$ for some I_1 minimally deviating from $Acc(\Phi, \Delta)$. Then I_1 contains $s(T,Y)$ and $s(F,Y)$ for all $y \in V_Y$. Hence, it is impossible to select exactly one of $s(T,Y)$ and $s(F,Y)$ in order that $\varphi(\alpha, V_X \cup V_Y, V_Z)$ would be false for all substitutions. Therefore, the formula $\forall \overline{y} \exists \overline{z} \alpha(\sigma \overline{x}, \overline{y}, \overline{z})$ is true for the substitution $\sigma x = true$ if $s(T,X) \in I_1$, and $\sigma x = false$ otherwise. So β is true.□

4.2 The case of varying signature

It is no wonder that without the premise of fixed signature the complexity grows exponentially.

Theorem 8 *When the signature varies, both problems* **OFIP** *and* **PFIP** *are* **EXPTIME**-*complete for the class of definite IC in partial and in total interpretations.*

Proof scheme: *Lower bound.* Let $x \in X$ be a problem in *EXPTIME*, and $M = (Q, \Sigma, P, q_0)$ be a Turing machine resolving X in exponential time. Let q_a be the accepting state of M. We presume that M makes on every input x exactly $2^{p(|x|)}$ steps for some polynomial p.

We reduce the problem "$x \in X?$" to **OFIP** as follows. Let us fix some x and set $N = p(|x|)$. We will write $\overline{x}$ instead of $x_1, \ldots, x_N$. The signature **S** will consist of predicate symbols $s^{(2N)}$, $Q^{(N+1)}$, $H^{(2N)}$, $L^{(2N+1)}$, $A^{(0)}$ and $B^{(0)}$ (we drop predicate arities in the sequel). We will use the set of constants $\{1, 0\} \cup \{q \in Q\} \cup \{a \in \Sigma\}$.

The IC Φ will have the following groups of clauses:

$$s(x_1, \ldots, x_i, 0, \underbrace{1, \ldots, 1}_{n-i-1}, x_1, \ldots, x_i, 1, \underbrace{0, \ldots, 0}_{n-i-1}) \leftarrow a$$

for all $i = 1, \ldots, n$;

for each instruction $aq \rightarrow bps \in P$, the corresponding group of clauses. For instance, for the shift $s = L$, the group:

$$Q(p, \overline{t}_2)) \leftarrow s(\overline{x}_1, \overline{x}_2), s(\overline{t}_1, \overline{t}_2), Q(q, \overline{t}_1), H(\overline{t}_1, \overline{x}_2), L(a, \overline{t}_1, \overline{x}_2), A$$
$$H(\overline{t}_2, \overline{x}_1) \leftarrow s(\overline{x}_1, \overline{x}_2), s(\overline{t}_1, \overline{t}_2), Q(q, \overline{t}_1), H(\overline{t}_1, \overline{x}_2), L(a, \overline{t}_1, \overline{x}_2), A$$
$$L(b, \overline{t}_2, \overline{x}_2) \leftarrow s(\overline{x}_1, \overline{x}_2), s(\overline{t}_1, \overline{t}_2), Q(q, \overline{t}_1), H(\overline{t}_1, \overline{x}_2), L(a, \overline{t}_1, \overline{x}_2), A$$
$$L(x_i, \overline{0}, \overline{i}) \leftarrow A,$$

where x_i is the i-th symbol of x, $\overline{0}$ is the N-tuple of 0, $\overline{i}$ is the N-tuple corresponding to the binary representation of i, where $i = 1, \ldots, 2^{|x|} - 1$.

$$L(\Lambda, \overline{0}, x_1, \ldots, x_{i-1}, 1, x_{i+1}, \ldots, x_N) \leftarrow \qquad (i < N - |x|),$$
$$H(\overline{0}, \overline{0}) \leftarrow A, \qquad Q(q_0, \overline{0}) \leftarrow A, \qquad B \leftarrow Q(q_a, \overline{x}).$$

We set $I = \emptyset$, $\Delta = (\{A\}, \emptyset)$, and $l = B$. Clearly, there is the unique $I_1 \in Acc(\Phi, \Delta)$

minimally deviating from I. This I_1 is the closure of $\{A\}$ with respect to Φ. It is easy to see that $B \in I_1$ iff M reaches q_a, i.e. $x \in X$. □

From this theorem and the proof of Corollary3 follows the general case complexity bound for partial interpretations.

Corollary 6 *When the signature varies, both problems* **OFIP** *and* **PFIP** *are* **EXPTIME**-*complete in partial interpretations.*

Finally, we demonstrate the following general case complexity bound for total interpretations.

Theorem 9 *When the signature varies,* **OFIP** *is* $\Sigma_2^{EXPTIME}$*-complete and* **PFIP** *is* $\Pi_2^{EXPTIME}$*-complete in total interpretations.*

Proof scheme: *Lower bound.* Let $x \in X$ be a problem in $\Sigma_2^{EXPTIME}$, and $M = (Q, \Sigma, P, q_0)$ be an alternating Turing machine resolving X in exponential time with single alternation from an existential state to a universal one. Let $Q = E \cup U$, E and U being the sets of existential and universal states respectively. We presume that P does not have instructions of the form $qa \to pbs$ for $q \in U$, $p \in E$. We also assume that for any pair $qa, q \in Q, a \in \Sigma$, there exist at most two instructions $qa \to p_1 b_1 s_1$, $qa \to p_2 b_2 s_2$ in P, and that their right sides can differ only by states, i.e. $b_1 = b_2$ and $s_1 = s_2$. Let q_a be the accepting state of M. We presume that M makes on every input x exactly $2^{p(|x|)}$ steps for some polynomial p.

We reduce the problem "$x \in X?$" to **OFIP** as follows. Let us fix some x and set $N = p(|x|)$. Signature **S** has the predicate symbols $s^{(2N)}$, $Q^{(N+1)}$, $H^{(2N)}$, $L^{(2N+1)}$, $A^{(0)}$ and $B^{(0)}$. We will use the set of constants $\{1, 0\} \cup \{q \in Q\} \cup \{a \in \Sigma\}$.

The IC Φ will have the following groups of clauses:

$$s(x_1, \ldots, x_i, 0, \underbrace{1, \ldots, 1}_{n-i-1}, x_1, \ldots, x_i, 1, \underbrace{0, \ldots, 0}_{n-i-1}) \leftarrow A$$

for all $i = 1, \ldots, n$;

for each pair of instructions $aq \to bp_1 s \in P$ and $aq \to bp_2 s \in P$ the corresponding group of clauses. For instance, for the shift $s = L$, the group:

$$Q(p_1, \bar{t}_2) \leftarrow s(\bar{x}_1, \bar{x}_2), s(\bar{t}_1, \bar{t}_2), Q(q, \bar{t}_1), H(\bar{t}_1, \bar{x}_2), L(a, \bar{t}_1, \bar{x}_2), \neg Q(p_2, \bar{t}_2), A$$
$$H(\bar{t}_2, \bar{x}_1) \leftarrow s(\bar{x}_1, \bar{x}_2), s(\bar{t}_1, \bar{t}_2), Q(q, \bar{t}_1), H(\bar{t}_1, \bar{x}_2), L(a, \bar{t}_1, \bar{x}_2), A$$
$$L(b, \bar{t}_2, \bar{x}_2) \leftarrow s(\bar{x}_1, \bar{x}_2), s(\bar{t}_1, \bar{t}_2), Q(q, \bar{t}_1), H(\bar{t}_1, \bar{x}_2), L(a, \bar{t}_1, \bar{x}_2), A$$
$$Q(q, \bar{t}) \leftarrow Q(p, \bar{t}), B \textit{ for any } p, q \in U$$
$$L(x_i, \bar{0}, \bar{i}) \leftarrow A,$$

where x_i is the i-th symbol of x, $\bar{0}$ is the N-tuple of 0, $\bar{i}$ is the N-tuple corresponding to the binary representation of i, $i = 1, \ldots, 2^{|x|} - 1$.

$$L(\Lambda, \bar{0}, x_1, \ldots, x_{i-1}, 1, x_{i+1}, \ldots, x_N) \leftarrow,$$

where $i < N - |x|$, and

$$H(\bar{0}, \bar{0}) \leftarrow A, \qquad Q(q_0, \bar{0}) \leftarrow A, \qquad B \leftarrow Q(q_a, \bar{x}).$$

Finally, we set $I = \emptyset$, $\Delta = (\{A\}, \emptyset)$, and $l = B$.

One can prove that M accepts x iff there exists $I_1 \in Acc(\Phi, \Delta)$ minimally deviating from I which contains B. □

We summarize our main results in the following tables.

Complexity of **OFIP** with fixed signature

	Partial		Total	
	ground	Non-ground	ground	Non-ground
Positive case	P	$NP/co\text{-}NP$	P	$NP/co\text{-}NP$
Definite IC	NP	Σ_2^P	NP	Σ_2^P
General case	NP	Σ_2^P	Σ_2^P	Σ_3^P

Complexity of **PFIP** with fixed signature

	Partial		Total	
	ground	Non-ground	ground	Non-ground
Positive case	P	$NP/co\text{-}NP$	P	$NP/co\text{-}NP$
Definite IC	$co\text{-}NP$	Π_2^P	$co\text{-}NP$	Π_2^P
General case	$co\text{-}NP$	Π_2^P	Π_2^P	Π_3^P

Complexity of **OFIP** and **PFIP** with varying signature

	OFIP		**PFIP**	
	partial	total	partial	total
Definite IC	$EXPTIME$	$EXPTIME$	$EXPTIME$	$EXPTIME$
General case	$EXPTIME$	$\Sigma_2^{EXPTIME}$	$EXPTIME$	$\Pi_2^{EXPTIME}$

5 Conclusion

Our analysis shows that in the worst case the "property" aspects of the EUP are very hard even under the fixed signature premise. Nevertheless, the case of ground IC in partial interpretations presents a rare and surprising exclusion, where the EUP problem itself receives a practical polynomial time solution. As it concerns the EUP-solutions marked by presence or absence of a given fact, in quite a practical situation of definite IC they can be found in polynomial time in the absence of negation/deletions in updates. In the general situation some special means should be used in order to optimize complete choice solutions of the EUP. Some methods of this kind are proposed in [10–13].

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. Abiteboul, S., Vianu, V.: Datalog Extensions for Database Queries and Updates. *JCSS,* **43**, n. 1 (1991) 62–124.
3. Alchourón, C.E., Gärdenfors, P., and Makinson, D.: On the logic of theory change: partial meet contraction and revision functions. *J. Symbolic Logic.* **50** (1985) 510-530.
4. 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.
5. Baral, C., Lobo, J.: Formal Characterization of Active Databases. In Pedreschi, D., Zaniolo, C., editors: *Logic in Databases. Intern. Workshop LID'96. Proceedings.* San Milano, Italy. (1996), 175-195.

6. Baralis E., Ceri S., Paraboschi S.: Improved rule analysis by means of triggering and activation graphs. In T.Sellis, editor: *Rules in Database Systems,* LNCS **985** (1995) 165-181.
7. Chandra A., Merlin P. : Optimal implementation of conjunctive queries in relational databases. In: *Ninth ACM Symp. on Theory of Computing.* (1977) 77-90.
8. Dayal, U., Hanson,E., and Widom, J.: Active database systems. In: W. Kim, editor, *Modern Database Systems.* Addison Wesley (1995) 436-456
9. 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.
10. 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.
11. 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.
12. Dekhtyar, M., Dikovsky, A., Dudakov, S., Spyratos, N.: Monotone Expansion of Updates in Logical Databases. In: *Proc. of 5th International Conference LPNMR'99.* LNAI **1730** (1999) 132-147.
13. Dekhtyar, M., Dikovsky, A., Dudakov, S., Spyratos, N.: Maximal Expansions of Database Updates. *To be published in the Proc. of FoIKS'2000.*
14. Eiter, T., Gottlob, G.: On the complexity of propositional knowledge base revision, updates, and counterfactuals. Artificial Intelligence **57** (1992) 227-270.
15. Gottlob, G., Moerkotte, G., Subrahmanian, V.S.: The PARK semantics for active databases. In: *Proceedings of EDBT'96.* Avignon, France (1996).
16. Guessoum A., Lloyd J.W.: Updating knowledge bases. New Generation Computing, **8** (1990), 71-89.
17. 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).
18. Kakas A.C., Mancarella P.: Database updates through abduction. IN: *Proc. 16th VLBD Conference.* (1990) 650-661.
19. Katsuno, H., Mendelzon, A. O.: Propositional knowledge base revision and minimal change. Artificial Intelligence **52** (1991) 253-294.
20. Lloyd, J.W., Foundations of Logic Programming. Second, Extended Edition. Springer-Verlag. (1993)
21. Marek, V.W., Truszczyński, M.: Revision programming, database updates and integrity constraints. In: *International Conference on Data Base theory, ICDT.* LNCS **893** (1995) 368-382.
22. Picouet, Ph., Vianu, V.: Expressiveness and Complexity of Active Databases. In: Afrati, F., Kolaitis, Ph., editors, *6th Int. Conf. on Database Theory, ICDT'97.* LNCS **1186** (1997) 155-172.
23. 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.
24. Raschid, L., Lobo, J.: Semantics for Update Rule Programs and Implementation in a Relational Database Management System. ACM Trans. on Database Systems **21** (December 1996) 526-571

25. Schewe, K.-D., Thalheim, B.: Consistency Enforcement in Active Databases. In: Chakravarty, S., Widom, L., editors: *Research Issues in Data Engineering - Active Databases*. Proceedings. Houston, (1994).
26. Winslet M.: Reasoning about action using a possible models approach. In:*Proc. AAAI'88, v. 1* (1988) 89-93.
27. Zaniolo, C.: Active database rules with transaction-conscious stable-model semantics. In:*Proceedings of Fourth International Conference, DOOD'95*. LNCS **1013** (1995) 55-72