Confidentiality-Preserving Publishing of EDPs for Credulous and Skeptical Users

Katsumi Inoue[1], Chiaki Sakama[2] and Lena Wiese[3]*

[1] National Institute of Informatics
2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo 101-8430, Japan
ki@nii.ac.jp
[2] Department of Computer and Communication Sciences, Wakayama University
930 Sakaedani, Wakayama 640-8510, Japan
sakama@sys.wakayama-u.ac.jp
[3] Institute of Computer Science, University of Hildesheim
Samelsonplatz 1, 31141 Hildesheim, Germany
lena.wiese@uni-hildesheim.de

Abstract. Publishing private data on external servers incurs the problem of how to avoid unwanted disclosure of confidential data. We study the problem of confidentiality-preservation when publishing extended disjunctive logic programs and show how it can be solved by extended abduction. In particular, we analyze how the differences between users who employ either credulous or skeptical non-monotonic reasoning affect confidentiality.

Keywords: Data publishing, confidentiality, privacy, extended abduction, answer set programming, negation as failure, non-monotonic reasoning

1 Introduction

Confidentiality of data (also called privacy or secrecy in some contexts) is a major security goal. Releasing data to a querying user without disclosing confidential information has long been investigated in areas like access control, k-anonymity, inference control, and data fragmentation. Such approaches prevent disclosure according to some security policy by restricting data access (denial, refusal), by modifying some data (perturbation, noise addition, cover stories, lying, weakening), or by breaking sensitive associations (fragmentation). Several approaches (like [3, 8, 14, 15, 2, 16]) employ logic-based mechanisms to ensure data confidentiality. In particular, [5] uses brave reasoning in default logic theories to solve a privacy problem in a classical database (a set of ground facts). For a non-classical knowledge base (where negation as failure *not* is allowed) [17] studies correctness of access rights. Confidentiality of predicates in collaborative multi-agent abduction is a topic in [11].

* Lena Wiese was partially supported by a postdoctoral research grant of the German Academic Exchange Service (DAAD) while preparing this work.

In this article we analyze **confidentiality-preserving data publishing** in a knowledge base setting: data as well as integrity constraints or deduction rules are represented as logical formulas. If such a knowledge base is released to the public for general querying (e.g., microcensus data) or outsourced to a storage provider (e.g., database-as-a-service in cloud computing), confidential data could be disclosed. This article is a revised and extended version of [10]; in particular, we extend [10] to also cover confidentiality-preserving data publishing for users who deduce information by *skeptical* non-monotonic reasoning. This article is one of only few papers (see [12, 17, 11]) covering confidentiality for logic programs. This formalism however has relevance in multi-agent communications where agent knowledge is modeled by logic programs. In our settings (as already in [10]), knowledge bases come in the form of extended disjunctive logic programs (EDPs) as defined below. Hence, with this formalism we achieve high expressiveness by allowing negation as failure *not* as well as disjunctions in rule heads. In this article, we assume that users accessing the published knowledge base use either credulous or skeptical reasoning to retrieve data from it; users also possess some invariant "a priori knowledge" that can be applied to these data to deduce further information – again by using either credulous or skeptical reasoning. On the knowledge base side, a confidentiality policy specifies which is the confidential information that must never be disclosed.

With **extended abduction** [13] we obtain a "secure version" of the knowledge base that can safely be published even when a priori knowledge is applied. In this article, we show how confidentiality-preservation for skeptical users differs from the one for credulous users. More precisely, while computing the secure version for a *credulous* user corresponds to finding a *skeptical* anti-explanation for all the elements of the confidentiality policy, computing the secure version for a *skeptical* user corresponds to finding a *credulous* anti-explanation for the elements of the confidentiality policy followed by an additional consistency check. Extended abduction has been used in different applications like for example providing a logical framework for dishonest reasoning [12]. It can be solved by computing the answer sets of an update program (see [13]); thus an implementation of extended abduction can profit from current answer set programming (ASP) solvers [4]. To retrieve the confidentiality-preserving knowledge base K^{pub} from the input knowledge base K, the a priori knowledge *prior* and the confidentiality policy *policy*, a sequence of transformations are applied; the overall approach is depicted in Figure 1.

In summary, this paper makes the following contributions:

- it formalizes confidentiality-preserving data publishing for users who retrieve data under either a credulous or a skeptical query response semantics.
- it devises a procedure to securely publish a logic program (with an expressiveness up to extended disjunctive logic programs) respecting a subset-minimal change semantics.
- it shows that confidentiality-preservation for credulous as well as skeptical users corresponds to finding anti-explanations and can be solved by extended abduction.

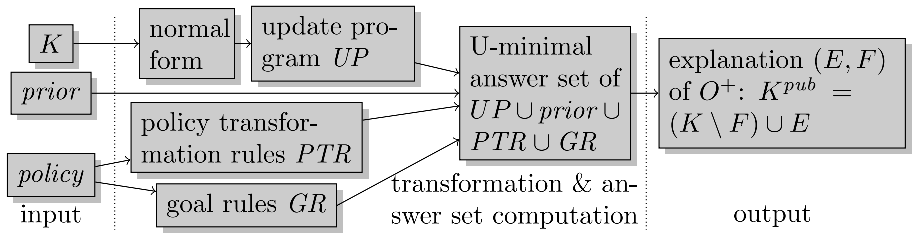
K
$prior$
$policy$
input
normal form
update program UP
policy transformation rules PTR
goal rules GR
U-minimal answer set of $UP \cup prior \cup PTR \cup GR$
transformation & answer set computation
explanation (E, F) of O^+: $K^{pub} = (K \setminus F) \cup E$
output

Fig. 1. Finding a confidentiality-preserving K^{pub}

In the remainder of this article, Section 2 provides background on extended disjunctive logic programs and answer set semantics; Section 3 defines the problem of confidentiality in data publishing; Section 4 recalls extended abduction and update programs; Section 5 shows how answer sets of update programs correspond to confidentiality-preserving knowledge bases; and Section 6 gives some discussion and concluding remarks.

2 EDPs and answer set semantics

In this article, a knowledge base K is represented by an *extended disjunctive logic program* (EDP) – a set of formulas called *rules* of the form:

$$L_1; \ldots; L_l \leftarrow L_{l+1}, \ldots, L_m, not L_{m+1}, \ldots, not L_n. \qquad (n \geq m \geq l \geq 0)$$

A rule contains literals L_i, disjunction ";", conjunction ",", negation as failure "*not*", and implication "$\leftarrow$". A literal is a first-order atom or an atom preceded by classical negation "$\neg$". $not L$ is called a *NAF-literal*. The disjunction left of the implication $\leftarrow$ is called the *head*, while the conjunction right of $\leftarrow$ is called the *body* of the rule. For a rule R, we write $head(R)$ to denote the set of literals $\{L_1, \ldots, L_l\}$ and $body(R)$ to denote the set of (NAF-)literals $\{L_{l+1}, \ldots, L_m, not L_{m+1}, \ldots, not L_n\}$. Rules consisting only of a singleton head $L \leftarrow$ are identified with the literal L and used interchangeably. An EDP is ground if it contains no variables. If an EDP contains variables, it is identified with the set of its ground instantiations: the elements of its Herbrand universe are substituted in for the variables in all possible ways. We assume that the language contains no function symbols, so that each rule with variables represents a finite set of ground rules. For a program K, we denote $\mathscr{L}_K$ the set of ground literals in the language of K. Note that EDPs offer a high expressiveness including disjunctive and non-monotonic reasoning.

Example 1. In a medical knowledge base $Ill(x, y)$ states that a patient x is ill with disease y; $Treat(x, y)$ states that x is treated with medicine y. Assume that if you read the record and find that one treatment (Medi1) is recorded and another one (Medi2) is not recorded, then you know that the patient is at least

ill with Aids or Flu (and possibly has other illnesses).
$K = \{Ill(x, \mathsf{Aids}); Ill(x, \mathsf{Flu}) \leftarrow Treat(x, \mathsf{Medi1}), not\, Treat(x, \mathsf{Medi2}).\ ,$
$Ill(\mathsf{Mary}, \mathsf{Aids}).\ ,\ \ Treat(\mathsf{Pete}, \mathsf{Medi1}).\}$ serves as a running example.

The semantics of K can be given by the answer set semantics [7]: A set $S \subseteq \mathscr{L}_K$ of ground literals *satisfies* a ground literal L if $L \in S$; S satisfies a conjunction if it satisfies every conjunct; S satisfies a disjunction if it satisfies at least one disjunct; S satisfies a ground rule if whenever the body literals are contained in S ($\{L_{l+1}, \ldots, L_m\} \subseteq S$) and all NAF-literals are not contained in S ($\{L_{m+1}, \ldots, L_n\} \cap S = \emptyset$), then at least one head literal is contained in S ($L_i \in S$ for an i such that $1 \leq i \leq l$). If an EDP K contains no NAF-literals ($m = n$), then such a set S is an *answer set* of K if S is a subset-minimal set such that

1. S satisfies every rule from the ground instantiation of K.
2. If S contains a pair of complementary literals L and $\neg L$, then $S = \mathscr{L}_K$.

This definition of an answer set can be extended to full EDPs (containing NAF-literals) as in [13]: For an EDP K and a set of ground literals $S \subseteq \mathscr{L}_K$, K can be transformed into a NAF-free program K^S as follows. For every ground rule from the ground instantiation of K (with respect to its Herbrand universe), the rule $L_1; \ldots; L_l \leftarrow L_{l+1}, \ldots, L_m$ is in K^S if $\{L_{m+1}, \ldots, L_n\} \cap S = \emptyset$. Then, S is an answer set of K if S is an answer set of K^S. An answer set is *consistent* if it is not $\mathscr{L}_K$. A program K is *consistent* if it has a consistent answer set; otherwise K is *inconsistent*.

Example 2. The example K has the following two consistent answer sets

$$S_1 = \{Ill(\mathsf{Mary}, \mathsf{Aids}),\ Treat(\mathsf{Pete}, \mathsf{Medi1}),\ Ill(\mathsf{Pete}, \mathsf{Aids})\},$$
$$S_2 = \{Ill(\mathsf{Mary}, \mathsf{Aids}),\ Treat(\mathsf{Pete}, \mathsf{Medi1}),\ Ill(\mathsf{Pete}, \mathsf{Flu})\}.$$

When adding the negative fact $\neg Ill(\mathsf{Pete}, \mathsf{Flu})$ to K, there is just one consistent answer set left: for $K' := K \cup \{\neg Ill(\mathsf{Pete}, \mathsf{Flu}).\}$ the only answer set is

$$S' = \{Ill(\mathsf{Mary}, \mathsf{Aids}), \neg Ill(\mathsf{Pete}, \mathsf{Flu}),\ Treat(\mathsf{Pete}, \mathsf{Medi1}),\ Ill(\mathsf{Pete}, \mathsf{Aids})\}.$$

If a rule R is satisfied in *every* answer set of K, we write $K \models R$. In particular, $K \models L$ if a literal L is included in every answer set of K.

3 Confidentiality-Preserving Knowledge Bases

When publishing a knowledge base K^{pub} while preserving confidentiality of some data in the original knowledge base K we do this according to

– the query response semantics that a user querying K^{pub} applies
– a confidentiality policy (denoted *policy*) describing confidential information that should not be released to the public
– background (a priori) knowledge (denoted *prior*) that a user can combine with query responses from the published knowledge base

First we define the credulous and the skeptical query response semantics: in the credulous case, a ground formula Q is *true* in K, if Q is satisfied in *some* answer set of K – that is, there might be answer sets that do not satisfy Q; in the skeptical case, a ground formula Q is *true* in K, if Q is satisfied in *every* answer set of K. If a rule Q is non-ground and contains some free variables, the response of K is the set of ground instantiations of Q that are *true* in K under either the credulous or skeptical semantics.

Definition 1 (Credulous and skeptical query response semantics). *Let U be the Herbrand universe of a consistent knowledge base K. The* credulous query responses *of formula $Q(X)$ (with a vector X of free variables) in K are*

$$\begin{aligned}cred(K, Q(X)) = \{Q(A) \mid\ & A \textit{ is a vector of elements of } U \textit{ and there} \\ & \textit{is an answer set of } K \textit{ that satisfies } Q(A)\}\end{aligned}$$

In particular, for a ground formula Q,

$$cred(K, Q) = \begin{cases} \{Q\} & \textit{if } K \textit{ has an answer set that satisfies } Q \\ \emptyset & \textit{otherwise} \end{cases}$$

The skeptical query responses *of $Q(X)$ in K are*

$$skep(K, Q(X)) = \{Q(A) \mid A \textit{ is a vector of elements of } U \textit{ and } K \models Q(A)\}$$

In particular, for a ground formula Q, $skep(K, Q) = \begin{cases} \{Q\} \textit{ if } K \models Q \\ \emptyset \textit{ otherwise} \end{cases}$

Example 3. Assume that the example K is queried for all patients x suffering from aids. Then, $cred(K, Ill(x, \mathsf{Aids})) = \{Ill(\mathsf{Mary}, \mathsf{Aids}), Ill(\mathsf{Pete}, \mathsf{Aids})\}$ and $skep(K, Ill(x, \mathsf{Aids})) = \{Ill(\mathsf{Mary}, \mathsf{Aids})\}$.

It is usually assumed that in addition to the query responses a user has some additional knowledge that he can apply to the query responses. Hence, we additionally assume given a set of rules as some *invariant* **a priori knowledge** *prior*; invariance is a common assumption (see [6]). We assume that *prior* is a consistent EDP. Thus, the a priori knowledge may consist of additional facts that the user assumes to hold in K, or some rules that the user can apply to data in K to deduce new information.

A **confidentiality policy** *policy* specifies confidential information. We assume that *policy* contains conjunctions of literals or NAF-literals. We do not only have to avoid that the published knowledge base contains confidential information but also prevent the user from deducing confidential information with the help of his a priori knowledge; this is known as the inference problem [6, 2].

Example 4. If we wish to declare the disease aids as confidential for any patient x we can do this with $policy = \{Ill(x, \mathsf{Aids}).\}$. A user querying K^{pub} might know that a person suffering from flu is not able to work. Hence $prior = \{\neg AbleToWork(x) \leftarrow Ill(x, \mathsf{Flu}).\}$. If we wish to also declare a lack of work ability as confidential, we can add this to the confidentiality policy: $policy' = \{Ill(x, \mathsf{Aids}).\ ,\ \neg AbleToWork(x).\}$.

Next, we establish a definition of confidentiality-preservation that allows for the answer set semantics as an inference mechanism and respects the credulous or skeptical query response semantics: when treating elements of the confidentiality policy as queries, the credulous or skeptical responses must be empty.

Definition 2 (Confidentiality-preservation for credulous and skeptical users). *A knowledge base* K^{pub} preserves confidentiality *of a given confidentiality policy under the* credulous *query response semantics and with respect to a given a priori knowledge prior, if for every conjunction* $C(X)$ *in the policy, the credulous query responses of* $C(X)$ *in* $K^{pub} \cup prior$ *are empty:* $cred(K^{pub} \cup prior, C(X)) = \emptyset$*. It* preserves confidentiality *under the* skeptical *query response semantics, if the skeptical query responses of* $C(X)$ *in* $K^{pub} \cup prior$ *are empty:* $skep(K^{pub} \cup prior, C(X)) = \emptyset$*.*

Note that the Herbrand universe of $K^{pub} \cup prior$ is applied in the query response semantics; hence, free variables in policy elements $C(X)$ are instantiated according to this universe. Moreover, $K^{pub} \cup prior$ must be consistent.

A goal secondary to confidentiality-preservation is **minimal change**: We want to publish as many data as possible and want to modify these data as little as possible. Different notions of minimal change are used in the literature (see for example [1] for a collection of minimal change semantics in a data integration setting). We apply a subset-minimal change semantics: we choose a K^{pub} that differs from K only subset-minimally. In other words, there is no other confidentiality-preserving knowledge base $K^{pub'}$ which inserts (or deletes) less rules to (from) K than K^{pub}.

Definition 3 (Subset-minimal change). *A confidentiality-preserving knowledge base* K^{pub} subset-minimally changes K *(or is* minimal*, for short) if there is no confidentiality-preserving knowledge base* $K^{pub'}$ *such that* $((K \setminus K^{pub'}) \cup (K^{pub'} \setminus K)) \subset ((K \setminus K^{pub}) \cup (K^{pub} \setminus K))$*.*

Example 5. For the example K and *policy* and no a priori knowledge, the fact $\mathit{Ill}(\mathsf{Mary}, \mathsf{Aids})$ has to be deleted under both the credulous and the skeptical query response semantics. Moreover, $\mathit{Ill}(\mathsf{Pete}, \mathsf{Aids})$ can be deduced credulously, because it is satisfied by answer set S_1. In order to avoid this, we have two options when only deletions are used: delete $\mathit{Treat}(\mathsf{Pete}, \mathsf{Medi1})$, or delete the non-literal rule in K; if insertions of literals are allowed, we have three options: insert $\mathit{Treat}(\mathsf{Pete}, \mathsf{Medi2})$, or insert $\mathit{Ill}(\mathsf{Pete}, \mathsf{Flu})$, or insert $\neg \mathit{Ill}(\mathsf{Pete}, \mathsf{Aids})$. Each of these options blocks the credulous deduction of $\mathit{Ill}(\mathsf{Pete}, \mathsf{Aids})$. In contrast, for K, *policy*$'$ and *prior*, the last two options (insert $\mathit{Ill}(\mathsf{Pete}, \mathsf{Flu})$, or insert $\neg \mathit{Ill}(\mathsf{Pete}, \mathsf{Aids})$) are not possible, because then the secret $\neg \mathit{AbleToWork}(Pete)$ could be deduced credulously. The same two options are impossible for K' (defined in Section 2) and *policy* because $\neg \mathit{Ill}(\mathsf{Pete}, \mathsf{Flu})$ is contained in K'.

In the following sections we obtain a minimal solution K^{pub} for a given input K, *prior* and *policy* by transforming the input into a problem of *extended abduction* and solving it with an appropriate update program.

4 Extended Abduction

Traditionally, given a knowledge base K and an observation formula O, *abduction* finds a "(positive) explanation" E – a set of hypothesis formulas – such that every answer set of the knowledge base and the explanation together satisfy the observation; that is, $K \cup E \models O$. Going beyond that [9, 13] use *extended* abduction with the notions of "negative observations", "negative explanations" F and "anti-explanations". An abduction problem in general can be restricted by specifying a designated set $\mathcal{A}$ of *abducibles*. This set poses syntactical restrictions on the explanation sets E and F. In particular, positive explanations are characterized by $E \subseteq \mathcal{A} \setminus K$ and negative explanations by $F \subseteq K \cap \mathcal{A}$. If $\mathcal{A}$ contains a formula with variables, it is meant as a shorthand for all ground instantiations of the formula. In this sense, an EDP K accompanied by an EDP $\mathcal{A}$ is called an *abductive program* written as $\langle K, \mathcal{A} \rangle$. The aim of extended abduction is then to find (anti-)explanations as follows:

- given a *positive* observation O, find a pair (E, F) where E is a positive explanation and F is a negative explanation such that
  1. [**explanation**]
     (a) [**skeptical**] O is satisfied in *every* answer set of $(K \setminus F) \cup E$; that is, $(K \setminus F) \cup E \models O$
     (b) [**credulous**] O is satisfied in *some* answer set of $(K \setminus F) \cup E$
  2. [**consistency**] $(K \setminus F) \cup E$ is consistent
  3. [**abducibility**] $E \subseteq \mathcal{A} \setminus K$ and $F \subseteq \mathcal{A} \cap K$
- given a *negative* observation O, find a pair (E, F) where E is a positive anti-explanation and F is a negative anti-explanation such that
  1. [**anti-explanation**]
     (a) [**skeptical**] *no* answer set of $(K \setminus F) \cup E$ satisfies O
     (b) [**credulous**] there is *some* answer set of $(K \setminus F) \cup E$ that does not satisfy O; that is, $(K \setminus F) \cup E \not\models O$
  2. [**consistency**] $(K \setminus F) \cup E$ is consistent
  3. [**abducibility**] $E \subseteq \mathcal{A} \setminus K$ and $F \subseteq \mathcal{A} \cap K$

Among (anti-)explanations, **minimal** (anti-)explanations characterize a subset-minimal alteration of the program K: an (anti-)explanation (E, F) of an observation O is called minimal if for any (anti-)explanation (E', F') of O, $E' \subseteq E$ and $F' \subseteq F$ imply $E' = E$ and $F' = F$.

For an abductive program $\langle K, \mathcal{A} \rangle$ both K and $\mathcal{A}$ are semantically identified with their ground instantiations with respect to the Herbrand universe, so that set operations over them are defined on the ground instances. Thus, when (E, F) contain formulas with variables, $(K \setminus F) \cup E$ means deleting every instance of formulas in F, and inserting any instance of formulas in E from/into K. When E contains formulas with variables, the set inclusion $E' \subseteq E$ is defined for any set E' of instances of formulas in E. Generally, given sets S and T of literals/rules containing variables, any set operation $\circ$ is defined as $S \circ T = \mathit{inst}(S) \circ \mathit{inst}(T)$ where $\mathit{inst}(S)$ is the ground instantiation of S. For example, when $p(x) \in T$, for any constant a occurring in T, it holds that $\{p(a)\} \subseteq T$, $\{p(a)\} \setminus T = \emptyset$, and $T \setminus \{p(a)\} = (T \setminus \{p(x)\}) \cup \{p(y) \mid y \neq a\}$, etc. Moreover, any literal/rule in a set is identified with its variants modulo variable renaming.

4.1 Extended Abduction and Confidentiality-Preservation

Now, the formal correspondence between confidentiality-preservation and extended abduction can be stated as follows. A confidentiality-preserving knowledge base K^{pub} can be obtained by deleting elements from the knowledge base K and by inserting rules that are made up of predicate symbols and constants occuring in $K \cup \mathit{prior}$; however, as we assume *prior* to be invariant, we cannot delete rules contained in *prior*. This is summarized in the following theorem.

Theorem 1. *Given a knowledge base K, prior and policy, $K^{pub} = (K \setminus F) \cup E$ is a (minimal) solution of confidentiality-preservation for credulous (resp. skeptical) users iff (E, F) is a (minimal) skeptical (resp. credulous) anti-explanation for every $C_i \in \mathit{policy}$ in the abductive program $\langle K \cup \mathit{prior}\, , \mathcal{A}_{K\cup \mathit{prior}} \setminus \mathit{prior}\rangle$ where $\mathcal{A}_{K\cup \mathit{prior}}$ is the set of all ground rules constructed in the language of $K \cup \mathit{prior}$.*

Proof. By Definition 1 we have that $\mathit{cred}((K \setminus F) \cup E, C_i) = \emptyset$ iff *no* answer set of $(K \setminus F) \cup E$ satisfies C_i (that is, (E, F) is a skeptical anti-explanation). Respectively, $\mathit{skep}((K \setminus F) \cup E, C_i) = \emptyset$ iff $(K \setminus F) \cup E \not\models C_i$ (that is, (E, F) is a credulous anti-explanation).

4.2 Normal form

Although extended abduction can handle the very general format of EDPs, some syntactic transformations are helpful. Based on [13] we will briefly describe how a semantically equivalent normal form of an abductive program $\langle K, \mathcal{A}\rangle$ is obtained; in the end, we obtain an equivalent abductive program with only literals as abducibles (instead of general rules). This makes an automatic handling of abductive programs easier; for example, abductive programs in normal form can be easily transformed into update programs as described in Section 4.3. The main step is that rules in $\mathcal{A}$ can be mapped to atoms by a naming function n. Let $\mathcal{R}_{\mathcal{A}}$ be the set of abducible *rules*:

$$\mathcal{R}_{\mathcal{A}} = \{\Sigma \leftarrow \Gamma \mid (\Sigma \leftarrow \Gamma) \in \mathcal{A} \text{ and } (\Sigma \leftarrow \Gamma) \text{ is not a literal}\}$$

Then the *normal form* $\langle K^n, \mathcal{A}^n\rangle$ is defined as follows where $n(R)$ maps each rule R to a fresh atom with the same free variables as R:

$$K^n = (K \setminus \mathcal{R}_{\mathcal{A}}) \cup \{\Sigma \leftarrow \Gamma, n(R) \mid R = (\Sigma \leftarrow \Gamma) \in \mathcal{R}_{\mathcal{A}}\} \cup \{n(R) \mid R \in K \cap \mathcal{R}_{\mathcal{A}}\}$$
$$\mathcal{A}^n = (\mathcal{A} \setminus \mathcal{R}_{\mathcal{A}}) \cup \{n(R) \mid R \in \mathcal{R}_{\mathcal{A}}\}$$

We define that any abducible literal L has name L, i.e., $n(L) = L$. It is shown in [13], that there is a 1-1 correspondence between (anti-)explanations with respect to $\langle K, A\rangle$ and those with respect to $\langle K^n, A^n\rangle$ for any observation O. That is, for $n(E) = \{n(R) \mid R \in E\}$ and $n(F) = \{n(R) \mid R \in F\}$: an observation O has a minimal (anti-)explanation (E, F) with respect to $\langle K, A\rangle$ iff O has a minimal (anti-)explanation $(n(E), n(F))$ with respect to $\langle K^n, A^n\rangle$. Hence, insertion (deletion) of a rule's name in the normal form corresponds to insertion (deletion) of the rule in the original program. In sum, with the normal form transformation, any abductive program with abducible *rules* is reduced to an abductive program with only abducible *literals*.

Example 6. We transform the example knowledge base K into its normal form based on a set of abducibles that is identical to K: that is $\mathcal{A} = K$; a similar setting will be used in Section 5.2 to achieve deletion of formulas from K. Hence we transform $\langle K, \mathcal{A}\rangle$ into its normal form $\langle K^n, \mathcal{A}^n\rangle$ as follows where we write $n(R)$ for the naming atom of the only rule in $\mathcal{A}$:

$$\begin{aligned}
K^n = \{&\mathit{Ill}(\mathsf{Mary}, \mathsf{Aids}).,\qquad \mathit{Treat}(\mathsf{Pete}, \mathsf{Medi1}).,\qquad n(R).,\\
&\mathit{Ill}(x, \mathsf{Aids}); \mathit{Ill}(x, \mathsf{Flu}) \leftarrow \mathit{Treat}(x, \mathsf{Medi1}), \mathit{not}\,\mathit{Treat}(x, \mathsf{Medi2}), n(R).\}\\
\mathcal{A}^n = \{&\mathit{Ill}(\mathsf{Mary}, \mathsf{Aids}),\qquad \mathit{Treat}(\mathsf{Pete}, \mathsf{Medi1}),\qquad n(R)\ \}
\end{aligned}$$

4.3 Update programs

Minimal (anti-)explanations can be computed with *update programs* (UPs) [13]. The *update-minimal* (U-minimal) answer sets of a UP describe which rules have to be deleted from the program, and which rules have to be inserted into the program, in order to (un-)explain an observation.

For the given EDP K and a given set of abducibles $\mathcal{A}$, a set of **update rules** UR is devised that describe how entries of K can be changed. This is done with the following three types of rules.

1. [**Abducible rules**] The rules for abducible literals state that an abducible is either true in K or not. For each $L \in \mathcal{A}$, a new atom $\bar{L}$ is introduced that has the same variables as L. The set of abducible rules for each L is
$$abd(L) = \{L \leftarrow not\bar{L}.\ ,\ \bar{L} \leftarrow notL.\}.$$
2. [**Insertion rules**] Abducible literals that are not contained in K might be inserted into K and hence might occur in the set E of the explanation (E, F). For each $L \in \mathcal{A} \setminus K$, a new atom $+L$ is introduced and the insertion rule is
$$+L \leftarrow L.$$
3. [**Deletion rules**] Abducible literals that are contained in K might be deleted from K and hence might occur in the set F of the explanation (E, F). For each $L \in \mathcal{A} \cap K$, a new atom $-L$ is introduced and the deletion rule is
$$-L \leftarrow notL.$$

The **update program** is then defined by replacing abducible literals in K with the update rules; that is, $UP = (K \setminus \mathcal{A}) \cup UR$.

Example 7. Continuing Example 6, from $\langle K^n, \mathcal{A}^n\rangle$ we obtain

$$\begin{aligned} UP = {} & abd(\mathit{Ill}(\mathsf{Mary},\mathsf{Aids})) \cup abd(\mathit{Treat}(\mathsf{Pete},\mathsf{Medi1})) \cup abd(n(R)) \cup \\ & \{-\mathit{Ill}(\mathsf{Mary},\mathsf{Aids}) \leftarrow \mathit{not}\,\mathit{Ill}(\mathsf{Mary},\mathsf{Aids}).,\\ & -\mathit{Treat}(\mathsf{Pete},\mathsf{Medi1}) \leftarrow \mathit{not}\,\mathit{Treat}(\mathsf{Pete},\mathsf{Medi1}).,\\ & -n(R) \leftarrow \mathit{not}\, n(R).,\\ & \mathit{Ill}(x,\mathsf{Aids}); \mathit{Ill}(x,\mathsf{Flu}) \leftarrow \mathit{Treat}(x,\mathsf{Medi1}), \mathit{not}\,\mathit{Treat}(x,\mathsf{Medi2}), n(R).\} \end{aligned}$$

The set of atoms $+L$ is the set $\mathcal{UA}^+$ of positive update atoms; the set of atoms $-L$ is the set $\mathcal{UA}^-$ of negative update atoms. The set of **update atoms** is $\mathcal{UA} = \mathcal{UA}^+ \cup \mathcal{UA}^-$. From all answer sets of an update program UP we can identify those that are **update minimal** (U-minimal): they contain less update atoms than others. Thus, S is U-minimal iff there is no answer set T such that $T \cap \mathcal{UA} \subset S \cap \mathcal{UA}$.

4.4 Ground observations

It is shown in [9] how in some situations the observation formulas O can be mapped to new positive ground observations. Non-ground atoms with variables can be mapped to a new ground observation. Several positive observations can be conjoined and mapped to a new ground observation. A negative observation (for which an anti-explanation is sought) can be mapped as a NAF-literal to a new positive observation (for which then an explanation has to be found). Moreover, several negative observations can be mapped as a conjunction of NAF-literals to one new positive observation such that its resulting explanation acts as an anti-explanation for all negative observations together. Hence, in extended abduction it is usually assumed that O is a positive ground observation for which an explanation has to be found. In case of finding a skeptical explanation, an inconsistency check has to be made on the resulting knowledge base. Transformations to a ground observation and inconsistency check will be detailed in Section 5.1 and applied to confidentiality-preservation.

5 Confidentiality-Preservation with UPs

We now show how to achieve confidentiality-preservation by extended abduction: we define the set of abducibles and describe how a confidentiality-preserving knowledge base can be obtained by computing U-minimal answer sets of the appropriate update program. We additionally distinguish between the case that we allow only deletions of formulas – that is, in the anti-explanation (E, F) the set E of positive anti-explanation formulas is empty – and the case that we also allow insertions of literals.

5.1 Policy transformation for credulous and skeptical users

Elements of the confidentiality policy will be treated as negative observations for which an anti-explanation has to be found while adding *prior* as invariable knowledge. Accordingly, we will transform policy elements to a set of rules containing new positive observations as sketched in Section 4.4. As these rules are distinct for credulous and skeptical users, we call them **policy transformation rules for credulous users** (PTR^{cred}) and **policy transformation rules for skeptical users** (PTR^{skep}), respectively. In the credulous user case, we aim to find a *skeptical* anti-explanation that unexplains all the policy elements at the same time; in other words, no answer set of the resulting knowledge base K^{pub} satisfies any of the policy elements. More formally, assume *policy* contains k elements. For each conjunction $C_i \in \mathit{policy}$ $(i = 1 \dots k)$, we introduce a new negative ground observation O_i^- and map C_i to O_i^-. As each C_i is a conjunction of (NAF-)literals, the resulting formula is an EDP rule. In the credulous case, as a last policy transformation rule, we add one rule that maps all new negative ground observations O_i^- (in their NAF version) to a positive observation O^+:

$$PTR^{cred} = \{O_i^- \leftarrow C_i. \mid C_i \in \mathit{policy}\} \cup \{O^+ \leftarrow \mathit{not}\, O_1^-, \dots, \mathit{not}\, O_k^-.\}.$$

In the skeptical case, we have to treat every policy element individually; more precisely, for each single policy element, we have to find a *credulous* anti-explanation. In other words, for every policy element there must be at least one answer set of K^{pub} where it is not satisfied. For different policy elements these answer sets can however be different. For the credulous anti-explanation this has the consequence that each policy element has to be treated independent of others in the update program. That is why we obtain a set of rules PTR_i^{skep}: each policy element alone is mapped to the new positive observation O^+. Hence,

$$PTR_i^{skep} = \{O_i^- \leftarrow C_i. \mid C_i \in \mathit{policy}\} \cup \{O^+ \leftarrow \mathit{not}\, O_i^-.\}.$$

Example 8. The sets of policy transformation rules for policy' are

$$\begin{aligned} PTR^{cred} &= \{O_1^- \leftarrow \mathit{Ill}(x, \mathsf{Aids}).\ ,\ O_2^- \leftarrow \neg \mathit{AbleToWork}(x).\ , \\ &\qquad O^+ \leftarrow \mathit{not}\, O_1^-, \mathit{not}\, O_2^-.\} \\ PTR_1^{skep} &= \{\ O_1^- \leftarrow \mathit{Ill}(x, \mathsf{Aids}).,\ \ O^+ \leftarrow \mathit{not}\, O_1^-.\ \} \\ PTR_2^{skep} &= \{\ O_2^- \leftarrow \neg \mathit{AbleToWork}(x).,\ O^+ \leftarrow \mathit{not}\, O_2^-.\ \} \end{aligned}$$

Lastly, in both cases we consider an additional **goal rule** GR that enforces the single positive observation O^+: $GR = \{\leftarrow \mathit{not}\, O^+.\}$.

5.2 Deletions for credulous users

As a simplified setting, we first of all assume that in the credulous user case only deletions are allowed to achieve confidentiality-preservation. This setting can informally be described as follows: For a given knowledge base K, if we only allow deletions of rules from K, we have to find a *negative explanation* F that explains the new positive observation O^+ while respecting *prior* as invariable a priori knowledge. The set of abducibles is thus identical to K as we want to choose formulas from K for deletion: $\mathcal{A} = K$. That is, in total we consider the abductive program $\langle K, \mathcal{A} \rangle$. Then, we transform it into normal form $\langle K^n, \mathcal{A}^n \rangle$, and compute its update program UP as described in Section 4.3. As for *prior*, we add this set to the update program UP in order to make sure that the resulting answer sets of the update program do not contradict *prior*.

For the credulous user case, we finally add all the policy transformation rules PTR^{cred} and the goal rule GR. The goal rule is then meant as a constraint that only allows those answer sets of $UP \cup \mathit{prior} \cup PTR^{cred}$ in which O^+ is *true*. We thus obtain a new program P^{cred} as

$$P^{cred} = UP \cup \mathit{prior} \cup PTR^{cred} \cup GR$$

and compute its U-minimal answer sets. If S is one of these answer sets, the negative explanation F is obtained from the negative update atoms contained in S: $F = \{L \mid -L \in S\}$.

To obtain a confidentiality-preserving knowledge base for a credulous user, we have to check for inconsistency with the negation of the positive observation O^+ (which makes F a *skeptical* explanation of O^+); and allow only answer sets of P that are U-minimal among those respecting this inconsistency property. More precisely, we check whether

$$(K \setminus F) \cup \mathit{prior} \cup PTR^{cred} \cup \{\leftarrow O^+.\} \text{ is inconsistent.} \tag{1}$$

Example 9. We combine the update program UP of K with *prior* and the policy transformation rules PTR^{cred} and goal rule GR. This leads to the following U-minimal answer sets satisfying inconsistency property (1): $S_1' = \{-\mathit{Ill}(\mathsf{Mary}, \mathsf{Aids}), -\mathit{Treat}(\mathsf{Pete}, \mathsf{Medi1}), n(R), \overline{\mathit{Ill}(\mathsf{Mary}, \mathsf{Aids})}, \overline{\mathit{Treat}(\mathsf{Pete}, \mathsf{Medi1})}, O^+\}$, as a first, and $S_2' = \{-\mathit{Ill}(\mathsf{Mary}, \mathsf{Aids}), \mathit{Treat}(\mathsf{Pete}, \mathsf{Medi1}), -n(R), \overline{\mathit{Ill}(\mathsf{Mary}, \mathsf{Aids})}, \overline{n(R)}, O^+\}$ as a second answer set. These two answer sets correspond to the two minimal solutions with only deletions from Example 5 where $\mathit{Ill}(\mathsf{Mary}, \mathsf{Aids})$ must be deleted from K together with either $\mathit{Treat}(\mathsf{Pete}, \mathsf{Medi1})$ or the rule named R. Note that the two resulting $(K \setminus F)$ indeed satisfy inconsistency property (1), because O^+ is contained in every answer set of $(K \setminus F) \cup \mathit{prior} \cup PTR^{cred}$.

From Theorem 1 and the correspondence between update programs and explanations shown in [13], the following proposition follows for deletions.

Proposition 1 (Correctness for deletions for credulous users). *A knowledge base $K^{pub} = K \setminus F$ preserves confidentiality under the credulous response semantics and changes K subset-minimally iff F is obtained by an answer set of the program P^{cred} that is U-minimal among those satisfying the inconsistency property (1).*

5.3 Deletions for skeptical users

For the skeptical user case, we first have to find those abducibles that have to be deleted from K such that confidentiality is preserved for each individual policy element; hence, we find a credulous anti-explanation for each individual negative observation O_i^- (which indeed corresponds to a credulous explanation of O^+) by computing U-minimal answer sets for the following programs:

$$P_i^{skep} = UP \cup \mathit{prior} \cup PTR_i^{skep} \cup GR.$$

If S_i is one of these answer sets, the negative explanation F_i is obtained from the negative update atoms contained in S_i: $F_i = \{L \mid -L \in S_i\}$. We collect all negative explanations of P_i^{skep} in the set $\mathcal{F}_i$:

$$\mathcal{F}_i = \{F_i \mid F_i \text{ is obtained from a U-minimal answer set of } P_i^{skep}\}$$

In order to obtain a publishable knowledge base K^{pub} that preserves confidentiality for all policy elements, we combine the individual explanations $F_i \in \mathcal{F}_i$ in every possible way – and take the subset-minimal ones; that is, we obtain

$$\mathcal{F} = \{F = F_1 \cup \ldots \cup F_k \mid F_i \in \mathcal{F}_i \text{ and there is no } F' = F_1' \cup \ldots \cup F_k' \\ (\text{for } F_i' \in \mathcal{F}_i) \text{ such that } F' \subset F\}$$

Lastly, we choose those sets F from $\mathcal{F}$ that satisfy the following *consistency* check: the resulting knowledge base must be consistent with the negation of each of the policy entries. More formally, we check whether

$$(K \setminus F) \cup \mathit{prior} \cup PTR_i^{skep} \cup \; \{\leftarrow O_i^- .\} \text{ is consistent for each } i = 1, \ldots, k. \quad (2)$$

In other words, we verify that the combined explanation set F indeed preserves confidentiality of each single policy element. In sum, we make sure that no policy element can be deduced skeptically from $K^{pub} = (K \setminus F)$ together with the given background knowledge *prior*: for every policy element there is at least one answer set in which it is not true.

Example 10. In our running example for K with *prior* and *policy*$'$, for PTR_1^{skep} $S_1'' = \{-\mathit{Ill}(\mathsf{Mary}, \mathsf{Aids}),\ \mathit{Treat}(\mathsf{Pete}, \mathsf{Medi1}),\ n(R), \overline{\mathit{Ill}(\mathsf{Mary}, \mathsf{Aids})}, \mathit{Ill}(\mathsf{Pete}, \mathsf{Flu}), \neg \mathit{AbleToWork}(\mathsf{Pete}), O^+\}$, is the only answer set; whereas for PTR_2^{skep} $S_2'' = \{\mathit{Ill}(\mathsf{Mary}, \mathsf{Aids}),\ \mathit{Treat}(\mathsf{Pete}, \mathsf{Medi1}), n(R), \mathit{Ill}(\mathsf{Pete}, \mathsf{Aids}), O^+\}$ is the only answer set. Only the update atom $-\mathit{Ill}(\mathsf{Mary}, \mathsf{Aids})$ appears in S_1''; and hence $\mathcal{F} = \{\mathit{Ill}(\mathsf{Mary}, \mathsf{Aids})\}$. Which means that we obtain the minimal solution from Example 5 by deleting $\mathit{Ill}(\mathsf{Mary}, \mathsf{Aids})$ from K. Note that the resulting $(K \setminus F)$ indeed satisfies consistency property (2), because each $(K \setminus F) \cup \mathit{prior} \cup PTR_i^{skep}$ has at least on answer set in which O_i^- is not contained.
Note that it is indeed necessary to compute each explanation individually: otherwise, for the example P^{cred} credulous and skeptical explanations coincide and hence would delete more entries from K than necessary.

Similar to Proposition 1, the following result follows for skeptical users.

Proposition 2 (Correctness for deletions for skeptical users). *A knowledge base $K^{pub} = K \setminus F$ preserves confidentiality under the skeptical response semantics and changes K subset-minimally iff F is obtained by combining update atoms of the answer sets of the programs P_i^{skep} that are U-minimal among those satisfying the consistency property (2) for each i.*

5.4 Deletions and literal insertions for credulous users

To obtain a confidentiality-preserving knowledge base, (incorrect) entries may also be inserted into the knowledge base. To allow for insertions of literals, a more complex set $\mathcal{A}$ of abducibles has to be chosen. We reinforce the point that the subset $\mathcal{A} \cap K$ of abducibles that are already contained in the knowledge base K are those that may be deleted while the subset $\mathcal{A} \setminus K$ of those abducibles that are not contained in K may be inserted. In general, for literal insertions we could take the whole set of atoms that can be obtained by considering predicate symbols from the knowledge base K and the a priori knowledge *prior*, and then instantiating them in all possible ways according to the Herbrand universe of K and *prior*. By taking all atoms and their negations we obtain a set of literals; all those literals that are not contained in K can be used as abducibles for a positive explanation E. In other words, they can potentially be inserted into K to avoid deduction of secrets.

However, we can reduce this number of new abducibles by analyzing which literals have influence on the policy elements at all. First of all, we assume that the policy transformation is applied as described in Section 5.1. Then, starting from the atoms in policy elements C_i, we trace back all rules in $K \cup \mathit{prior}$ that influence these policy atoms and collect all atoms in the bodies as well as heads of these rules. In other words, we construct a dependency graph (similar to [17]). However, in contrast to the traditional dependency graph, (as EDPs allow disjunction in rule heads) we do not only consider body atoms but also the head atoms as well as all their negations. More formally, let P_0 be the set of literals that can be obtained from atoms that appear in the policy:

$$P_0 = \{A, \neg A \mid A \text{ is an atom in a literal or NAF-literal in a policy element}\}$$

Next we iterate and collect all the literals that the P_0 literals depend on:

$$P_{j+1} = \{A, \neg A \mid A \text{ is an atom in a literal or NAF-literal in the head or body of a rule } R \text{ where } R \in K \cup \mathit{prior} \text{ and } \mathit{head}(R) \cap P_j \neq \emptyset\}$$

and combine all these literals in a set $\mathcal{P} = (\bigcup_{j=0}^{\infty} P_j)$.

As we also want to have the option to delete rules from K (not only the literals in $\mathcal{P}$), we define the set of abducibles as the set $\mathcal{P}$ plus all those rules in K whose head depends on literals in $\mathcal{P}$:

$$\mathcal{A} = \mathcal{P} \cup \{R \mid R \in K \text{ and } \mathit{head}(R) \cap \mathcal{P} \neq \emptyset\}$$

Example 11. For the example $K \cup \mathit{prior} \cup \mathit{PTR}^{cred}$, the dependency graph on atoms is shown in Figure 2. We note that the policy atom $\mathit{Ill}(x, \mathsf{Aids})$ directly depends on the atoms $\mathit{Ill}(x, \mathsf{Flu})$, $\mathit{Treat}(x, \mathsf{Medi1})$ and $\mathit{Treat}(x, \mathsf{Medi2})$; the policy atom $\mathit{AbleToWork}(x)$ directly depends on the atom $\mathit{Ill}(x, \mathsf{Flu})$ which again depends on $\mathit{Ill}(x, \mathsf{Aids})$, $\mathit{Treat}(x, \mathsf{Medi1})$ and $\mathit{Treat}(x, \mathsf{Medi2})$. In the end, considering negations of these atoms $\mathcal{P} = \{(\neg)\mathit{Ill}(x, \mathsf{Aids}), (\neg)\mathit{AbleToWork}(x), (\neg)\mathit{Ill}(x, \mathsf{Flu}), (\neg)\mathit{Treat}(x, \mathsf{Medi1}), (\neg)\mathit{Treat}(x, \mathsf{Medi2})\}$ is obtained. Lastly, we also have to add the rule R from K to $\mathcal{A}$ because literals in its head are contained in $\mathcal{P}$.

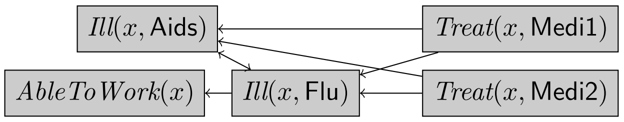

Fig. 2. Dependency graph for literals in *policy* wrt. $K \cup prior$

We obtain the normal form and then the update program UP for K and the new set of abducibles $\mathcal{A}$. The process of finding a skeptical explanation (for the new positive observation O^+) proceeds with finding an answer set of program P^{cred} as in Section 5.2 where additionally the positive explanation E is obtained as $E = \{L \mid +L \in S\}$ and S is U-minimal among those satisfying

$$(K \setminus F) \cup E \cup prior \cup PTR^{cred} \cup \{\leftarrow O^+.\} \text{ is inconsistent.} \tag{3}$$

Example 12. For UP from Example 9 the new set of abducibles leads to new insertion rules. The insertion rules for the new abducibles $Treat(\mathsf{Pete}, \mathsf{Medi2})$, $\neg Ill(x, \mathsf{Aids})$ and $Ill(x, \mathsf{Flu})$ are $+Treat(\mathsf{Pete}, \mathsf{Medi2}) \leftarrow Treat(\mathsf{Pete}, \mathsf{Medi2})$, as well as $+\neg Ill(x, \mathsf{Aids}) \leftarrow \neg Ill(x, \mathsf{Aids})$ and $+Ill(x, \mathsf{Flu}) \leftarrow Ill(x, \mathsf{Flu})$. With these new rules included in UP, we also obtain the solutions of Example 5 where the appropriate facts are inserted into K (together with deletion of $Ill(\mathsf{Mary}, \mathsf{Aids})$).

Proposition 3 (Correctness for deletions & literal insertions for credulous users). *A knowledge base $K^{pub} = (K \setminus F) \cup E$ preserves confidentiality and changes K subset-minimally iff (E, F) is obtained by an answer set of program P^{cred} that is U-minimal among those satisfying inconsistency property (3).*

5.5 Deletions and literal insertions for skeptical users

For skeptical users, we have to obtain the same new set of abducibles as for credulous users by tracing back all dependencies. But in the skeptical case we again have to find an anti-explanation for each policy element individually to avoid changing the knowledge base K more than necessary. Hence, we obtain the update programs UP based on the new set of abducibles and compute U-minimal answer sets of the following individual programs:

$$P_i^{skep} = UP \cup prior \cup PTR_i^{skep} \cup GR.$$

These answer sets may now contain positive explanations E_i as well as negative explanations F_i. If S_i is one of these answer sets, F_i is obtained from the negative update atoms contained in S_i: $F_i = \{L \mid -L \in S_i\}$ whereas E_i is $E_i = \{L \mid +L \in S_i\}$. We collect these explanations of P_i^{skep} in the set Exp_i:

$$Exp_i = \{(E_i, F_i) \mid (E_i, F_i) \text{ is obtained from a U-minimal answer set of } P_i^{skep}\}$$

Similar to the deletion only case, we combine the individual explanations $(E_i, F_i) \in \mathit{Exp}_i$ in every possible way – and take the subset-minimal ones; that is, we obtain

$$\begin{aligned}\mathit{Exp} = \{(E, F) \mid\ & F = F_1 \cup \ldots \cup F_k, E = E_1 \cup \ldots \cup E_k, (E_i, F_i) \in \mathit{Exp}_i \\ & \text{and there is no } F' = F'_1 \cup \ldots \cup F'_k \text{ and no } E' = E'_1 \cup \ldots \cup E'_k \\ & (\text{for } (E'_i, F'_i) \in \mathit{Exp}_i) \text{ such that } F' \cup E' \subset F \cup E\}\end{aligned}$$

We choose those sets (E, F) from Exp that satisfy the following *consistency* check: the resulting knowledge base must be consistent with the negation of each of the policy entries. More formally, we check whether

$$(K \setminus F) \cup E \cup \mathit{prior} \cup \mathit{PTR}_i^{\mathit{skep}} \cup \{\leftarrow O_i^-.\} \text{ is consistent for each } i = 1, \ldots, k. \quad (4)$$

That is, we again verify that the combined explanation (E, F) preserves confidentiality of each single policy element, and hence no policy element can be deduced skeptically from $K^{pub} = (K \setminus F) \cup E$ together with the given background knowledge prior.

Example 13. To give an example for literal insertions for skeptical users, we consider the given example K and *policy* as well as a new a priori knowledge $\mathit{prior}' = \{\neg \mathit{Ill}(\mathsf{Pete}, \mathsf{Flu})\}$. In this case, from $K \cup \mathit{prior}'$ the secrets $\mathit{Ill}(\mathsf{Mary}, \mathsf{Aids})$ and $\mathit{Ill}(\mathsf{Pete}, \mathsf{Aids})$ can both be deduced skeptically. There is only one *policy* element and hence only one program $\mathit{PTR}_1^{\mathit{skep}}$. This program has two U-minimal answer sets; one containing $-\neg \mathit{Ill}(\mathsf{Mary}, \mathsf{Aids})$ and $-\mathit{Treat}(\mathsf{Pete}, \mathsf{Medi1})$ and a second one containing $-\neg \mathit{Ill}(\mathsf{Mary}, \mathsf{Aids})$ and $+\mathit{Treat}(\mathsf{Pete}, \mathsf{Medi2})$. Hence we now also have the option to insert $\mathit{Treat}(\mathsf{Pete}, \mathsf{Medi2})$ in order protect the secret $\mathit{Ill}(\mathsf{Pete}, \mathsf{Aids})$.

Proposition 4 (Correctness for deletions & literal insertions for skeptical users). *A knowledge base $K^{pub} = (K \setminus F) \cup E$ preserves confidentiality and changes K subset-minimally iff (E, F) is obtained by combining update atoms of the answer sets of the programs P_i^{skep} that are U-minimal among those satisfying the consistency property (4) for each i.*

6 Discussion and Conclusion

This article showed that when publishing an extended disjunctive logic program, confidentiality-preservation can be ensured by extended abduction; more precisely, we showed that under the credulous and skeptical query response semantics it reduces to finding anti-explanations with update programs. This is an application of data modification, because a user can be misled by the published knowledge base to believe incorrect information; we hence apply dishonesties [12] as a security mechanism. This is in contrast to [17] whose aim is to avoid incorrect deductions while enforcing access control on a knowledge base. Another difference to [17] is that they do not allow disjunctions in rule heads; hence, to the best of our knowledge this article is the first one to handle a confidentiality problem for EDPs. In [3] the authors study databases that may provide users with incorrect answers to preserve security in a multi-user environment. Differently from our approach, they consider a database as a set of formulas of propositional logic and formulate the problem using modal logic. In analogy to [13], a complexity analysis for our approach can be achieved by reduction of extended abduction to normal abduction. More precisely, using the correspondence between extended abduction and confidentiality-preservation in Theorem 1, we can obtain computational complexity results for decision problems in confidentiality-preserving data publishing, based on the complexity results of extended abduction reported in [13]. Future work might handle insertion of non-literal rules. Moreover, the whole system could be extended by preferences among the possible solutions. Generally, we can consider preferences such that deleting facts is preferred to deleting rules, or inserting facts with non-confidential predicates is preferred to inserting facts with confidential ones.

References

1. Foto N. Afrati and Phokion G. Kolaitis. Repair checking in inconsistent databases: algorithms and complexity. In *ICDT2009*, volume 361 of *ACM International Conference Proceeding Series*, pages 31–41. ACM, 2009.
2. Joachim Biskup. Usability confinement of server reactions: Maintaining inference-proof client views by controlled interaction execution. In *DNIS 2010*, volume 5999 of *LNCS*, pages 80–106. Springer, 2010.
3. Piero A. Bonatti, Sarit Kraus, and V. S. Subrahmanian. Foundations of secure deductive databases. *IEEE Trans. Knowl. Data Eng.*, 7(3):406–422, 1995.
4. Francesco Calimeri, Giovambattista Ianni, Francesco Ricca, Mario Alviano, Annamaria Bria, Gelsomina Catalano, Susanna Cozza, Wolfgang Faber, Onofrio Febbraro, Nicola Leone, Marco Manna, Alessandra Martello, Claudio Panetta, Simona Perri, Kristian Reale, Maria Carmela Santoro, Marco Sirianni, Giorgio Terracina, and Pierfrancesco Veltri. The third answer set programming competition: Preliminary report of the system competition track. In *LPNMR 2011*, volume 6645 of *LNCS*, pages 388–403. Springer, 2011.
5. Jürgen Dix, Wolfgang Faber, and V. S. Subrahmanian. The relationship between reasoning about privacy and default logics. In *LPAR 2005*, volume 3835 of *Lecture Notes in Computer Science*, pages 637–650. Springer, 2005.
6. Csilla Farkas and Sushil Jajodia. The inference problem: A survey. *SIGKDD Explorations*, 4(2):6–11, 2002.
7. Michael Gelfond and Vladimir Lifschitz. Classical negation in logic programs and disjunctive databases. *New Generation Computing*, 9(3/4):365–386, 1991.
8. Bernardo Cuenca Grau and Ian Horrocks. Privacy-preserving query answering in logic-based information systems. In *ECAI2008*, volume 178 of *Frontiers in Artificial Intelligence and Applications*, pages 40–44. IOS Press, 2008.
9. Katsumi Inoue and Chiaki Sakama. Abductive framework for nonmonotonic theory change. In *Fourteenth International Joint Conference on Artificial Intelligence (IJCAI 95)*, volume 1, pages 204–210. Morgan Kaufmann, 1995.
10. Katsumi Inoue, Chiaki Sakama, and Lena Wiese. Confidentiality-preserving data publishing for credulous users by extended abduction. *The Computing Research*

Repository (CoRR) abs/1108.5825, Proceedings of the 19th International Conference on Applications of Declarative Programming and Knowledge Management (INAP), 2011.

11. Jiefei Ma, Alessandra Russo, Krysia Broda, and Emil Lupu. Multi-agent confidential abductive reasoning. In *ICLP (Technical Communications)*, volume 11 of *LIPIcs*, pages 175–186. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2011.
12. Chiaki Sakama. Dishonest reasoning by abduction. In *22nd International Joint Conference on Artificial Intelligence (IJCAI 2011)*, pages 1063–1064. IJCAI/AAAI, 2011.
13. Chiaki Sakama and Katsumi Inoue. An abductive framework for computing knowledge base updates. *Theory and Practice of Logic Programming*, 3(6):671–713, 2003.
14. Phiniki Stouppa and Thomas Studer. Data privacy for knowledge bases. In Sergei N. Artëmov and Anil Nerode, editors, *LFCS2009*, volume 5407 of *LNCS*, pages 409–421. Springer, 2009.
15. Tyrone S. Toland, Csilla Farkas, and Caroline M. Eastman. The inference problem: Maintaining maximal availability in the presence of database updates. *Computers & Security*, 29(1):88–103, 2010.
16. Lena Wiese. Horizontal fragmentation for data outsourcing with formula-based confidentiality constraints. In *IWSEC 2010*, volume 6434 of *LNCS*, pages 101–116. Springer, 2010.
17. Lingzhong Zhao, Junyan Qian, Liang Chang, and Guoyong Cai. Using ASP for knowledge management with user authorization. *Data & Knowl. Eng.*, 69(8):737–762, 2010.