COPYRIGHT NOTICE

Outlier Detection Using Default Reasoning ⋆

Fabrizio Angiulli [a], Rachel Ben-Eliyahu - Zohary [b,1,∗], Luigi Palopoli [a]

[a] *DEIS, Università della Calabria, Via Pietro Bucci 41C, 87036 Rende (CS), Italy*

[b] *Communication Systems Engineering Dept., Ben-Gurion University, Beer-Sheva, Israel*

Abstract

Default logics are usually used to describe the regular behavior and normal properties of domain elements. In this paper we suggest, conversely, that the framework of default logics can be exploited for detecting *outliers*. Outliers are observations expressed by sets of literals that feature unexpected semantical characteristics. These sets of literals are selected among those explicitly embodied in the given knowledge base. Hence, essentially we perceive outlier detection as a knowledge discovery technique. This paper defines the notion of *outlier* in two related formalisms for specifying defaults: Reiter's default logic and extended disjunctive logic programs. For each of the two formalisms, we show that finding outliers is quite complex. Indeed, we prove that several versions of the outlier detection problem lie over the second level of the polynomial hierarchy. We believe that a thorough complexity analysis, as done here, is a useful preliminary step towards developing effective heuristics and exploring tractable subsets of outlier detection problems.

Key words: default logic, disjunctive logic programming, knowledge representation, nonmonotonic reasoning, computational complexity, data mining, outlier detection

1. Introduction

This paper is about detecting outliers. In this work, outliers are unexpected observations, e.g., strange characteristics of individuals, in a given application domain. Exceptionality is determined here with respect to a given trustable knowledge base, with which a given set of elements does not comply. The issue that we address is how to locate such unusual elements automatically.

A first step towards automatically detecting outliers is to state their formal definition. In this work, it is assumed that the given knowledge base is expressed using a default reasoning language and hence we formalize our definition of outliers in this framework. The languages mainly dealt with are propositional default logics and extended disjunctive logic programs.

⋆ This manuscript is an extended and comprehensive report of results of which part have appeared in IJCAI-03 under the title "Outlier Detection Using Default Logic" and in ECAI-04 under the title "Outlier Detection Using Disjunctive Logic Programming".

∗ Corresponding author.

Email addresses: `f.angiulli@deis.unical.it` (Fabrizio Angiulli), `rbz@jce.ac.il` (Rachel Ben-Eliyahu - Zohary), `palopoli@deis.unical.it` (Luigi Palopoli).

1 Part of this work was done while the author was a visiting scholar in the Division of Engineering and Applied Sciences, Harvard University, Cambridge, Massachusetts.

Default logic was originally developed as a tool for working with incomplete knowledge. Default rules allow one to describe a normal behavior of a system and to draw consequent conclusions. As such, default rules can also be exploited for detecting outliers – observations that are unexpected according to the default theory at hand. This is the basic idea behind this paper. We refer to outliers as sets of observations that demonstrate some properties contrasting with those that can be logically "justified" according to the given knowledge base. Along with outliers, their "witnesses" are singled out– those unexpected properties that characterize outliers.

To illustrate, some informal application examples for outlier detection are described below.

Using outliers for diagnosis of computer hardware. Suppose that it usually takes about four seconds to download a gigabyte file from a server, but one day the system becomes slower, instead, eight seconds are needed to perform the same task. While eight seconds may indicate a good performance, it is, nonetheless, helpful to find the source of the delay in order to prevent more critical faults in the future. In this case, the download operation is the outlier while the delay is its witness.

Mechanical failure. Assume that someone's car brakes are making a strange noise. Although they seem to be functioning properly, this is not a normal behavior and the car is brought in for servicing. In this case, the car brakes are the outlier and the noise is a witness for it.

Knowledge base integrity. If an abnormal property is discovered in a database, the source that reported this information would have to be checked. Detecting abnormal properties, that is, detecting outliers, can also lead to an update of default rules in a knowledge base. For example, suppose we have the rule that birds fly, and we observe a bird that does not fly. This occurrence of such an outlier in the theory would be reported to the knowledge engineer. The engineer investigates the case, finds out that the bird is actually a penguin, therefore he updates the knowledge base with the default "penguins do not fly."

According to our approach, exceptions are not explicitly recorded in the knowledge base as "abnormals," as is often done in logical-based abduction [47,16,23]. Rather, their "abnormality" is singled out precisely because some of the properties characterizing them cannot be justified within the given theory.

In this paper we formally define outliers in both the related formalisms of *Reiter's default logic* and *Extended disjunctive logic programming* (EDLP).

Reiter's Default Logic is a powerful nonmonotonic formalism to deal with incomplete information, while logic programming is a practical tool that is widely employed in KR&R. The paper mostly deals with the propositional fragment of these logics. However, first-order default theories shall be also briefly discussed in the paper (see Section 5 below).

In the logic programming framework, we focus on Answer Set Semantics, which is used in most advanced systems for knowledge representation [38,43,40]. *Extended logic programs* (ELP) under Answer Set Semantics allow both negation as failure and classical negation to be used. These programs can be naturally embedded into default theories and therefore can be considered as a subset of default logic. As a consequence, our results for default theories carry over quite simply to ELPs. However, unlike ELP, extended *disjunctive logic programs* (EDLP) under Answer Set Semantics, in which also head-disjunction is allowed, cannot be viewed as a subset of default logic, although default logic in its full volume does include disjunction. Indeed, part of the motivation for developing disjunctive logic programming lies in the limitations of default logic in handling disjunctive knowledge (see the paper by Poole [47]). In this context, EDLP can be considered as a convenient tool for representing and manipulating complex knowledge [38] due to its declarativity and expressive power.

In what follows, we first introduce our formal definition of outliers. Then, we analyze the complexities involved in incorporating the outlier detection mechanism into knowledge bases expressed in default logic and extended disjunctive logic programs. We believe that a thorough complexity analysis is a useful step towards singling out the more complex subtasks involved in outlier detection. This first step is conducive to designing effective algorithms for implementation purposes.

According to the view adopted in this work, the witness that an observation is an outlier is a property or a behavior that is *explicitly* the opposite of what is expected. Representing such contradicting properties requires the usage of classical negation. Both default logic and extended logic programs make use of classical negation. Hence, these two languages represent a natural setting for outlier detection. A different approach, which does not require that the negation of the exceptional property is explicitly inferred but, rather, that it is simply not entailed by a logic program, is taken in [5]. As explained thoroughly in this paper, the anomalies that can be singled out by the definition of [5] are quite different than the outliers detected by the work presented here. This is mirrored in the different complexity figures we obtained: most of the outlier detection problems investigated here lie at the third level of the polynomial hierarchy, whereas the most complex of the problems considered in [5] are contained in its second level. In the sequel we will further elaborate on these differences.

The rest of this paper is organized as follows: Section 2 provides preliminary definitions and Section 3 defines outliers and related notions. Section 4 discusses the complexity of finding outliers in general propositional as well as in disjunction-free default logics. Section 5 deals with first-order defaults. Section 6 discusses related work – in particular, the relationship between outlier detection and abduction. Finally, Section 7 draws conclusions.

2. Preliminary Definitions

In this section we briefly review preliminary definitions used in default logic and extended (disjunctive) logic programs. Note that only the propositional fragment of these logics is considered here. Outlier detection in first-order default languages shall be briefly discussed in Section 5. Thus, whenever a default theory or a logic program with variables is used, it is referred to as an abbreviation of its grounded version.

2.1. *Default Logic*

Default logic was first introduced by Reiter in a first-order setting [50]. Next we recall basic definitions concerning the propositional fragment of default logic. Let T be a propositional theory and S a set of propositional formulas. Then, we denote by T^* the logical closure of T and by $\neg S$ the set $\{\neg(s) | s \in S\}$. A set of literals L is *inconsistent* if, for some literal $\ell \in L$, $\neg\ell \in L$.

A *propositional default theory* Δ is a pair (D, W) consisting of a set W of propositional formulas and a set D of default rules. In this paper we deal with finite default theories. A default theory $\Delta = (D, W)$ is finite if both the set of default rules D and the set of propositional formulas W are finite. A *default rule* δ has the form

$$\frac{\alpha : \beta_1, \ldots, \beta_m}{\gamma} \tag{1}$$

where α, each β_i, $1 \leq i \leq m$, and γ are propositional formulas. In particular, α is called the *prerequisite*, $\beta_1, \ldots, \beta_m$ are called the *justifications*, and γ is called the *consequent* (or *conclusion*) of δ. The prerequisite may be missing but the justification and the consequent are required (an empty justification is tantamount to have the identically true literal **true** [49] specified in its place). If the conclusion of a default rule is included in its justification, the rule is called *semi-normal* [25], while if the conclusion is identical to the justification the rule is called *normal*. A default theory containing only (semi-)normal defaults is called *(semi-)normal*. Given a default rule δ, $pre(\delta)$, $just(\delta)$, and $concl(\delta)$ denote the prerequisite, justification, and consequent of δ, respectively. Analogously, given a set of default rules $D = \{\delta_1, \ldots, \delta_n\}$, $pre(D)$, $just(D)$, and $concl(D)$ denote , respectively, the sets $\{pre(\delta_1), \ldots, pre(\delta_n)\}$, $\{just(\delta_1), \ldots, just(\delta_n)\}$, and $\{concl(\delta_1), \ldots, concl(\delta_n)\}$.

A propositional default theory $\Delta = (D, W)$ is *disjunction free* (DF for short) [33], if W is a set of literals, and $pre(\delta)$, $just(\delta)$, and $concl(\delta)$ are conjunctions of literals.

The informal meaning of a default rule δ can be stated as follows: If $pre(\delta)$ is known to hold and if it is consistent to assume $just(\delta)$, then infer $concl(\delta)$. The formal semantics of a default theory Δ is defined in terms of *extensions*, which denote maximal sets of conclusions that can be drawn from Δ. Thus, $\mathcal{E}$ is an extension for a theory $\Delta = (D, W)$ if it satisfies the following set of equations:

- $E_0 = W$,
- for $i \geq 0$, $E_{i+1} = E_i^* \cup \left\{\gamma \mid \frac{\alpha:\beta_1,\ldots,\beta_m}{\gamma} \in D, \alpha \in E_i, \neg\beta_1 \notin \mathcal{E}, \ldots, \neg\beta_m \notin \mathcal{E}\right\}$,
- $\mathcal{E} = \bigcup_{i=0}^{\infty} E_i$.

An extension $\mathcal{E}$ of a finite propositional default theory $\Delta = (D, W)$ can be finitely characterized through the set $D_{\mathcal{E}}$ of the *generating defaults* for $\mathcal{E}$ w.r.t. Δ [50,59]. In fact, [59] shows that a finite propositional default theory $\Delta = (D, W)$ has an extension $\mathcal{E}$ iff there exists a set $D_{\mathcal{E}} \subseteq D$, the generating defaults for $\mathcal{E}$ w.r.t. Δ, that can be partitioned into a finite number of strata $D_{\mathcal{E}}^{(0)}, D_{\mathcal{E}}^{(1)}, \ldots, D_{\mathcal{E}}^{(n)}$, such that:

– $D_{\mathcal{E}}^{(0)} = \{\delta \mid \delta \in D_{\mathcal{E}}, pre(\delta) \in W^*\}$,
– for each i, $1 \leq i \leq n$, $D_{\mathcal{E}}^{(i)} = \{\delta \mid \delta \in D_{\mathcal{E}} - \bigcup_{j=0}^{i-1} D_{\mathcal{E}}^{(j)}, pre(\delta) \in (W \cup concl(\bigcup_{j=0}^{i-1} D_{\mathcal{E}}^{(j)}))^*\}$,
– $(\forall \delta \in D_{\mathcal{E}})(\forall \beta \in just(\delta))(\neg\beta \notin (W \cup concl(D_{\mathcal{E}}))^*)$, and
– $(\forall \delta \in D)(pre(\delta) \in (W \cup concl(D_{\mathcal{E}}))^* \wedge (\forall \beta \in just(\delta))(\neg\beta \notin (W \cup concl(D_{\mathcal{E}}))^* \Rightarrow \delta \in D_{\mathcal{E}})$.

If such a set $D_{\mathcal{E}}$ exists, then $\mathcal{E} = (W \cup concl(D_{\mathcal{E}}))^*$ is an extension of Δ.

For the case of DF theories, it is useful to rewrite the definition of extension, as done in [33]. Let $\Delta = (D, W)$ be a DF default theory. Then $\mathcal{E}$ is an extension of Δ if there exists a sequence of defaults $\delta_1, ..., \delta_n$ from D and a sequence of sets $E_0, E_1, ..., E_n$, such that for all $i > 0$:
– $E_0 = W$,
– $E_i = E_{i-1} \cup concl(\delta_i)$,
– $pre(\delta_i) \subseteq E_{i-1}$,
– $(\nexists c \in just(\delta_i))(\neg c \in E_n)$,
– $(\nexists \delta \in D)(pre(\delta) \subseteq E_n \wedge concl(\delta) \not\subseteq E_n \wedge (\nexists c \in just(\delta))(\neg c \in E_n))$,
– $\mathcal{E}$ is the logical closure of E_n

where E_n is called the *signature set* of $\mathcal{E}$ and is denoted $lits(\mathcal{E})$ and the sequence of rules $\delta_1, ..., \delta_n$ is the set $D_{\mathcal{E}}$ of generating defaults of $\mathcal{E}$.

Although default theories are *nonmonotonic*, normal default theories satisfy the property of *semi-monotonicity* (see Theorem 3.2 of [50]). That is: Let $\Delta = (D, W)$ and $\Delta' = (D', W)$ be two default theories such that $D \subseteq D'$; then for each extension E of Δ there is an extension E' of Δ' such that $E \subseteq E'$.

A default theory may not have any extensions, like in the theory $(\{\frac{:\beta}{\neg\beta}\}, \emptyset)$. Then, a default theory is called *coherent* if it has at least one extension, and incoherent otherwise. Normal default theories are coherent. A coherent default theory $\Delta = (D, W)$ is called *inconsistent* if it has just one extension which is inconsistent. By Theorem 2.2 of [50], the theory Δ is inconsistent iff W is inconsistent.

The *entailment problem* is one of the basic problems in KR formalisms. For default theories, it is as follows: Given a default theory Δ and a propositional formula ϕ, does every extension of Δ contain ϕ? In the affirmative case, we write $\Delta \models \phi$. For a set of propositional formulas S, we analogously write $\Delta \models S$ to denote $(\forall \phi \in S)(\Delta \models \phi)$. The entailment realizes the form of reasoning called *skeptical* (or *cautious*) *reasoning* [18].

2.2. *Extended Disjunctive Logic Programs*

A literal is an expression of the form ℓ or $\neg\ell$ where ℓ is a propositional letter and the symbol "$\neg$" denotes classical negation. A propositional EDLP is a collection of rules of the form

$$L_1 | \dots | L_k \leftarrow L_{k+1}, \dots, L_m, not\ L_{m+1}, \dots, not\ L_n$$

where $n, m, k \geq 0$, the symbol "*not* " denotes negation by default and each L_i is a literal. If $k = 0$, then the rule is called an *integrity clause*. If $0 \leq k \leq 1$ then the rule is said to be *non-disjunctive*. A propositional ELP is a collection of non-disjunctive rules.

An EDLP is given semantics using *answer sets* [26], which are defined as follows: Let $\mathrm{Lit}(P)$ denote the set of literals obtained using the propositional letters occurring in P. By a *context* [12] we mean any subset of $\mathrm{Lit}(P)$. Let P be a *negation-by-default-free* EDLP. A context is S *closed under* P if, for each rule $L_1 | \dots | L_k \leftarrow L_{k+1}, \dots, L_m$ in P, if $L_{k+1}, \dots, L_m \in S$ then, for some $i = 1, \dots, k$, $L_i \in S$. An *answer set* of P is any minimal context S such that (1) S is closed under P and (2) if S is inconsistent, then $S = \mathrm{Lit}(P)$.

For general EDLPs answer sets are defined as follows: Let the *reduct of* P *w.r.t. the context* S, denoted by $Red(P, S)$, be the EDLP obtained from P by deleting (i) each rule that has $not\ L$ in its body, for some $L \in S$, and (ii) all remaining subformulas of the form $not\ L$ from rule bodies. Then, any context S which is an answer set of $Red(P, S)$ is an *answer set* of P.

The collection of all consistent answer sets of an EDLP P is denoted $\mathrm{ANSW}(P)$. An EDLP P is ANSW-*consistent* iff $\mathrm{ANSW}(P) \neq \emptyset$. An EDLP P entails a propositional formula F, written $P \models F$, if $F \in S^*$ for each $S \in \mathrm{ANSW}(P)$. $P \models G$, for a set of propositional formulas G, means that $(\forall F \in G)(P \models F)$.

2.3. *Complexity Theory*

Some basic definitions in complexity theory are recalled next, particularly that of the polynomial time hierarchy. The reader is referred to [31,45] for more details.

Decision problems are mappings from strings (encoding the input instance over a suitable alphabet) to $\{$"yes", "no"$\}$. The language associated with a decision problem is the set of all and only the strings over the alphabet such that the decision problem outputs "yes" on them. A (possibly nondeterministic) Turing machine M answers a decision problem if on a given input x, (*i*) a branch of M halts in an accepting state iff x is a "yes" instance, and (*ii*) all the branches of M halt in some rejecting state iff x is a "no"instance.

The class P is the set of decision problems that can be answered by a deterministic Turing machine in polynomial time. The class of decision problems that can be solved by a nondeterministic Turing machine in polynomial time is denoted by NP, while the class of decision problems whose complementary problem is in NP, is denoted by co-NP.

More generally, classes $\Sigma_{\mathrm{k}}^{\mathrm{P}}$ and $\Pi_{\mathrm{k}}^{\mathrm{P}}$, which form the *polynomial hierarchy*, are defined as follows: $\Sigma_0^{\mathrm{P}} = \Pi_0^{\mathrm{P}} = \mathrm{P}$ and for all $k \geq 1$, $\Sigma_{\mathrm{k}}^{\mathrm{P}} = \mathrm{NP}^{\Sigma_{\mathrm{k-1}}^{\mathrm{P}}}$, and $\Pi_{\mathrm{k}}^{\mathrm{P}} = \text{co-}\Sigma_{\mathrm{k}}^{\mathrm{P}}$. $\Sigma_{\mathrm{k}}^{\mathrm{P}}$ is modeled computability by a nondeterministic polynomial time Turing machine which may use an oracle for solving a problem in $\Sigma_{\mathrm{k-1}}^{\mathrm{P}}$. An oracle is, loosely speaking, a subprogram that can be run with a constant computational cost. Thus, NP = Σ_1^{P}, and co-NP = Π_1^{P}. The class $\mathrm{D}_{\mathrm{k}}^{\mathrm{P}}$, $k \geq 1$, is the class of problems that are defined as the conjunction of two independent problems, one from $\Sigma_{\mathrm{k}}^{\mathrm{P}}$ and one from $\Pi_{\mathrm{k}}^{\mathrm{P}}$. That is, a problem associated with a language L is in $\mathrm{D}_{\mathrm{k}}^{\mathrm{P}}$ if and only if there are two languages L_1, associated with a problem in $\Sigma_{\mathrm{k}}^{\mathrm{P}}$, and L_2, associated with a problem in Π_2^{P}, such that $L = L_1 \cap L_2$. Note that, for all $k \geq 1$, $\Sigma_{\mathrm{k}}^{\mathrm{P}} \subseteq \mathrm{D}_{\mathrm{k}}^{\mathrm{P}} \subseteq \Sigma_{\mathrm{k+1}}^{\mathrm{P}}$.

Finally, we need to recall the notion of reduction. A decision problem A_1 is *polynomially reducible* to a decision problem A_2 if there is a polynomial time computable function h such that for every x, $h(x)$ is defined and A_1 output "yes" on input x iff A_2 outputs "yes" on input $h(x)$. A decision problem A is *complete* for the class $\mathcal{C}$ of the polynomial hierarchy iff A belongs to $\mathcal{C}$ and every problem in $\mathcal{C}$ is polynomially reducible to A.

A well-known $\Sigma_{\mathrm{k}}^{\mathrm{P}}$-complete problem is to decide the satisfiability of a formula $\mathrm{QBE}_{\mathrm{k},\exists}$, that is, a formula of the form $\exists X_1 \forall X_2 \ldots Q X_k f(X_1, \ldots, X_k)$, where Q is $\exists$ if k is odd and $\forall$ if k is even, $X_1, \ldots, X_k$ are disjoint sets of variables, and $f(X_1, \ldots, X_k)$ is a propositional formula in conjunctive normal form if k is odd and in disjunctive normal form if k is even, on the set of variables $X_1, \ldots, X_k$. Analogously, deciding the validity of a formula $\mathrm{QBE}_{\mathrm{k},\forall}$, that is a formula of the form $\forall X_1 \exists X_2 \ldots Q X_k f(X_1, \ldots, X_k)$, where Q is $\forall$ if k is odd and is $\exists$ if k is even, and $f(X_1, \ldots, X_k)$ is a propositional formula in disjunctive normal form if k is odd and in conjunctive normal form if k is even, on the set of variables $X_1, \ldots, X_n$, is complete for $\Pi_{\mathrm{k}}^{\mathrm{P}}$. Finally, deciding the conjunction $\Phi \wedge \Psi$, where Φ is a $QBE_{k,\exists}$ formula and Ψ is a $\mathrm{QBE}_{\mathrm{k},\forall}$ formula, is complete for $\mathrm{D}_{\mathrm{k}}^{\mathrm{P}}$.

3. Outliers

Next, we shall formally define the notion of outlier in the context of default logic and extended disjunctive logic programming. Also, we shall introduce a number of significant decision problems (which we shall call *queries*) associated with singling outliers out.

3.1. *Outliers in Default Logic*

We start by defining the concept of outlier in default logic. To motivate the definition and clarify it, we present several examples.

Example 3.1 Consider the following default theory which represents the knowledge that birds normally fly, but penguins normally do not fly. Moreover, we know that penguins are birds. Also, we have observed that Tweety is a bird, Pini is a penguin, and Tweety does not fly.

$$D = \left\{ \frac{Bird(x) : Fly(x)}{Fly(x)}, \frac{Penguin(x) : \neg Fly(x)}{\neg Fly(x)}, \right\}$$

$$W = \{Bird(Tweety), Penguin(Pini), \neg Fly(Tweety)\} \cup \{Penguin(X) \rightarrow Bird(X)\}$$

This default theory has two extensions. One extension is the logical closure of $W \cup \{Bird(Pini), \neg Fly(Pini)\}$ and the other is the logical closure of $W \cup \{Bird(Pini), Fly(Pini)\}$. As $\neg Fly(Tweety) \in W$, both extensions include this literal. But Tweety's not flying is quite strange. Indeed, it is known that birds normally fly, Tweety is a bird and there is no apparent justification for the fact that Tweety does not fly (other than $\neg Fly(Tweety)$ belonging to W). Were Tweety a penguin, Tweety's not flying would be promptly explained. But, as the theory stands, Tweety's not flying is inexplicable. Moreover, if we try to nail down what makes all that exceptional, we may notice that if we had dropped the observation $\neg Fly(Tweety)$ from W, the exact opposite would have been concluded, namely, that Tweety does fly. But if both $\neg Fly(Tweety)$ and $Bird(Tweety)$ are dropped from W, it can be no longer concluded that Tweety flies. Hence, $Fly(Tweety)$ can be looked at as a "consequence" of the fact that Tweety is a bird. Thus $Bird(Tweety)$ is the observation to be considered exceptional and $\neg Fly(Tweety)$ determines this exceptionality. A set of literals like $\{Bird(Tweety)\}$ will be called an *outlier*, whereas a set of literals like $\{\neg Fly(Tweety)\}$ will be called its *witness set* in the following.

In sum, we can define an outlier as an observation characterized by some exceptional semantical property. In the logic, this observation will be denoted by a set of literals. Such sets of literals are going to denote anomalous characteristics of elements of the world that our knowledge base encodes (e.g., a *bird* named *Tweety* in the example above). Therefore, in what follows, though we may sometime talk informally about outliers as individuals, it should be clear that, formally, outliers are observations as encoded by sets of literals.

We can now give a formal definition of outlier. We make use of the following notation: given a set W and a list of sets $S_1, \ldots, S_n$, $W_{S_1,\ldots,S_n}$ denotes the set $W \setminus (S_1 \cup S_2 \cup \ldots \cup S_n)$.

Definition 3.2 (Outliers and Outlier Witness Set in Default Logic) Let $\Delta = (D, W)$ be a propositional default theory and let $L \subseteq W$ be a set of literals [2] . If there exists a non-empty set of literals $S \subseteq W_L$ such that:

(i) $(D, W_S) \models \neg S$, and

(ii) $(D, W_{S,L}) \not\models \neg S$

then we say that L is an *outlier* set in Δ and S is an *outlier witness* set for L in Δ. If there is no $L' \subset L$ and $S' \subseteq W_{L'}$ such that L' is an outlier with witness set S' in Δ, then we say that L is a *minimal* outlier set.

According to this definition, in the default theory of Example 3.1 we can conclude that $\{Bird(Tweety)\}$ denotes an outlier set and $\{\neg Fly(Tweety)\}$ is its witness.

Remark 3.3

(i) We point out that we regard outlier detection as a kind of data mining technique. Therefore, we mine from explicitly observed facts and, accordingly, outliers (as well as witnesses) are defined as sets of literals that are explicitly included in the set of observations W.

(ii) In some situations it may be useful for the analyst to be allowed to provide a specific set which outliers and witnesses should be mined from. This is certainly a sensible and interesting idea from an application viewpoint. However, if we use a different definition of outliers according to this idea, it will make no difference in the conceptual and theoretical development we are going to present in the following.

Next, we shall illustrate our definition by several further examples.

Example 3.4 A well-known center for rare diseases is located in the small city of Lamezia in Calabria. One hot day in summer you are walking along the nice streets of Lamezia when you notice a young man wearing a heavy coat going in the same direction. In this situation, if you are a student in a school of medicine interested in genetic diseases, you might want to ask that man about his rare illness. Another way to put it is to say that the fact that the man is wearing a coat in a hot summer day makes him an outlier, and one of the probable explanations at that time and place for such behavior is that this man has a rare genetic disease. A default theory Δ that describes this episode might be as follows:

$$D = \left\{ \frac{Day(x) \wedge Warm(x) \wedge Person(y) : \neg WearCoat(y, x)}{\neg WearCoat(y, x)} \right\}$$

$$W = \{Day(Tuesday), Warm(Tuesday), Person(Jim), WearCoat(Jim, Tuesday)\}$$

This theory claims that normally a person would not wear a coat on a warm day. The observations are that Tuesday is a day and Tuesday is warm and Jim is a person who is wearing a coat on Tuesday. This system would preferably

[2] Note that in a preliminary version of this work [4], an outlier was defined as a single literal. In this work, we generalize that definition since, as we will show in the sequel, in some scenarios the original definition might be too restrictive.

conclude that Jim is the argument of an outlier. Indeed, the reader can verify that the following facts are true:

(i) $(D, W_{\{WearCoat(Jim,Tuesday)\}}) \models \neg WearCoat(Jim, Tuesday)$

(ii) $(D, W_{\{WearCoat(Jim,Tuesday)\},\{Person(Jim)\}}) \not\models \neg WearCoat(Jim, Tuesday)$.

(iii) $(D, W_{\{WearCoat(Jim,Tuesday)\},\{Day(Tuesday)\}}) \not\models \neg WearCoat(Jim, Tuesday)$.

(iv) $(D, W_{\{WearCoat(Jim,Tuesday)\},\{Warm(Tuesday)\}}) \not\models \neg WearCoat(Jim, Tuesday)$.

Hence, $\{Person(Jim)\}$, $\{Day(Tuesday)\}$ and $\{Warm(Tuesday)\}$ are all the minimal outliers found in our theory and $\{WearCoat(Jim, Tuesday)\}$ is the unique outlier witness set for each of them. □

Outlier witnesses have been defined as sets because, in general, a single literal may not suffice to form a witness for a given outlier. We illustrate this in the following example.

Example 3.5 Consider the default theory $\Delta = (D, W)$, where the set of default rules D conveys the following information about weather and traffic:

(i) $\frac{SummerWeekend \wedge Traffic\_Jam : Accident \vee Tornado}{Accident \vee Tornado}$ – that is, normally, if there is a traffic jam during a summer weekend then an accident has occurred or a tornado hit the freeway.

(ii) $\frac{Accident : Police \wedge Ambulance}{Police \wedge Ambulance}$ – that is, normally, if an accident occurred then the police and ambulances are around.

(iii) $\frac{Tornado : Police \wedge Ambulance}{Police \wedge Ambulance}$ – that is, normally, if a tornado hits the freeway then the police and ambulances are around.

Suppose also that $W = \{SummerWeekend, Traffic\_Jam, \neg Police, \neg Ambulance\}$. Then, the set $S = \{Police, Ambulance\}$ is an outlier witness for the outlier $L = \{SummerWeekend\}$ (and for the outlier $L' = \{Traffic\_Jam\}$ as well). Note that there is no singleton witness for this outlier. □

Example 3.6 Consider the following default theory Δ:

$$D = \left\{ \frac{PlantOwner(x) : MakesMoney(x)}{MakesMoney(x)}, \frac{GoodWilling(x) : WantsReducedPollution(x)}{WantsReducedPollution(x)}, \frac{PlantOwner(x) \wedge GoodWilling(x) : Donates(x)}{Donates(x)} \right\}, \text{ and}$$

$$W = \{PlantOwner(Johnny), GoodWilling(Johnny), \neg MakesMoney(Johnny), Donates(Johnny), \neg WantsReducedPollution(Johnny)\}.$$

This theory claims that normally plant owners make money and that good-willed plant owners are interested in reduced pollution and in donations. The observations are that Johnny is a good-willed plant owner who does not make money and is not interested in reduced pollution, but anyway makes donations. Therefore it would be interesting to have a KR system that could automatically conclude that Johnny is the argument of two outliers. Indeed, the reader can verify that the following facts are true:

(i) $(D, W_{\{\neg MakesMoney(Johnny)\}}) \models MakesMoney(Johnny)$,

(ii) $(D, W_{\{\neg WantsReducedPollution(Johnny)\}}) \models WantsReducedPollution(Johnny)$,

(iii) $(D, W_{\{\neg MakesMoney(Johnny)\},\{PlantOwner(Johnny)\}}) \not\models MakesMoney(Johnny)$, and

(iv) $(D, W_{\{\neg WantsReducedPollution(Johnny)\},\{GoodWilling(Johnny)\}}) \not\models WantsReducedPollution(Johnny)$.

Hence, both $\{\neg MakesMoney(Johnny)\}$ and $\{\neg WantsReducedPollution(Johnny)\}$ are outlier witnesses, while $\{PlantOwner(Johnny)\}$ and $\{GoodWilling(Johnny)\}$ are the corresponding outliers. □

Finally, the following example demonstrates why it is sensible to define an outlier as a set, and not as a single literal.

Example 3.7 Consider a set of default rules D conveying the following information about watering the grass:

(i) $\frac{SprinklerOn : WetGrass}{WetGrass}$ – normally, if the sprinkler is on, the grass is wet.

(ii) $\frac{Rain : WetGrass}{WetGrass}$ – normally, when it is raining the grass is wet.

(iii) $\frac{SprinklerOn : \neg WinterTime}{\neg WinterTime}$ – the sprinkler does not normally operate during the winter.

(iv) $\frac{\neg WinterTime : \neg ChimneySmoke}{\neg ChimneySmoke}$ – normally, there is no smoke in the chimney when it is not winter time (since the fireplace is not used).

Now, suppose outliers have to be defined to be single literals. Then, for the observation $W_1 = \{Rain, \neg WetGrass\}$ we would have that $\{Rain\}$ is an outlier with witness set $\{\neg WetGrass\}$. Similarly, for the observations $W_2 = \{SprinklerOn, \neg WetGrass\}$, we would have that $\{SprinklerOn\}$ is an outlier with witness set $\{\neg WetGrass\}$. However, for the observation $W_3 = \{SprinklerOn, Rain, \neg WetGrass\}$ no outliers can be singled out. This is because:

(i) $(D, W_{\{\neg WetGrass\}}) \models WetGrass$, *but*
(ii) $(D, W_{\{\neg WetGrass, Rain\}}) \models WetGrass$ and also $(D, W_{\{\neg WetGrass, SprinklerOn\}}) \models WetGrass$.
However, if *both* $Rain$ and $SprinklerOn$ are dropped from W, it is no longer possible to infer $WetGrass$ and, as such, $\{Rain, WetGrass\}$ is an outlier according to our definition with witness set $\{\neg WetGrass\}$.

For another example, suppose we have the set D as above, and $W_4 = \{SprinklerOn, ChimneySmoke\}$. Then $\{SprinklerOn\}$ is an outlier, with the witness $\{ChimneySmoke\}$. If the observations are $W_5 = \{\neg WinterTime, ChimneySmoke\}$, then $\{\neg WinterTime\}$ is an outlier, with the witness $\{ChimneySmoke\}$. However, for the observations $W_6 = \{\neg WinterTime, SprinklerOn, ChimneySmoke\}$ we will not get any outliers if outlier would have been defined as a single literal while $\{\neg WinterTime, SprinklerOn\}$ is an outlier according to our definition with outlier witness $\{ChimneySmoke\}$.

Thus, definition that restricts the outlier to be a single literal may be too restrictive. □

3.2. *Outlier Detection in Extended Disjunctive Logic Programs*

We now define the concept of outlier in the context of EDLP. Within this reasoning framework, we assume that the general knowledge about the world is encoded as an extended (disjunctive) logic program D, called the *rule program*, and that the factual evidence about some aspects of the current status of the world is encoded in a set of literals W, called the *observations set*.

Thus, a *rule-observations program* P is a pair $P = (D, W)$ consisting of a rules program and an observations set. Intuitively, a rule-observations program relates the general knowledge encoded in D with the evidence about the world encoded in W.

In the following we denote by $P = (D, W)$ the EDLP $D \cup W$. Also, given two disjoint subsets L and S of W, we denote by P_S the logic program $D \cup (W \setminus S)$ and by $P_{S,L}$ the logic program $D \cup (W \setminus (S \cup L))$. In the context of EDLP knowledge bases, the definition of outlier is analogous to that given in the framework of default logic (Definition 3.2).

Definition 3.8 (Outliers and Outlier Witness Set in Extended Logic Programs) Let a rule-observations program $P = (D, W)$ and a set of literals $L \subseteq W$ be given. If there exists a non-empty set of literals $S \subseteq W_L$ such that:
(i) $P_S \models \neg S$, and
(ii) $P_{S,L} \not\models \neg S$
then we say that L is an *outlier set* and S is an *outlier witness* set for L in P.

All the motivating examples given in Section 3.1 for the default logic framework, except Example 3.5, can be translated to EDLP. For instance, the default theory given in Example 3.1 can be translated to an EDLP as follows: Let $P = (D, W)$, where $W = \{Bird(Tweety), \neg Fly(Tweety)\} \cup \{Bird(X) \leftarrow Penguin(X)\}$ and D is the set

$$\{\ Fly(X) \leftarrow Bird(X), not\ \neg Fly(X),$$
$$\neg Fly(X) \leftarrow Penguin(X), not\ Fly(X)\ \}.$$

Analogous to what we showed in Section 3.1, if we are trying to understand what makes Tweety an exception, we notice that if we drop the observation $\neg Fly(Tweety)$, we would conclude the exact opposite, namely, that Tweety does fly. Thus, $\{\neg Fly(Tweety)\}$ is a witness according to Definition 3.8. Furthermore, if we drop both the observations $\neg Fly(Tweety)$ and $Bird(Tweety)$, we are no longer able to conclude that Tweety flies. This implies that $Fly(Tweety)$ is a consequence of the fact that Tweety is a bird, and thus $Bird(Tweety)$ is an outlier.

In the next example, we use disjunctive information represented in an EDLP which is not head-cycle-free [11]. We are interested in a full-fledged EDLP for two main reasons. First, as noted by [11], head-cycle-free EDLPs can be faithfully translated into disjunction-free logic programs. Second, disjunction-free programs are equivalent to a subset of default logic and their expressive power is strictly less than that of general disjunctive programs [18].

The following example is adapted from [15]. We assume a situation where goods from a set G are produced by companies in a set C owned by a set of stock holders H. Each good is produced by at most two companies and each company may produce several goods. Suppose that currently H produces all goods in G by means of its companies, and this represents a business advantage over its competitors. Hence, the owners' policy prescribes that for a company $c \in C$ to be *safely* sold, H should not loose its capability of producing all goods. Therefore, the owners consider not safely sellable any company that belongs to all minimal sets of companies producing all goods. The situation is further complicated by the presence of a *control* relationship amongst companies: a company c might be controlled by a triplet of companies c_1, c_2, c_3. If this is the case, then c is considered safely sellable only if at least one among its controlling companies c_1, c_2, c_3 is safely sellable as well. Call *strategic* a company that cannot be safely sold according to the owners' policy. The owners need to know which companies can be safely sold. The situation can be formalized in an EDLP as follows. There are literals of the form $prod(g, c_1, c_2)$, one for each good g, to denote that good g is produced by companies c_1 and c_2. We will use the literal $prod(g, c, c)$ to denote that good g is produced by only one company c. There are literals of the form $contr(c, c_1, c_2, c_3)$ to denote that c is controlled by c_1, c_2, c_3. Rules in P are as follows:

– a company is not strategic if it is consistent to assume so

$$\neg strategic(X) \leftarrow not\ strategic(X)$$

– at least one of the companies producing a good is strategic

$$strategic(Y) \mid strategic(Z) \leftarrow prod(X, Y, Z)$$

– a company controlled by three strategic companies is strategic as well

$$strategic(W) \leftarrow contr(W, X, Y, Z), strategic(X), strategic(Y), strategic(Z)$$

– normally strategic companies are not sellable

$$unsell(X) \leftarrow strategic(X), not\ \neg unsell(X)$$

We recall that to establish whether a company c is indeed strategic in the above setting is a Π_2^{P}-complete problem [15] and, as such, cannot be expressed by means of any disjunction-free logic program (this is because disjunction-free logic programs can express only problems that are at most as complex as co-NP). Now assume that the following literals have been observed

$$\begin{aligned} W = \{\ & prod(g_1, c_1, c_2), prod(g_2, c_1, c_3), prod(g_3, c_2, c_3), prod(g_4, c_2, c_4), prod(g_5, c_3, c_4), prod(g_6, c_1, c_4), \\ & prod(g_7, c_2, c_5), contr(c_5, c_1, c_2, c_3), contr(c_5, c_1, c_2, c_4), contr(c_5, c_2, c_3, c_4), \neg unsell(c_5)\ \}. \end{aligned}$$

Then, it is evident that there is an outlier among the observations. Indeed, according to our formal definition, we have that $\{\neg unsell(c_5)\}$ is an outlier witness with outlier $\{contr(c_5, c_1, c_2, c_3)\}$.

3.3. *Some More Extensive Examples*

In this section we describe in detail some further and more extensive application examples of the proposed framework.

3.3.1. *Learning and Knowledge Base Integrity*

Assume a database of examples is given – both positive and negative examples. We want to acquire knowledge that abstracts the examples. A way to go is to learn rules encoding the knowledge. Clear enough, the more expressive the rule language employed for the learning purposes is, the richer the description of the example database properties as learnt in this process will be. In this context, techniques to induce defaults from examples can be applied, which guarantee the capability of encoding defeasible Σ_2^{P} properties of the database (whereas, for instance, learning Horn rules would result in the possibility of encoding "certain" polynomial time properties). For instance, the techniques of [20,42] can be applied here. Once the set of default rules has been learned, outlier detection techniques might be applied for checking abnormality occurring in the knowledge base.

Next we show an example of such application using the framework for learning default theories proposed in [20]. First, we recall the definition of a learned default theory provided there. [3]

Definition 3.9 (Learning a Default Theory) [20] Given a set of positive examples $E^+ = \{e_1, e_2, \ldots, e_n\}$ of the predicate p (that is, $p(e)$ is assumed to be true for all $e \in E^+$), a set of negative examples $E^- = \{e'_1, e'_2, \ldots, e'_m\}$ (that is, $\neg p(e)$ is assumed be true for all $e \in E^-$) and an initial consistent set of first order formulas W containing no occurrence of p, *learning a default theory* consists of finding a set D of defaults such that

[3] A slightly different definition was also provided by the same authors in [42].

$$(D, W) \models \left(\bigwedge_{e \in E^+} p(e) \right) \wedge \left(\bigwedge_{e \in E^-} \neg p(e) \right).$$

Informally speaking, [4] default rules learnt by the method reported in [20] have the following form:

$$\frac{\varphi(X) : p(X) \wedge \neg\psi(X)}{p(X)} \left(\frac{\varphi(X) : \neg p(X) \wedge \neg\psi(X)}{\neg p(X)}, \text{ resp.} \right)$$

where the formula $\varphi(X)$ generalizes some positive (negative, resp.) examples, and the formula $\psi(X)$ generalizes all the exceptions to $\varphi(X)$, that are the *negative* (*positive*, resp.) examples which are generalized by $\varphi(X)$.

Consider the following set of first-order formulas W:

$$\begin{aligned}
&pen(1), pen(2), bird(3), bird(4), bird(5), mam(6), mam(7), mam(8), mam(9), bat(10), superpen(11)\\
&pen(X) \rightarrow bird(X)\\
&superpen(X) \rightarrow pen(X)\\
&bat(X) \rightarrow mam(X)
\end{aligned}$$

where *pen* stands for penguin and *mam* stands for mammal. Assume the following set of positive and negative examples concerning the predicate *flies* are given:

$$\begin{aligned}
E^+ &= \{3, 4, 5, 10, 11\} \equiv \{flies(3), flies(4), flies(5), flies(10), flies(11)\}\\
E^- &= \{1, 2, 6, 7, 8, 9\} \equiv \{\neg flies(1), \neg flies(2), \neg flies(6), \neg flies(7), \neg flies(8), \neg flies(9)\}
\end{aligned}$$

Using the technique of [20] we will learn the following set of defaults D:

$$\begin{aligned}
&\delta_1 = \frac{bird(X) : flies(X) \wedge \neg pen(X)}{flies(X)}, \delta_2 = \frac{pen(X) : \neg flies(X) \wedge \neg superpen(X)}{\neg flies(X)},\\
&\delta_3 = \frac{superpen(X) : flies(X)}{flies(X)}, \delta_4 = \frac{bat(X) : flies(X)}{flies(X)}, \delta_5 = \frac{mam(X) : \neg flies(X) \wedge \neg bat(X)}{\neg flies(X)}
\end{aligned}$$

Let Δ be (D, W) where D and W are as defined above. Assume that the set of facts $W^{\text{ins}} = \{bird(12), \neg flies(12)\}$ is added to the theory Δ so that the theory $\Delta^{\text{ins}} = (D, W \cup W^{\text{ins}})$ is obtained. Then, in the theory Δ^{ins} the set $L = \{bird(12)\}$ is an outlier with witness $S = \{\neg flies(12)\}$. Indeed, $\Delta' = (D, W \cup W^{\text{ins}}_S)$ is such that $\Delta' \models flies(12)$ by means of the default rule $\frac{bird(X):flies(X) \wedge \neg pen(X)}{flies(X)}$, and $\Delta'' = (D, W \cup W^{\text{ins}}_{S,L})$ is such that $\Delta'' \not\models flies(12)$, since $bird(12)$ is no longer entailed by Δ''.

An outlier may indicate that something is functioning wrong and that some actions are to be taken. In the example at hand, the individual 12 could be unhealthy and thus requires to be cured. If it is believed that the outlier must be "removed" from the knowledge base, then this can be basically accomplished using two different procedures, that we describe next in the context of the example at hand. According to the first procedure, it is acknowledged that the bird 12 is a penguin and, hence, the literal $pen(12)$ is added to the theory Δ^{ins}. According to the second procedure, it is recognized that 12 cannot fly since one of its wings is broken. Consequently, the fact $sick(12)$ is added to the set W, 12 is added to the set of negative examples, and a novel set of defaults D^{rev} (taking into account this kind of exception) will be learned. In particular, the set D^{rev} will differ from the set D since the default rules δ_1 and δ_2 will be replaced by the two following ones:

$$\begin{aligned}
\delta_1^{\text{rev}} &= \frac{bird(X) : flies(X) \wedge \neg pen(X) \wedge \neg sick(X)}{flies(X)}, \text{ and}\\
\delta_2^{\text{rev}} &= \frac{pen(X) \vee sick(X) : \neg flies(X) \wedge \neg superpen(X)}{\neg flies(X)}.
\end{aligned}$$

[4] The reader is referred to [20] for more details.

It is interesting to stress the relationships between outliers and learned default theories. Let $\Delta = (D, W)$ be a learned theory, and assume some set of facts W' is added to it. Assume that $\{\neg p(c)\} \subseteq W'$ is an outlier witness set for an outlier $L \subseteq (W \cup W')$. Then, it is the case that $(D, W \cup W'_{\{\neg p(c)\}}) \models p(c)$, that is, that c behaves as a positive example, while we have stated the exact opposite, that is, we have $\neg p(c)$ as a negative example provided with the update set W'. A similar reasoning can be followed in case where the witness set is $\{p(c)\}$.

3.3.2. *A Biological Rule Database*

For several reasons, the realm of biology is quite interesting for applying outlier detection techniques . First of all, *rules* in biology often have exceptions. Second, the domain is not completely known. Third, knowledge base and data base tools in bioinformatics applications are critically needed [44].

With the aid of a biologist, we formalized the following knowledge base about the "central dogma" of molecular biology, that is, the process according to which DNA sequences are translated into proteins. The knowledge base $\Delta^{\text{bio}} = (D^{\text{bio}}, W^{\text{bio}})$ is as follows:

$Rule_1 : \frac{DNA(S):\neg transRNA(S)}{\neg transRNA(S)}$	Normally a generic DNA sequence is not transcribed to RNA.
$Rule_2 : \frac{DNA(S) \wedge prom(S):transRNA(S)}{transRNA(S)}$	Normally a DNA sequence having a promotor is transcribed to RNA.
$Rule_3 : \frac{transRNA(S) \wedge RBS(S):transProtein(S)}{transProtein(S)}$	Normally a transcribed DNA sequence that has a Ribosomal Binding Site is translated into a protein.
$Rule_4 : \frac{transProtein(S):\neg deg(S)}{\neg deg(S)}$	Normally a translated protein is not degradated.
$Rule_5 : \frac{DNA(S):\neg prom(S)}{\neg prom(S)}$	Normally a DNA sequence doesn't have a promotor.
$Rule_6 : \frac{transRNA(S):\neg RBS(S)}{\neg RBS(S)}$	Normally a transcriptable RNA sequence has not a Ribosomal Binding Site.
$Rule_7 : \frac{recDeg(S):foundSubseq(S)}{foundSubseq(S)}$	Normally, if a protein degradated just recently, some broken subsequences are found in the wet lab experimental sample.
$Rule_8 : \frac{transProtein(S) \wedge \neg deg(S):foundProtein(S)}{foundProtein(S)}$	Normally when a sequence is translated into a protein and the protein is not degradated, the protein is found in the experimental sample.
$Rule_9 : \frac{:\neg foundProtein(S)}{\neg foundProtein(S)}$	Normally proteins are not found in experimental samples.
$Rule_{10} : \frac{old(S):\neg foundSubseq(S)}{\neg foundSubseq(S)}$	Normally if a degradated protein is an old one no subsequences are found.
$Rule_{11} : prom(S) \wedge rep(S) \rightarrow \neg transRNA(S)$	It is known that if a repressor binds the promotor region of a DNA sequence, then the sequence is not transcribed.
$Rule_{12} : \neg prom(S) \rightarrow \neg transRNA(S)$	It is known that if no promoter region is in a DNA sequence, it is not transcribed.
$Rule_{13} : deg(S) \rightarrow recDeg(S) \vee old(S)$	It is known that degradated proteins are those which are either degradated recently or very old ones.
$Rule_{14} : foundSubseq(S) \rightarrow deg(S)$	It is known that if some broken subsequences are found in the experimental sample the protein is degradated.

Assume, now, that the results of two different wet lab experiments are encoded in the two following sets of literals:

$W_1^{\text{exp}} = \{DNA(seqA), prom(seqA), RBS(seqA), \neg foundProtein(seqA)\}$, and

$W_2^{\text{exp}} = \{DNA(seqB), prom(seqB), rep(seqB), RBS(seqB), foundSubseq(seqB), foundProtein(seqB)\}$.

Then in the theory $(D^{\text{bio}}, W^{\text{bio}} \cup W_1^{\text{exp}})$ the sets $\{DNA(seqA)\}$, $\{prom(seqA)\}$, and $\{RBS(seqA)\}$ are all outliers with witness $\{\neg foundProtein(seqA)\}$. Indeed, these sets include all the indications that protein should be found. Therefore, it might be concluded that a repressor binds the promotor region of the DNA sequence or the protein is degradated.

Also in the theory $(D^{\text{bio}}, W^{\text{bio}} \cup W_2^{\text{exp}})$ there are outliers. In particular, $L = \{rep(seqB), foundSubseq(seqB)\}$ is a minimal outlier having witness $S = \{foundProtein(seqB)\}$. In fact, it is surprising that the protein is found whereas a repressor is present and some broken subsequences are found in the lab sample.

4. Complexity Results

In this section we study the computational complexity underlying outlier detection problems. Formal proofs of the results we present are reported in Appendix A (for results of Section 4.2 concerning default theories) and Appendix B (for results of Section 4.3 concerning extended (disjunctive) logic programs). Below we provide all the theorems together with an informal outline of the proofs. To start with, we define the outlier detection problems we are going to analyze.

4.1. *Outlier Detection Queries and Result Summary*

In order to analyze the computational complexity underlying outlier detection we refer to some decision problems, called *queries*, which are defined next. These queries refer to a given knowledge base $\mathcal{KB}$, where $\mathcal{KB}$ is either a default theory $\Delta = (D, W)$ or an EDLP rule-observations program $P = (D, W)$:
– OUTLIER: Given $\mathcal{KB}$, does there exist at least one outlier set in $\mathcal{KB}$?
– OUTLIER[k]: Let k be a constant positive integer. Given $\mathcal{KB}$, does there exist at least one outlier set with cardinality of at most k in $\mathcal{KB}$?
– OUTLIER(L): Given $\mathcal{KB}$ and a set of literals $L \subseteq W$, is L an outlier in $\mathcal{KB}$?
– OUTLIER(S): Given $\mathcal{KB}$ and a set of literals $S \subseteq W$, is S a witness for some outlier set L in $\mathcal{KB}$?
– OUTLIER[k](S): Let k be a constant positive integer. Given $\mathcal{KB}$ and a set of literals $S \subseteq W$, is S a witness for any outlier set L of cardinality of at most k in $\mathcal{KB}$?
– OUTLIER(S)(L): Given $\mathcal{KB}$, a set of literals $S \subseteq W$, and a set of literals $L \subseteq W$, is L an outlier with witness S in $\mathcal{KB}$?
Furthermore, we are interested in the complexity of some of the above-defined problems when only *minimal* outliers are to be singled out. Thus, we also consider the following two additional queries:
– OUTLIER-MIN(L): Given $\mathcal{KB}$ and a set of literals $L \subseteq W$, is L a minimal outlier in $\mathcal{KB}$ (that is, there is no other outlier L' in $\mathcal{KB}$ such that $L' \subseteq L$)?
– OUTLIER-MIN(S)(L): Given $\mathcal{KB}$, a set of literals $S \subseteq W$, and a set of literals $L \subseteq W$, is L a minimal outlier and is S a witness set for L in $\mathcal{KB}$?
Note that the complexity of a query asking for the existence of a minimal outlier set is obviously the same as that of query OUTLIER, since an outlier exists in a given theory if and only if a minimal one is there as well.

Complexity results are summarized in Table 1 and explained below. It is clear from Table 1 that answering outlier detection problems on propositional normal general (disjunction-free, resp.) default theories turns out to be as hard as answering them on propositional disjunctive (non-disjunctive, resp.) extended logic programs.

4.2. *Complexity of outlier detection using Reiter's Default Logic*

In this subsection we analyze the complexity associated with detecting outliers in general and and in DF propositional default logic. We preliminarily notice that all the membership results have been established for general theories and general disjunction-free theories, while hardness results have been established for a strict subset of them, that is, normal theories. Hence, complexity results hold overall for both normal default theories and general default theories.

We start our analysis with the query OUTLIER, the most general of the set. Given a default theory, this query asks whether there exists an outlier in the theory.

Theorem 4.1 *On propositional default theories,* OUTLIER *is* Σ_3^{P}*-complete for general default theories, and* Σ_2^{P}*-complete for* DF *theories.*

Proof Outline. It follows from Definition 3.2 that the $\mathcal{C}$-membership of the query OUTLIER given on a propositional default theory $\Delta = (D, W)$, where $\mathcal{C}$ denotes a suitable complexity class, can be proved by building a nondeterministic Turing machine T that simultaneously guesses two disjoint subsets L and $S = \{s_1, \ldots, s_n\}$ of W, and then verifies that

Theory / Query	*Normal Default* EDLP	*Normal* DF *Default* ELP
OUTLIER	Σ_3^{P}-complete	Σ_2^{P}-complete
OUTLIER[k]	Σ_3^{P}-complete	Σ_2^{P}-complete
OUTLIER(L)	Σ_3^{P}-complete	Σ_2^{P}-complete
OUTLIER(S)	$\mathrm{D}_2^{\mathrm{P}}$-complete	D^{P}-complete
OUTLIER[k](S)	$\mathrm{D}_2^{\mathrm{P}}$-complete	D^{P}-complete
OUTLIER(S)(L)	$\mathrm{D}_2^{\mathrm{P}}$-complete	D^{P}-complete
OUTLIER-MIN(L)	$\mathrm{D}_3^{\mathrm{P}}$-complete	$\mathrm{D}_2^{\mathrm{P}}$-complete
OUTLIER-MIN(S)(L)	Π_3^{P}-complete	Π_2^{P}-complete

Table 1
Complexity results for outlier detection

– $(D, W_S) \models \neg s_1 \wedge \ldots \wedge \neg s_n$ (entailment problem q'), and
– $(D, W_{S,L}) \not\models \neg s_1 \wedge \ldots \wedge \neg s_n$ (entailment problem q'').

Let $\mathcal{C}_e$ be the complexity class of the entailment problem for Δ. Then the problem q' is in the class $\mathcal{C}_e$, while the problem q'' is in the class co-$\mathcal{C}_e$. Thus, T can employ a $\mathcal{C}_e$ oracle to solve both q' and q''. Hence, the query OUTLIER is in the class $\mathcal{C} = NP^{\mathcal{C}_e}$. Recall that the entailment problem is in $\Pi_2^P = $ co-Σ_2^P for general propositional default theories [55,27], and is in co-NP for DF propositional default theories [33]. As a consequence, query OUTLIER belongs to the classes $\mathrm{NP}^{\Sigma_2^P} = \Sigma_3^P$ and $\mathrm{NP}^{\mathrm{NP}} = \Sigma_2^P$ for general propositional default theories and for DF propositional default theories, respectively.

Completeness is proved by reducing the Σ_k^P-complete ($k \in \{2,3\}$) problem of deciding the validity of a $QBE_{k,\exists}$ formula to OUTLIER. The reader is referred to the Appendix for the detailed proof. □

Let us now turn to analyzing the second query. Given a theory and a positive integer number k, the query OUTLIER[k] asks for the existence of an outlier of size at most k in the theory. The complexity of this query is stated below.

Theorem 4.2 *On propositional default theories,* OUTLIER[k] *is* Σ_3^{P}*-complete for general theories, and* Σ_2^{P}*-complete for propositional* DF *default theories.*

Proof Outline. Bounding the size of the outlier does not change the complexity of singling it out. Indeed, as for the membership, both a witness set S and an outlier set L such that $|L| \leq k$ can be guessed, and the rest of the proof follows the same line of reasoning outlined above for Theorem 4.1. As for hardness, the construction referred to in Theorem 4.1 still holds as well, since the outlier set L we employ in the proof has size 1 and, hence, complies with any possible value of k. □

Next, we focus on query OUTLIER(L). It turns out that knowing the outlier set L in advance does not reduce the complexity w.r.t. the general OUTLIER query since, in particular, the number of possible outlier witness sets $S \subseteq W_L$ for L is still exponential. This is summarized in the following theorem.

Theorem 4.3 *On propositional default theories,* OUTLIER(L) *is* Σ_3^{P}*-complete for general theories, and* Σ_2^{P}*-complete for* DF *theories.*

Proof Outline. The same as for Theorem 4.1. □

Given a default theory and a set of literals S, query OUTLIER(S) asks whether S is a witness set for any outlier in the theory. It turns out that the complexity of OUTLIER(S) is lower than the complexity of OUTLIER. This is so because, once the candidate outlier witness set S is given, there is no need to check all the potential outlier witnesses (and there is an exponential number of potential witnesses). The result is given in the following theorem.

Theorem 4.4 *On propositional default theories,* OUTLIER(S) *is* $\mathrm{D}_2^{\mathrm{P}}$*-complete for general theories and* D^{P}*-complete for* DF *theories.*

Proof Outline. In order to prove membership we refer, again, to the entailment problems q' and q'' introduced in the proof outline of Theorem 4.1, but this time the outlier witness set S is fixed in advance. We recall that for general propositional default theories, q' is in Π_2^P, while for DF propositional default theories, it is in co-NP. As for problem

q'', it is in Σ_2^P and in NP for general theories and for DF theories, respectively, provided that L is known (these are standard entailment problems for general and DF default theories). However, it is possible to show that q'' belongs to those classes also when L is unknown. Indeed, q'' can be answered by showing that there exists a set L in W_S and an extension E of the theory $(D, W_{S,L})$ such that $\neg S \notin E$. Thus, a nondeterministic polynomial time Turing machine TM can be built that guesses simultaneously the set $L \subseteq W_S$, the subset $D_E \subseteq D$ of generating defaults of an extension E of $(D, W_{S,L})$, and an ordering of the rules in D_E. Then, TM proceeds as follows:

- for general propositional default theories, TM uses an NP oracle (a) to check the conditions that D_E must satisfy to be a set of generating defaults for E, and (b) to verify that $\neg s_1 \wedge \ldots \wedge \neg s_n \notin E$. These steps can be performed by executing a polynomially bounded number of calls to the NP oracle;
- for DF default theories, TM (a) checks the conditions that D_E must satisfy to be a set of generating defaults for an extension E of a disjunction-free theory, and (b) verifies that $\neg s_1 \wedge \ldots \wedge \neg s_n \notin E$, by checking that for every i, $1 \leq i \leq n$, $\neg s_i$ is not the conclusion of any default in D_E. These steps can be performed in polynomial time.

Thus, for general default theories, OUTLIER(S) is the conjunction of two independent problems belonging to Π_2^P and Σ_2^P and, therefore, it is in D_2^P. For DF default theories, OUTLIER(S) is the conjunction of two independent problems, one from co-NP, the other from NP and, hence, OUTLIER(S) is in D^P.

As for hardness, we consider the following decision problem, which we call *problem* q. Given two independent default theories Δ_1 and Δ_2, and two letters s_1 and s_2, the problem q is to verify whether the following is true:

$$(\Delta_1 \models s_1) \wedge (\Delta_2 \not\models s_2) \text{ (problem } q).$$

For general propositional default theories, q is a D_2^P-complete problem, since testing whether $\Delta_1 \models s_1$ is Π_2^P-complete, while the problem of testing $\Delta_2 \not\models s_2$ is Σ_2^P-complete [27,55]. For DF propositional default theories, q is a D^P-complete problem, since testing $\Delta_1 \models s_1$ is co-NP-complete, while testing $\Delta_2 \not\models s_2$ is NP-complete [33]. Then, hardness of query OUTLIER(S) is proven by reducing q to query OUTLIER(S). Given an instance of q, a default theory $\Delta(q) = (D(q), W(q))$ is associated with q such that $\neg s_1, s_2 \in W(q)$, and q is true iff $\{\neg s_1\}$ is an outlier witness set for $\{s_2\}$ in $\Delta(q)$. □

The following result shows that, similarly to what was shown for query OUTLIER, bounding the size of the outlier set to be associated with the provided witness set S in advance does not change the complexity figures.

Theorem 4.5 *On propositional default theories,* OUTLIER[k](S) *is* D_2^P*-complete for general theories, and* D^P*-complete for* DF *theories.*

Proof Outline. Both membership and hardness can be proved as discussed above for Theorem 4.4. To prove membership it suffices to guess only outlier sets with size of at most k. As for hardness, the reduction proceeds as described in the proof outline of Theorem 4.4, since outlier witness sets employed in the construction are singleton sets. □

Next, we analyze the query OUTLIER(S)(L). Note that this query is important because it might be the basic operator in a system for outlier detection using propositional default theories. We recall that, given a default theory and two sets of literals S and L, this query "simply" asks if S is an outlier witness set for the outlier L in that theory.

Theorem 4.6 *On propositional default theories,* OUTLIER(S)(L) *is* D_2^P*-complete for general theories, and* D^P*-complete for* DF *theories.*

Proof Outline. Complexity of query OUTLIER(S)(L) is the same as that of query OUTLIER(S). Similarly to what happens for query OUTLIER(L) with respect to query OUTLIER, knowing the outlier set L in advance does not reduce the complexity of OUTLIER(S)(L) with respect to query OUTLIER(S). Indeed, recall that even if sets L and S are fixed, it is nonetheless needed to solve entailment problems q' and q'' defined in the proof outline of Theorem 4.1, which are, respectively, Π_2^P-complete (co-NP-complete, resp.) and Σ_2^P-complete (NP-complete, resp.) for propositional general (DF, resp.) default theories. Hence, the result. □

Given a default theory and a set of literals L, the query OUTLIER-MIN(L) asks whether L is a minimal outlier set in the theory. Although the outlier set is given as input to this query, OUTLIER-MIN(L) turns out to be the most complex of the outlier detection problems considered in this work, even more complex than the general OUTLIER query.

Theorem 4.7 OUTLIER-MIN(L) *on propositional default theories is* D_3^P*-complete for general theories, and* D_2^P*-complete for* DF *theories.*

Proof Outline The membership part of the theorem can be proved as follows. First, consider the problem of verifying that L is indeed an outlier, that is, that there exists a witness set $S \subseteq W_L$ for L. It has been shown in Theorem 4.3 that OUTLIER(L) is Σ_3^P-complete for general default theories and Σ_2^P-complete for DF theories. Once verified that L is an outlier it must be further shown that L is a *minimal* outlier, i.e., that for each nonempty subset L' of L and for each subset S' of $W_{L'}$, L' and S' together do not satisfy Definition 3.2. The negation of the latter problem can be solved by a polynomial time nondeterministic Turing machine with an oracle for the entailment problem that guesses the two subsets L' and S' and then verifies if they indeed satisfy Definition 3.2. As a consequence, the problem is in Π_3^{P} for general default theories and in Π_2^{P} for DF theories.

It can be concluded that the overall query OUTLIER-MIN(L) on general (resp. DF) theories is the conjunction of two independent problems, one from Σ_3^{P} (resp. Σ_2^P) and one from Π_3^{P} (resp. Π_2^{P}) and, thus it lies in $\mathrm{D}_3^{\mathrm{P}}$ (resp. $\mathrm{D}_2^{\mathrm{P}}$).

Hardness of query OUTLIER-MIN(L) for general theories is proved by reducing the problem of deciding the validity of a formula

$$F = ((\exists X)(\forall Y)(\exists Z) f(X,Y,Z)) \wedge ((\forall W)(\exists U)(\forall V) g(W,U,V))$$

to the problem OUTLIER-MIN(L). Within F, $f(X,Y,Z)$ is a Boolean formula in conjunctive normal form and $g(X,Y,Z)$ is a Boolean formula in disjunctive normal form. Formula F is the conjunction of a $QBE_{3,\exists}$ and a $QBE_{3,\forall}$ and, hence, this reduction establishes the completeness of OUTLIER-MIN(L) on general theories for the class D_3^P. A similar reduction, but exploiting a conjunction of a $QBE_{2,\exists}$ formula and a $QBE_{2,\forall}$ formula, is used to prove the hardness of OUTLIER-MIN(L) for DF default theories. □

Let us now consider the query OUTLIER-MIN(S)(L). Recall that given a default theory and two disjoint sets S and L, this query asks if L is a minimal outlier set having S as a witness set in the theory. Note that this query is at least as complex as query OUTLIER(S)(L) that checks whether L and S represent a pair of outlier (not necessarily a minimal one) and a witness. The precise complexity is stated next.

Theorem 4.8 OUTLIER-MIN(S)(L) *on propositional default theories is* Π_3^{P}*-complete for general theories, and* Π_2^{P}*-complete for* DF *theories.*

Proof Outline. Let us first consider membership. In order to answer the query OUTLIER-MIN(S)(L) it must be verified that (a) S and L satisfy Definition 3.2, i.e., $(D, W_S) \models \neg S$ and $(D, W_{S,L}) \not\models \neg S$, and (b) for each subset L' of L, and S' of $W_{L'}$, S' and L' do not satisfy Definition 3.2, i.e. $(D, W_{S'}) \not\models \neg S'$ or $(D, W_{S',L'}) \models \neg S'$. The former query coincides with OUTLIER(S)(L), and hence it is in $\mathrm{D}_2^{\mathrm{P}}$ (resp. D^{P}) for general (resp. DF) theories. Vice versa, the query at (b) is in Π_3^{P} (resp. Π_2^{P}) for general (resp. DF) theories, since its negation can be answered by a nondeterministic polynomial time Turing machine that guesses a pair of disjoint subsets $L' \subseteq L$ and $S' \subseteq W_{L'}$ and then checks that they form an outlier and witness pair by using an oracle in Σ_2^{P}. Thus, the overall problem is in Π_3^{P} (resp. Π_2^{P}).

As for hardness, in the case of general (resp., DF) theories, it is proved by reducing the problem of deciding the validity of a $\mathrm{QBE}_{3,\forall}$ (resp., $\mathrm{QBE}_{2,\forall}$) formula to query OUTLIER-MIN(S)(L). □

4.3. *Complexity Results for Extended (Disjunctive) Logic Programs*

This section discusses the complexity of detecting outliers when ELPs (Section 4.3.1) and EDLPs (Section 4.3.2) are considered.

4.3.1. *Complexity of Outlier Detection in Extended Logic Programs*

Extended logic programs, for which disjunction is not allowed, correspond to a subset of default theories. The correspondence between the two languages is as follows [26]. For each ELP rule r:

$$L_0 \leftarrow L_1, \ldots, L_m, not\ L_{m+1}, \ldots, not\ L_n$$

let $\delta(r)$ denote the following default rule:

$$\frac{L_1 \wedge \ldots \wedge L_m : \neg L_{m+1}, \ldots, \neg L_n}{L_0},$$

where the justification of $\delta(r)$ is the identically true literal **true** if r has no negation as failure literals (i.e., if $n = m$). Then, with every ELP P, we can associate a default theory $\Delta^P = (\{\delta(r) \mid r \in P\}, \emptyset)$ such that the following holds [26]:

(i) If M is an answer set of P, then the deductive closure of M is an extension of Δ^P, and

(ii) every extension of Δ^P is the deductive closure of exactly one answer set of P.

In the sequel, given an ELP rule-observations program $P = (D, W)$, $\Delta(P)$ will denote the associated default theory $(\{\delta(r) \mid r \in D\}, W)$. Note that $\Delta(P)$ is disjunction-free. Similarly, given a disjunction-free default theory $\Delta =$

(D, W) such that for each $\delta \in D$, the consequent of δ is a literal and there is no conjunction in the justification of δ, $P(\Delta)$ will denote the associated rule-observations program $(\{r \mid \delta(r) \in D\}, W)$.

In order to state following complexity results, a technical Lemma is needed.

Lemma 4.9 *Let $P = (D, W)$ be an ELP rule-observations program and let $\Delta(P)$ be its associated DF default theory. Then L is an outlier set in P with witness set S iff L is an outlier set in $\Delta(P)$ with witness set S.*

Proof: By the relationship holding between the answer sets of the ELP P and the extensions of the default theory Δ^P stated in [26] (and recalled above), it follows that given an ELP rule-observations program $P = (D, W)$, for each subset Z of W the answer sets of P_Z are in one-to-one correspondence with the extensions of (D', W_Z) where $(D', W) = \Delta(P)$. $\square$

All that given, the complexity results for general ELPs can be summarized in the following theorem.

Theorem 4.10 *The complexities of outlier detection problems over ELP are as follows:*
- OUTLIER, OUTLIER[k], *and* OUTLIER(L) *are* Σ_2^P*-complete;*
- OUTLIER-MIN *is* D_2^P*-complete and* OUTLIER-MIN(S)(L) *is* Π_2^P*-complete, and*
- OUTLIER(S), OUTLIER[k](S), *and* OUTLIER(S)(L) *are* D^P*-complete.*

Proof: (Membership) Given an ELP rule-observations program $P = (D, W)$, by construction, $\Delta(P)$ is a disjunction-free default theory whose size is polynomially-bounded in the size of P. Thus, the claim follows from Lemma 4.9 and the membership parts of Theorems 4.1, 4.2, 4.3, 4.4, 4.5, 4.6, 4.7, and 4.8.

(Hardness) Given a normal DF default theory $\Delta = (D, W)$ such that for each $\delta \in D$, δ has the form $\frac{L_1 \wedge \ldots \wedge L_m : L_0}{L_0}$, $P(\Delta)$ is an ELP rule-observations program whose size is, by construction, polynomially bounded in the size of Δ. Hence, hardness follows from Lemma 4.9 and the hardness parts of Theorems 4.1, 4.2, 4.3, 4.4, 4.5, 4.6, 4.7, and 4.8 concerning DF default theories. Indeed, we recall that all the hardness parts of these theorems make use of a normal DF default theory such that the conclusion, and hence the justification, of each default rule occurring in it consists in a single literal. $\square$

4.3.2. *Complexity of Outlier Detection in Disjunctive Extended Logic Programs*

We have analyzed the complexity of outlier detection problems for extended logic programs by exploiting results obtained for default theories. However, the more general class of extended *disjunctive* logic programs cannot be mapped to the language of default theories. Therefore, we are not able to directly exploit the complexity results proved for default logics in order to derive correspondent results for EDLPs. Nevertheless, it turns out that outlier detection problems on EDLPs are precisely just as hard as the corresponding tasks evaluated on default logics, as summarized in the following theorem.

Theorem 4.11 *For general EDLPs, queries* (1) OUTLIER, (2) OUTLIER[k], *and* (3) OUTLIER(L) *are* Σ_3^{P}*-complete, queries* (4) OUTLIER(S), (5) OUTLIER[k](S), *and* (6) OUTLIER(S)(L) *are* $\mathrm{D}_2^{\mathrm{P}}$*-complete, query* (7) OUTLIER-MIN(L) *is* $\mathrm{D}_3^{\mathrm{P}}$*-complete, and query* (8) OUTLIER-MIN(S)(L) *is* Π_3^{P}*-complete.*

Proof Outline. (1) Membership in Σ_3^{P} follows since, given a rule-observation pair $P = (D, W)$, the query can be answered by a polynomial-time nondeterministic Turing machine that guesses the outlier set L and the witness set W and then employs a Σ_2^{P} oracle to decide $P_S \models \neg S$ (a Π_2^{P} problem) and $P_{S,L} \not\models \neg S$ (a Σ_2^{P} problem).
Σ_3^{P}-hardness is proved by reducing the problem of the validity of a $\mathrm{QBE}_{3,\exists}$ formula to query OUTLIER. The reduction associates a negation-free EDLP rule-observation pair $P(\Phi) = (D(\Phi), W(\Phi))$ with a $\mathrm{QBE}_{3,\exists}$ formula $\Phi = \exists X \forall Y \exists Z f(X, Y, Z)$ in conjunctive normal form. The properties of $P(\Phi)$ are analogous to the properties of the default theory $\Delta(\Phi)$ which were described when discussing query OUTLIER on default theories (see Theorem 4.1). However, differently from Theorem 4.1, a saturation technique is employed to guarantee those properties.

(2) *and* (3) The proof uses the same ideas illustrated for query OUTLIER.

(4) Membership in $\mathrm{D}_2^{\mathrm{P}}$ can be proved by taking into account that the problem corresponds to the conjunction of two independent problems, namely, deciding whether $P_S \models \neg S$ and whether $P_{S,L} \not\models \neg S$. While the former problem is in Π_2^{P}, it can be shown that the latter is in Σ_2^{P} even though L is not provided as input to the problem. Indeed, one can proceed by guessing together the outlier set L and a (minimal) model M of $P_{S,L}$ such that $\neg S \not\subseteq M$ and then checking that it is indeed minimal by exploiting an NP oracle.
D_2^P-hardness is proved by reducing to query OUTLIER(S) the problem of deciding whether a program P' is consistent (a Σ_2^{P} check) and a program P'' is inconsistent (a Π_2^{P} check).

(5) *and* (6) The results can be proved using the same line of reasoning as illustrated above for query OUTLIER(S).

(7) Membership is analogous to that of Theorem 4.7. As for the hardness, the line of reasoning is analogous to that of the hardness part of Point 1 of Theorem 4.7, but, from a technical point of view, the construction used in Point 1 of Theorem 4.11, where a Σ_3^P-complete problem is taken into account, is employed. Loosely speaking, two sets of rules, r' and r'', representing two distinct copies of the program used in the reduction shown in Point 1 of this Theorem, are used in order to build a logic rule-observations program $P(F)$ such that that the conjunction $F = \Phi \wedge \Psi$ of a $\mathrm{QBE}_{3,\exists}$ formula Φ (encoded via rules r') and of a $\mathrm{QBE}_{3,\forall}$ formula Ψ (encoded via rules r'') is valid if and only if a suitable set L is a minimal outlier set in $P(F)$.

(8) Both membership and hardness are analogous to that of Theorem 4.8 and the hardness exploits rules r'' used in Point 7 of this proof. □

5. The First-Order Case

We now discuss the extension of our framework to the first-order case, where variables are allowed to occur in defaults. We will use a formal framework which is similar in several aspects to the one set down by Cadoli et al. [15].

A finite *first-order default theory* $\Delta = (D, W)$ consists of a finite set W of first-order formulas and of a finite set D of default rules of the form (1), with the prerequisite, justifications, and consequence being first-order formulas with free variables among those in $\mathbf{X} = X_1, \ldots, X_n$. A default is *closed* if none of α, β_1, …, β_n, and γ contains free variables. A default theory is closed if all the formulas in W and in D are closed. A default or default theory which is not closed is called *open*.

It is assumed that the Herbrand Universe $\mathcal{U}$ of a finite default theory is nonempty and finite (hence, no function symbols are allowed to occur in the theory). In the following we will consider only finite first-order default theories.

The semantics of a closed first-order default theory is based on the notion of extension, whose definition is analogous to the definition of extension provided in Section 2.1 in the context of a propositional default theory [15]. The definition of extension is applied to open default theories by assuming that the defaults with free variables implicitly stand for the set of closed defaults obtained by replacing those free variables with terms of the Herbrand Universe $\mathcal{U}$ of the default theory.

Let $\phi(\mathbf{X})$ be a formula whose free variables are among $\mathbf{X} = X_1, \ldots, X_n$, and let $\zeta = \zeta_1, \ldots, \zeta_n$ be a list of objects from $\mathcal{U}$. Then, we denote by $\phi[\mathbf{X}/\zeta]$ the result of simultaneously substituting ζ_i for X_i in ϕ, for all $i = 1, \ldots, n$. Let $\Delta = (D, W)$ be a first-order default theory. We denote by $\mathrm{INST}(W)$ the *instantiation* of W, which is the set of closed formulas

$$\{\phi[\mathbf{X}/\zeta] \mid \phi(\mathbf{X}) \in W,\ \zeta \in \mathcal{U}^n\}.$$

Similarly, we denote by $\mathrm{INST}(D)$ the *instantiation* of D, which is the set of closed defaults

$$\left\{ \frac{\alpha[\mathbf{X}/\zeta] : \beta_1[\mathbf{X}/\zeta], \ldots, \beta_m[\mathbf{X}/\zeta]}{\gamma[\mathbf{X}/\zeta]} \mid \frac{\alpha(\mathbf{X}) : \beta_1(\mathbf{X}), \ldots, \beta_m(\mathbf{X})}{\gamma(\mathbf{X})} \in D,\ \zeta \in \mathcal{U}^n \right\},$$

and serves the purpose of eliminating free variables from the formulas.

The instantiation $\mathrm{INST}(\Delta)$ of Δ is $(\mathrm{INST}(D), \mathrm{INST}(W))$. For example, consider a group of friends who have to decide whether to go together to watch a movie or not, given that some of them do not like going to the movies. This is encoded in the theory $\Delta^{\mathrm{ex}} = (D, W)$, where

$$D = \left\{ \frac{(\exists X)(\neg likes(X, Y)) : \neg watch(Y)}{\neg watch(Y)} \right\}, \text{ and}$$
$$W = \{likes(mary, movie), \neg likes(jennie, movie), watch(movie)\}.$$

Then $\mathcal{U} = \{mary, jennie, movie\}$, and the instantiation $\mathrm{INST}(\Delta^{\mathrm{ex}})$ of Δ^{ex} is such that $\mathrm{INST}(W) = W$ and $\mathrm{INST}(D)$ is

$$\left\{ \frac{(\exists X)(\neg likes(X, mary)) : \neg watch(mary)}{\neg watch(mary)}, \frac{(\exists X)(\neg likes(X, jennie)) : \neg watch(jennie)}{\neg watch(jennie)}, \right.$$
$$\left. \frac{(\exists X)(\neg likes(X, movie)) : \neg watch(movie)}{\neg watch(movie)} \right\}.$$

Now we are in a position to extend the definition of outlier in the context of first-order default theories.

Definition 5.1 (First-Order Outliers and Outlier Witness Set) Let $\Delta = (D, W)$ be a first-order default theory and let $L \subseteq W$ be a set of *ground literals*. If there exists a non-empty set of *ground literals* $S \subseteq W_L$ such that:

(i) $(\mathrm{INST}(D), \mathrm{INST}(W)_S) \models \neg S$, and

(ii) $(\mathrm{INST}(D), \mathrm{INST}(W)_{S,L}) \not\models \neg S$

then we say that L is an *outlier* set in Δ and S is an *outlier witness* set for L in Δ.

For example, consider the theory Δ^{ex} above. Then $L = \{\neg likes(jennie, movie)\}$ is an outlier with the witness $S = \{watch(movie)\}$.

Given a finite default theory Δ, the instantiation of Δ contains only closed formulas *but it is not in general a ground theory*, due to the possible presence of quantifiers. A finite propositional default theory can be anyway associated with Δ as follows.

Let $F = \forall X \phi(X)$ ($F = \exists X \phi(X)$, resp.) be a universally (existentially, resp.) quantified formula. Then, the *propositional version* $\mathrm{PROP}(F)$ of F (under Domain Closure) is the formula $\bigwedge_{\zeta \in \mathcal{U}} \phi(\zeta)$ ($\bigvee_{\zeta \in \mathcal{U}} \phi(\zeta)$, resp.). The propositional version $\mathrm{PROP}(\phi)$ of a quantifier-free formula ϕ is the formula ϕ itself. Let ϕ be a formula, then *propositional version* $\mathrm{PROP}(\phi)$ of ϕ is obtained by recursively substituting each subformula ψ of ϕ with its propositional version $\mathrm{PROP}(\psi)$.

Let $\Delta = (D, W)$ be a first-order default theory. We denote by $\mathrm{PROP}(W)$ the *propositional version* of W, which is the set of propositional formulas

$$\{\mathrm{PROP}(\phi) \mid \phi \in \mathrm{INST}(W)\}.$$

Similarly, we denote by $\mathrm{INST}(D)$ the *propositional version* of D, which is the set of propositional defaults

$$\left\{ \frac{\mathrm{PROP}(\alpha) : \mathrm{PROP}(\beta_1), \ldots, \mathrm{PROP}(\beta_m)}{\mathrm{PROP}(\gamma)} \mid \frac{\alpha : \beta_1, \ldots, \beta_m}{\gamma} \in \mathrm{INST}(D) \right\}.$$

The *propositional version* $\mathrm{PROP}(\Delta)$ of Δ is the propositional default theory $(\mathrm{PROP}(D), \mathrm{PROP}(W))$. Since it has been assumed that the Herbrand Universe $\mathcal{U}$ of the theory Δ is finite, the propositional default theory $\mathrm{PROP}(\Delta)$ is finite. For example, consider the theory Δ^{ex} above. The propositional version $\mathrm{PROP}(\Delta^{\mathrm{ex}})$ of Δ^{ex} is such that $\mathrm{PROP}(W) = W$ and $\mathrm{PROP}(D)$ is

$$\left\{ \frac{\neg likes(mary, movie) \vee \neg likes(jennie, movie) \vee \neg likes(movie, movie) : \neg watch(movie)}{\neg watch(movie)}, \right.$$
$$\frac{\neg likes(mary, mary) \vee \neg likes(jennie, mary) \vee \neg likes(movie, mary) : \neg watch(mary)}{\neg watch(mary)},$$
$$\left. \frac{\neg likes(mary, jennie) \vee \neg likes(john, jennie) \vee \neg likes(movie, jennie) : \neg watch(jennie)}{\neg watch(jennie)} \right\}.$$

Let $\Delta = (D, W)$ be a first-order default theory, and let $S, L \subseteq W$ two disjoint subsets of ground literals. Then, by construction of $\mathrm{PROP}(\Delta)$, if L is an outlier with witness S in Δ then L in an outlier with witness S in the propositional default theory $\mathrm{PROP}(\Delta)$. The converse holds provided that outliers and witnesses are constrained to be subsets of $W \cap \mathrm{PROP}(W)$. [5] We have already remarked that constraining outliers and witnesses to be singled out from a given subset of set of literals in W does not change the complexity figures. Thus, if $\mathrm{PROP}(\Delta)$ is a finite propositional default theory, complexity results of Section 4.2 can be directly applied to the case in which the input theory is $\mathrm{PROP}(\Delta)$. Thus, the complexity analysis presented in Section 4.2 allows us to characterize the difficulty of the mining problem at hand once the propositional version of the first-order theory is available.

For example, consider the theory $\Delta^{\mathrm{ex}} = (D, W)$. Since $\mathrm{PROP}(W) = W$, the outliers in the propositional theory $\mathrm{PROP}(\Delta^{\mathrm{ex}})$ are in one-to-one correspondence with the outliers in the first-order theory Δ^{ex}.

As a matter of fact, it must be recalled that the theory $\mathrm{PROP}(\Delta)$ can be exponentially larger than Δ. And, indeed, it can be seen that, due to the exponential increase of the size of the theory, the complexity of deciding the existence of an outlier in a first-order theory is in $\mathrm{NEXPTIME}^{\Sigma_2^{\mathrm{P}}}$, which is the exponential analogue for $\Sigma_3^{\mathrm{P}} = \mathrm{NP}^{\Sigma_2^{\mathrm{P}}}$.

[5] From a technical point of view, the transformation $\mathrm{PROP}(\Delta)$ can be modified so that both directions hold immediately. To this aim it suffices to replace open (and, in the case there is a single constant in $\mathcal{U}$, also non-ground) literals ℓ in W with formulas of the form $f \vee \neg f \rightarrow \ell$.

Furthermore, as we have already pointed out, our approach is to regard outlier detection as a data mining technique, and therefore we mine from explicitly observed facts belonging to W. These facts can be very naturally regarded as tuples of a database from which we are interested in singling out anomalies. In the database scenario, a rather pertinent issue is to characterize the *data complexity* [58], i.e., the complexity of query evaluation when the database is assumed to vary, whereas the query expression is assumed to be fixed. Next, we are going to address this issue in the context of outlier detection in default theories scenario.

Let $\Delta = (D, W)$ be a first-order default theory. We denote by W^{fact} the *extensional component* of Δ, that is the subset of W consisting in all the ground literals in W. The extensional component can be assimilated to a relational database. The *first-order component* of Δ is, conversely, the set $W^{\mathrm{rule}} = W \setminus W^{\mathrm{fact}}$.

Definition 5.2 (Data Complexity of Outlier Detection Queries) *The* data complexity *of an outlier query Q, where Q is one of the queries defined in Section 4.1, when the knowledge base is a first-order default theory* $\Delta = (D, W^{\mathrm{rule}} \cup W^{\mathrm{fact}})$*, is the complexity of deciding Q measured in the size of the extensional component* W^{fact} *of* Δ.

In other words, the data complexity of Q is the complexity of answering Q on a first-order default theory, under the assumption that the set of default rules and the first-order component are held fixed and the only component allowed to vary is the extensional one.

Next we show that under data complexity measure, the query OUTLIER is Σ_3^{P}-complete, that is, it has the same complexity as that of its propositional counterpart.

Theorem 5.3 *The data complexity of* OUTLIER *is* Σ_3^{P}*-complete.*

Proof Outline. Consider membership. Given a first-order default theory $\Delta = (D, W^{\mathrm{rule}} \cup W^{\mathrm{fact}})$, let k be the maximum number of variables occurring in a default rule in D or in a formula in W^{rule}, and let n be the number of distinct elements in the Herbrand Universe $\mathcal{U}$ of Δ. The number n is at most linear in the size of the extensional component W^{fact} of Δ, that is $n = O(|W^{\mathrm{fact}}|)$. Hence, assuming that D and W^{rule} are not part of the input, the size of the theory $\mathrm{PROP}(\Delta)$ is $O(n^k)$, hence polynomial in the size of W^{fact} which is the input of the OUTLIER query. In order to complete the proof, it is sufficient to recall that the query OUTLIER for finite general propositional default theories is in Σ_3^{P}, as shown in the membership part of Theorem 4.1, Point 1.

As for hardness, it can be shown (see the appendix for details) that there exists a fixed set of default rules D^{FO} and a fixed set of non atomic formulas W^{FO}, together with a mapping which given as input a $\mathrm{QBE}_{3,\exists}$ formula Φ outputs a set $W(\Phi)$ of ground atoms, such that the theory $\mathrm{PROP}((\Delta^{\mathrm{FO}}, W^{\mathrm{FO}} \cup W(\Phi)))$ is equivalent to the propositional theory $\Delta(\Phi)$ described in the hardness part of Theorem 4.1, Point 1. The rest of the proof then follows from the above mentioned theorem. Details of the reduction are reported in the Appendix. □

The construction described in the proof of Theorem 5.3 can be used to adapt the other reductions depicted in the hardness part of theorems concerning general propositional default theories in order to obtain reductions valid under the data complexity measure. Therefore, we obtain:

Theorem 5.4 *The data complexity of* (i) OUTLIER *is* Σ_3^P*-complete;* (ii) OUTLIER[k] *and* OUTLIER(L) *is* Σ_3^P*-complete;* (iii) OUTLIER(S), OUTLIER[k](S), *and* OUTLIER(S)(L) *is* D_2^P*-complete;* (iv) OUTLIER-MIN(L) *is* $\mathrm{D}_3^{\mathrm{P}}$*-complete, and* (v) OUTLIER-MIN(S)(L) *is* Π_3^{P}*-complete.*

By construction, both Theorems 5.3 and 5.4 hold for finite normal first-order default theories.

6. Related Work

Research related to the work presented in this paper can be divided into three groups: (i) abduction, (ii) outlier detection from data, and (iii) outlier detection using logic programming under stable model semantics.

6.1. *Abduction*

The research on logic-based abduction [47,16,23,53,22,19,32,52,39] is related to outlier detection. Generally speaking, in the framework of logic-based abduction, the domain knowledge is described using a logical theory T. A subset X of hypotheses is an abduction explanation to a set of manifestations M if $T \cup X$ is a consistent theory that entails M.

The work by Eiter, Gottlob, and Leone on abduction from default theories [24] is very relevant to the work presented here. In that paper, the authors present a basic model of abduction from default logic and analyze the complexity of some associated abductive reasoning tasks. They also present two modes of abduction: one based on brave reasoning and the other on cautious reasoning. According to [24], a *default abduction problem* (DAP) is a tuple $\langle H, M, W, D\rangle$ where H is a set of ground literals called *hypotheses*, M is a set of ground literals called *observations*, and (D, W) is a default theory. The goal is to explain observations from M by using hypotheses in the context of the default theory (D, W). The authors propose the following definition for an explanation:

Definition 6.1 ([24]) *Let $P = \langle H, M, D, W\rangle$ be a DAP and let $E \subseteq H$. Then, E is a* skeptical explanation *for P iff*
- *(i) $(D, W \cup E) \models M$, and*
- *(ii) $(D, W \cup E)$ has a consistent extension.*

The relationship between outlier detection on *normal* propositional default theories and skeptical explanations is summarized by the following theorem.

Theorem 6.2 *Let $\Delta = (D, W)$ be a normal default theory, where W is consistent. Also, let $L \subseteq W$ and $S \subseteq W$ be two disjoint sets. Then S is an outlier witness set for L in Δ if and only if L is a minimal nonempty skeptical explanation for $\neg S$ in the DAP $P = \langle L, \neg S, D, W_{S,L}\rangle$.*

Proof:

(i) ("Only If") Let $\Delta = (D, W)$ be a normal default theory, $L \subseteq W$, and $S \subseteq W$ an outlier witness set for L. By our definition of outlier, it must be the case that $(D, W_S) \models \neg S$, or in other words, $(D, W_{S,L} \cup L) \models \neg S$. Moreover, since (D, W) is a normal default theory, so is $(D, W_{S,L} \cup L)$. In addition, since W is consistent, so is W_S. Hence, (D, W_S) has a consistent extension. Therefore, by Definition 6.1, L is a skeptical explanation for $\neg S$ in the DAP P.

(ii) ("If") Suppose L is a minimal nonempty skeptical explanation for $\neg S$ in the DAP $P = \langle L, \neg S, D, W_{S,L}\rangle$. By definition, we have:
  (a) $(D, W_S) \models \neg S$, and
  (b) (D, W_S) has a consistent extension.

  Moreover, since L is a minimal nonempty explanation, at least one of the following must be true:
  (a) $(D, W_{S,L}) \not\models \neg S$, or
  (b) $(D, W_{S,L})$ does not have a consistent extension.

  Since $\Delta = (D, W)$ is a normal default theory and W is a consistent theory, it must be the case that $\Delta = (D, W_{S,L})$ is also a normal default theory and $W_{S,L}$ is consistent. Hence, the default theory $(D, W_{S,L})$ has a consistent extension. Therefore it must be the case that $(D, W_{S,L}) \not\models \neg S$, and it can be concluded that S is an outlier witness set for L in (D, W).

$\square$

In sum, it follows that some sort of duality does hold in the context of normal default theories between abduction and outlier detection problems. Nonetheless, in outlier detection problems, the outlier witness set S (which according to Theorem 6.2 is the analog to the set of observations in abduction problems) has to be guessed, while the set of observations in abduction is given in the input. We have shown in Section 4 that the high complexity of the outlier detection problems arises from the fact that the set S is not given in advance. Hence, the fact that the witness set S is not given in input is quite significant from the computational complexity point of view and prevents us from borrowing complexity results from abduction problems.

Remark 6.3 *Theorem 6.2 is valid also for ordered* semi-normal *default theories. This is because the second condition in Definition 6.1 requires a default theory which has at least one consistent extension, and ordered semi-normal default theories are guaranteed to have this property (see [25]). For other subclasses of default theories, however, Theorem 6.2 might not hold. Consider for example the following default theory $\Delta = (D, W)$, where $D = \{\frac{l:\neg s}{\neg s}, \frac{\neg s:\neg q\wedge p}{p}, \frac{\neg s:\neg p\wedge l}{l}, \frac{\neg s:\neg l\wedge q}{q}\}$ and $W = \{l, s\}$. This default theory is semi-normal but not ordered. The reader can verify that $\{l\}$ is an outlier set and $\{s\}$ is its witness set. However, since $(D, W_{\{s\}})$ does not have a consistent extension, we cannot say that $\{l\}$ is a minimal nonempty skeptical explanation for $\neg\{s\}$ in the DAP $P = \langle\{l\}, \neg\{s\}, D, W_{\{s\},\{l\}}\rangle$.*

Research on logic-based abduction from disjunctive logic programs [22,53] is also related to outlier detection, and has comprehensibly been studied in the context of logic programming (see [19] for a survey). This issue has been explored in two directions. The first line of work has used logic programs as an AI tool for knowledge representation and reasoning about abduction, while the second approach has used the concept of abduction for defining the semantics of logic programs. In the context of disjunctive logic programming, research has focused on the relationship between semantics of DLP and abduction-based semantics of logic programs (see, for example, [53]).

Eiter et al. [22] have studied abduction in the context of *normal* and *disjunctive* logic programs that have only one type of negation, namely, negation by failure, but have not considered extended logic programs where both classical negation and negation by failure are allowed. In addition, unlike our framework, the model of Eiter et al. does not allow for an explicit exploitation of integrity constraints. Their abduction schema assumes that the inference operator is provided as an input. Eiter et al. define a *logic programming abduction problem* as follows:

Definition 6.4 ([22]) *Let V be a set of propositional atoms. A logic programming abduction problem* (LPAP) *P over V consists of a tuple $\langle H, M, LP, \models \rangle$, where $H \subseteq V$ is a finite set of hypotheses, $M \subseteq V \cup \{\neg v \mid v \in V\}$ is a finite set of manifestations, LP is a propositional logic program on V and $\models$ is an inference operator.*

They define a *solution* to an *LPAP* as:

Definition 6.5 ([22]) *Let P=$\langle H, M, LP, \models \rangle$ be an LPAP, and let $S \subseteq H$. Then S is a* solution *(or* explanation*) to P iff $LP \cup S \models M$.*

According to [22], abductive conclusions should not lead to inconsistency. Hence, they use a variant of skeptical inference in which some answer set must exist. We cannot establish a formal relationship between outlier detection and abduction on disjunctive logic programs because the definition of outliers requires the ability to prove negative literals in the classical negation sense, while in the semantics used in [22], one can never prove negative literals since the program itself does not use any negative literal.

Sakama and Inoue have defined abduction in the context of EDLPs, and suggested a program transformation between "abductive programs" and disjunctive programs [53]. However, their techniques and complexity results do not apply to the case of outlier detection since they have investigated *credulous* reasoning rather than skeptical reasoning, and since in their framework an observation is constrained to be a single literal.

However, if we use the framework of Eiter et al. as described in Definitions 6.4 and 6.5, and adapt it to a "skeptical" version of the work of Sakama and Inoue, we can show the following.

Theorem 6.6 *Let $P = (D, W)$ be a rule-observations program and let $L, S \subseteq W$. Then the following holds:*

(i) If S is an outlier witness set for L in P, then L is an explanation in the LPAP $A = \langle W, \neg S, P' = (D, W_{S,L}), \models \rangle$, where $\models$ is entailment from EDLPs as defined in Section 2.2, and

(ii) If L is a minimal explanation in the LPAP $A = \langle W, \neg S, P' = (D, W_{S,L}), \models \rangle$, where $\models$ is entailment from EDLPs as defined in Section 2.2, then S is an outlier witness set for L in P.

Proof:

(i) Let $P = (D, W)$ be a rule-observations program and let $L, S \subseteq W$. Assume S is an outlier witness set for L in P. By definition of outlier, it must be the case that $(D, W_{S,L} \cup L) \models \neg S$. Then, by Definition 6.5, L is an explanation for $\neg S$ in the LPAP A.

(ii) Suppose L is a minimal explanation for $\neg S$ in the LPAP $A = \langle W, \neg S, P' = (D, W_{\{S,L\}}), \models \rangle$. By Definition 6.5, it is known that:

$$(D, W_{S,L} \cup L) \models \neg S$$

Therefore $(D, W_S) \models \neg S$. Moreover, since L is a *minimal* explanation, the following must be true:

$$(D, W_{S,L}) \not\models \neg S$$

Hence, it can be concluded that S is an outlier witness set for L in (D, W).

□

As with default logic, there is a clear difference between those two frameworks. The construction given in the proof of Theorem 6.2 and 6.6 does not provide a technique to solve outlier detection problems using abduction, since for outlier detection both the outlier L and its outlier witness set S have to be singled out, while in abduction both hypotheses and observations are fixed sets. In fact, outlier detection is a knowledge discovery technique: the task in outlier detection is to learn the exceptional observations along with the information witnessing for it.

6.2. *Outlier detection from data*

The vast body of literature concerning outlier detection in databases largely exploits techniques borrowed from statistics, machine learning and other fields [9,28,57]. In almost all cases, the techniques deal with data organized as a single relational table. Often only numerical attributes are handled and a metrics relating pairs of rows in the table are first required. These approaches can be classified as *supervised*-learning methods, where each example must be labeled as exceptional or not [37,51], and *unsupervised*-learning methods, where such labels are not required. The latter approach is obviously more general. As the technique proposed in our work is unsupervised, the sequel of this section will focus on unsupervised techniques. These techniques can be categorized into various groups.

Statistical-based methods, which assume that the given data set has a distribution model. Outliers, then, are those objects that satisfy a discordancy test, that is, that are significantly larger (or smaller) w.r.t. the values they are supposed to assume according to the hypothesized distribution [9].

Deviation-based techniques identify outliers by inspecting the typical characteristics of objects and defines them as objects that deviate from those features [8,54].

A rather different technique, which finds outliers by observing *low dimensional projections* of the search space, is presented in [1]. In that paper, a point is considered an outlier if it is located in some low density subspace.

Yu et al.[17] introduced a method based on *wavelet transform*, that identifies outliers by removing clusters from the original data set. Wavelet transform has also been used in [56] to detect outliers in stochastic processes.

A further group of methods use *density-based* techniques [14] and exploit a notion of *locality* that measures the plausibility for an object to be an outlier with respect to the density of the local neighborhood. To reduce the computational load, Jin et al. [30] proposed a method to determine only the top-n local outliers.

Distance-based outlier detection was introduced by Knorr and Ng [34,35] to overcome the limitations of statistical methods. A *distance-based* outlier is defined as follows: *A point p in a data set is an outlier with respect to parameters k and R if at least k points in the data set lie at a distance greater than R from p.* This definition generalizes the definition of outlier in statistics and is appropriate when the data set does not fit any standard distribution. Ramaswamy et al.[48] modified the above definition of outlier. They do not provide any ranking for outliers that are singled out. The definition they suggest is based on the distance of the k-th nearest neighbor of a point p, denoted by $D^k(p)$, and proceeds as follows: *Given k and n, a point p is an outlier if no more than $n-1$ other points q in the data set have a higher value for $D^k(q)$ than p.* This means that the points q having the n greatest $D^k(q)$ values are singled out as outliers.

A definition of outlier that considers for each point the sum of the distances from its k nearest neighbors is proposed in [6,7,2]. The authors present an algorithm that uses the Hilbert space-filling curve which exhibits scaling results close to linear. Similarly, a near-linear time algorithm for detection of distance-based outliers exploiting randomization is described in [10].

The general differences and analogies between the approaches described above and the one suggested in the present work are significant. In fact, those approaches deal with "knowledge," as encoded within one single relational table. In contrast, our technique deals with complex knowledge bases, which though comprising relational-like information, generally also include semantically richer forms of knowledge, such as axioms, default rules and so forth. Hence, in the framework analyzed in this paper, complex relations relating objects of the underlying theory can be expressed. As a consequence, even if the intuitive and general sense of computing outliers in the two contexts is analogous, the conceptual and technical developments are quite different as well as the formal properties of the computed outliers.

Sometimes domain knowledge can help to single out outliers that would otherwise be difficult to identify via methods like the ones surveyed above. The following example is intended to provide some intuition about this. The example also serves to highlight the different types of knowledge that can be mined using our approach as opposed to these others. For this purpose, we make a comparison between our approach and a typical distance-based approach. To facilitate the comparison, we will use an example where literals included in the evidential knowledge denote facts concerning integer numbers.

Example 6.7 Let $I = \{0, 1, 2, \ldots, 99, 100\}$. Consider a binary predicate $p(x, y)$, which normally is used to represent pairs $(x, y) \in I^2$ such that (a) $x + y = 100$ and (b) $x \neq y$, and assume the following set of observations is available:

$$DB = \{p(0, 100), p(1, 99), p(2, 98), \ldots, p(49, 51), p(50, 50), p(51, 49), \ldots, p(98, 2), p(99, 1), p(100, 0)\}.$$

According to the knowledge informally stated above, the literal $p(50, 50)$ is associated with an anomaly in DB, since the pair of integers $x = 50$ and $y = 50$ satisfies condition (a) but not condition (b).

Now, suppose one wants to single out anomalous observations in DB. In that case, in the absence of domain knowledge, an unsupervised data mining technique could be used to mine outliers in DB. For example, the *distance-based outlier* definition given in [34] would be suitable for this purpose. Assume that the Euclidean distance is employed and that parameters k and R are set to $k = 3$ and $R = \sqrt{2}$ (which is the distance separating each point (x, y) of DB from its nearest neighbor (x', y') in DB). According to these parameters, there are two outliers in DB, namely, $(0, 100)$ and $(100, 0)$. These two points are precisely the two extremes of the distribution associated with points in DB.

Notice that, according to the distance-based definition, point $(50, 50)$ is the worst candidate to represent an outlier in DB, since it is, in fact, the centermost point of DB. And indeed, for each combination of values for the parameter k and R, it holds that the point $(50, 50)$ is a distance-based outlier in DB if and only if all the points in DB are distance-based outliers. A similar situation would characterize the other methods surveyed above in this subsection.

Assume, conversely, that the domain knowledge informally stated above is available as encoded in a default theory $\Delta = (D, W)$ such that

$$W = DB \cup \{s(x, y, x + y) \mid x, y \in I\} \cup \{e(x, x) \mid x \in I\}, \text{ and}$$
$$D = \left\{ \frac{\neg s(x, y, 100) \vee e(x, y) : \neg p(x, y)}{\neg p(x, y)} \mid x, y \in I \right\} \cup \left\{ \frac{: \neg s(x, y, z)}{\neg s(x, y, z)} \mid x, y, z \in I \right\},$$

where the ternary predicate $s(x, y, z)$ represents the sum ($s(x, y, z)$ is in W if $z = x + y$) and the binary predicate $e(x, y)$ represents the equality ($e(x, y)$ is in W if $x = y$). By exploiting Definition 3.2, the abnormality perceived in the knowledge encoded by theory Δ can be singled out. Indeed
$(D, W_{\{p(50,50)\}}) \models \neg p(50, 50)$, and
$(D, W_{\{p(50,50)\},\{e(50,50)\}}) \not\models \neg p(50, 50)$,
and the set $S = \{p(50, 50)\}$ is an outlier witness set for the outlier set $L = \{e(50, 50)\}$. Loosely speaking, $\{e(50, 50)\}$ being an outlier can be interpreted as the number 50 behaving abnormally, while $\{p(50, 50)\}$ being a witness can be interpreted as the explanation of its abnormality, that is that the point $(50, 50)$ does not satisfy both conditions (a) and (b) but is, nonetheless, included in the set of observations DB concerning the predicate p. $\square$
Before concluding, we want to emphasize that the framework developed in this paper relies heavily on the concept of default rules. The default rules describe what is normally expected, and hence can be exploited for detecting observations which are not normal according to the default theory at hand. Such default rules are not given in advance in the frameworks discussed in this section. Furthermore, along with outliers, we single out their "witnesses" – those unexpected properties that characterize outliers. As far as we know, no such justifications for outliers are provided by the approaches described above.

6.3. *Outlier Detection using Stable Model Semantics*

We have originally presented outlier detection in the context of default logics [4]. Outlier detection was successively studied in the context of non-disjunctive logic programs under stable model semantics in [5]. Next, we recall the notion of stable model and the definition of outlier provided in [5].

A propositional *logic program* (LP, for short) is a collection of classical-negation-free non-disjunctive propositional rules. Clearly, from a syntactic viewpoint, LPs form a subset of ELPs (precisely those consisting of all classical-negation-free ELPs).

The stable model semantics of a LP P assigns to P the set of its *stable models* $\mathrm{SM}(P)$, that corresponds to the set of the answer sets of P.

Let S be a set of propositional letters. Then, program P *entails* S (resp. $\neg S$), denoted by $P \models S$ (resp. $P \models \neg S$), if *for each* model $M \in \mathrm{SM}(P)$ and for each letter L in S, $L \in M$ (resp. $L \notin M$).

So, loosely speaking, logic programs rely on the *closed world assumption* (CWA for short), which states that everything which is not explicitly inferred is false. Extended logic programs, on the other hand, relay on *open world assumption* (OWA for short), which states that only what can be explicitly inferred is true, while all the rest is unknown.

In the following, unless it is clear from the context, in order to differentiate the operator defined on logic programs (relying on the CWA) from the operator defined on extended logic programs (relying, vice versa, on the OWA), we will denote the former by $\models^{\mathrm{cwa}}$ and the latter by $\models^{\mathrm{owa}}$.

A LP rule-observation pair $P = (D, W)$ is defined in [5] analogously to an ELP rule-observation pair, but D is a logic program and W is a set of letters (positive literals). Next, the definition of outlier in the context of logic programs under stable model semantics is recalled.

Definition 6.8 (given in [5]) Let $P = (D, W)$ be a LP rule-observations program, and let $L \subseteq W$ be a set of literals. If there exists a non-empty set of literals $S \subseteq W_L$ such that:

(i) $P_S \models^{\rm cwa} \neg S$, and

(ii) $P_{S,L} \not\models^{\rm cwa} \neg S$

then we say that L is an *outlier set* and S is an *outlier witness* set for L in P.

As already noted in the introduction, in [5] a weaker assumption is adopted to define outliers. Indeed, normal logic programs under stable model semantics do not allow classical negation, but only negation as failure. As a consequence, in [5], the negation of the witness is not required to be explicitly inferred but, rather, that the witness *is not entailed* by the logic program. This is in contrast to the view adopted in the present paper, by which the witnessing set for an outlier is a property that is explicitly observed and opposite to that which is expected. Hence, the approach of [5] singles out anomalies of a different nature.

These differences can be substantiated from a formal point of view by showing that Definitions 3.2 and 6.8 cannot be reduced one to the other by means of natural program transformations, which are usually employed in order to prove that normal logic programs are semantically equivalent to extended logic programs [6] [26]. Given an (extended) logic program-observation pair P, we investigate two program transformations: the *positive form* P^+ of P which, informally speaking, is the LP representing the counterpart of P under the stable model semantics, and the *two-valued form* P^- of P, formally defined next, which, loosely speaking, is the ELP representing the closed-world interpretation of P.

Definition 6.9 *Let $P = (D, W)$ be an ELP rule-observation pair and let $\mathcal{L}$ be the set of the propositional letters occurring in P. For each $\ell \in \mathcal{L}$, let $\ell^\neg$ denote a novel propositional letter. Let $L \in \mathcal{L}$ be a positive literal ℓ (resp., negative literal $\neg\ell$), then by L^+ we denote the propositional letter ℓ (resp., the propositional letter $\ell^\neg$). Let S be a subset of $\mathcal{L} \cup \neg\mathcal{L}$. Then by S^+ we denote the set $\{L^+ \mid L \in S\}$. For each rule $r = L_1 \leftarrow L_2, \ldots, L_m, not\ L_{m+1}, \ldots, not\ L_n$, let r^+ denote the rule $L_1^+ \leftarrow L_2^+, \ldots, L_m^+, not\ L_{m+1}^+, \ldots, not\ L_n^+$. The* positive form *$P^+$ of P is the LP rule-observation pair $P^+ = (D^+, W^+)$ such that $D^+ = \{r^+ \mid r \in D\} \cup \{C \leftarrow \ell, \ell^\neg \mid \ell \in \mathcal{L}\} \cup \{\ell \leftarrow C \mid \ell \in \mathcal{L}\} \cup \{\ell^\neg \leftarrow C \mid \ell \in \mathcal{L}\}$, where C is a new letter. The last three sets are introduced in order to assure that whenever a literal and its negation belong to an answer set, then the answer set is forced to include all the literals appearing in the logic program.*

By the results of [26], it holds that M is a consistent (resp., inconsistent) answer set for P if and only if M^+ ($M^+ \cup \{C\}$, resp.) is a stable model for P^+.

The definition of a two-valued form of an extended logic program (ELP) follows.

Definition 6.10 *Let $P = (D, W)$ be an ELP rule-observation pair and let $\mathcal{L}$ be the set of the propositional letters occurring in P. The* two-valued form *P^- of P is the ELP rule-observation pair $P^- = (D^-, W^-)$ such that $D^- = D \cup \{\neg p \leftarrow not\ p \mid p \in \mathcal{L}\}$ and $W^- = W$.*

Given an ELP (LP, resp.) rule-observation pair P, we denote by $\Sigma^{\rm owa}(P)$ ($\Sigma^{\rm cwa}(P)$, resp.) the set of pairs (S, L) such that L is an outlier set in P and S is an outlier witness set for L in P according to Definition 3.2 (6.8, resp.).

We show that the following three properties hold.

- OP1: Let P be an ELP rule-observation pair, $(S, L) \in \Sigma^{\rm owa}(P) \not\Rightarrow (S^+, L^+) \in \Sigma^{\rm cwa}(P^+)$ and $(S^+, L^+) \in \Sigma^{\rm cwa}(P^+) \not\Rightarrow (S, L) \in \Sigma^{\rm owa}(P)$,
- OP2: Let P be an ELP rule-observation pair, $(S, L) \in \Sigma^{\rm owa}(P) \not\Rightarrow (S, L) \in \Sigma^{\rm owa}(P^-)$ and $(S, L) \in \Sigma^{\rm owa}(P^-) \not\Rightarrow (S, L) \in \Sigma^{\rm owa}(P)$,
- OP3: Let P be a LP rule-observation pair, $(S, L) \in \Sigma^{\rm cwa}(P) \Leftrightarrow (S, L) \in \Sigma^{\rm owa}(P^-)$.

OP1 states that the outliers according to Definition 3.2 in a generic ELP program-observation pair P are incomparable with the outliers according to Definition 6.8 in the positive form P^+ of P; OP2 states that the outliers according to Definition 3.2 in a generic ELP program-observation pair P are incomparable with the outliers according to Definition 3.2 in the two-valued form P^- of P; and, finally, OP3 states that Definition 3.2 can be used to single out the outliers in a LP rule-observation pair P according to Definition 6.8, by means of the transformation P^-.

It follows from these properties that the framework for mining outliers described here is more general than that described in [5].

OP1, OP2 and OP3 are proved next.

[6] We are grateful to an anonymous referee for suggesting this line of comparison.

Theorem 6.11 (OP1) *Let P be an* ELP *rule-observation pair. Then* $(S, L) \in \Sigma^{\text{owa}}(P) \not\Rightarrow (S^+, L^+) \in \Sigma^{\text{cwa}}(P^+)$ *and* $(S^+, L^+) \in \Sigma^{\text{cwa}}(P^+) \not\Rightarrow (S, L) \in \Sigma^{\text{owa}}(P)$.

Next we prove the above statement by providing two counterexamples. Let us start with the first implication. Consider the ELP pair $\widehat{P} = (D, W)$, where D includes the only rule

$$\neg p \leftarrow not\ p, q.$$

and W is the set $\{p, q\}$. Then, $\widehat{P}^+ = (D^+, W^+)$ is such that D^+ is

$$p^\neg \leftarrow not\ p, q. \quad C \leftarrow p, p^\neg. \quad C \leftarrow q, q^\neg. \quad p \leftarrow C. \quad p^\neg \leftarrow C. \quad q \leftarrow C. \quad q^\neg \leftarrow C.$$

and W^+ is $\{p, q\}$. Consider the outlier $L = \{q\}$ in $\widehat{P}$ with the associated witness $S = \{p\}$. It is clear that L^+ and S^+ do not form an outlier-witness pair in $\widehat{P}^+$. Indeed, while $\widehat{P}^+_{\{p\}} \models^{\text{cwa}} \neg p$, since p does not belong to the unique stable model $M_1 = \{p^\neg, q\}$ of $\widehat{P}^+_{\{p\}}$, the unique stable model M_2 of the latter logic program is the empty set, and hence $\widehat{P}^+_{\{p\},\{q\}} \models^{\text{cwa}} \neg p$.

As for the second implication in OP1, consider the ELP pair $\widetilde{P} = (D, W)$, where D includes only the rule

$$p \leftarrow not\ q.$$

and W is the set $\{p, q\}$. Then $\widetilde{P}^+ = (D^+, W^+)$ is such that D^+ is

$$p \leftarrow not\ q. \quad C \leftarrow p, p^\neg. \quad C \leftarrow q, q^\neg. \quad p \leftarrow C. \quad p^\neg \leftarrow C. \quad q \leftarrow C. \quad q^\neg \leftarrow C.$$

and W^+ is $\{p, q\}$. Consider the outlier $L^+ = \{q\}$ in $\widetilde{P}^+$ with the associated witness $S^+ = \{p\}$. It is clear that L and S do not form an outlier-witness pair in P. Indeed, $\widetilde{P}_{\{p\}} \not\models^{\text{owa}} \neg p$, since the unique answer set of the ELP $\widetilde{P}_{\{p\}}$ is $\{q\}$.

Theorem 6.12 (OP2) *Let P be an* ELP *rule-observation pair. Then* $(S, L) \in \Sigma^{\text{owa}}(P) \not\Rightarrow (S, L) \in \Sigma^{\text{owa}}(P^-)$ *and* $(S, L) \in \Sigma^{\text{owa}}(P^-) \not\Rightarrow (S, L) \in \Sigma^{\text{owa}}(P)$.

Next we prove the above statement by providing two counterexamples. Let us start with the first implication in OP2. Consider again the ELP pair $\widehat{P}$ of the preceding example, and the corresponding two-valued form $\widehat{P}^- = (D^-, W^-)$, where D^- is

$$\neg p \leftarrow not\ p, q. \quad \neg p \leftarrow not\ p. \quad \neg q \leftarrow not\ q.$$

and W^- is $\{p, q\}$. Consider the outlier $L = \{q\}$ in $\widehat{P}$ with the associated witness $S = \{p\}$. Also in this case, L and S do not form an outlier-witness pair in $\widehat{P}^-$. Indeed, while $\widehat{P}^-_{\{p\}} \models^{\text{owa}} \neg p$, the ELP $\widehat{P}^-_{\{p\},\{q\}}$ has the unique answer set $\{\neg p, \neg q\}$, and hence $\widehat{P}^-_{\{p\},\{q\}} \models^{\text{owa}} \neg p$.

As for the second implication in OP2, consider again the ELP pair $\widetilde{P}$ of the preceding example, and the corresponding two-valued form $\widetilde{P}^- = (D^-, W^-)$, where D^- is

$$p \leftarrow not\ q. \quad \neg p \leftarrow not\ p. \quad \neg q \leftarrow not\ q.$$

and W^- is $\{p, q\}$. Consider the outlier $L = \{q\}$ in $\widetilde{P}^-$ with the associated witness $S = \{p\}$. Then L and S do not form an outlier-witness pair in $\widetilde{P}$. Indeed, the unique answer set of $\widetilde{P}_{\{p\}}$ is $\{q\}$ and hence $\widetilde{P}_{\{p\}} \not\models^{\text{owa}} \neg p$.

Next we provide an intuition as to why those implications cited in Theorems 6.11 and 6.13 actually fail to hold.

As for the positive form transformation (see Theorem 6.11), consider condition (ii) of Definition 3.2:

$$P_{S,L} \not\models^{\text{owa}} \neg S \equiv (\exists M \in \text{ANSW}(P_{S,L}))(\exists s \in S)(\neg s \notin M).$$

Following results from [26] and assuming consistency, condition (ii) of Definition 6.8 can be formulated as:

$$P^+_{S^+,L^+} \not\models^{\text{cwa}} \neg S^+ \equiv (\exists M \in \text{ANSW}(P_{S,L}))(\exists s \in S^+)(s \in M^+).$$

Clearly, the first condition does not imply in general the second one since it might exist an answer set M such that both $\neg s$ and s do not occur in M. Consequently, $(S, L) \in \Sigma^{\text{owa}}(P) \not\Rightarrow (S^+, L^+) \in \Sigma^{\text{cwa}}(P^+)$. For the inverse implication a similar line of reasoning can be followed.

As for the two-valued form transformation (see Theorem 6.13), consider again condition (ii) of Definition 3.2. Clearly, this condition does not exclude that there exists an answer set M such that $s \notin M$. Therefore, since in the program P^- there is the rule $\neg s \leftarrow not\ s$, it might be the case that M is associated with an answer set M^- of P^- such that $\neg s \in M^-$. Consequently $(S, L) \in \Sigma^{\rm owa}(P) \not\Rightarrow (S, L) \in \Sigma^{\rm owa}(P^-)$. For the inverse implication a similar line of reasoning can be drawn.

Conversely, a faithful correspondence holds for $\Sigma^{\rm cwa}(P)$ and $\Sigma^{\rm owa}(P^-)$, as shown next.

Theorem 6.13 (OP3) *Let P be a* LP *rule-observation pair. Then* $(S, L) \in \Sigma^{\rm cwa}(P) \Leftrightarrow (S, L) \in \Sigma^{\rm owa}(P^-)$.

Proof: It is well-known that if M is a stable model of a LP P, then

$$M^- = M \cup \{\neg\ell \mid \ell \in (\mathcal{L} \setminus M)\},$$

where $\mathcal{L}$ is the set of all the propositional letters occurring in P, is an answer set of P^- [26]. Moreover, every answer set of P^- can be represented in the form shown above, where M is a stable model of P. Thus,

$$P_S \models^{\rm cwa} \neg S \Leftrightarrow (\forall M \in {\rm SM}(P_S))(s \notin M) \Leftrightarrow (\forall M^- \in {\rm ANSW}(P_S^-))(\neg s \in M^-) \Leftrightarrow P_S^- \models^{\rm owa} \neg S, \text{ and}$$
$$P_{S,L} \not\models^{\rm cwa} \neg S \Leftrightarrow (\exists M \in {\rm SM}(P_{S,L}))(s \in M) \Leftrightarrow (\exists M^- \in {\rm ANSW}(P_{S,L}^-))(\neg s \notin M^-) \Leftrightarrow P_{S,L}^- \not\models^{\rm owa} \neg S.$$

This completes the proof. □

For example, consider the logic program $P = (D, W)$, where D consists of the only rule

$q \leftarrow m, not\ e.$

and W is $\{q, m, e\}$. The ELP program $P^- = (D^-, W^-)$ is such that D^- is

$q \leftarrow m, not\ e. \quad \neg q \leftarrow not\ q. \quad \neg m \leftarrow not\ m. \quad \neg e \leftarrow not\ e.$

and W^- is $\{p, m, e\}$. Consider the outlier $L = \{e\}$ in P with the associated witness set $\{q\}$. L and P form an outlier-witness pair also in P^-. Indeed it can be verified that $P^-_{\{q\}} \models^{\rm owa} \neg q$, due to the rule $\neg q \leftarrow not\ q$, and also that $P^-_{\{q\},\{e\}} \not\models^{\rm owa} \neg q$, since q belongs to the unique answer set of this program and consequently $\neg q$ does not belong to it.

The following example helps in further clarifying the dissimilarities between the two frameworks. Consider a railroad crossing scenario, and assume that the current state of the world viewed by Agent A is modeled by the proposition red_light, which represents the knowledge that the semaphore located near the railroad is red, and $cross(B)$, representing the knowledge that Agent B is passing through the railroad track. The knowledge base of Agent A supposedly has a rule asserting that normally the railroad should not be crossed when the light is red. Let us consider two formalisms for representing the knowledge base. First, assume that the knowledge of Agent A is encoded in a default theory, consisting of the single default rule $\frac{red\_light:\neg cross(x)}{\neg cross(x)}$ (or, equivalently, by the ELP rule under the answer set semantics $\neg cross(x) \leftarrow not\ cross(x), red\_light$). Equipped with this knowledge, Agent A might conclude that something is wrong, since $\{cross(B)\}$ is a witness for the outlier $\{red\_light\}$, in that removing $cross(B)$ from the current state of the world explicitly derives the exact opposite, i.e. $\neg cross(B)$, while removing both $cross(B)$ and red_light does not entail $\neg cross(B)$. To conclude it is worth pointing out that the rule $\neg cross(x) \leftarrow not\ cross(x), red\_light$ plays the role of program $\widehat{P}$ introduced in one of the counterexamples used to prove Theorem 6.11.

Second, assume that instead of being encoded in default logic, the knowledge of Agent A is encoded in a logic program under stable model semantics. In this case, Agent A must rely on a kind of knowledge which is somewhat different and weaker than the knowledge encoded by the default rule above. Indeed, since there is no explicit negation in this formalism, a logic programming rule like $cross(x) \leftarrow not\ red\_light$ may be used. This rule models the somewhat hazardous decision: "if it is unknown that the semaphore light is red, then the railroad can be crossed". According to the work of [5], in this situation we will also conclude that red_light is an outlier and $cross(B)$ is its witness. This is because after removing $cross(B)$ from the current state of the world, $cross(B)$ can no longer be explicitly inferred, and, consequently, $\neg cross(B)$ is entailed, while after removing both $cross(B)$ and red_light, $cross(B)$ is explicitly inferred, and, consequently, $\neg cross(B)$ is not entailed. Again, it is worth noticing the connection to the theorems proved above: in this case, the rule $cross(x) \leftarrow not\ red\_light$ plays the role of program $\widetilde{P}$ introduced in one of the counterexamples used to prove Theorem 6.11.

This example demonstrates basic differences with the formalism adopted by [5]. Indeed, in the previous example, the default rule $\frac{red\_light:\neg cross(x)}{\neg cross(x)}$ (or, equivalently, by the ELP rule under the *answer set semantics* $\neg cross(x) \leftarrow not\ cross(x), red\_light$) captures the knowledge that normally the railroad should not be crossed when the light is red (unless, for example, there is some state of emergency). One could claim that the rule we used under stable model semantics does not faithfully correspond to the default, since the rule states under which conditions one *should* cross, instead of saying, as done in the default, when one *should not* cross. The problem is that there is no natural way of saying anything similar in the language of normal logic programs under stable model semantics because this language has no classical negation and, by the previous theorem, classical negation cannot be faithfully simulated. To illustrate the other way around, note that with outlier detection under stable model semantics of [5] all observations are encoded as positive literals. Now consider one such example where we look for outliers under stable model semantics. If we were to construct a perfectly corresponding example under default reasoning, with the same outliers and witnesses to be exactly there, then the involved default rules should have strong negated literals as consequences according to Definition 3.2, but such consequences cannot be directly coded under stable model semantics, since strong negation is not allowed.

It is also worth pointing out that complexity pictures derived for the fragment of non-disjunctive logic programs under stable model semantics cannot be exploited to derive complexity results concerning disjunctive logic programs under answer set semantics, since the latter are more general. And, in fact, outlier detection using EDLPs is more complex than outlier detection when disjunction-free logic programs under stable model semantics are considered. For instance, while it has been shown in previous sections that the outlier existence problem is Σ_3^P-complete in the former case, the same problem is Σ_2^P-complete in the latter case (see [5]).

7. Conclusion

Suppose you are walking down the street and you see a blind person going in the opposite direction. You believe he is blind because he is feeling his way with a walking stick. Suddenly something falls out of his bag and to your surprise, he finds it immediately without probing about with his fingers, as one would expect from a blind person. This kind of behavior renders that person to be blind a suspicious situation.

In this paper we provided a formal framework for this type of commonsense reasoning. Our framework is essentially a data mining technique as it allows for automatic discovery of outliers. The user of a system built using our outlier detection method does not have to specify the suspicious observations or provide a list of potential outliers. Rather, the system identifies the exceptional observations by analyzing the data and the default rules in the knowledge base at hand. This method can be used in several applications such as diagnosis and homeland security.

We used two related logics for our formal analysis: Reiter's default logic and extended disjunctive logic programming under the answer set semantics. Reiter's default logic was chosen because it constitutes a powerful nonmonotonic formalism that deals with incomplete information, while logic programming was chosen for its practicality. Logic programming as a tool of KR&R is widely employed in Artificial Intelligence, and some efficient implemented systems are available [38,43,40].

We addressed several main issues related to this idea of outlier detection in the framework of default logics and extended disjunctive logic programs under the answer set semantics. First, we introduced a formal definition of outliers and justified it using several application examples. Second, we analyzed the complexities involved in incorporating the outlier detection mechanism into knowledge bases expressed using default logic or extended logic programs. In our view, a thorough complexity analysis is essential for singling out the more complex subtasks involved in outlier detection. This first step is conducive to designing effective algorithms for implementation purposes. We have also provided an extensive comparison between our formalism and other methods that exist in the literature for outlier detection.

Our approach relies on the existence of default rules in the knowledge base. Hence methods to automatically create defaults are of interest for the application of our technique. For instance, the techniques developed by Nicholas and Duval [20,42], allow for learning default theories from examples. In Section 3.3 we demonstrated an example of outlier detection in the context of learned default theories. Other techniques for learning rules deserve attention, and coupling our method with these techniques could be an interesting topic for future research. Among these other techniques, inductive logic programming deals with learning (variants of) logic programs in various settings[36]. For instance, Inoue and Kudoh [29] worked on learning extended logic programs. Learning rules can also be realized in semi-supervised settings using an approach like metaquerying [3,13].

This work can be extended in several directions. First, the concept of outliers in other frameworks of default reasoning, like System Z [46], and Circumscription [41] should be studied. Second, intelligent heuristics that will enable the heavy computational task involved in efficient outlier detection should be investigated. In this respect, the identification of tractable subsets for the task of outlier detection might be one possible step towards making the computation task more efficient. Third, the ideas developed here might be exploited for using default logic for specifying semantically rich integrity constraints on relational databases such that any tuple that does not comply with the constraints will be an outlier (cf. Section 3.3).

Private and public organizations are overwhelmed with vast quantities of data and knowledge. Procedures that efficiently analyze the data and report only essential information are in great need. Our framework is one such knowledge discovery procedure as it is capable of identifying abnormal properties and abnormal observations automatically. It remains to find out how the ideas developed in this paper will work in practice. To this end, we hope to develop a system for outlier detection and test it on real-world data.

Acknowledgments

The authors thank Michael Gelfond for fruitful discussions and Francesco Scarcello and Gianluigi Greco for providing useful insights into some of the computational complexity issues we have raised. The second author is grateful to Barbara Grosz for inviting her to spend a year at Harvard DEAS as a visiting scholar. The authors are thankful to the anonymous reviewers for their valuable and constructive comments which have allowed to significantly improve the paper. We also thank Estelle Schulgasser for proofreading this manuscript.

Appendix A. Proofs for Section 4

The following notations are used in this sequel.

Let L be a set of literals. Then we denote by L^+ the set of positive literals occurring in L, and with L^- the set of negative literals occurring in L.

Let T be a truth assignment to the set $x_1, \ldots, x_n$ of variables. Then $Lit(T)$ denotes the set of literals $\{\ell_1, \ldots, \ell_n\}$, such that ℓ_i is x_i if $T(x_i) = \mathbf{true}$ and is $\neg x_i$ if $T(x_i) = \mathbf{false}$, for $i = 1, \ldots, n$.

Let T_1 and T_2 be two truth assignments on the disjoint sets X and Y of variables, respectively. Then $T_1 \cup T_2$ denotes the truth assignment T on the set of variables $X \cup Y$ such that $T(x) = T_1(x)$, if $x \in X$, and $T(x) = T_2(x)$, if $x \in Y$.

Proof of Theorem 4.1:
OUTLIER on propositional default theories is
1. Σ_3^P-complete, for general theories, and
2. Σ_2^P-complete, for DF theories.

Proof:
1. (Membership) Given a theory $\Delta = (D, W)$, we must show that there exist two disjoint subsets L and $S = \{s_1, \ldots, s_n\}$ of W such that $(D, W_S) \models \neg s_1 \wedge \ldots \wedge \neg s_n$ (problem q') and $(D, W_{S,l}) \not\models \neg s_1 \wedge \ldots \wedge \neg s_n$ (problem q''). Problem q' is Π_2^P-complete, while problem q'' is Σ_2^P-complete [27,55]. Thus, a polynomial-time nondeterministic Turing machine can be built with a Σ_2^P oracle, which solves query OUTLIER as follows: the machine guesses both the sets L and S and then solves problems q' and q'' by two calls to the oracle.

   (Hardness) To prove the completeness of the query OUTLIER, the Σ_3^P-complete problem of deciding the validity of a $QBE_{3,\exists}$ formula is reduced to it. A $QBE_{3,\exists}$ formula Φ has the form

$$\exists X \forall Y \exists Z f(X, Y, Z)$$

   where X, Y, Z are disjoint sets of variables, and $f(X, Y, Z)$ is a propositional formula on X, Y, Z. Intuitively, the reduction associates the default theory $\Delta(\Phi) = (D(\Phi), W(\Phi))$ with the formula Φ so that:
   - there exists one and only one literal l in $W(\Phi)$ that may belong to an outlier set, but not to any witness set;
   - there exists a bijection between each of the possible outlier witness sets S from $W(\Phi)$ and all the potential truth assignments of the variables in the set X;

– $\Delta_S(\Phi) = (D(\Phi), W(\Phi)_S)$ encodes Φ so that $\Delta_S(\Phi) \models \neg S$ iff $\forall Y \exists Z f(X,Y,Z)$ is valid, subject to the truth assignment of X induced by S;
– $\{l\}$ acts as a switch, that is, if removed from $W(\Phi)_S$ then $(D(\Phi), W(\Phi)_{S,\{l\}}) \not\models \neg S$ for each outlier witness set S.

More formally, let $\Phi = \exists X \forall Y \exists Z f(X,Y,Z)$ be a quantified Boolean formula, where $X = x_1, \ldots, x_n$, $Y = y_1, \ldots, y_m$, and Z are disjoint set of variables. With Φ, the default theory $\Delta(\Phi) = (D(\Phi), W(\Phi))$ is associated, where $W(\Phi)$ is the set of literals $\{l, \neg\phi, x_1, \ldots, x_n\}$, and $D(\Phi) = D_1 \cup D_2 \cup D_3 \cup D_4$, where:

$$D_1 = \left\{ \delta_{1,1,i} = \frac{x_i : x_i'}{x_i'}, \delta_{1,2,i} = \frac{: \neg x_i}{\neg x_i}, \delta_{1,3,i} = \frac{\neg\phi : x_i}{x_i}, \delta_{1,2,4} = \frac{\neg x_i : x_i'}{x_i'} \mid i = 1, \ldots, n \right\}$$

$$D_2 = \left\{ \delta_{2,1} = \frac{: \phi}{\phi}, \delta_{2,2} = \frac{: \neg l}{\neg l}, \delta_{2,3} = \frac{\neg l : \neg\phi}{\neg\phi} \right\}$$

$$D_3 = \left\{ \delta_{3,1,j} = \frac{: y_j}{y_j}, \delta_{3,2,j} = \frac{: \neg y_j}{\neg y_j} \mid j = 1, \ldots, m \right\}$$

$$D_4 = \left\{ \delta_4 = \frac{x_1' \wedge \ldots \wedge x_n' \wedge \neg f(X,Y,Z) : \neg\phi}{\neg\phi} \right\}$$

and $X' = x_1', \ldots, x_n'$ are new letters. Occurrences of these letters can be removed from the defaults in $D(\Phi)$ without affecting the correctness of the reduction. However, their use makes the reduction easier to understand. Clearly, $\Delta(\Phi)$ can be built in polynomial time. Now it is shown that Φ is valid iff there exists an outlier in $\Delta(\Phi)$.

First of all, we note that each extension E of the default theory $(D(\Phi), W(\Phi)_S)$, where S is an arbitrary subset of $W(\Phi)$, is the logical closure of a maximal consistent subset of the set $U = (X \cup X' \cup Y \cup Y' \cup \{l, \phi\}) \cup \neg(X \cup Y \cup \{l, \phi\})$.

Claims 1-3 below take into account the role of the defaults belonging to the set $D(\Phi)$ defined above.

– **Claim 1** *Let S be a subset of $W(\Phi)$, and $\Delta' = (D(\Phi), W(\Phi)_S)$. Then $l \in S$ implies that $\Delta' \not\models \neg s$ for each $s \in W(\Phi) \setminus \{l\}$.*

  Proof of Claim 1: Assume that $l \in S$. Then, by applying rules $\delta_{2,2}$ and $\delta_{2,3}$, there exists an extension E of Δ' such that $\neg\phi \in E$. Consequently, for each $x_i \in S$, by defaults $\delta_{1,3,i}$ ($1 \leq i \leq n$), there exists an extension E_i of Δ' such that $x_i \in E_i$. □

– **Claim 2** *Let $S \subseteq W(\Phi)$ be an outlier witness for an outlier set $L \subseteq W(\Phi)$ in $\Delta(\Phi)$. Then $\{\neg\phi\} \subseteq S \subseteq W(\Phi) \setminus \{l\}$.*

  Proof of Claim 2: Let Δ' be the theory $(D(\Phi), W(\Phi)_S)$ and let Δ'' be the theory $(D(\Phi), W(\Phi)_{S,L})$. Suppose that $l \in S$. From Claim 1, $l \in S$ implies that $\Delta' \not\models \neg s$, for each $s \in W(\Phi) \setminus \{l\}$. Hence, we can conclude that $l \in S$ implies that $S = \{l\}$. But, because of rule $\delta_{2,3}$ and as there does not exist a rule in $D(\Phi)$ having l as its consequence, both $\Delta' \models \neg l$ and $\Delta'' \models \neg l$, no matter what the value of L is. Thus, $\{l\}$ cannot be an outlier witness for any set in $W(\Phi)$, and $S \subseteq W(\Phi) \setminus \{l\}$.

  Suppose now that $\neg\phi$ does not belong to S, i.e., that $S \subseteq \{x_1, \ldots, x_n\}$. Since the literal $\neg\phi$ belongs to $W(\Phi)_S$, there exists an extension E of Δ' such that $E \supseteq \{x_1, \ldots, x_n\}$ which is obtained by applying defaults $\delta_{1,3,i}$ ($1 \leq i \leq n$), and S is not an outlier witness set, a contradiction. □

– **Claim 3** *Let T_X be a truth assignment on the set X of variables and let $S = \{x_i \in X \mid T_X(x_i) =$ **false**$\} \cup \{\neg\phi\}$. Then, for each extension E of $(D(\Phi), W(\Phi)_S)$, it holds that $E \supset Lit(T_X)$.*

  Proof of Claim 3: Assume that there exists an extension E of $(D(\Phi), W(\Phi)_S)$ such that $E \not\supseteq Lit(T_X)$. Then it is the case that there exists a letter x_i, which is false according to T_X, such that $x_i \in E$. Since $x_i \in S$, it is the case that rule $\delta_{1,3,i}$ is a generating default of E (while rule $\delta_{1,2,i}$ is not). Notice now that the precondition of rule $\delta_{1,3,i}$ is $\neg\phi$. As $\neg\phi \in S$, it is the case that rule δ_4 is also a generating default of E applied before rule $\delta_{1,3,i}$. Since rule δ_4 has the letter x_i' in its precondition, it can be concluded that $\delta_{1,2,i}$ is a generating default of E applied before δ_4 and that $\neg x_i \in E$. Thus, $\delta_{1,3,i}$ cannot be a generating default of E, or, equivalently, $x_i \notin E$, a contradiction. □

We now continue with the main proof.

($\Rightarrow$) Suppose that Φ is valid. Then there exists a truth assignment T_X on the set X of variables such that T_X satisfies $\forall Y \exists Z f(X,Y,Z)$. Let $S = \{x_i \in X \mid T_X(x_i) = \textbf{false}\} \cup \{\neg\phi\}$. We will show that S is an outlier witness for $\{l\}$ in $\Delta(\Phi)$.

For each truth assignment T_Y on the set of variables Y, consider the subset $\mathcal{E}_Y$ of the extensions of $(D(\Phi), W(\Phi)_S)$ consisting of the extensions E_Y such that $E_Y \supset Lit(T_Y)$. In particular, by Claim 3, each E_Y is such that $E_Y \supset Lit(T_X) \cup Lit(T_Y)$. Since Φ is valid, it is the case that there exists a truth assignment T_Z to the set of variables Z such that $T_X \cup T_Y \cup T_Z$ satisfies $f(X, Y, Z)$. Hence, for each $E_Y \in \mathcal{E}_Y$, it holds that $\neg f(X, Y, Z) \notin E_Y$, and the rule δ_4 cannot be a generating default of the extension E_Y. To conclude, the sets $\mathcal{E}_Y$ induce a partition of the set of all extensions of $(D(\Phi), W(\Phi)_S)$, and hence $(D(\Phi), W(\Phi)_S) \models \neg S$.

($\Leftarrow$) Suppose that there exists an outlier set L in $\Delta(\Phi)$. Then there exists a nonempty set of literals S such that S is an outlier witness set for L in $\Delta(\Phi)$ and, from Claim 2, such that $\{\neg\phi\} \subseteq S \subseteq W(\Phi) \setminus \{l\}$.

Let T_X be the truth assignment to the set of variables X such that $T_X(x_i) = \textbf{false}$, if $x_i \in S$, and $T_X(x_i) = \mathbf{true}$, if $x_i \notin S$. Then, by Claim 3, for each extension E of $(D(\Phi), W(\Phi)_S)$, it holds that $E \supseteq Lit(T_X)$. Now it is shown that T_X satisfies $\forall Y \exists Z f(X, Y, Z)$, i.e., that Φ is valid. For each truth assignment T_Y to the set of variables Y, there exists a subset $\mathcal{E}_Y$ of the extensions of $(D(\Phi), W(\Phi)_S)$ that is the set of all the extensions E_Y such that $E_Y \supset Lit(T_Y)$. We also recall that $E_Y \supset Lit(T_X)$.

Thus, in order for L to be an outlier in $\Delta(\Phi)$, it must be the case that, for each set of literals $Lit(T_Y)$; for each set of extensions $\mathcal{E}_Y$; and for each extension $E_Y \in \mathcal{E}_Y$, it holds that $\phi \in E_Y$. By defaults $\delta_{2,1}$ and δ_4, $\phi \in E_Y$ if and only if $\neg\phi \notin E_Y$ if and only if δ_4 is not a generating default of E_Y if and only if there exists a truth assignment to the set of variables Z such that $T_X \cup T_Y \cup T_Z$ satisfies $f(X, Y, Z)$. As the sets of extensions $\mathcal{E}_Y$ form a partition of the extensions of $(D(\Phi), W(\Phi)_S)$, we can conclude that Φ is valid.

As for the outlier set L, note that S is always an outlier witness set for $L = \{l\}$ in $\Delta(\Phi)$. Indeed, consider the theory $\Delta'' = (D(\Phi), W(\Phi)_{S,\{l\}})$. It follows from Claim 1 that $\Delta'' \not\models \neg S$.

2. (Membership) Analogous to Point 1 of Theorem 4.1, the only difference being that an NP oracle is used in place of a Σ_2^{P} oracle.

   (Hardness) Let $\Phi = \exists X \forall Y f(X, Y)$ be a quantified Boolean formula in disjunctive normal form, where $X = x_1, \ldots, x_n$ and $Y = y_1, \ldots, y_m$ are disjoint set of variables, and $f(X, Y) = D_1 \vee \ldots \vee D_r$, with $D_k = t_{k,1} \wedge t_{k,2} \wedge t_{k,3}$, and each $t_{k,1}, t_{k,2}, t_{k,3}$ is a literal on the set $X \cup Y$ ($k = 1, \ldots, r$). The default theory $\Delta(\Phi) = (D(\Phi), W(\Phi))$ is associated with Φ, where $W(\Phi) = \{l, \neg\phi, x_1, \ldots, x_n\}$ is a set of letters, with l and ϕ being new letters that are distinct from those occurring in Φ, and $D(\Phi) = D_1 \cup D_2 \cup D_3 \cup D_4$ is identical to the set of defaults reported in point 1 of Theorem 4.1 except for set D_4 which is:

$$D_4 = \left\{ \delta_{4,h,k} = \frac{x_1' \wedge \ldots \wedge x_n' \wedge \neg t_{k,h} : \neg d_k}{\neg d_k} \mid k = 1, \ldots, r; h = 1, 2, 3 \right\} \cup$$
$$\cup \left\{ \delta_4 = \frac{\neg d_1 \wedge \ldots \wedge \neg d_r : \neg\phi}{\neg\phi} \right\}$$

   where $d_1, \ldots, d_r$ are new letters that are distinct from those occurring in Φ. Clearly, $\Delta(\Phi)$ can be built in polynomial time. By exactly following the same line of reasoning as in Theorem 4.1 point 1, it can be shown that Φ is valid iff there exists an outlier in $\Delta(\Phi)$.

□

Proof of Theorem 4.3:

OUTLIER(L) on propositional default theories is

1. Σ_3^P-complete, for general theories, and
2. Σ_2^P-complete, for DF theories.

Proof:

1. (Membership) The proof is analogous to that used in Point 1 of Theorem 4.1.

   (Hardness) The reduction is the same as that in Point 1 of Theorem 4.1. Clearly, Φ is valid iff $\{l\}$ is an outlier set for $\Delta(\Phi)$.
2. The proof is analogous to that used in Point 2 of Theorem 4.1.

□

Proof of Theorem 4.4:

OUTLIER(S) on propositional default theories is

1. D_2^P-complete, for general theories, and
2. D^P-complete, for DF theories.

Proof:

1. (Membership) See Section 4.2.

   (Hardness) Let $\Delta_1 = (D_1, W_1)$ and $\Delta_2 = (D_2, W_2)$ be two normal propositional default theories, let s_1, s_2 be two letters, and let q be the problem $((\Delta_1 \models s_1) \wedge (\Delta_2 \not\models s_2))$. W.l.o.g, it can be assumed that Δ_1 and Δ_2 contain different letters.
   Notice that, given a normal default theory (D, W) and a literal q, the problem of deciding if $\Delta \models q$ is Π_2^P-complete even in the case in which W is empty. Thus, in the following we assume that both W_1 and W_2 are empty sets.

   Problem q is associated with the default theory $\Delta(q) = (D(q), W(q))$ which is defined as follows. Let $D(q) = \{ \frac{s_2 \wedge \alpha : \beta}{\beta} \mid \frac{\alpha : \beta}{\beta} \in D_1 \} \cup D_2$, and $W(q) = \{\neg s_1, s_2\}$. We will show that q is true iff $\{\neg s_1\}$ is a witness for some outlier in $\Delta(q)$. Note that q is the conjunction of a Π_2^P-hard and a Σ_2^P-hard problem, which proves D_2^P-hardness.

   ($\Rightarrow$) Suppose that q is true. We will show that $\{\neg s_1\}$ is an outlier witness for $\{s_2\}$ in $\Delta(q)$. Consider the theory $\Delta' = (D(q), W(q)_{\{\neg s_1\}})$. First, we note that $\Delta' \models s_1$. Indeed, from $\Delta_1 \models s_1$ and $s_2 \in W(q)_{\{\neg s_1\}}$, it can be concluded that $\Delta' \models s_1$ by means of defaults coming from the set D_1.
   Consider now the theory $\Delta'' = (D(q), W(q)_{\{\neg s_1\},\{s_2\}})$. As $\Delta_2 \not\models s_2$, there exists an extension E of Δ'' such that s_2 does not belong to E, and its associated set D_E of generating defaults does not contain any rule from the set of defaults $D(q) \setminus D_2$. We also note that Δ'' is consistent, as both Δ_1 and Δ_2 are consistent. Thus it can be concluded that $\Delta'' \not\models s_1$. Hence, $\{\neg s_1\}$ is an outlier witness for $\{s_2\}$ in $\Delta(q)$.

   ($\Leftarrow$) Suppose that $\{\neg s_1\}$ is a witness for some outlier set L in $\Delta(q)$. Let Δ' and Δ'' denote the theories $(D(q), W(q)_{\{\neg s_1\}})$ and $(D(q), W(q)_{\{\neg s_1\},L})$ respectively.
   First, we note that $\Delta' \models s_1$. As the literal s_1 occurs only in the rules of $D(q)$ coming from D_1, and the rules in D_2 have no letters in common with these rules, except for letter s_2, and $s_2 \in W(q)_{\{\neg s_1\}}$, then it is the case that $\Delta_1 \models s_1$.
   In order for $\Delta'' \not\models s_1$ to hold, L must include the letter s_2. Since $W(q) = \{\neg s_1, s_2\}$, L must be equal to $\{s_2\}$. Clearly, it must also be the case that $\Delta'' \not\models s_2$, i.e., that $\Delta_2 \not\models s_2$. This proves that the problem q is indeed verified.
2. Both membership and hardness are analogous to Point 1 of Theorem 4.4.

$\square$

Proof of Theorem 4.6:
OUTLIER(S)(L) on propositional default theories is

1. D_2^P-complete, for general theories, and
2. D^P-complete, for DF theories.

Proof: The membership proof is identical to that of Theorem 4.4. The hardness proof is identical to that of Theorem 4.4, but with a minor addendum. Indeed, while in the problem considered in Theorem 4.4, the outlier set L is unknown and the outlier witness set S is part of the input, in the problem considered here both the outlier set L and the witness set S are part of the input. Note that the reduction employed in Theorem 4.4 outputs a default theory $\Delta(q) = (D(q), W(q))$ (we refer to that theorem for the form of the formula q), with $W(q) = \{s_1, \neg s_2\}$, such that $S = \{\neg s_1\}$ is a witness set in $\Delta(q)$ if and only if q is valid. It can be inferred also that S is a witness set if and only if $L = \{s_2\}$ is an outlier set. Thus, to conclude the hardness proof, it is sufficient to include the sets $L = \{s_2\}$ and $S = \{\neg s_1\}$ together with the theory $\Delta(q)$ as part of the input for the problem OUTLIER(S)(L). $\square$

Proof of Theorem 4.7:
OUTLIER-MIN(L) on general propositional default theories is

1. D_3^P-complete, for general theories, and
2. D_2^P-complete, for DF theories.

Proof:

1. (Membership) Given a default theory $\Delta = (D, W)$ and a set of literals $L \subseteq W$, we must show (i) that L is an outlier in Δ, i.e., that there exists a set $S \subseteq W_L$ such that $(D, W_S) \models \neg S$ and $(D, W_{S,L}) \not\models \neg S$ (query q'),

and (ii) that L is a minimal outlier, i.e., that for each nonempty subset $L' \subset L$, L' is not an outlier in Δ, i.e., that for each subset $S' \subset (W \setminus L')$, $(((D, W_{S'}) \not\models \neg S') \vee ((D, W_{S',L'}) \models \neg S'))$ holds (query q'').

Query q' can be solved by a polynomial time nondeterministic Turing machine with a Σ_2^P oracle that guesses the set S and then calls the oracle twice to decide $((D, W_S) \models \neg S) \wedge ((D, W_{S,L}) \not\models \neg S)$, and hence it is in Σ_3^P. Furthermore, the negation of query q'' can be decided by a polynomial time nondeterministic Turing machine with a Σ_2^P oracle, that guesses the two sets L' and S' and then calls the oracle two times to decide whether Q holds, and hence it is in Π_3^P. Summarizing, the problem is the conjunction of two independent problems, one from Σ_3^P and the other from Π_3^P, which implies that the problem is in D_3^P.

(Hardness) Completeness of query OUTLIER-MIN(L) for general theories can be proved by reducing the problem of deciding the validity of the formula

$$F = ((\exists X)(\forall Y)(\exists Z) f(X, Y, Z)) \wedge ((\forall W)(\exists U)(\forall V) g(W, U, V))$$

to the problem OUTLIER-MIN(L), where $f(X, Y, Z)$ is a Boolean formula in conjunctive normal form and $g(X, Y, Z)$ is a Boolean formula in disjunctive normal form. Formula F is the conjunction of a $QBE_{3,\exists}$ and a $QBE_{3,\forall}$ formula, a complete problem for the class D_3^P. A similar reduction, but considering the conjunction of a $QBE_{2,\exists}$ formula and a $QBE_{2,\forall}$ formula, can be used to prove the result for DF default theories. In particular, the default theory $(D(F), W(F))$ is associated to the formula F such that:

(a) the outlier set L precisely consists of two literals, that is $L = \{l_1, l_2\}$;
(b) $W(F)_L$ is partitioned into two subsets S_X and S_W such that, for each subset S of $W(F)_L$, there exists a bijection between the sets $S \cap S_X$ and $S \cap S_W$ and each possible truth assignment for the variables X and W, respectively;
(c) $D(F)$ is such that
   - $(D(F), W(F)_S) \models \neg(S \cap S_X)$, and
   - $(D(F), W(F)_{S,\{l_1\}}) \models \neg(S \cap S_X)$, and
   - $(D(F), W(F)_{S,\{l_2\}}) \models \neg(S \cap S_X)$, and
   - $(D(F), W(F)_{S,\{l_1,l_2\}}) \not\models \neg(S \cap S_X)$

   iff $(\forall Y)(\exists Z) f(X, Y, Z)$ holds true subject to the truth assignment for the variables in the set X induced by $S \cap S_X$;
(d) $D(F)$ is such that
   - $(D(F), W(F)_S) \models \neg(S \cap S_W)$, and
   - $(D(F), W(F)_{S,\{l_1\}}) \not\models \neg(S \cap S_W)$, and
   - $(D(F), W(F)_{S,\{l_2\}}) \not\models \neg(S \cap S_W)$

   iff $(\forall U)(\exists V) \neg g(X, Y, Z)$ holds true subject to the truth assignment for the variables in the set W induced by $S \cap S_W$.

It follows from point (d) above that, in order for $\{l_1, l_2\}$ to be a minimal outlier set, then it is the case that there does not exist a truth assignment to the variables of the set W such that $(\forall U)(\exists V) \neg g(X, Y, Z)$ is true, that is, that the formula $(\forall W)(\exists U)(\forall V) g(W, U, V)$ is true. As a consequence, a witness S for L must be such that $S \subseteq S_X$ and, from point 3 above, S encodes a truth assignment for the variables of the set X such that $(\forall Y)(\exists Z) f(X, Y, Z)$ is true. Note that by point (c) above, a subset S of S_X represents a witness for neither $\{l_1\}$ nor $\{l_2\}$, and this finally proves that L is a minimal outlier set in $(D(F), W(F))$ iff the formula F is true.

We now proceed to the detailed proof. Let $\Phi = \exists X \forall Y \exists Z f(X, Y, Z)$ and $\Psi = \forall W \exists U \forall V g(W, U, V)$ be two quantified Boolean formulas, where $X = x_1, \ldots, x_n$, $Y = y_1, \ldots, y_m$, Z, $W = w_1, \ldots, w_p$, $U = u_1, \ldots, u_q$, and V are disjoint set of variables. Let F be the formula $\Phi \wedge \Psi$.

The normal default theory $\Delta(F) = (D(F), W(F))$ is associated With F, where $W(F)$ is the set of letters $\{l_1, l_2, \neg\phi, x_1, \ldots, x_n, \psi, w_1, \ldots, w_p\}$ where l_1, l_2, ϕ, and ψ are new letters distinct from those occurring in F, and $D(F)$ is $D_1 \cup \ldots \cup D_8$, where:

$$D_1 = \left\{ \frac{x_i : x_i'}{x_i'}, \frac{: \neg x_i}{\neg x_i}, \frac{\neg x_i : x_i'}{x_i'}, \frac{\neg\phi : x_i}{x_i} \mid i = 1, \ldots, n \right\}$$

$$D_2 = \left\{ \frac{: y_j}{y_j}, \frac{: \neg y_j}{\neg y_j} \mid j = 1, \ldots, m \right\}$$

$$D_3 = \left\{ \frac{l_1 \wedge x'_1 \wedge \ldots \wedge x'_n \wedge \neg f(X,Y,Z) : \neg\phi}{\neg\phi}, \frac{l_2 \wedge x'_1 \wedge \ldots \wedge x'_n \wedge \neg f(X,Y,Z) : \neg\phi}{\neg\phi} \right\}$$

$$D_4 = \left\{ \frac{l_1 : \phi}{\phi}, \frac{l_2 : \phi}{\phi} \right\}$$

$$D_5 = \left\{ \frac{w_k : w'_k}{w'_k}, \frac{: \neg w_k}{\neg w_k}, \frac{\neg w_k : w'_k}{w'_k}, \frac{\psi : w_k}{w_k} \mid k = 1, \ldots, p \right\}$$

$$D_6 = \left\{ \frac{: u_h}{u_h}, \frac{: \neg u_h}{\neg u_h} \mid h = 1, \ldots, q \right\}$$

$$D_7 = \left\{ \frac{l_1 \wedge l_2 \wedge w'_1 \wedge \ldots \wedge w'_p \wedge g(W,U,V) : \psi}{\psi} \right\}$$

$$D_8 = \left\{ \frac{l_1 \wedge l_2 : \neg\psi}{\neg\psi} \right\}$$

Clearly $W(F)$ is consistent and $\Delta(F)$ can be built in polynomial time. Now it is shown that F is valid iff $\{l_1, l_2\}$ is a minimal outlier set in $\Delta(F)$.

($\Rightarrow$) Suppose that F is valid. We will show that $\{l_1, l_2\}$ is a minimal outlier in $\Delta(F)$.
Since F is valid, there exists a truth assignment T_X to the variables in the set X such that T_X satisfies $\forall Y \exists Z f(X,Y,Z)$. Let $S = \{\neg\phi\} \cup \{x_i \mid T_X(x_i) = \text{false}\}$. We will show that S is a witness for the outlier set $\{l_0, l_1\}$. First note that, by rules in the set D_1, each extension E of $(D(F), W(F)_S)$ is such that $x_i \in E$ ($\neg x_i \in E$ resp.) if x_i is true (false resp.) according to T_X. Furthermore, by rules in the set D_2, each extension E of $(D(F), W(F)_S)$ can be associated with a truth assignment T_Y to the set Y of variables. In order for the literal $\neg\phi$ to belong to some extension of the theory $(D(F), W(F)_S)$, it must be the case that, by the rules in the set D_3, there exists a truth assignment T_Y to the set Y of variables such that $(\forall Z)\neg f(X,Y,Z)$ is true subject to the truth assignment $T_X \cup T_Y$, or, equivalently, that the formula $\neg(\forall Y)(\exists Z) f(X,Y,Z)$ is true subject to the truth assignment T_X, and that contradicts the definition of T_X. It follows then that $\neg\phi$ does not belong to every extension of $(D(F), W(F)_S)$. By the rules in the sets D_1 and D_4, we can conclude that the negation of the variables in the set S is entailed by the default theory $(D(F), W(F)_S)$.
Finally, the default theory $(D(F), W(F)_{S,\{l_1,l_2\}})$ does not entail $\neg S$, since prerequisites of both the two rules in the set D_4 are removed from $W(F)$ and do not appear in the conclusion of any default rule in $\Delta(F)$. Thus, $\{l_1, l_2\}$ is an outlier set in $\Delta(F)$.

Next, it is shown that $\{l_1, l_2\}$ is a minimal outlier set in $\Delta(F)$, that is that neither $\{l_1\}$ nor $\{l_2\}$ are witness sets in $\Delta(F)$. First of all, note that neither l_1 nor l_2 can belong to a witness set (recall that both $\neg l_1$ and $\neg l_2$ do not appear in the consequence of any rule in $D(F)$). Furthermore, for each subset S of $W(F)$, let S' denote the set $S \cap \{\neg\phi, x_1, \ldots, x_n\}$ and S'' denote the set $S \cap \{\psi, w_1, \ldots, w_p\}$. Note that if $S'' = \emptyset$ then $S = S' \subseteq \{\neg\phi, x_1, \ldots, x_n\}$ cannot be a witness set for $\{l_1\}$ ($\{l_2\}$ resp.). Indeed, as shown above, in order for $(D(F), W(F)_{S'}) \models \neg S'$ to hold, it is the case that S' encodes a truth assignment to the variables in the set X and, consequently, that ϕ belongs to every extension of the default theory $(D(F), W(F)_{S'})$. Nevertheless, in this scenario, the default theory $(D(F), W(F)_{S',\{l_1\}})$ ($(D(F), W(F)_{S',\{l_2\}})$ resp.) will continue to entail ϕ due to rules belonging to the set D_4.
It can be concluded that if $\{l_1\}$ ($\{l_2\}$ resp.) is an outlier set in $\Delta(F)$, then its associated witness set S must be such that $S'' = S \cap \{\psi, w_1, \ldots, w_p\}$ is not empty. Note that ψ must belong to S'', for otherwise S does not encode a witness by rules $\frac{\psi : w_k}{w_k}$ ($1 \leq k \leq p$) in the set D_5. Now, for the sake of contradiction assume that $\{l_1\}$ ($\{l_2\}$ resp.) is an outlier set in $\Delta(F)$. Notice that by rules in the set D_5, for each extension E of $(D(F), W(F)_S)$, $w_k \in E$ ($\neg w_k \in E$ resp.) if $s'_k \notin S''$ ($s'_k \in S''$ resp.). Hence, the set S'' encodes a truth assignment to the variables in the set W. Furthermore, by the rules in the set D_6, each extension E of $(D(F), W(F)_S)$ can be associated with a truth assignment T_U to the variables in the set U. Since $\{l_1\}$ ($\{l_2\}$ resp.) is an outlier set, then the default theory $(D(F), W(F)_S)$ entails $\neg S''$, and $\neg\psi$ belongs to every extension E of $(D(F), W(F)_S)$. By rules in the sets D_7, it can be concluded that, for each truth assignment T_U to the set of variables U, there exists a truth variable assignment to the set of variables V which makes the formula $g(W,U,V)$ false, for otherwise the literal ψ belongs to at least one extension of the default theory

$(D(F), W(F)_S)$. It can be concluded that there exists a truth assignment T_W to the set of variables W such that $(\forall U)(\exists V)\neg g(W, U, V)$ is true, that is, that $\neg\Psi$ is valid, which contradicts the fact that $F = \Phi \wedge \Psi$ is valid. Hence, we can conclude that neither $\{l_1\}$ nor $\{l_2\}$ are outlier sets in $\Delta(F)$. Thus, $\{l_1, l_2\}$ is a minimal outlier set in $\Delta(F)$.

($\Leftarrow$) Suppose that $\{l_1, l_2\}$ is a minimal outlier in $\Delta(F)$. Since $\{l_1, l_2\}$ is an outlier set, there exists a subset S of $\{\neg\phi, x_1, \ldots, x_n, \psi, w_1, \ldots, w_p\}$ such that $(D(F), W(F)_S) \models \neg S$. Let $S' = S \cap \{\neg\phi, x_1, \ldots, x_n\}$. Assume by contradiction that there exists a nonempty subset S'' of $\{\psi, w_1, \ldots, w_p\}$ such that $S = S' \cup S''$ is a witness set for $\{l_1, l_2\}$. The set S'' must contain the literal ψ, by rules $\frac{\psi : w_k}{w_k}$ ($1 \leq k \leq p$) in the set D_5. In order for $(D(F), W(F)_S) \models \neg S''$ to hold, it must be the case that $\neg\psi$ is entailed by the default theory $(D(F), W(F)_S)$. Since $\neg\psi$ is entailed only by the rule in the set D_8, it can be concluded that both $\{l_1\}$ and $\{l_2\}$ are outlier sets in $\Delta(F)$, having the associated witness set S'', which contradicts the fact that $\{l_1, l_2\}$ is a minimal outlier set. Hence, the set S'' must be empty.

As already observed in the previous point, rules in the set D_5 associate with each subset S'' a truth assignment T_W to the set of variables W and rules in the set D_6 associate with each extension of $(D(F), W(F)_S)$ a truth assignment to the variables in the set U, while rules in the set D_7 evaluate whether T_W implies $\forall U \exists V \neg g(W, U, V)$ or not. Hence, from the fact that S'' must be an empty set, it can also be concluded that there is no truth assignment T_W such that $\forall U \exists V \neg g(W, U, V)$, i.e. that $\Psi = \forall W \exists U \forall V g(W, U, V)$ is valid.

Since S'' is empty, it is the case that $S = S' \subseteq \{s_0, \ldots, s_n\}$. In order for S to be a witness set, it must be the case that for each extension E of $(D(F), W(F)_{S'})$, $\neg S' \in E$ holds and, hence, that $\phi \in E$. Now, note that by the rules in the set D_1, $x_i \in E$ ($\neg x_i \in E$ resp.) if $s_i \notin S'$ ($s_i \in S'$ resp.). Let T_X be the truth assignment to the set of variables X such that $T_X(x_i) = \text{true}$ ($T_X(x_i) = \text{false}$ resp.) if $s_i \notin S'$ ($s_i \in S'$ resp.). By the rules in the set D_2, each E can be associated with a truth assignment T_E to the set of variables Y. In particular, T_E is such that $T_E(y_j) = \textbf{true}$ ($T_E(y_j) = \textbf{false}$ resp.) if $y_j \in E$ ($y_j \notin E$ resp.). Since, for each extension E of $(D(F), W(F)_{S'})$, it holds that $\phi \in E$, by the rules in the set D_3 it follows that $\neg f(X, Y, Z) \notin E$, and hence it follows that for each truth assignment T_E to the set of variables Y there exists a truth assignment T_{Z_E} to the set of variables Z such that $T_X \cup T_E \cup T_{Z_E}$ satisfies the formula $f(X, Y, Z)$. To conclude, it is the case that S' encodes a truth assignment T_X for the variables in the set X such that T_X implies $\forall Y \exists Z f(X, Y, Z)$, and hence it follows that the formula Φ is valid. It can be therefore eventually concluded that $F = \Phi \wedge \Psi$ is a valid formula.

2. (Membership) This part is analogous to the membership part of Point 1 of this theorem.

   (Hardness) Let $\Phi = \exists X \forall Y f(X, Y)$ and $\Psi = \forall Z \exists W g(Z, W)$ be two quantified Boolean formulas, where $X = x_1, \ldots, x_n$, $Y = y_1, \ldots, y_m$, $Z = z_1, \ldots, z_l$, $W = w_1, \ldots, w_p$ are disjoint sets of variables; $f(X, Y) = d_1 \vee \ldots \vee d_r$ is a formula in disjunctive normal form, where each disjunct d_j ($1 \leq j \leq r$) is the conjunction of three literals, that is, $d_j = t_{j,1} \wedge t_{j,2} \wedge t_{j,3}$; and $g(Z, W) = c_1 \wedge \ldots \wedge c_s$ is a formula in conjunctive normal form, where each conjunct c_h ($1 \leq h \leq s$) is the disjunction of three literals, that is, $c_h = u_{h,1} \vee u_{h,2} \vee u_{h,3}$.

   Let F be the formula $\Phi \wedge \Psi$. We associate with F the default theory $\Delta(F) = (D(F), W(F))$, which is identical to the theory described in Point 1 of this theorem, except for the sets of rules D_3 and D_7 that are as follows:

$$D_3 = \left\{ \frac{l_i \wedge x'_1 \wedge \ldots \wedge x'_n \wedge \neg t_{j,k} : \neg d_k}{\neg d_k} \mid i = 1, 2; j = 1, \ldots, r; k = 1, 2, 3 \right\} \cup$$
$$\cup \left\{ \frac{\neg d_1 \wedge \ldots \wedge \neg d_p : \neg\phi}{\neg\phi} \right\}$$
$$D_7 = \left\{ \frac{l_1 \wedge l_2 \wedge w'_1 \wedge \ldots \wedge w'_p \wedge \neg u_{h,1} \wedge \neg u_{h,2} \wedge \neg u_{h,3} : \psi}{\psi} \mid h = 1, \ldots, s \right\}$$

   The rest of the proof is analogous to that of Point 1 of this theorem.

□

Proof of Theorem 4.8:
OUTLIER-MIN(S)(L) on general propositional default theories is
1. Π_3^{P}-complete, for general theories, and

2. Π_2^{P}-complete, for DF theories.

Proof:

1. (Membership) In order to answer query OUTLIER-MIN(S)(L), it must be verified that (problem q') S and L satisfy Definition 3.2, i.e., $(D, W_S) \models \neg S$ and $(D, W_{S,L}) \not\models \neg S$, and that (problem q'') for each proper subset L' of L, and S' of $W_{L'}$, S' and L' do not satisfy Definition 3.2, i.e., $(D, W_{S'}) \not\models \neg S'$ or $(D, W_{S',L'}) \models \neg S'$.

   Problem q' coincides with OUTLIER(S)(L), and hence it is in $\mathrm{D}_2^{\mathrm{P}}$ (D^{P} resp.) for general (DF resp.) theories. Furthermore, problem q'' is in Π_3^{P} (Π_2^{P} resp.) for general (DF resp.) theories, since its negation can be answered by a nondeterministic polynomial time Turing machine that guesses a pair of disjoint subsets $L' \subseteq L$ and $S' \subseteq W_{L'}$ and then checks that they form an outlier and witness pair by using an oracle in Σ_2^{P}. Thus, the overall problem is in Π_3^{P} (Π_2^{P} resp.).

   (Hardness) Let $\Psi = \forall W \exists U \forall V g(W, U, V)$ be a quantified Boolean formula, where $W = w_1, \ldots, w_p$, $U = u_1, \ldots, u_p$, and V are disjoint set of variables. With Ψ, the default theory $\Delta(\Psi) = (D(\Psi), W(\Psi))$ is associated, where $W(\Psi)$ is the set of literals $\{l_1, l_2, s, \psi, w_1, \ldots, w_p\}$ where l_1, l_2, s and ψ are new letters distinct from those occurring in Ψ, and $D(\Psi)$ is composed of the default rules in the sets D_5, D_6, D_7, and D_8 reported in the reduction shown in Point 1 of Theorem 4.7, plus the set of rules $D_0 = \{\frac{l_1:\neg s}{\neg s}, \frac{l_2:\neg s}{\neg s}, \frac{l_1 \wedge l_2:\neg s}{\neg s}\}$.

   Next, it is shown that Ψ is valid iff $L = \{l_1, l_2\}$ is a minimal outlier set with outlier witness set $S = \{s\}$ in $\Delta(\Psi)$.

   It follows from Point 1 of Theorem 4.7, that $\{l_1\}$ and $\{l_2\}$ are outlier sets in $(D(\Psi) - D_0, W(\Psi))$, with associated witness set $S_W \subseteq \{\psi, w_1, \ldots, w_p\}$, if and only if Ψ is not valid. Thus, in order for L to be a minimal outlier set, it must be the case that Ψ is valid. Finally, by rules in the set D_0, it holds that

   – $(D(\Psi), W(\Psi)_{\{s\}}) \models \neg s$,
   – $(D(\Psi), W(\Psi)_{\{l_1\},\{s\}}) \models \neg s$,
   – $(D(\Psi), W(\Psi)_{\{l_2\},\{s\}}) \models \neg s$, and
   – $(D(\Psi), W(\Psi)_{\{l_1,l_2\},\{s\}}) \not\models \neg s$.

   Hence, the result follows.
2. Both membership and hardness are analogous to Point 1 of this theorem.

□

Appendix B. Proofs for Section 4.3

Proof of Theorem 4.11:

For general EDPLs,

1. OUTLIER *is* Σ_3^P*-complete,*
2. OUTLIER[k] *is* Σ_3^P*-complete,*
3. OUTLIER(L) *is* Σ_3^P*-complete,*
4. OUTLIER(S) *is* D_2^P*-complete,*
5. OUTLIER[k](S) *is* D_2^P*-complete,*
6. OUTLIER(S)(L) *is* D_2^P*-complete,*
7. OUTLIER-MIN(L) *is* $\mathrm{D}_3^{\mathrm{P}}$*-complete, and*
9. OUTLIER-MIN(S)(L) *is* Π_3^{P}*-complete.*

Proof:

1. (Membership) Given a rule-observations program $P = (D, W)$, we must show that there exist two disjoint sets $S \subseteq W$ and $L \subseteq W$ such that $P_S \models \neg S$ (problem q') and $P_{S,L} \not\models \neg S$ (problem q''). Problem q' is Π_2^P-complete, while problem q'' is Σ_2^P-complete [21]. Thus, we can build a polynomial-time nondeterministic Turing machine equipped with a Σ_2^P oracle that solves query OUTLIER as follows: the machine will first guess the sets S and L and then solve queries q' and q'' by calling the oracle twice.

   (Hardness) Let $\Phi = \exists X \forall Y \exists Z f(X, Y, Z)$ be a quantified Boolean formula, where $X = x_1, \ldots, x_n$, $Y = y_1, \ldots, y_m$ and $Z = z_1, \ldots, z_l$ are disjoint set of variables, and $f(X, Y, Z)$ is a Boolean formula in conjunctive normal form, i.e., $f(X, Y, Z) = C_1 \wedge \ldots \wedge C_N$, with $C_h = t_{h,1} \vee t_{h,2} \vee t_{h,3}$, and each $t_{h,1}, t_{h,2}, t_{h,3}$ is a literal in the set $X \cup Y \cup Z$, for $h = 1, \ldots, N$. We associate with Φ the rule-observations program $P(\Phi) =$

$\langle D(\Phi), W(\Phi)\rangle$, where $W(\Phi) = \{o, x_0, x_1, \ldots, x_n\}$ consists of the letters in the set X plus the new letters x_0 and o which are distinct from those occurring in Φ, and $D(\Phi)$ is

$$
\begin{array}{lll}
r_0 & : \neg\phi \leftarrow \mathit{not}\ o & \\
r_{1,i} & : \neg x_i \mid x_i \leftarrow \neg\phi & (0 \le i \le n) \\
r_{2,i} & : \neg x_i \leftarrow \mathit{not}\ x_i, \mathit{not}\ \neg\phi & (0 \le i \le n) \\
r_{3,i} & : x'_i \leftarrow \mathit{not}\ x_i & (1 \le i \le n) \\
r_{4,j} & : y_j \mid y'_j \leftarrow & (1 \le j \le m) \\
r_{5,k} & : z_k \mid z'_k \leftarrow & (1 \le k \le l) \\
r_{6,k} & : z_k \leftarrow \neg\phi & (1 \le k \le l) \\
r_{7,k} & : z'_k \leftarrow \neg\phi & (1 \le k \le l) \\
r_{8,h} & : \neg\phi \leftarrow \sigma(\neg t_{h,1}), \sigma(\neg t_{h,2}), \sigma(\neg t_{h,3}) & (1 \le h \le N)
\end{array}
$$

where also $X' = x'_1, \ldots, x'_n$, $Y' = y'_1, \ldots, y'_m$, and $Z' = z'_1, \ldots, z'_l$ are new letters distinct from those occurring in Φ, and $\sigma : X \cup \neg X \cup Y \cup \neg Y \cup Z \cup \neg Z \mapsto X \cup X' \cup Y \cup Y' \cup Z \cup Z'$ is the following mapping:

$$
\sigma(t) = \begin{cases} x'_i & \text{if } t = \neg x_i\ (1 \le i \le n) \\ y'_j & \text{if } t = \neg y_j\ (1 \le j \le m) \\ z'_k & \text{if } t = \neg z_k\ (1 \le k \le l) \\ t & \text{otherwise} \end{cases}
$$

Clearly, $P(\Phi)$ can be built in polynomial time. Now we show that Φ is valid iff there exists an outlier in $P(\Phi)$. Given a truth assignment T on a subset of $X \cup Y \cup Z$, let $\mathcal{I}(T)$ denote the context $\sigma(\mathrm{Lit}(T))$.

($\Rightarrow$) Suppose that Φ is valid. We shall show that $L = \{o\}$ is an outlier in $P(\Phi)$. Then there exists a truth assignment T_X to the variables in the set X such that T_X satisfies $\forall Y \exists Z f(X, Y, Z)$. Let $S = \{x_0\} \cup \neg(\mathrm{Lit}(T_X)^-)$, We will show that $P(\Phi)_S \models \neg S$.

By contradiction, suppose that there exists an answer set M' of $P(\Phi)_S$ such that $\neg S \not\subset M'$. By rules $r_{2,i}$, it is the case that $\neg\phi \in M'$. Furthermore, by rules $r_{1,i}$, M' is of the form $\mathcal{I}(T_X) \cup \mathcal{I}(T_Y) \cup Z \cup Z' \cup \{\neg\phi, o\} \cup S'$, where T_Y denotes a truth assignment to the set of variables in Y and S' is a set of the form $\{s \mid x \in S \wedge (s = x \vee s = \neg x)\}$ such that $S' \neq \neg S$. As a consequence, the set $M = \mathcal{I}(T_X) \cup \mathcal{I}(T_Y) \cup Z \cup Z' \cup \{\neg\phi, o\} \cup \neg S$, where T_Y denotes a truth assignment to the set of variables in Y, is an answer set of $P(\Phi)_S$.

As M is an answer set of $P(\Phi)_S$, and hence a minimal context closed under $\mathit{Red}(P(\Phi)_S, M)$, it follows that for each truth assignment T_Z to the variables in the set Z, the subset $M'' = \mathcal{I}(T_X) \cup \mathcal{I}(T_Y) \cup \mathcal{I}(T_Z) \cup \{o\} \cup \neg S$ of M is not a context closed under $\mathit{Red}(P(\Phi)_S, M)$. Thus, for each T_Z there exists an $h \in \{1, \ldots, N\}$ such that $\sigma(\neg t_{h,1}), \sigma(\neg t_{h,2}), \sigma(\neg t_{h,3}) \in M''$. We can conclude that there exists an answer set M' of $P(\Phi)_S$ such that $\neg S \not\subset M'$ if and only if T_X satisfies

$$
\exists Y \forall Z \bigvee_{r=1}^{N} (\neg t_{r,1} \wedge \neg t_{r,2} \wedge \neg t_{r,3}) \equiv \neg \forall Y \exists Z f(X, Y, Z),
$$

which contradicts the fact that T_X satisfies $\forall Y \exists Z f(X, Y, Z)$. Hence, $P(\Phi)_S \models \neg S$. Let $L = \{o\}$; then, by rules r_9 and $r_{1,i}$, $P(\Phi)_{S,L} \not\models \neg S$. Thus, $L = \{o\}$ is an outlier set with outlier witness set S.

($\Leftarrow$) Suppose that there exists an outlier $L \subseteq W(\Phi)$ with witness $S \subseteq W(\Phi)$ in P. As $\neg o$ cannot be entailed by $P(\Phi)_S$, it must be the case that $S \subseteq X \cup \{x_0\}$. From what is stated above, $P(\Phi)_S \models \neg S$ implies that the truth value assignment T_X on the set of variables X such that $T_X(x_i) =$ **false** if and only if $x_i \in S$, satisfies $\forall Y \exists Z f(X, Y, Z)$, i.e., that Φ is valid. To conclude, the literal $L = \{o\}$ is always an outlier having such a witness.

2. The proof is analogous to that of Point 1.
3. The proof is analogous to that of Point 1.
4. (Membership) Given the rule-observations program $P = (D, W)$, and a subset $S \subseteq W$, we should verify that there exists $L \subseteq W_S$ such that $P_S \models \neg S$ (problem q') and $P_{S,L} \not\models \neg S$ (problem q''). We have already noted that problem q' is Π_2^{P}-complete. As for problem q'', it is Σ_2^{P}-complete as it can be answered by a polynomial-time nondeterministic Turing machine with an NP oracle as follows: the machine guesses both an outlier $L \subseteq W_S$ and a consistent context M of $P_{S,L}$ such that $\neg S \not\subseteq M$, verifies that M is closed under $Red(P_{S,L}, M)$, and decides whether $M' \subset M$ exists such that M' is closed under $Red(P_{S,L}, M)$ with a call to the NP oracle. Hence, we have to decide the conjunction $q' \wedge q''$, i.e., a $\mathrm{D}_2^{\mathrm{P}}$ problem.

   (Hardness) Let r be an EDLP rule, and let $\mathbf{h}(r)$ and $\mathbf{b}(r)$ denote respectively the head and the body of r. Let P' and P'' be two EDLPs. W.l.o.g. assume that P' and P'' contain no common letters and also that they do not contain the letters s and l. Consider the rule-observations program $P = (D, W)$ where $D = \{\mathbf{h}(r) \leftarrow \mathbf{b}(r), l : r \in P'\} \cup \{\mathbf{h}(r) \leftarrow \mathbf{b}(r), \mathit{not}\ l : r \in P''\}$ and $W = \{s, l\}$. By noting that $\mathrm{ANSW}(P') = \mathrm{ANSW}(P_{\{s\}})$ and $\mathrm{ANSW}(P'') = \mathrm{ANSW}(P_{\{s\},\{l\}})$, it follows that P' is inconsistent (a Π_2^P-complete check, see [21], Theorem 39) and P'' is consistent (a Σ_2^P-complete check, see [21], Theorem 40) iff $\{l\}$ is an outlier with witness $\{s\}$ in P.
5. The proof is analogous to that of Point 4.
6. The proof is analogous to that of Point 4.
7. (Membership) Analogous to the membership part of Theorem 4.7.

   (Hardness) Let $\Phi = \exists X \forall Y \exists Z f(X, Y, Z)$ be a quantified Boolean formula, where $X = x_1, \ldots, x_n$, $Y = y_1, \ldots, y_m$, and $Z = z_1$, are disjoint sets of variables, $f(X, Y, Z)$ is the Boolean formula in conjunctive normal form $C_1 \wedge \ldots \wedge C_N$, with $C_h = t_{h,1} \vee t_{h,2} \vee t_{h,3}$, and each $t_{h,1}, t_{h,2}, t_{h,3}$ is a literal in the set $X \cup Y \cup Z$, for $h = 1, \ldots, N$.

   Let $\Psi = \forall W \exists U \forall V g(W, U, V)$ be another quantified Boolean formula, where $W = w_1, \ldots, w_p$, $U = u_1, \ldots, u_q$, and $V = v_1, \ldots, v_r$ are disjoint sets of variables, and $g(W, U, V)$ is the Boolean formula in disjunctive normal form $D_1 \vee \ldots \vee D_M$, with $D_h = s_{h,1} \wedge s_{h,2} \wedge s_{h,3}$, and each $s_{h,1}, s_{h,2}, s_{h,3}$ is a literal in the set $W \cup U \cup V$, for $h = 1, \ldots, M$.

   Let F be the formula $\Phi \wedge \Psi$. We associate with F the rule-observations program $P(F) = \langle D(F), W(F) \rangle$, where

$$W(F) = \{o_1, o_2, x_0, x_1, \ldots, x_n, w_0, w_1, \ldots, w_p\}$$

   consists of the letters in the set $X \cup W$ plus the new letters x_0, w_0, o_1 and o_2 that are distinct from those occurring in F, and $D(F)$ is

			$r''_{0,1}$	$: \neg\psi \leftarrow \mathit{not}\ o_1$	
r'_0	$: \neg\phi \leftarrow \mathit{not}\ o_1, \mathit{not}\ o_2$		$r''_{0,2}$	$: \neg\psi \leftarrow \mathit{not}\ o_2$	
$r'_{1,i}$	$: \neg x_i \mid x_i \leftarrow \neg\phi$	$(0 \leq i \leq n)$	$r''_{1,i}$	$: \neg w_i \mid w_i \leftarrow \neg\psi$	$(0 \leq i \leq p)$
$r'_{2,i}$	$: \neg x_i \leftarrow \mathit{not}\ x_i, \mathit{not}\ \neg\phi$	$(0 \leq i \leq n)$	$r''_{2,i}$	$: \neg w_i \leftarrow \mathit{not}\ w_i, \mathit{not}\ \neg\psi$	$(0 \leq i \leq p)$
$r'_{3,i}$	$: x'_i \leftarrow \mathit{not}\ x_i$	$(1 \leq i \leq n)$	$r''_{3,i}$	$: w'_i \leftarrow \mathit{not}\ w_i$	$(1 \leq i \leq p)$
$r'_{4,j}$	$: y_j \mid y'_j \leftarrow$	$(1 \leq j \leq m)$	$r''_{4,j}$	$: u_j \mid u'_j \leftarrow$	$(1 \leq j \leq q)$
$r'_{5,k}$	$: z_k \mid z'_k \leftarrow$	$(1 \leq k \leq l)$	$r''_{5,k}$	$: v_k \mid v'_k \leftarrow$	$(1 \leq k \leq r)$
$r'_{6,k}$	$: z_k \leftarrow \neg\phi$	$(1 \leq k \leq l)$	$r''_{6,k}$	$: v_k \leftarrow \neg\psi$	$(1 \leq k \leq r)$
$r'_{7,k}$	$: z'_k \leftarrow \neg\phi$	$(1 \leq k \leq l)$	$r''_{7,k}$	$: v'_k \leftarrow \neg\psi$	$(1 \leq k \leq r)$
$r'_{8,h}$	$: \neg\phi \leftarrow \sigma(\neg t_{h,1}), \sigma(\neg t_{h,2}), \sigma(\neg t_{h,3})$	$(1 \leq h \leq N)$	$r'_{8,0}$	$: \neg\psi \leftarrow \neg d_1, \ldots, \neg d_M$	
			$r''_{8,1,h}$	$: \neg d_h \leftarrow \sigma(\neg s_{h,1})$	$(1 \leq h \leq M)$
			$r''_{8,2,h}$	$: \neg d_h \leftarrow \sigma(\neg s_{h,2})$	$(1 \leq h \leq M)$
			$r''_{8,3,h}$	$: \neg d_h \leftarrow \sigma(\neg s_{h,3})$	$(1 \leq h \leq M)$

   where also $X' = x'_1, \ldots, x'_n$, $Y' = y'_1, \ldots, y'_m$, $Z' = z'_1, \ldots, z'_l$, $W' = w'_1, \ldots, w'_p$, $U' = u'_1, \ldots, u'_q$, and $V' = v'_1, \ldots, v'_r$ are new letters distinct from those occurring in F, and $\sigma : X \cup \neg X \cup Y \cup \neg Y \cup Z \cup \neg Z \cup$

$W \cup \neg W \cup U \cup \neg U \cup V \cup \neg V \mapsto X \cup X' \cup Y \cup Y' \cup Z \cup Z' \cup W \cup W' \cup U \cup U' \cup V \cup V'$ is the following mapping:

$$\sigma(t) = \begin{cases} x'_i, & \text{if } t = \neg x_i \ (1 \leq i \leq n) \\ y'_j, & \text{if } t = \neg y_j \ (1 \leq j \leq m) \\ z'_k, & \text{if } t = \neg z_k \ (1 \leq k \leq l) \\ w'_i, & \text{if } t = \neg w_i \ (1 \leq i \leq n) \\ u'_j, & \text{if } t = \neg u_j \ (1 \leq j \leq m) \\ v'_k, & \text{if } t = \neg v_k \ (1 \leq k \leq l) \\ t, & \text{otherwise} \end{cases}$$

Clearly, $P(F)$ can be built in polynomial time. Now we show that F is valid iff $\{o_1, o_2\}$ is a minimal outlier set in $P(F)$.

The line of reasoning employed to prove the result is analogous to that of the hardness part of Theorem 4.7, Point 1. The reader is referred to the discussion preceding the reduction therein for the explanation of that line of reasoning. Next, technicalities concerning the reduction presented here are pointed out.

First of all, in order to understand the role of rules r' (r'', resp.) in the reduction depicted above, we note that these rules have the same structure as the rules used in the reduction given in Point 1 of this proof, where a Σ_3^{P}-complete problem is considered.

In particular, rules r' serve the purpose of guaranteeing that $(D(F), W(F)_S) \models \neg(S \cap S_X)$ if and only if the formula $\Phi = \exists X \forall Y \exists Z f(X, Y, Z)$ is satisfiable (see Point (c) of the discussion recalled above), while rules r'' serve the purpose of guaranteeing that $(D(F), W(F)_S) \models \neg(S \cap S_W)$ if and only if the formula $\Psi = \exists W \forall U \exists V \neg g(W, U, V)$ is satisfiable (see Point (d) of the same discussion).

As for the relationship between the minimality of the outlier set $\{o_1, o_2\}$ and the satisfiability of the formula $\Phi \wedge \Psi$, this is taken care of by rules r'_0, $r''_{0,1}$ and $r''_{0,2}$ (once again, refer to the discussion of Theorem 4.7, Point 1, and to Point 1 of this proof for details).

8. Both membership and hardness are analogous to that of Theorem 4.8. For the hardness part, we can make use of rules r'' defined in Point 7 of this proof, plus rules equivalent to the defaults in the set D_0 defined in Point 1 of Theorem 4.8.

□

Appendix C. Proofs for Section 5

Proof of Theorem 5.3:
The data complexity of OUTLIER is Σ_3^{P}-complete.
Proof: (Membership) See Section 5.

(Hardness) To prove the completeness of the query OUTLIER, the Σ_3^P-complete problem of deciding the validity of a $\mathrm{QBE}_{3,\exists}$ formula is reduced to it. A $\mathrm{QBE}_{3,\exists}$ formula Φ has the form $\exists X \forall Y \exists Z f(X, Y, Z)$, where $X = x_1, \ldots, x_n$, $Y = y_1, \ldots, y_m$, and Z are disjoint sets of variables, and $f(X, Y, Z)$ is a propositional formula on X, Y, Z. Without loss of generality, it can be assumed that the Boolean formula $f(X, Y, Z)$ is in conjunctive normal form with exactly three literals per clause, that is, that $f(X, Y, Z) = c_1 \wedge c_2 \wedge \ldots \wedge c_r$, with $c_k = t_{k,1} \vee t_{k,2} \vee t_{k,3}$, for $k = 1, \ldots, r$.

We now describe a fixed default theory $\Delta^{\mathrm{FO}} = (D^{\mathrm{FO}}, W^{\mathrm{FO}})$, where W^{FO} does not contain atomic formulas, and a mapping $W(\Phi)$ from any $\mathrm{QBE}_{3,\exists}$ formula Φ to a set of ground literals, such that there exists an outlier in $\Delta^{\mathrm{FO}}(\Phi) = (D^{\mathrm{FO}}, W^{\mathrm{FO}} \cup W(\Phi))$ if and only if the formula Φ is satisfiable. We encode the formula Φ in the extensional component $W(\Phi)$ of $\Delta^{\mathrm{FO}}(\Phi)$ by means of the following sets of atoms:

$$
\begin{aligned}
W_X = & \ \{e(x_i, x_{i+1}) \mid i = 1, \ldots, n-1\} \cup \{e(x_n, x_0)\}, \\
W_Y = & \ \{u(y_j, y_{j+1}) \mid j = 1, \ldots, m-1\} \cup \{u(y_m, y_0)\}, \text{ and} \\
W_f = & \ \{c(c_k, c_{k+1}, sgn(t_{k,1}), let(t_{k,1}), sgn(t_{k,2}), let(t_{k,2}), sgn(t_{k,3}), let(t_{k,3})) \mid k = 1, \ldots, r-1\} \cup \\
& \ \{c(c_r, c_0, sgn(t_{r,1}), let(t_{r,1}), sgn(t_{r,2}), let(t_{r,2}), sgn(t_{r,3}), let(t_{r,3}))\},
\end{aligned}
$$

where $sgn(\ell)$ is the constant p if ℓ is a positive literal, and the constant n if ℓ is a negative literal, while $let(\ell)$ is the propositional letter occurring in the literal ℓ. Intuitively, atoms with functor e list existential variables in the set X, atoms with functor u list universal variables in the set Y, and atoms with functor c list clauses of the formula $f(X, Y, Z)$. Variables in the set Z are not explicitly listed, since this is not needed for the sake of the reduction. In the following the set $W_X \cup W_Y \cup W_f$ is denoted by W_Φ.

It must be avoided that atoms in the set W_Φ become part of an outlier or a witness, for otherwise the above encoding of the formula Φ is invalidated. With this aim, the following rules, forming the set W^{FO}, are employed:

$$
\begin{aligned}
& r_0 : f \vee \neg f \rightarrow t \\
& r_1 : t \rightarrow t(x_0) \\
& r_2 : t \rightarrow e(x_0, x_1) \\
& r_3 : t \rightarrow u(y_0, y_1) \\
& r_4 : t \rightarrow c(c_0, c_1, p, x_0, p, x_0, p, x_0) \\
& r_5 : (\forall A)(\forall B)(\exists C)(e(A, B) \rightarrow e(B, C)) \\
& r_6 : (\forall A)(\forall B)(\exists C)(u(A, B) \rightarrow u(B, C)) \\
& r_7 : (\forall A)(\forall B)(\exists C)(\exists V_1) \ldots (\exists V_{12})(c(A, B, V_1, V_2, V_3, V_4, V_5, V_6) \rightarrow c(B, C, V_7, V_8, V_9, V_{10}, V_{12}, V_{12}))
\end{aligned}
$$

Rules r_2, r_3, and r_4 introduce one additional existential variable, a universal variable, and a clause, respectively, whose roles are clarified below. Rules r_5, r_6, and r_7 serve the purpose of making the overall theory inconsistent whenever at least one atom in the set W_Φ is removed from $W(\Phi)$. Indeed, assume that an atom of the form $e(x_i, x_{i+1})$ is removed from $W(\Phi)$. Then the formula $e(x_{i-1}, x_i) \rightarrow e(x_i, x_{i+1})$, and consequently also the formula r_5, evaluates to false. In particular, formula r_5 is true if and only if the atomic formulas of the form $e(x_i, x_{(i+1) \bmod (n+1)})$ are either all true or all false. Hence, it follows from formula r_2 that if at least one of the atoms in the set W_X is removed, then the theory is inconsistent. The same reasoning can be applied for all the other atoms in the set W_Φ. Assume that L is an outlier with witness S in $\Delta^{\mathrm{FO}}(\Phi)$ such that $(L \cup S) \cap W_\Phi \neq \emptyset$. Then, the default theory $(D^{\mathrm{FO}}, W^{\mathrm{FO}} \cup W(\Phi)_{S,L})$ is inconsistent and $(D^{\mathrm{FO}}, W^{\mathrm{FO}} \cup W(\Phi)_{S,L}) \models \neg S$, a contradiction. Thus, outliers and witnesses cannot contain atoms belonging to the set W_Φ.

Given a propositional variable v, we use the literals $t(v)$ and $\neg t(v)$ to represent the information that v is true and that v is false, respectively, in a certain truth value assignment. The truth value of a 3CNF formula whose clauses are encoded by means of the atoms above can be thus evaluated by using the following first-order formula:

$$
\begin{aligned}
F = (\forall A)(\forall B)(\forall T_1)(\forall T_2)(\forall T_3) \Big(& \big(c(A, B, p, T_1, p, T_2, p, T_3) \rightarrow t(T_1) \vee t(T_2) \vee t(T_3)\big) \wedge \\
& \big(c(A, B, n, T_1, p, T_2, p, T_3) \rightarrow \neg t(T_1) \vee t(T_2) \vee t(T_3)\big) \wedge \\
& \big(c(A, B, p, T_1, n, T_2, p, T_3) \rightarrow t(T_1) \vee \neg t(T_2) \vee t(T_3)\big) \wedge \\
& \big(c(A, B, n, T_1, n, T_2, p, T_3) \rightarrow \neg t(T_1) \vee \neg t(T_2) \vee t(T_3)\big) \wedge \\
& \big(c(A, B, p, T_1, p, T_2, n, T_3) \rightarrow t(T_1) \vee t(T_2) \vee \neg t(T_3)\big) \wedge \\
& \big(c(A, B, n, T_1, p, T_2, n, T_3) \rightarrow \neg t(T_1) \vee t(T_2) \vee \neg t(T_3)\big) \wedge \\
& \big(c(A, B, p, T_1, n, T_2, n, T_3) \rightarrow t(T_1) \vee \neg t(T_2) \vee \neg t(T_3)\big) \wedge \\
& \big(c(A, B, n, T_1, n, T_2, n, T_3) \rightarrow \neg t(T_1) \vee \neg t(T_2) \vee \neg t(T_3)\big) \Big).
\end{aligned}
$$

Indeed, according to formula F, for each clause c_k at least one literal among $t_{k,1}$, $t_{k,2}$, $t_{k,3}$ must be true in the considered truth value assignment.

Now, we are in a position to complete the reduction $\Delta^{\mathrm{FO}}(\Phi)$. The set $W(\Phi)$ is given by

$$W_\Phi \cup \{l, \neg\phi, t(x_1), \ldots, t(x_n)\},$$

while D^{FO} is the fixed set of defaults $D_1 \cup D_2 \cup D_3 \cup D_4$, where:

$$D_1 = \left\{ \frac{e(A,C) \wedge t(A) : ok(A)}{ok(A)}, \frac{e(A,C) : \neg t(A)}{\neg t(A)}, \frac{e(A,C) \wedge \neg\phi : t(A)}{t(A)}, \frac{e(A,C) \wedge \neg t(A) : ok(A)}{ok(A)} \right\}$$

$$D_2 = \left\{ \frac{: \phi}{\phi}, \frac{: \neg l}{\neg l}, \frac{\neg l : \neg\phi}{\neg\phi} \right\}$$

$$D_3 = \left\{ \frac{u(B,C) : t(B)}{t(B)}, \frac{u(B,C) : \neg t(B)}{\neg t(B)} \right\}$$

$$D_4 = \left\{ \frac{((\forall A)(e(A,C) \rightarrow ok(A))) \wedge \neg F : \neg\phi}{\neg\phi} \right\}$$

The reader can verify that the grounded version $\mathrm{PROP}(\Delta^{\mathrm{FO}}(\Phi))$, which is a finite propositional general default theory, is equivalent to the default theory $\Delta(\Phi)$ described in the hardness part of Theorem 4.1, Point 1. Hence, from what is stated above, and from Theorem 4.1, the result follows. $\Box$

References

[1] Charu C. Aggarwal and Philip S. Yu. Outlier detection for high dimensional data. In *Proceedings of the ACM Int. Conf. on Management of Data*, pages 37–46, Santa Barbara, California, USA, 2001.

[2] F. Angiulli, S. Basta, and C. Pizzuti. Distance-based detection and prediction of outliers. *IEEE Transactions Knowledge and Data Engineering*, 18(2):145–160, 2006.

[3] F. Angiulli, R. Ben-Eliyahu-Zohary, G. Ianni, and L. Palopoli. Computational properties of metaquerying problems. *ACM Transactions on Computational Logic*, 4(3):149–180, 2003.

[4] F. Angiulli, R. Ben-Eliyahu-Zohary, and L. Palopoli. Outlier detection using default logic. In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI)*, pages 833–838, 2003.

[5] F. Angiulli, G. Greco, and L. Palopoli. Outlier detection by logic programming. *ACM Transactions on Computational Logic*, 9(1), 2007.

[6] F. Angiulli and C. Pizzuti. Fast outlier detection in high dimensional spaces. In *European Conference on Principles of Data Mining and Knowledge Discovery*, pages 15–26, Helsinki, 2002.

[7] F. Angiulli and C. Pizzuti. Outlier mining in large high dimensional data sets. *IEEE Transactions on Knowledge and Data Engineering*, 17(2):203–215, 2005.

[8] A. Arning, C. Aggarwal, and P. Raghavan. A linear method for deviation detection in large databases. In *Proc. Int. Conf. on Knowledge Discovery and Data Mining*, pages 164–169, Portland, Oregon, USA, 1996.

[9] V. Barnett and T. Lewis. *Outliers in Statistical Data*. John Wiley & Sons, 1994.

[10] S.D. Bay and M. Schwabacher. Mining distance-based outliers in near linear time with randomization and a simple pruning rule. In *Proc. Int. Conf. on Knowledge Discovery in Databases*, pages 29–38, 2003.

[11] Rachel Ben-Eliyahu and Rina Dechter. Propositional semantics for disjunctive logic programs. *Annals of Mathematics and Artificial Intelligence*, 12:53–87, 1994.

[12] R. Ben-Eliyahu-Zohary and R. Dechter. Propositional semantics for disjunctive logic programs. In *Proceedings of the Eigthteenth Joint International Conference (JICSLP-92)*, pages 813–827, 1992.

[13] R. Ben-Eliyahu-Zohary, E. Gudes, and G. Ianni. Metaqueries: Semantics, complexity, and efficient algorithms. *Artificial Intelligence*, 149(1):61–87, 2003.

[14] M. M. Breunig, H. Kriegel, R.T. Ng, and J. Sander. Lof: Identifying density-based local outliers. In *Proc. ACM Int. Conf. on Managment of Data*, pages 93–104, Dallas, Texas, USA, 2000.

[15] Marco Cadoli, Thomas Eiter, and Georg Gottlob. Default logic as a query language. *IEEE Trans. in Knowl. and Data Eng.*, 9.

[16] Luca Console, Daniele Theseider Dupré, and Pietro Torasso. On the relationship between abduction and deduction. *Journal of Logic and Computation*, 1(5):661–690, 1991.

[17] Yu D., Sheikholeslami S., and A. Zhang. Findout: Finding outliers in very large datasets. In *Tech. Report, 99-03, Univ. of New York, Buffalo*, pages 1–19, 1999.

[18] E. Dantsin, T. Eiter, G. Gottlob, and A. Voronkov. Complexity and expressive power of logic programming. *ACM Computing Surveys*, 33(3):374–425, 2001.

[19] M. Denecker and C. Kakas. Abductive logic programming. *J. Log. Program.*, 44(1-3):1–4, 2000.

[20] Béatrice Duval and Pascal Nicolas. Learning default theories. In *ESCQARU*, pages 148–159, 1999.

[21] T. Eiter and G. Gottlob. On the computational cost of disjunctive logic programming: Propositional case. *Annals of Mathematics and Artificial Intelligence*, 15:289–323, 1995.

[22] T. Eiter, G. Gottlob, and N. Leone. Abduction from logic programs: semantics and complexity. *Theoretical Computer Science*, 189(1–2):129–177, 1997.

[23] Thomas Eiter and Georg Gottlob. The complexity of logic-based abduction. *J. of ACM*, 42(1):3–42, 1995.

[24] Thomas Eiter, Georg Gottlob, and Nicola Leone. Semantics and complexity of abduction from default theories. *Artificial Intelligence*, 90(1-2):177–223, 1997.

[25] David W. Etherington. Formalizing nonmonotonic reasoning systems. *Artificial Intelligence*, 31(1):41–85, 1987.

[26] Michael Gelfond and Vladimir Lifschitz. Classical negation in logic programs and disjunctive databases. *New Generation Computing*, 9:365–385, 1991.

[27] George Gottlob. Complexity results for nonmonotonic logics. *Journal of Logic and Computation*, 2(3):397–425, 1992.

[28] V. Hodge and J. Austin. A survey of outlier detection methodologies. *Artif. Intell. Rev.*, 22(2):85–126, 2004.

[29] Katsumi Inoue and Yoshimitsu Kudoh. Learning extended logic programs. In *IJCAI*, pages 176–181, 1997.

[30] Wen Jin, Anthony K. H. Tung, and Jiawei Han. Mining top-n local outliers in large databases. In *Proceedings of the ACM Int. Conf. on Knowledge Discovery and Data Mining*, pages 293–298, San Francisco, California, USA, 2001.

[31] D. S. Johnson. A catalog of complexity classes. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science,* Volume A: Algorithms and Complexity, chapter 9, pages 67–161. Elsevier and The MIT Press (co-publishers), 1990.

[32] A. Kakas, R. Kowalski, and F. Toni. *The role of abduction in logic programming*. Kakas, A. C., Kowalski, R. A. and Toni, F.: The role of abduction in logic programming, Handbook of logic in Artificial Intelligence and Logic Programming 5, Oxford University Press, 1998, 235–324. 1998.

[33] Henry A. Kautz and Bart Selman. Hard problems for simple default logics. *Artificial Intelligence*, 49(1-3):243–279, 1991.

[34] E. Knorr and R. Ng. Algorithms for mining distance-based outliers in large datasets. In *Proc. Int. Conf. on Very Large Databases*, pages 392–403, New York, USA, 1998.

[35] E. Knorr, R. Ng, and V. Tucakov. Distance-based outlier: algorithms and applications. *VLDB Journal*, 8(3-4):237–253, 2000.

[36] N. Lavrac and S. Dzeroski. *Inductive Logic Programming: Techniques and Applications*. Ellis Horwood, New York, 1994.

[37] W. Lee, S.J. Stolfo, and K.W. Mok. Mining audit data to build intrusion detection models. In *Proc. ACM Int. Conf on Knowledge Discovery and Data Mining*, pages 66–72, New York, USA, 1998.

[38] N. Leone, G. Pfeifer, W. Faber, T. Eiter, G. Gottlob, S. Perri, and F. Scarcello. The dlv system for knowledge representation and reasoning. *ACM Transactions on Computational Logic*, To appear.

[39] F. Lin and J.-H. You. Abduction in logic programming: A new definition and an abductive procedure based on rewriting. In *IJCAI*, pages 655–666, 2001.

[40] F. Lin and Y. Zhao. ASSAT: Computing answer sets of a logic program by SAT solvers. In *Proc. of the 8th National Conference on Artificial Intelligence and Fourteenth Conference on Innovative Applications of Artificial Intelligence (AAAI/IAAI-02)*, pages 112–118, Menlo Parc, CA, USA, 2002.

[41] John McCarthy. Circumscription - a form of non-monotonic reasoning. *Artificial Intelligence*, 13(1-2):27–39, 1980.

[42] Pascal Nicolas and Béatrice Duval. Representation of incomplete knowledge by induction of default theories. In *LPNMR*, pages 160–172, 2001.

[43] I. Niemelä and P. Simons. Smodels: An implementation of the stable model and well-founded semantics for normal LP. In *Proc. of the 4th International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR)*, pages 420–429, Berlin, 1997.

[44] Luigi Palopoli, Simona E. Rombo, and Giorgio Terracina. Flexible pattern discovery with (extended) disjunctive logic programming. In *ISMIS*, pages 504–513, 2005.

[45] Christos H. Papadimitriou. *Computatational Complexity*. Addison-Wesley, Reading, Mass., 1994.

[46] Judea Pearl. System z: a natural ordering of defaults with tractable applications to nonmonotonic reasoning. In *Proceedings of the 3rd Conference on Theoretical Aspects of Reasoning about Knowledge*, pages 121–135, Monterey, CA, 1990.

[47] David Poole. Normality and faults in logic-based diagnosis. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 1304–1310, Detroit, Michigan, USA, 1989.

[48] S. Ramaswamy, R. Rastogi, and K. Shim. Efficient algorithms for mining outliers from large data sets. In *Proc. ACM Int. Conf. on Managment of Data*, pages 427–438, Dallas, Texas, USA, 2000.

[49] Ray Reiter, 1992.

[50] Raymond Reiter. A logic for default reasoning. *Artificial Intelligence*, 13(1-2):81–132, 1980.

[51] S. Rosset, U. Murad, E. Neumann, Y. Idan, and G. Pinkas. Discovery of fraud rules for telecommunications-challenges and solutions. In *Proc. ACM Int. Conf on Knowledge Discovery and Data Mining*, pages 409–413, San Diego, California, USA, 1999.

[52] C. Sakama and K. Inoue. On the equivalence between disjunctive and abductive logic programs. In *International Conference on Logic Programming*, pages 489–503, 1994.

[53] C. Sakama and K. Inoue. Abductive logic programming and disjunctive logic programming: their relationship and transferability. *Journal of Logic Programming*, 44(1-3):75–100, 2000.

[54] S. Sarawagi, R. Agrawal, and N. Megiddo. Discovery-driven exploration of olap data cubes. In *Proc. Sixth Int. Conf on Extending Database Thecnology*, Valencia, Spain, March 1998.

[55] Jonathan Stillman. The complexity of propositional default logics. In *Proceedings of the 10th National Conference on Artificial Intelligence*, pages 794–800, San Jose, California, USA, July 1992.

[56] Z.R. Struzik and A. Siebes. Outliers detection and localisation with wavelet based multifractal formalism. In *Tech. Report INS-R0008, CWI, Amsterdam*, 2000.

[57] P.-N. Tan, M. Steinbach, and V. Kumar. *Introduction to Data Mining*. Addison-Wesley, 2005.

[58] Moshe Y. Vardi. The complexity of relational query language. In *Proc. 14th ACM Symp. on Theory of computing*, pages 137–146, San Francisco, CA, USA, May 1982.

[59] A. Zhang and W. Marek. On the classification and existence of structures in default logic. *Fundamenta Informaticae*, 13(4):485–499, 1990.