Outlier Detection using Disjunctive Logic Programming

Fabrizio Angiulli[1] and **Rachel Ben-Eliyahu - Zohary**[2] and **Luigi Palopoli**[3]

Abstract. Assume your general knowledge about the world is encoded in a disjunctive logic program P via Answer Set Semantics. Then, assume you get factual evidence about some aspects of the current status of the world encoded in a second disjunctive logic program, say P'. A fundamental question to be answered is the following: "Does the general knowledge encoded in P agree with the evidence about the world as encoded in P'"? In this paper we first define a formal framework suitable to discuss the question and then illustrate how difficult it is to answer the question.

1 Introduction

Assume your general knowledge about the world is encoded in a disjunctive logic program P via Answer Set Semantics. Then, assume you get factual evidence about some aspects of the current status of the world encoded in a second disjunctive logic program, say P'. A fundamental question to be answered is the following: "Does the general knowledge encoded in P agree with the evidence about the world as encoded in P'"? In the following, we shall formally state such a question by defining the concepts of an *outlier* and an *outlier witness*. Intuitively, an outlier is a piece of knowledge from the currently acquired evidence (encoded as a disjunctive logic program) which is not coherent with the given background theory (this latter too encoded in the form of a disjunctive logic program) and the witness represents the clue for such incoherence. In this sense, an outlier denotes some factual properties that cannot be logically justified in terms of the background theory.

Clear enough, detecting outliers can be useful for instance in order to be prompted to revise the background knowledge base as determined by the newly acquired knowledge or, vice versa, simply to get evidence that something abnormal is taking place. Note that in our approach such "exceptional" pieces of knowledge are not explicitly listed in the background theory as "abnormals", as is usually done with logic-based abduction [14, 3, 17, 6, 10, 18, 12]. Rather, such pieces of knowledge (the outliers) are characterized as "exceptional" since they denote evidence for which no logical justification is found in the background theory at hand.

Just to give a simple example of the matter, consider a situation where someone's car brakes work properly but they make a strange noise and that noise is not, as expected, justified in a theory describing the normal behavior of the car brakes. In this case, the car brakes would be identified as outliers and their noise is the associated witness.

[1] ICAR-CNR, Via Pietro Bucci 41C, 87030 Rende (CS), Italy, email: angiulli@icar.cnr.it

[2] Comm. Systems Engineering Dept., Ben-Gurion University of the Negev, Beer-Sheva 84105, Israel, email: rachel@bgumail.bgu.ac.il

[3] DEIS, Università della Calabria, Via P. Bucci 41C, 87030 Rende, Italy, email: palopoli@deis.unical.it

The aim of this paper is to formally state the concept of outlier and outlier witness in the context of disjunctive logic programming. This means that both the background knowledge, that is, what is known in general about the world and the factual evidences, that is, what is currently perceived about (possibly a portion of) the world are encoded in the form of disjunctive logic programs.

Before closing this introductory section, we briefly present next some related works.

These authors investigated outlier detection in the context of default logics [1]. In that context we have a propositional default theory $\Delta = (D, W)$, where D is a set of defaults and W is a set of propositional formulas. An outlier is then defined as a literal in W which is not justified in Δ with respect to a witness $S \subseteq W$. In that paper, the complexity of singling out outliers was studied for several fragments of Reiter's propositional default logics [16], the resulting complexity ranging from P to Σ_3^{P}.

It is well known that unlike extended logic programs with no disjunctive heads, disjunctive logic programs (DLP) under stable model semantics cannot be viewed as a subset of default logic and hence have a significant role as a KR tool. The so called *extended logic programs* (logic programs with two kinds of negation: negation as failure and classical negation) can be embedded very naturally into default theories and therefore can be considered as a subset of default logic. On the other hand, disjunctive extended logic programs cannot be viewed as a subset of default logic although default logic in its full volume does include disjunction. This is well explained in the paper by Gelfond and Lifschitz [7] where stable model semantics for DLP was introduced. For example, a DLP having a single rule $\{P|Q \longleftarrow\}$ has two models: $\{P\}$ and $\{Q\}$, while a default logic where W is empty and D has a single default $\frac{True:True}{P \vee Q}$ has a single extension, which is the logical closure of $\{P \vee Q\}$. Part of the motivation for developing stable model semantics for DLP was the limitations of default logic in handling disjunctive knowledge (See a paper by Poole [15]).

As already noted elsewhere [1], there is a main semantically significant difference between abduction and outlier detection. In abduction, we have a set of manifestations (either "exceptional" or not) and a set of justifications for them is to be find. Intuitively this set will include logical sentences which, if added to the given theory, would make this latter imply the manifestations. Outlier detection, on the other hand, is closer to *learning* which manifestations belonging to a given set thereof are to be considered exceptional (the outliers) and to find justifications (the witnesses) to their exceptionality.

A second bunch of work which is related to outlier detection is that on logic-based abduction from disjunctive logic programs [6]. In the context of logic programming, abduction is a well-studied issue (see [4] for a detailed survey). Abduction in the logic-programming framework was explored in two directions. The first line of work has used logic programs as an AI tool for knowledge representation and reasoning about abduction, and the second line of work has used the concept of abduction for defining semantics of logic programs. With respect to DLP, there was also research on the relationship between semantics of DLP and abduction-based semantics of logic programs (see for example [18]) . As explained above, there is a major difference between abduction and outlier detection. In addition, it is clear that in this paper no attempt is done to develop yet another semantics to DLP. Hence, while the work on abductive logic programs might look at a first glance very relevant to the work reported here, there is clearly a major difference between the two lines of research.

The rest of this paper is organized as follows. In Section 2, we give preliminary definitions. In Section 3, we introduce the formal definition of the concept of outlier. Section 4 analyzes the complexity of detecting outliers in extended disjunctive logic programs. Conclusions are given in Section 5.

2 Preliminaries

2.1 Extended Disjunctive Logic Programs

A propositional *Extended Disjunctive Logic Program* (EDLP) is a collection of rules of the form

$$L_1 | \ldots | L_k \leftarrow L_{k+1}, \ldots, L_m, not\ L_{m+1}, \ldots, not\ L_n, \qquad n \geq m \geq k \geq 0,$$

where the symbol "not" denotes negation by default, and each L_i is a literal, i.e. an expression of the form ℓ or $\neg\ell$ with ℓ a propositional letter and the symbol "$\neg$" denotes classical negation. If $k = 0$ then the rule is called *integrity clause*.

Answer sets [8] of an EDLP P are defined as follows. Let $\mathrm{Lit}(P)$ denote the set of all the literals obtained using the propositional letters occurring in P. Let a *context* [2] be any subset of $\mathrm{Lit}(P)$. Let P be a *negation-by-default-free* EDLP. Call a context S *closed under* P iff for each rule $L_1 | \ldots | L_k \leftarrow L_{k+1}, \ldots, L_m$ in P, if $L_{k+1}, \ldots, L_m \in S$, then for some $i = 1, \ldots, k$, $L_i \in S$. An *answer set* of P is any minimal context S such that (1) S is closed under P and (2) if S is inconsistent, that is if there exists a propositional letter ℓ such that both $\ell \in S$ and $\neg\ell \in S$, then $S = \mathrm{Lit}(P)$. An answer set of a general EDLP is 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 subformulae of the form $not\ L$ of the bodies of the remaining rules. Any context S which is an answer set of $Red(P, S)$ is an *answer set* of P. By $\mathrm{ANSW}(P)$ we denote the collection of all consistent answer sets of an EDLP P. An EDLP P is ANSW-consistent iff $\mathrm{ANSW}(P) \neq \emptyset$, and P entails a propositional formula F, written $P \models F$, iff $F \in Cn(S)$ for each $S \in \mathrm{ANSW}(P)$, where $Cn(\mathcal{F})$ denotes the set of all logical consequences from the set of formulae $\mathcal{F}$.

Because of both its declarativity and its expressive power, EDLP is largely considered a convenient tool for representing and manipulating complex knowledge [11].

2.2 Complexity Theory

We recall some basic definitions about complexity theory, particularly, the polynomial time hierarchy. The reader is referred to [9, 13] for more on this.

The class P is the set of decision problems that can be answered by a deterministic Turing machine in polynomial time.

The classes Σ_k^P and Π_k^P, forming the *polynomial hierarchy*, are defined as follows: $\Sigma_0^P = \Pi_0^P = \mathrm{P}$ and for all $k \geq 1$, $\Sigma_k^P = \mathrm{NP}^{\Sigma_{k-1}^P}$, and $\Pi_k^P = \text{co-}\Sigma_k^P$. Σ_k^P models computability by a nondeterministic polynomial time Turing machine which may use an oracle, that is, loosely speaking, a subprogram, that can be run with no computational cost, for solving a problem in Σ_{k-1}^P. 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. The class D_k^P, $k \geq 1$, is defined as the class of problems that consist of the conjunction of two independent problems from Σ_k^P and Π_k^P, respectively. Note that, for all $k \geq 1$, $\Sigma_k^P \subseteq D_k^P \subseteq \Sigma_{k+1}^P$.

Let Γ denote the set of all the strings over a given finite set of symbols. A function $f : \Gamma \mapsto \Gamma$ is said to be *polynomial time computable* if there exists a deterministic polynomial time Turing machine that computes it. Let L_1, L_2 be two subsets of Γ. A polynomial time computable function $\tau : \Gamma \mapsto \Gamma$ is called a *polynomial time transformation* from L_1 to L_2 if for each $x \in \Gamma$ the following holds: $x \in L_1$ iff $\tau(x) \in L_2$. The *language* $L(A)$ associated to a decision problem A, is the set consisting of the strings $x \in \Gamma$ such that A returns "yes" on the input x. A problem A is *polynomially reducible* to a problem B if there exists a polynomial time transformation from $L(A)$ to $L(B)$. A 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 Σ_k^P-complete problem is to decide the validity of a formula $QBE_{k,\exists}$, that is, a formula of the form $\exists X_1 \forall X_2 \ldots QX_k f(X_1, \ldots, X_k)$, where Q is $\exists$ if k is odd and is $\forall$ 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 $X_1, \ldots, X_k$. Analogously, the validity of a formula $QBE_{k,\forall}$, that is a formula of the form $\forall X_1 \exists X_2 \ldots QX_k f(X_1, \ldots, X_k)$, where Q is $\forall$ if k is odd and is $\exists$ if k is even, is complete for Π_k^P. Deciding the conjunction $\Phi \wedge \Psi$, where Φ is a $QBE_{k,\exists}$ formula and Ψ is a $QBE_{k,\forall}$ formula, is complete for D_k^P.

3 Defining Outliers

Definition 1 Let P^{rls} be an EDLP, encoding general knowledge about the world, called *rules program*, and let P^{obs} a negation-by-default-free EDLP, encoding some aspects of the current status of the world, called *observations program*.

A *rule-observation pair* $\mathcal{P}$ is a pair $\mathcal{P} = \langle P^{\mathrm{rls}}, P^{\mathrm{obs}} \rangle$ composed by a rules program and an observations program. Intuitively, a rule-observation pair relates the general knowledge encoded in P^{rls} with the evidence about the world encoded in P^{obs}.

In the following, we will denote by $P(\mathcal{P})$ the EDLP $P^{\mathrm{rls}} \cup P^{\mathrm{obs}}$.

Definition 2 Let $\mathcal{P} = \langle P^{\mathrm{rls}}, P^{\mathrm{obs}} \rangle$ be a rule-observation pair, and let $\mathcal{O} \subseteq P^{\mathrm{obs}}$ be a set of clauses. If there exists a non empty set of clauses $\mathcal{W} \subseteq P^{\mathrm{obs}}$ such that:

1. $\mathrm{ANSW}(P(\mathcal{P})_{\mathcal{W}}) \models \neg\mathcal{W}$, and
2. $\mathrm{ANSW}(P(\mathcal{P})_{\mathcal{W},\mathcal{O}}) \not\models \neg\mathcal{W}$,

where $P(\mathcal{P})_{\mathcal{W}} = P(\mathcal{P}) \setminus \mathcal{W}$ and $P(\mathcal{P})_{\mathcal{W},\mathcal{O}} = P(\mathcal{P})_{\mathcal{W}} \setminus \mathcal{O}$, then we say that $\mathcal{O}$ is an *outlier* in $\mathcal{P}$ and $\mathcal{W}$ is an *outlier witness* for $\mathcal{O}$ in $\mathcal{P}$ under the ANSW semantics.

In order to motivate the above definition and make it easy to understand, we look at the following example. Consider the rule-observation pair $\mathcal{P} = \langle P^{\rm rls}, P^{\rm obs}\rangle$, where $P^{\rm obs} = \{Bird(Tweety), \neg Fly(Tweety)\}$ and $P^{\rm rls}$ is the set

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

If we look carefully at the program $P(\mathcal{P})$, we note that Tweety not flying is quite strange, since we know that birds fly and Tweety is a bird. If we are trying to nail down what makes Tweety an exception, we notice that if we would have dropped the observation $\neg Fly(Tweety)$, we would have concluded the exact opposite, namely, that Tweety does fly. Thus, $\neg Fly(Tweety)$ is a witness according to the definition above. 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 the outlier.

One may wonder whether it is always possible to find outliers which are singleton sets of clauses. The following example shows that this is not the case. Consider the following rule-observation pair $\mathcal{P}$:

$$P^{\rm obs} \left\{ \begin{array}{l} w \\ o_1 \\ o_2 \end{array}\right. \qquad P^{\rm rls} \left\{ \begin{array}{l} \leftarrow q \\ q \leftarrow o_1, not\ w \\ q \leftarrow o_2, not\ w \\ w \leftarrow not\ o_1, not\ o_2 \end{array}\right.$$

We note that $P(\mathcal{P})$ is consistent, and that $\{w\}$ is an outlier witness for $\{o_1, o_2\}$. Furthermore, nor $\{o_1\}$ neither $\{o_2\}$ are outliers.

4 Complexity Results

In this section we discuss the complexity of detecting and verifying outliers (queries $Q0 - Q3$ below).

In order to state the computational complexity of detecting outliers, in the rest of the work we refer to the following problems (also referred to as *queries*) defined for an input rule-observation pair $\mathcal{P} = \langle P^{\rm rls}, P^{\rm obs}\rangle$:

- $Q0(\rm ANSW)$: does there exist an outlier in $\mathcal{P}$ under the ANSW semantics ?
- $Q1(\rm ANSW)$: given $\mathcal{O} \subseteq P^{\rm obs}$, is there any outlier witness set for $\mathcal{O}$ in $\mathcal{P}$ under the ANSW semantics ?
- $Q2(\rm ANSW)$: given $\mathcal{W} \subseteq P^{\rm obs}$, is $\mathcal{W}$ a witness for any outlier $\mathcal{O}$ in $\mathcal{P}$ under the ANSW semantics ?
- $Q3(\rm ANSW)$: given $\mathcal{O}, \mathcal{W} \subseteq P^{\rm obs}$, is $\mathcal{O}$ an outlier in $\mathcal{P}$ with witness $\mathcal{W}$ under the ANSW semantics ?

Let L be a consistent set of literals. We denote with $\mathcal{T}_L$ the truth assignment on the set of letters occurring in L such that, for each positive literal $p \in L$, $\mathcal{T}_L(p) = \mathbf{true}$, and for each negative literal $\neg p \in L$, $\mathcal{T}_L(p) = \mathbf{false}$.

Let L be a set of literals. Then we denote with 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 of the set $\{x_1, \ldots, x_n\}$ of boolean variables. Then we denote with $\mathrm{Lit}(T)$ 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 $i = 1, \ldots, n$.

Theorem 1 *$Q0(\rm ANSW)$ is Σ_3^P-complete under polynomial time transformations. Σ_3^P-completeness holds even if classical negation does not occur in the rule-observation pair $\mathcal{P}$ and $\mathcal{P}$ contains no integrity clauses.*

Proof: (Membership) Given a rule-observation pair $\mathcal{P} = \langle P^{\rm rls}, P^{\rm obs}\rangle$ such that classical negation does not occur in $\mathcal{P}$, we must show that there exist two sets $\mathcal{W}, \mathcal{O} \subseteq P^{\rm obs}$ such that $\mathrm{ANSW}(P(\mathcal{P})_{\mathcal{W}}) \models \neg\mathcal{W}$ (query q') and $\mathrm{ANSW}(P(\mathcal{P})_{\mathcal{W},\mathcal{O}}) \not\models \neg\mathcal{W}$ (query q''). Query q' is Π_2^P-complete, while query q'' is Σ_2^P-complete [5]. Thus, we can build a polynomial-time nondeterministic Turing machine with a Σ_2^P oracle, solving query $Q0(\rm ANSW)$ as follows: the machine guesses both the sets $\mathcal{W}$ and $\mathcal{O}$ and then solves queries q' and q'' by two calls to the oracle.

(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 $C_r = t_{r,1} \vee t_{r,2} \vee t_{r,3}$, and each $t_{r,1}, t_{r,2}, t_{r,3}$ is a literal on the set $X \cup Y \cup Z$, for $r = 1, \ldots, N$. We associate with Φ the rule-observation pair $\mathcal{P}(\Phi) = \langle P^{\rm rls}(\Phi), P^{\rm obs}(\Phi)\rangle$, where $P^{\rm obs}(\Phi) = \{o, x_0, x_1, \ldots, x_n\}$ consists of the letters in the set X plus the new letters x_0 and o distinct from those occurring in Φ, and $P^{\rm rls}(\Phi)$ is

$$\begin{array}{ll} a_i \leftarrow not\ x_i & 1 \leq i \leq n \\ y_j \mid b_j \leftarrow & 1 \leq j \leq m \\ z_k \mid c_k \leftarrow & 1 \leq k \leq l \\ z_k \leftarrow f & 1 \leq k \leq l \\ c_k \leftarrow f & 1 \leq k \leq l \\ f \leftarrow \sigma(\neg t_{r,1}), \sigma(\neg t_{r,2}), \sigma(\neg t_{r,3}) & 1 \leq r \leq N \\ f \leftarrow not\ f & \\ f \leftarrow x_0 & \\ f \leftarrow not\ o & \end{array}$$

where also $A = a_1, \ldots, a_n$, $B = b_1, \ldots, b_m$, and $C = c_1, \ldots, c_l$ and f are new letters distinct from those occurring in Φ, and $\sigma : X \cup Y \cup Z \mapsto X \cup Y \cup Z \cup A \cup B \cup C$ is the following mapping:

$$\sigma(t) = \left\{ \begin{array}{ll} a_i & \text{if } t = \neg x_i,\ 1 \leq i \leq n \\ b_j & \text{if } t = \neg y_j,\ 1 \leq j \leq m \\ c_k & \text{if } t = \neg z_k,\ 1 \leq k \leq l \\ t & \text{otherwise} \end{array}\right.$$

Clearly, $\mathcal{P}(\Phi)$ can be built in polynomial time. Now we show that Φ is valid iff there exists an outlier in $\mathcal{P}(\Phi)$.

Given a truth assignment T on a subset of $X \cup Y \cup Z$, let $\mathcal{I}(T)$ denote the context $\mathrm{Lit}(T)^+ \cup \sigma(\mathrm{Lit}(T)^-)$.

($\Rightarrow$) Suppose that Φ is valid. 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 $\mathcal{W} = \{x_0\} \cup \neg(\mathrm{Lit}(T_X)^-)$, now we show that $P(\mathcal{P}(\Phi))_{\mathcal{W}}$ is inconsistent. By contradiction, suppose that there exists an answer set S of $P(\mathcal{P}(\Phi))_{\mathcal{W}}$, then $S = \mathcal{I}(T_X) \cup \mathcal{I}(T_Y) \cup Z \cup C \cup \{f, o\}$, where T_Y denotes an arbitrary truth assignment to the set of variables in Y. Note that $f \in S$ is assured by the clause $f \leftarrow not\ f$. As S is an answer set of $P(\mathcal{P}(\Phi))_{\mathcal{W}}$, hence a minimal context closed under $Red(P(\mathcal{P}(\Phi))_{\mathcal{W}}, S)$, it follows that for each truth assignment T_Z to the variables in the set Z, the subset $S' = \mathcal{I}(T_X) \cup \mathcal{I}(T_Y) \cup \mathcal{I}(T_Z) \cup \{o\}$ of S is not a context closed under $Red(P(\mathcal{P}(\Phi))_{\mathcal{W}}, S)$. Thus, for each T_Z there exists an $r \in \{1, \ldots, N\}$ such that $\sigma(\neg t_{r,1}), \sigma(\neg t_{r,2}), \sigma(\neg t_{r,3}) \in S'$. We can conclude that there exists an answer set S of $P(\mathcal{P}(\Phi))_{\mathcal{W}}$ iff T_X satisfies $\exists Y \forall Z \bigvee_{r=1}^{N} (\neg t_{r,1} \wedge \neg t_{r,2} \wedge \neg t_{r,3}) \equiv \exists Y \forall Z \neg f(X, Y, Z)$, which contradicts the fact that T_X satisfies $\forall Y \exists Z f(X,Y,Z)$. Hence $P(\mathcal{P}(\Phi))_{\mathcal{W}}$ is inconsistent and $\mathrm{ANSW}(P(\mathcal{P}(\Phi))_{\mathcal{W}}) \models \neg\mathcal{W}$. Let $\mathcal{O} = \{o\}$, then $P(\mathcal{P}(\Phi))_{\mathcal{W},\mathcal{O}}$ is consistent and such that $\mathrm{ANSW}(P(\mathcal{P}(\Phi))_{\mathcal{W},\mathcal{O}}) \not\models \neg\mathcal{W}$. Thus, $\mathcal{O}$ is an outlier with outlier witness set $\mathcal{W}$.

($\Leftarrow$) Suppose that there exists an outlier $\mathcal{O} \subseteq P^{\mathrm{obs}}(\Phi)$ with witness $\mathcal{W} \subseteq P^{\mathrm{obs}}(\Phi)$ in $\mathcal{P}$. As classical negation does not occur in $P(\mathcal{P}(\Phi))$, then it is the case that $\mathrm{ANSW}(P(\mathcal{P}(\Phi))_{\mathcal{W}}) \models \neg\mathcal{W}$ iff $P(\mathcal{P}(\Phi))_{\mathcal{W}}$ is inconsistent. Thus, $\mathcal{W} \subseteq X \cup \{x_0\}$ and, from what is stated above, $P(\mathcal{P}(\Phi))_{\mathcal{W}}$ is inconsistent implies that $T_{(X\setminus\mathcal{W})\cup\neg\mathcal{W}}$ satisfies $\forall Y \exists Z f(X,Y,Z)$, i.e. that Φ is valid. To conclude, the set $\mathcal{O} = \{o\}$ is always an outlier having such a witness. □

Theorem 2 $Q1(\mathrm{ANSW})$ *is* Σ_3^P*-complete under polynomial time transformations.* Σ_3^P*-completeness holds even if classical negation does not occur in the rule-observation pair* $\mathcal{P}$ *and* $\mathcal{P}$ *contains no integrity clauses.*

Proof: The proof is similar to that of Theorem 1. □

Theorem 3 $Q2(\mathrm{ANSW})$ *is* D_2^P*-complete under polynomial time transformations.* D_2^P*-completeness holds even if classical negation does not occur in the rule-observation pair* $\mathcal{P}$ *and* $\mathcal{P}$ *contains no integrity clauses.*

Proof sketch: (Membership) Given the rule-observation pair $\mathcal{P} = \langle P^{\mathrm{rls}}, P^{\mathrm{obs}} \rangle$, with P^{rls} and P^{obs} containing no classical negation and no integrity clauses, and a subset $\mathcal{W} \subseteq P^{\mathrm{obs}}$, we should verify that there exists $\mathcal{O} \subseteq P^{\mathrm{obs}}$ such that $\mathrm{ANSW}(P(\mathcal{P})_{\mathcal{W}}) \models \neg\mathcal{W}$ (query q') and that $\mathrm{ANSW}(P(\mathcal{P})_{\mathcal{W},\mathcal{O}}) \not\models \neg\mathcal{W}$ (query q''). We have already noted that query q' is Π_2^P-complete. As for query 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 $\mathcal{O} \subseteq P^{\mathrm{obs}}$ and a consistent context S of $P(\mathcal{P})_{\mathcal{W},\mathcal{O}}$ such that $\neg\mathcal{W} \not\subseteq S$, verifies that S is closed under $Red(P(\mathcal{P})_{\mathcal{W},\mathcal{O}}, S)$, and decides whether $S' \subset S$ exists such that S' is closed under $Red(P(\mathcal{P})_{\mathcal{W},\mathcal{O}}, S)$ with a call to the NP oracle. Hence, we have to decide the conjunction $q' \wedge q''$, i.e. a D_2^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 without classical negation and integrity clauses. W.l.o.g. assume that P' and P'' contain no common letters and also that they do not contain the letters w and o. Consider the EDLPs $P^{\mathrm{rls}} = \{\mathbf{h}(r) \leftarrow \mathbf{b}(r), o : r \in P'\} \cup \{\mathbf{h}(r) \leftarrow \mathbf{b}(r), not\ o : r \in P''\}$ and $P^{\mathrm{obs}} = \{w, o\}$, and the rule-observation pair $\mathcal{P} = \langle P^{\mathrm{rls}}, P^{\mathrm{obs}} \rangle$. Now we show that P' is inconsistent (a Σ_2^P-complete check, see [5], Theorem 39) and P'' is consistent (a Π_2^P-complete check, see [5], Theorem 40) iff $\{o\}$ is an outlier with witness $\{w\}$ in $\mathcal{P}$ under the ANSW semantics. This result follows immediately by noting that $\mathrm{ANSW}(P') = \mathrm{ANSW}(P(\mathcal{P})_{\{w\}})$ and $\mathrm{ANSW}(P'') = \mathrm{ANSW}(P(\mathcal{P})_{\{w\},\{o\}})$. □

Theorem 4 $Q3(\mathrm{ANSW})$ *is* D_2^P*-complete under polynomial time transformations.* D_2^P*-completeness holds even if classical negation does not occur in the rule-observation pair* $\mathcal{P}$ *and* $\mathcal{P}$ *contains no integrity clauses.*

Proof: The proof is analogous to that of Theorem 3. □

5 Conclusions

In this paper, we have defined outliers and associated witnesses. These concepts serve the purpose of denoting situations where some piece of knowledge coming from factual observations is not justified in the context of the general knowledge about the world as encoded in a given background theory. In the framework we have presented, both the background knowledge and the observations are encoded in the form of ELDP. We have also studied the complexity of detecting outlier and shown that this problem is quite demanding from the computational viewpoint. Presently, we are working towards extending the presented framework to other languages and semantics and singling out significant tractable cases of the outlier detection problem.

Acknowledgements. This research has been partially funded by MIUR in the context of the research programme CFIN'03 "Tecniche di induzione di regole, metaquerying ed estrazione di pattern strutturati su database biologici".

REFERENCES

[1] 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)*, pp. 833–838, (2003).

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

[3] L Console, D. Theseider Dupré, and P. Torasso, 'On the relationship between abduction and deduction', *Journal of Logic and Computation*, **1**(5), 661–690, (1991).

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

[5] 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).

[6] T. Eiter, G. Gottlob, and N. Leone, 'Abduction from logic programs: semantics and complexity', *Theoretical Computer Science*, **189**(1–2), 129–177, (1997).

[7] M. Gelfond and V. Lifschitz, 'The stable model semantics for logic programming', in *Proceedings of the Fifth International Conference on Logic Programming*, pp. 1070–1080, (1988).

[8] M. Gelfond and V. Lifschitz, 'Classical negation in logic programs and disjunctive databases', *New Generation Comput.*, **9**(3-4), 365–386, (1991).

[9] D.S. Johnson, 'A catalog of complexity classes', in *Handbook of Theoretical Computer Science*, Volume A: Algorithms and Complexity, ed., J. van Leeuwen, chapter 9, 67–161, Elsevier and The MIT Press (copublishers), (1990).

[10] 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.

[11] N. Leone, G. Pfeifer, W. Faber, T. Eiter, G. Gottlob, S. Perri, and F. Scarcello, 'The dlv system for knowledge representation and reasoning', in *Technical Report cs.AI/0211004, arXiv.org*, (2002).

[12] F. Lin and J.-H. You, 'Abduction in logic programming: A new definition and an abductive procedure based on rewriting', in *IJCAI*, pp. 655–666, (2001).

[13] C.H. Papadimitriou, *Computatational Complexity*, Addison-Wesley, Reading, Mass., 1994.

[14] D. Poole, 'Normality and faults in logic-based diagnosis', in *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 1304–1310, (1989).

[15] D. Poole, 'What the lottery paradox tells us about default reasoning', in *KR*, pp. 333–340, (1989).

[16] R. Reiter, 'A logic for default reasoning', *Artificial Intelligence*, **13**, 81–132, (1980).

[17] C. Sakama and K. Inoue, 'On the equivalence between disjunctive and abductive logic programs', in *International Conference on Logic Programming*, pp. 489–503, (1994).

[18] 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).