Computing Abductive Explanations

Luciano Caroprese, Ester Zumpano, Bart Bogaerts

Abstract—We study the computation of constrained explanations in the framework of abductive logic programming. A general characteristic of abductive reasoning is the existence of multiple abductive explanations. Therefore, identifying a subclass of "preferred explanations" is a relevant problem. A typical approach is to "prefer" explanations that are, in some sense, *simple*. Several concepts of simplicity were considered in the literature, most notably those based on minimality with respect to inclusion and cardinality. We adopt, as a measure of the quality of an explanation, its degree of *arbitrariness* that can be briefly described as the number of arbitrary assumptions that have been made to derive the explanation. The more arbitrary the explanation, the less appealing it is, with explanations having no arbitrariness, called *constrained*, being the preferred ones. In this paper we present a technique that, for a special class of theories, computes constrained explanations. It is based on a rewriting of the theory and the observation into a disjunctive logic program with negation so that the constrained explanations correspond to a subset of its stable models. The proposed technique lays the foundation for using ASP solvers to compute constrained explanations.

Index Terms—Artificial Intelligence,Computing Methodologies, Knowledge Representation Formalisms and Methods.

INTRODUCTION

In the context of logic programming, abduction was first studied by Eshghi and Kowalski [1], and then by Kakas and Mancarella [2] under the *brave reasoning* variant of the stable-model semantics. That work established *abductive logic programming* as an important subarea of abduction. In abductive logic programming, the background theory is represented by a logic program, often with negation in the bodies and disjunction in the heads, and any of the standard logic programming semantics could be used to provide the meaning [3], [4]. A general characteristic of abductive reasoning is the existence of multiple abductive explanations. These explanations are typically not equally compelling. Therefore, identifying a subclass, possibly narrow, of "preferred" explanations is an important problem. A typical approach is to identify as preferred those explanations that are, in some sense, *simple*, rooted in objects present in the background theory and an observation. Several concepts of simplicity were considered in the literature, most notably those based on minimality with respect to inclusion and cardinality. This paper continues the work of Caroprese et al. [5], who studied the problem of "preferred" explanations in the framework of abductive logic programming. Caroprese et al. [5] proposed an orthogonal measure of the simplicity (quality) of an explanation which they called the degree of *arbitrariness*. The less arbitrary the explanation (the lower its degree of arbitrariness), the more appealing it is, with explanations having no arbitrariness, called *constrained*, being the most preferred. A constrained explanation connects the structural information present in the theory and the knowledge embedded in the observation in a non-arbitrary (*constrained*) way, without assuming the existence of new objects. Informally, it makes no arbitrary assumptions. Let us consider the following scenario.

Example 1. Let us assume that a security breach at a component of an information system may only occur when a person with an account makes an unapproved access. Regular staff personnel have accounts on the system if they complete training and have their security clearance current. Visitors may also be granted an account but only with an approval by the head of the IT department. This situation can be described by the following program:

$$\begin{aligned} account(X) &\leftarrow staff(X), trained(X), current(X).\\ account(X) &\leftarrow visitor(X), approved(X).\\ breach(W) &\leftarrow unapprovedAccess(W,X), account(X).\end{aligned}$$

Let us also assume that *tom* and *mary* are regular staff members and *dan* is a visitor. Finally, let us assume that the system has information that *tom* completed training. That is, the program also contains the facts:

$$staff(tom).\;\; staff(mary).\;\; visitor(dan).\;\; trained(tom).$$

If we observe *breach*(*warehouse*) (the security of *warehouse* was compromised), there are several possible explanations. Below we list some of them:

$$\begin{aligned} E_{tom} &= \{unapprovedAccess(warehouse,tom), current(tom)\}\\ E_{mary} &= \{unapprovedAccess(warehouse,mary), trained(mary),\\ &\quad current(mary)\}\\ E_u &= \{unapprovedAccess(warehouse,u), staff(u), trained(u),\\ &\quad current(u)\},\end{aligned}$$

where u is a name in the domain,

$$\begin{aligned} E_{dan} &= \{unapprovedAccess(warehouse,dan), approved(dan)\}\\ E_v &= \{unapprovedAccess(warehouse,v), visitor(v),\\ &\quad approved(v)\},\end{aligned}$$

where v is a name in the domain,

$$\begin{aligned} E_{tom,dan} = \{&unapprovedAccess(warehouse,tom), current(tom),\\ &unapprovedAccess(warehouse,dan),\\ &approved(dan)\}.\end{aligned}$$ □

- L. Caroprese and E. Zumpano are with:
DIMES, University of Calabria, Via P. Bucci 42C, 87036 Rende, Italy.
E-mail: {l.caroprese,e.zumpano}@dimes.unical.it
B. Bogaerts is with:
AI Lab, Vrije Universiteit Brussel, Pleinlaan 9, 1050 Elsene, Belgium.
E-mail: bart.bogaerts@vub.be

Which of these explanations are more compelling or, to put it differently, more plausible than the others? Most approaches to the problem of selecting preferred explanations follow the Occam's principle of parsimony that entities should not be multiplied unnecessarily and that among possible explanations the simplest one tends to be the right one. However, simplicity is a notoriously complex concept and different formalizations of it are possible. They range from the subset minimality, to those that require minimum cardinality, minimum weight, or minimality under prioritization of individual hypotheses [6].

In Example 1, the explanations E_{tom}, E_{mary}, E_u, $u \notin \{tom, mary\}$, E_{dan}, and E_v, $v \neq dan$, are subset minimal and so, *preferred* under the subset minimality criterion. On the other hand, the explanation $E_{tom,dan}$ is not subset minimal. If we use a more restrictive criterion of minimum cardinality, the preferred explanations are E_{tom} and E_{dan}.

Let us assume that there are reasons to view each of the latter two as wrong (*tom* and *dan* can conclusively demonstrate they were not involved). Under the subset minimality criterion, we now prefer explanations E_{mary}, E_u, $u \notin \{tom, mary\}$, and E_v, $v \neq dan$, while under the minimum cardinality criterion we prefer E_{mary} and E_v, $v \neq dan$. Let us look more carefully at the explanations E_u, $u \notin \{tom, mary\}$, and E_v, $v \neq dan$. They select an *arbitrary* individual in the domain with no particular reason to choose one over another. On the other hand, the explanations E_{tom}, E_{dan} and E_{mary} connect the structural information present in the program and the knowledge provided by the observation in a non-arbitrary (*constrained*) way. About $E_{tom,dan}$, we observe that it implicitly provides two ways to derive the observation. Then in principle one of the two constants *tom* and *dan* could be replaced with a different arbitrary constant.

Caroprese et al. [5], [7] formalized these observations into the concept of the degree of arbitrariness. That degree is 1 for the explanations E_u, $u \notin \{tom, mary\}$, E_v, $v \neq dan$ and $E_{tom,dan}$ and it is 0 for the explanations E_{tom}, E_{dan}, E_{mary}; they are constrained. The principle of minimum arbitrariness can be used with all types of explanations and is "orthogonal" to other criteria one might consider when selecting preferred explanations such as the subset or cardinality minimality.

Abductive reasoning is the basis of many decisions that people make every day. Many of these decisions are critical because they affect their own life or that of other individuals. Think for example of the diagnosis process carried out by a doctor who must derive the causes of the pathology from the *symptoms* (*observation*) and his own *medical knowledge* (*theory*). It should be noted that in this case it is essential that the doctor returns a diagnosis as close as possible to the anamnesis carried out, without inventing hypothetical scenarios. A defendant's trial, is still another example of abductive reasoning. Jurors must consider the details of the offense (textit observation), the evidence collected, and their textit theory (textit knowledge) of the industry in which the accused is involved. Also in this case it is essential that the sentence is not based on arbitrary interpretations of the jurors.

This paper presents a technique to compute constrained explanations of an observation, following the theoretical framework developed by Caroprese et al. [5] for a subclass of abductive theories. It is based on a rewriting of the theory and the observation into a disjunctive logic program with negation. Stable models of this program correspond to constrained explanations.

Abductive Explanations

This section recalls the definitions or arbitrary and constrained explanations [5]. We consider a fixed vocabulary σ consisting of *relation*, *constant*, and *variable symbols*. We write $\mathscr{R}$, $\mathscr{C}$, and $\mathscr{V}$ for the sets of these symbols, respectively. We assume that $\mathscr{C}$ is a countable set. For a set $\mathscr{S} \subseteq \mathscr{R}$ of predicate symbols, we define $\mathscr{S}^{\mathscr{C}}$ to be the set of all ground atoms (*facts*) whose predicate symbols are in $\mathscr{S}$ (i.e. expressions $p(c_1, \ldots, c_k)$, where $p \in \mathscr{S}$ and all $c_i \in \mathscr{C}$). In particular, $\mathscr{R}^{\mathscr{C}}$ is the *Herbrand base* of σ and it is denoted as $\mathscr{H}$.

A *(disjunctive logic) rule* is an expression:

$$h_1(\overline{X_1}) \vee \ldots \vee h_n(\overline{X_n}) \leftarrow \mathscr{P}(\overline{X}, \overline{Y}), \mathscr{N}(\overline{Z})$$

where:

- $\overline{X_i}$, for all $i \in [1..n]$, $\overline{X}$, $\overline{Y}$ and $\overline{Z}$ are tuples of variables;
- each variable in $\overline{X_i}$ also occurs in $\overline{X}$, for all $i \in [1..n]$;
- each variable in $\overline{Z}$ also occurs in $\overline{X}$ or $\overline{Y}$;
- $h_i(\overline{X_i})$ is an atom, for all $i \in [1..n]$;
- $\mathscr{P}(\overline{X}, \overline{Y})$ is a conjunction of atoms and $\mathscr{N}(\overline{Z})$ is a conjunction of negative literals.

The disjunction $h_1(\overline{X_1}) \vee \ldots \vee h_n(\overline{X_n})$ and the conjunction $\mathscr{P}(\overline{X}, \overline{Y}), \mathscr{N}(\overline{Z})$ are respectively the *head* and the *body* of the rule. If $n = 0$, the head is denoted as $\perp$ and the rule is called *denial constraint*. A *normal rule* is a rule whose head consists of a single atom ($n = 1$). A *Horn rule* is a normal rule whose body is positive. The set $\mathscr{R}$ of predicate symbols in σ is commonly partitioned into two sets $\mathscr{R}_{\mathscr{I}}$ and $\mathscr{R}_{\mathscr{E}}$ of *intensional* and *extensional* predicate symbols, respectively. Programs are finite sets of rules, with the head predicate symbols from $\mathscr{R}_{\mathscr{I}}$, and facts over predicate symbols from $\mathscr{R}_{\mathscr{E}}$. A program $\mathscr{P}$ is *normal* (resp. *Horn*) if each rule $r \in \mathscr{P}$ is *normal* (resp. *Horn*). When describing programs, we use two shorthands:

1) $h(\overline{X}) \leftarrow \bigvee_{i \in [1..n]} (\mathscr{P}_i(\overline{X}, \overline{Y_i}), \mathscr{N}_i(\overline{Z_i}))$
   represents the set of rules:
   $\{h(\overline{X}) \leftarrow \mathscr{P}_i(\overline{X}, \overline{Y_i}), \mathscr{N}_i(\overline{Z}) \mid i \in [1..n]\}$;
2) $h_1(\overline{X_1}), \ldots, h_n(\overline{X_n}) \leftarrow \mathscr{P}(\overline{X}, \overline{Y}), \mathscr{N}(\overline{Z})$,
   where $\overline{X_i} \subseteq \overline{X}$, for all $i \in [1..n]$, stands for the set of rules:
   $\{h_i(\overline{X_i}) \leftarrow \mathscr{P}(\overline{X}, \overline{Y}), \mathscr{N}(\overline{Z}) \mid i \in [1..n]\}$.

With a little abuse of notation we also call *rule* each of these shorthands.

By S we denote a semantics of logic programs (for instance, the stable-model semantics). We assume that S is given in terms of subsets of $\mathscr{H}$. For a logic program $\mathscr{P}$, we denote by $sems_S(\mathscr{P})$ the collection of subsets of $\mathscr{H}$ that are models of $\mathscr{P}$ according to the semantics S. The general framework of Caroprese et al. [5] can be applied with any of the standard semantics of logic programs. In this paper we commit to the most common choice for S by selecting the stable-model semantics [8].

Definition 1 (Abductive Theory). An *abductive theory* $\mathscr{T}$ over a vocabulary σ, with the set of predicate symbols $\mathscr{R}$ partitioned into the sets $\mathscr{R}_{\mathscr{I}}$ and $\mathscr{R}_{\mathscr{E}}$ of intensional and extensional predicate symbols, is a triple $\langle \mathscr{P}, \mathscr{A}, \mathscr{I} \rangle$, where:

- $\mathscr{P}$ is a normal program;
- $\mathscr{A} \subseteq \mathscr{R}_{\mathscr{E}}$ is a finite set of predicate symbols called *abducible predicates*;
- $\mathscr{I}$ is a finite set of *denial constraints*.

Informally, the program $\mathscr{P}$ and the integrity constraints $\mathscr{I}$ model the problem domain. $\mathscr{P}$ defines intensional predicates in terms of extensional predicates. Some of the extensional predicates (those in $\mathscr{A}$) are *abducible*. Information about extensional predicates is given in terms of facts (contained in $\mathscr{P}$). Facts based on abducible predicates are *abducibles*. The integrity constraints in $\mathscr{I}$ impose domain constraints on predicates in the language.

An *observation* is a set of facts based on *non-abducible* predicates. An observation may "agree" with the program $\mathscr{P}$ and the integrity constraints $\mathscr{I}$. But if it does not, we assume that this "disagreement" is caused by the incorrect information about the properties modeled by the *abducible* predicates. Abductive reasoning consists of inferring updates to the set of abducibles in the program (removal of some and inclusion of some new ones) so that the updated program, the integrity constraints and the observation "agree". Each update that yields an agreement constitutes a possible *explanation* of the observation.

Different concepts of "agreement" and consequently different definition of abductive explanations have been proposed in the literature [4], [9]. In this paper, we assume that an agreement exists if at least one model of the program satisfies the integrity constraints [5] and the observation holds in every model of the program satisfying the integrity constraints.

Definition 2 (ABDUCTIVE EXPLANATION). Let $\mathscr{T} = \langle \mathscr{P}, \mathscr{A}, \mathscr{I} \rangle$ be an abductive theory and O an observation. A pair $\Delta = (E, F)$, where E and F are disjoint finite sets of abducibles and $F \subseteq \mathscr{P}$, is an *abductive explanation* of O, if O *agrees* with $\mathscr{P}^{\Delta} = (\mathscr{P} \cup E) \setminus F$ and $\mathscr{I}$, that is:

1) there is $M \in sem_S(\mathscr{P}^{\Delta})$ s.t. $M \models \mathscr{I}$, and
2) for every $M \in sem_S(\mathscr{P}^{\Delta})$ s.t. $M \models \mathscr{I}$, $M \models O$. □

Given an explanation $\Delta = (E, F)$, we define $E(\Delta) = E$ and $F(\Delta) = F$. In general, abductive explanations form a rich space, with some of them being more plausible than others. In this paper, we are primarily interested in constrained explanations. Formally, the notions of arbitrariness and constrainedness are based on the idea of "replaceability" of constants. Here we recall the key definitions [5].

Definition 3.

- OCCURRENCE.
  Let $p(\bar{x})$ be a fact, where p has arity n and $k \in [1..n]$. We denote by $p(\bar{x})[k]$ the constant in position k in $p(\bar{x})$. If E is a set of facts, an *occurrence* of a constant c in E is an expression of the form $p(\bar{x})^k$, where $p(\bar{x})$ is a fact in E, and $p(\bar{x})[k] = c$.
- REPLACEMENT FUNCTION.
  Let E be a set of facts and c a constant occurring in E. A *replacement function* for E and c w.r.t. a *non-empty* set C of some (not necessarily all) occurrences of c in E, is a function $f_{E,C} : \mathscr{C} \rightarrow 2^{\mathscr{H}}$ such that for each $x \in \mathscr{C}$, $f_{E,C}(x)$ is the set E' obtained by replacing with x each constant c in E referred by an occurrence in C.
- INDEPENDENCE OF REPLACEMENT FUNCTIONS.
  Let c_1 and c_2 be constants, and C_1 and C_2 sets of occurrences (possibly not all) of c_1 and c_2. Replacement functions f_{E,C_1} and f_{E,C_2} for a set $E \subseteq \mathscr{H}$ are *independent* if $c_1 \neq c_2$ or if $C_1 \cap C_2 = \emptyset$.
- DEGREE OF ARBITRARINESS.
  Let $\mathscr{T} = \langle \mathscr{P}, \mathscr{A}, \mathscr{I} \rangle$ be an abductive theory, O an observation, $\Delta = (E, F)$ an explanation for O w.r.t. $\mathscr{T}$, and ξ an arbitrary constant in $\mathscr{C}$ not occurring in $\mathscr{T}$, E nor O. The *degree of arbitrariness* of Δ, denoted as $\delta(\Delta)$, is the maximum number of *pairwise independent* replacement functions $f_{E,C}$ (not necessarily all for the same constant) such that $\Delta' = (f_{E,C}(\xi), F)$ is an explanation for O w.r.t. $\mathscr{T}$. □

Since the domain $\mathscr{C}$ is infinite, it we always can find a constant ξ not occurring in $\mathscr{T}$, E nor O. Moreover, the specific choice of the replacement constant ξ does not affect the maximum number of *pairwise independent* replacement functions. Thus, the degree of arbitrariness is well defined.

The following example illustrates the concepts we have introduced above.

Example 2. Let $\mathscr{T} = \langle \mathscr{P}, \mathscr{A}, \emptyset \rangle$, where the program $\mathscr{P}$ contains the rule $t \leftarrow p(X)$, *not* $q(X)$ and the facts $p(1), p(2), q(1), q(2), q(3)$. Let us suppose that p and q are abducible predicates and that $O = \{t\}$. The following pairs of sets of abducibles are explanations for O w.r.t. $\mathscr{T}$:

$\Delta_1 = (\emptyset, \{q(1)\})$.
$\Delta_2 = (\emptyset, \{q(2)\})$.
$\Delta_3 = (\{p(3)\}, \{q(3)\})$.
$\Delta_x = (\{p(x)\}, \emptyset)$, where $x \notin \{1, 2, 3\}$.

Let's consider the explanation Δ_3. The only occurrence of the constant 3 in $p(3)$ is denoted as $p(3)^1$. The only possible replacement function for $E(\Delta_3) = \{p(3)\}$ is $f_{\{p(3)\},\{p(3)^1\}}$. Therefore, $f_{\{p(3)\},\{p(3)^1\}}(\xi) = \{p(\xi)\}$.
We can verify that $\delta(\Delta_1) = \delta(\Delta_2) = 0$ and $\delta(\Delta_3) = \delta(\Delta_x) = 1$. In fact, $E(\Delta_1)$ and $E(\Delta_2)$ are empty, while the only constant in $E(\Delta_3)$ and $E(\Delta_x)$ (3 and x respectively) can be replaced with a fresh constant ξ and the result is a new explanation. Interestingly, Δ_3 shows that a replacement may change a minimal explanation into a non-minimal one. □

In Example 2, the explanation Δ_3 is not satisfactory. Once we decide to remove $q(3)$, there is no reason why we have to add $p(3)$. Adding any atom $p(\xi)$, with $\xi \notin \{1, 2\}$, works equally well. Thus, the choice of the constant 3 in $p(3)$ is arbitrary and not grounded in the information available in the theory. Similarly, Δ_x, where $x \notin \{1, 2, 3\}$, is not satisfactory either. Here too, the choice of x is not grounded in the abductive theory and the observation. The explanations Δ_1 and Δ_2 do not show this arbitrariness.

Definition 4 **(CONSTRAINED/ARBITRARY EXPLANATIONS).** Let $\mathscr{T}$ be an abductive theory $\langle \mathscr{P}, \mathscr{A}, \mathscr{I} \rangle$, O an observation, and Δ an explanation for O w.r.t. $\mathscr{T}$. We say that Δ is *constrained* if $\delta(\Delta) = 0$. Otherwise, Δ is *arbitrary*. □

The degree of arbitrariness of an explanation (E, F) only depends on the "add" part E; the "delete" component, F, has no effect on arbitrariness. Intuitively, the reason is that we can delete only those atoms that are in $\mathscr{P}$. Thus, if we replace a constant in an atom p in F with a fresh constant ξ, the effect simply is that p is no longer deleted.

Additionally, note that constrained explanations use only constants occurring in the abductive theory or in observation [5]. It is important as it allows us to restrict the scope of search for constrained explanations.

COMPUTING ABDUCTIVE EXPLANATIONS

In this section we present a rewriting technique allowing to compute constrained explanations for a subclass of abductive theories.

Definition 5 (DEPENDENCY GRAPHS). The *dependency graph* of a Horn program $\mathscr{P}$ is a directed graph $G_{\mathscr{P}} = (\mathscr{R}, \mathscr{E})$ where $\mathscr{R}$ (*nodes*) is the set of predicate symbols occurring in $\mathscr{P}$ and $\mathscr{E}$ (*edges*) is the set of pairs (p,q) s.t. there is at least a rule in $\mathscr{P}$ whose head predicate is p and in whose body q occurs. □

Definition 6 (DEPENDENT PREDICATES). Given a Horn program $\mathscr{P}$, the predicate p *depends* on the predicate q if (p,q) is an edge of the transitive closure of $G_{\mathscr{P}}$.
The predicates p and q are *dependent* if p depends on q or q depends on p. □

The rewriting technique presented is this section has been designed for abductive theories $\mathscr{T} = \langle \mathscr{P}, \mathscr{A}, \emptyset \rangle$, where $\mathscr{P}$ is a *non-recursive Horn program not containing any rule in whose body two dependent predicates occur*.

We report two complexity results, presented in [5], for abductive theories $\mathscr{T} = \langle \mathscr{P}, \mathscr{A}, \emptyset \rangle$, where $\mathscr{P}$ is a non-recursive *Horn program.*

Theorem 1 (Caroprese et al. [5]). Let $\mathscr{P}$ be a non-recursive Horn program and $\mathscr{A}$ a set of abducible predicates. The following problems are in P:

- Given an observation O and a pair of sets of abducibles (E,F), decide whether (E,F) is a constrained explanation for O w.r.t. $\langle \mathscr{P}, \mathscr{A}, \emptyset \rangle$.
- Given an observation O, decide whether a constrained explanation for O w.r.t. $\langle \mathscr{P}, \mathscr{A}, \emptyset \rangle$ exists.

Without loss of generality, we assume that each intensional predicate is defined by means of exactly one rule of the form:

$$h(\overline{X}) \leftarrow \bigvee_{i \in [1..n]} \mathscr{P}_i(\overline{X}, \overline{Y_i}) \qquad (1)$$

To show how the technique works, we will use the following examples.

Example 3. Let $\mathscr{T} = \langle \mathscr{P}, \mathscr{A}, \emptyset \rangle$, where $\mathscr{A} = \{q,r,t\}$ and $\mathscr{P}$ consists of the rules:

$$\begin{aligned} \mathscr{R} = \{ & p(X) \leftarrow q(X,Y), s(X,Y,Z); \\ & s(X,Y,Z) \leftarrow r(X,Y,Z), t(X,Z)\} \end{aligned}$$

and the facts:

$$B = \{q(a,b), q(a,c), r(a,b,c)\}.$$

Suppose $O = \{p(a)\}$. One can check that each of the following pairs of sets of abducibles is an explanation:

$$\begin{aligned} \Delta_{x_1,x_2} &= (\{q(a,x_1), r(a,x_1,x_2), t(a,x_2)\}, \emptyset), \\ & \quad \text{where } x_1 \neq c,\ x_1 \neq b \text{ and } x_2 \neq c \\ \Delta_{x_3} &= (\{r(a,b,x_3), t(a,x_3)\}, \emptyset), \\ & \quad \text{where } x_3 \neq c \\ \Delta_{x_4} &= (\{r(a,c,x_4), t(a,x_4)\}, \emptyset) \\ \Delta &= (\{t(a,c)\}, \emptyset). \end{aligned}$$

One can check that $\delta(\Delta_{x_1,x_2}) = 2$. Indeed changing all occurrences of x_1 or all occurrences of x_2 to a new constant ξ results in an explanation. In addition, the corresponding replacement functions for each constant and all its occurrences are obviously independent. Similarly, one can see that $\delta(\Delta_{x_3}) = 1$ (resp. $\delta(\Delta_{x_4}) = 1$), because all occurrences of x_3 (resp. x_4) are free for a simultaneous change, and $\delta(\Delta) = 0$, because neither a nor c can be changed to a fresh constant. □

Example 4. Let $\mathscr{T} = \langle \mathscr{P}, \mathscr{A}, \emptyset \rangle$, where $\mathscr{A} = \{r\}$ and $\mathscr{P}$ consists of the rules $p(a) \leftarrow r(X,b)$ and $q(a) \leftarrow r(a,Y)$ and contains no facts. Let us suppose $O = \{p(a), q(a)\}$. One can check that each of the following pairs of sets of abducibles is an explanation:

$$\begin{aligned} \Delta_{x_1,x_2} &= (\{r(a,x_1), r(x_2,b)\}, \emptyset), \text{ where } x_1 \neq b \text{ and } x_2 \neq a \\ \Delta &= (\{r(a,b)\}, \emptyset). \end{aligned}$$

One can check that $\delta(\Delta_{x_1,x_2}) = 2$ and $\delta(\Delta) = 0$. □

Rewriting into a Disjunctive Logic Program with Negation

This section presents a method for computing constrained explanations of observations given an abductive theory of the form discussed above. It consists of a transformation of the abductive theory and the observation into a disjunctive logic program with negation. The stable models of the program correspond to the constrained explanations.

The rewriting implements a *backward process* that starts from the observation and, from *true heads* of logic rules in the theory (*consequences*), derives the atoms in their bodies (*causes*). *Arbitrary constants* introduced during the process are replaced (*unified*), when it is possible, with *actual* (non-arbitrary) constants occurring in the theory. If in a stable model each arbitrary constant is unified with an actual constant, that stable model corresponds to a constrained explanation.

The derivation of a fact not already present in the theory implies that it must be inserted. If there exists a stable model not containing any insertion, then the theory as it is, already explains the observation. Indeed, the rewriting will derive all possible constrained ways to explain the observation by means of the abductive theory, including those that do not require any changes in the theory (empty explanations). The proposed rewriting can be submitted and tested on disjunctive ASP solvers such as DLV (https://www.dlvsystem.it/dlvsite/).

Let $\mathscr{T} = \langle \mathscr{R} \cup B, \mathscr{A}, \emptyset \rangle$ be an abductive theory such that $\mathscr{R}$ is a non-recursive Horn program not containing any rule in whose body two dependent predicates occur, B is a finite set of facts, and O is an observation. We describe the rewriting *Rew* of $\mathscr{T}$ and O in a set of definitions. The rewriting uses new predicate symbols. In particular, for every predicate symbol p in the language, we have a fresh predicate symbol p^* of the same arity as p. If p is a base (resp. derived) predicate, we say that p^* is a *starred* base (resp. derived) predicate. Similarly, if $p(\overline{X})$ is a base (resp. derived) atom, we say that $p^*(\overline{X})$ is a *starred* base (resp. derived) atom. Moreover, given an atom a (resp. set of atoms A), we will denote the corresponding starred atom (resp. set of starred atoms) as a^* (resp. A^*).

We assume that each constant is stored in the unary relation *constant* and we write $Const(\mathscr{T})$ for the set of its facts w.r.t. to constants in $\mathscr{T}$.

Definition 7 (REWRITING OF THE OBSERVATION). Given the observation $O = \{o_1, \ldots, o_n\}$, $Rew(O) = \{o_1^*, \ldots, o_n^*\}$ where, for $i \in [1..n]$, o_i^* is obtained from o_i by replacing its predicate symbol p with p^*. □

Definition 8 (REWRITING OF THE DATABASE). Given the database $B = \{b_1, \ldots, b_n\}$, $Rew(B) = \{b_1^*, \ldots, b_n^*\}$ where, for $i \in [1..n]$, b_i^* is obtained from b_i by replacing its predicate symbol p with p^*. □

Definition 9 ***(***REWRITING OF A RULE). Given a rule r of the form (1), $\mathit{Rew}(r)$ is the set containing the following rules:

1) $h_1^*(\overline{X}) \vee \cdots \vee h_n^*(\overline{X}) \leftarrow h^*(\overline{X})$
2) $\mathscr{P}_i^*(\overline{X},\ y_{i,1}(h,\overline{X}),\ \ldots,\ y_{i,m_i}(h,\overline{X})) \leftarrow h_i^*(\overline{X}), \quad \forall i \in [1..n]$
3) $\mathit{arbitrary}(y_{i,1}(h,\overline{X})),\ \ldots,\ \mathit{arbitrary}(y_{i,m_i}(h,\overline{X})) \leftarrow h_i^*(\overline{X}),$ $\forall i \in [1..n]$

where the conjunction $\mathscr{P}_i^*(\overline{X},\ y_{i,1}(h,\overline{X}),\ \ldots,\ y_{i,m_i}(h,\ \overline{X}))$ is obtained from $\mathscr{P}_i(\overline{X},\overline{Y}_i)$ by replacing each predicate symbol p with p^* and each variable $Y_{i,k}$ with the functional term $y_{i,k}(h,\overline{X})$. □

The operator $\mathit{Rew}(\cdot)$ is extended to sets of rules in the standard way. Previous rewriting is the core of our technique as it implements the inversion of the rules. Arbitrary constants introduced in the process are represented by functional terms $y_{i,k}(h,\overline{X})$ and are stored in the relation *arbitrary*. These constants will be "unified" by means of rules we introduce below with actual constants in the theory.

Definition 10 ***(***UNIFICATION). Let $\mathit{Unification}(\mathscr{T})$ be the set of next rules, setting the candidate values and an assignment for each arbitrary constant:

4) $\mathit{term}(X) \leftarrow \mathit{constant}(X)$
5) $\mathit{term}(X) \leftarrow \mathit{arbitrary}(X)$
6) $\mathit{candidate}(X,X) \leftarrow \mathit{term}(X)$
7) $\mathit{candidate}(X_i,Y_i) \leftarrow r^*(X_1,\ldots,X_i,\ldots,X_{n_r}),$
   $r^*(Y_1,\ldots,Y_i,\ldots,Y_{n_r}),$
   $\mathit{arbitrary}(X_i), \mathit{constant}(Y_i),$
   $\bigwedge_{j\in[1..n_r]\setminus\{i\}} \mathit{compatible}(X_j,Y_j)$
   for each predicate $r(X_1,\ldots,X_{n_r})$ and for each $i \in [1..n_r]$.
8) $\mathit{compatible}(X,X) \leftarrow \mathit{term}(X)$
9) $\mathit{compatible}(X,Y) \leftarrow \mathit{compatible}(Y,X)$
10) $\mathit{compatible}(X,Y) \leftarrow \mathit{assign}(X,Z), \mathit{assign}(Y,Z)$
11) $\mathit{compatible}(X,Y) \leftarrow \mathit{arbitrary}(X), \mathit{constant}(Y),$
    $\mathit{not\ incompatible}(X,Y)$
12) $\mathit{incompatible}(X,Y) \leftarrow \mathit{arbitrary}(X), \mathit{constant}(Y),$
    $\mathit{assign}(X,Z), Y \neq Z$
13) $\mathit{discard}(X,Y) \vee \mathit{discard(X,Z)} \leftarrow \mathit{candidate}(X,Y),$
    $\mathit{candidate}(X,Z), Y \neq Z.$
14) $\mathit{assign}(X,Y) \leftarrow \mathit{candidate}(X,Y), \mathit{not\ discard}(X,Y)$
15) $r^*(Y_1,\ldots,Y_n) \leftarrow r^*(X_1,\ldots,X_n), \mathit{assign}(X_1,Y_1),\ldots,$
    $\mathit{assign}(X_n,Y_n)$
    for each *extensional predicate symbol r*

□

Additional rules have to be added in order to guarantee the correct computation of constrained explanations.

Definition 11 ***(***CONSTRAINEDNESS). We define the rules $\mathit{Constrained}(\mathscr{T})$ to compute whether a solution is constrained or not:

16) $\mathit{evaluated}(X) \leftarrow \mathit{assign}(X,Y), \mathit{not\ arbitrary}(Y)$
17) $\mathit{unevaluated}(X) \leftarrow \mathit{term}(X), \mathit{not\ evaluated}(X)$
18) $\mathit{arbitraryExplanation} \leftarrow \mathit{unevaluated}(X)$
19) $\mathit{constrainedExplanation} \leftarrow \mathit{not\ arbitraryExplanation}$ □

Rule 16 derives the terms evaluated assigning to them arbitrary constants. The next rule derives the unevaluated arbitrary constants. Next two rules derives whether the explanation is arbitrary or not. In particular, an explanation is arbitrary if there is a non evaluated arbitrary constant. In this case the atom *arbitraryExplanation* is derived. Otherwise, it is constrained and the atom *constrainedExplanation* is derived.

Definition 12 ***(***UPDATE). We define the rules $\mathit{Update}(\mathscr{T})$, modeling the abducibles that will be inserted (when they have to be present and they are not in the theory):

20) $r^+(Y_1,\ldots,Y_n) \leftarrow r^*(X_1,\ldots,X_n),$
    $\mathit{assign}(X_1,Y_1),\ldots,\mathit{assign}(X_n,Y_n),$
    $\mathit{not}\ r(Y_1,\ldots,Y_n)$
21) $\bot \leftarrow r^+(X_1,\ldots,X_n)$ if $r \notin \mathscr{A}$

for each extensional predicate r.

The constraint in latest item prevents insertions of facts with non abducible predicates.

Definition 13. Given an abductive theory $\mathscr{T} = \langle \mathscr{R} \cup B, \mathscr{A}, \emptyset \rangle$, where $\mathscr{R}$ is a non-recursive Horn program not containing any rule in whose body two dependent predicates occur and B is a finite set of facts, and an observation O, $\mathit{Rew}(\mathscr{T},O) = \mathit{Rew}(O) \cup \mathit{Rew}(\mathscr{R}) \cup \mathit{Rew}(B) \cup B \cup \mathit{Const}(\mathscr{T}) \cup \mathit{Unification}(\mathscr{T}) \cup \mathit{Constrained}(\mathscr{T}) \cup \mathit{Update}(\mathscr{T})$. □

Given a stable model M of $\mathit{Rew}(\mathscr{T},O)$, we define $F(M) = \{r(x_1,\ldots,x_n) \mid r^+(x_1,\ldots,x_n) \in M\}$.

We only consider explanations with no deletions as deletions are not needed for the type of theories we consider.

Theorem 2. Let $\mathscr{T} = \langle \mathscr{R} \cup B, \mathscr{A}, \emptyset \rangle$ be an abductive theory, where $\mathscr{R}$ is a non-recursive Horn program not containing any rule in whose body two dependent predicates occur and B is a finite set of facts, and O an observation. Then:

1) If $\Delta = (E,\emptyset)$ is a constrained explanation for O w.r.t. $\mathscr{T}$, then there is a stable model M of $\mathit{Rew}(\mathscr{T},O)$ containing the fact *constrainedExplanation* s.t. $F(M) = E$.
2) If M is a stable model of $\mathit{Rew}(\mathscr{T},O)$ containing the fact *constrainedExplanation*, $E = F(M)$ and E is minimal, i.e. there is no $E' \subset E$ s.t. $\mathscr{R} \cup B \cup E' \models O$, then $\Delta = (E,\emptyset)$ is a constrained explanation for O w.r.t. $\mathscr{T}$. □

Proof (sketch)[1]***.***
1) From E it is possible to define a set M and show that M is a stable model of $\mathscr{Q} = \mathit{Rew}(\mathscr{T},O)$. First it can be proved that M is a model of the reduct $\mathscr{Q}^M$ and then that M is minimal. Minimality can be proved by contradiction. Assuming that M is not a minimal model of $\mathscr{Q}^M$, there must be $W \subset M$ s.t. $W \models \mathscr{Q}^M$. From W a new model M' for $\mathit{Res}(\mathscr{T},O)$ can be obtained. Let $E' = F(M')$. It can be proved that $\Delta' = (E',\emptyset)$ is an explanation for O w.r.t. $\mathscr{T}$ and that E' can be obtained by replacing some constants in E with new constants not occurring in $\mathscr{T}$ and O. Therefore $(E,\emptyset)$ is not constrained. This is a contradiction.
2) First, it is possible to prove that, given $E = F(M)$, $\Delta = (E,\emptyset)$ is an explanation, that is $\mathscr{R} \cup B \cup E \models O$. Then, it is possible to prove by contradiction that Δ is constrained. Assuming that Δ is not constrained, it can be proved that M is not a stable model of $\mathscr{R} \cup B \cup E \models O$. This is a contradiction. □

Theorem 2 is the main result of our work. It demonstrates the correctness of the rewriting and suggests the algorithm to compute the constrained explanations.

Algorithm

1) Compute the set $\mathscr{M}$ of stable models of $\mathit{Rew}(\mathscr{T},O)$).
2) Compute the set $\mathscr{E} = \{F(M) \mid M \in \mathscr{M}$ *and constrained Explanation* $\in M\}$.

[1] The full proof is reported in the appendix of an extended version of the paper at *https://github.com/caroprese/abduction*.

3) Let $\mathscr{I}$ the class of minimal sets in $\mathscr{E}$. The constrained explanations of O w.r.t. $\mathscr{T}$ are $\{(E,\emptyset) \mid E \in \mathscr{I}\}$. □

This approach allows to compute the constrained explanations in two steps. The first step computes the set $\mathscr{E}$ containing $F(M)$ for each stable model M of $Rew(\mathscr{T},O)$ that includes the atom $constrainedExplanation$. The second step selects the minimal sets in $\mathscr{E}$. They correspond exactly to the constrained explanations we are looking for.
This approach is much more efficient than a *guess and check* procedure because greatly reduces the search space.

Example 5. Let $\mathscr{T} = \langle \mathscr{R} \cup B, \mathscr{A}, \emptyset \rangle$, where $\mathscr{A} = \{p\}$, $\mathscr{R} = \{o \leftarrow m(X), n(Y);\ n(X) \leftarrow p(X), s(X);\ m(X) \leftarrow p(X), s(X)\}$ and $B = \{p(a), p(b)\}$. Let us assume $O = \{o\}$. The set $Rew(\mathscr{R})$ contains the following rules:

- $o^*_1 \leftarrow o^*$
- $m^*(y_{1,1}(o)) \leftarrow o^*_1$
- $n^*(y_{1,2}(o)) \leftarrow o^*_1$
- $arbitrary(y_{1,1}(o)) \leftarrow o^*_1$
- $arbitrary(y_{1,2}(o)) \leftarrow o^*_1$
- $n_1(X)^* \leftarrow n^*(X)$
- $p(X)^* \leftarrow n^*_1(X)$
- $s(X)^* \leftarrow n^*_1(X)$
- $m_1(X)^* \leftarrow m^*(X)$
- $p(X)^* \leftarrow m^*_1(X)$
- $s(X)^* \leftarrow m^*_1(X)$

We do not report $Rew(O)$, $Rew(B)$, $Const(\mathscr{T})$, $Unification(\mathscr{T})$, $Constrained(\mathscr{T})$ and $Update(\mathscr{T})$ as they are trivial. One can check that $\mathscr{E} = \{\{s(a)\},\{s(b)\},\{s(a),s(b)\}\}$ and then $\mathscr{I} = \{\{s(a)\},\{s(b)\}\}$ that corresponds to the constrained explanations $\Delta_1 = (\{s(a)\},\emptyset)$ and $\Delta_2 = (\{s(b)\},\emptyset)$. □

The rewritings of the abductive theories and the observations presented in Example 3, Example 4 and Example 5 can be found at *https://github.com/caroprese/abduction*. This repository contains the DLV system, two source files, the related batch files allowing to run the experiments on Windows systems and the results of the experiments (the stable models of the rewritings and the corresponding abductive explanations).

Discussion and Concluding Remarks

Abduction was introduced to artificial intelligence in early 1970s by Harry Pople Jr. [10]. Over the years several criteria have been proposed to identify the preferred (best) explanations, all rooted in the Occam's razor (parsimony) principle. The abduction reasoning formalism we study in the paper uses logic programs to represent background knowledge in abductive theories. It is referred to as *abductive logic programming* [1], [3]. Abductive explanations which allow the removal of hypotheses are first introduced by Inoue and Sakama [11]. The importance of abductive logic programming to knowledge representation was argued by Denecker and Schreye [12]. It was applied in diagnosis [13], planning [14] and natural language understanding [15]. Denecker and Kakas [4] provide a comprehensive survey of the area. Eiter et al. [6] studied the complexity of reasoning tasks in the abductive logic programming setting. In [16] and [17] an algorithm for computing abductive explanations for *propositional Horn theories* is presented. The concept of *simplicity* adopted in this paper is based on minimality with respect to set inclusion. In [18] an extension of abduction where explanations are jointly computed by sets of interacting agents is investigated. Also in this paper only the propositional case is analyzed and the use of answer set engines such as DLV. to calculate the explanation is left as a topic for further research.

None of the earlier works on abduction considered the concepts of constrainedness or arbitrariness. These concepts were originally proposed for the setting of view updates in deductive databases [19], [20]. View updating consists of modifying base relations to impose properties on view relations, that is, relations defined on the database by queries. The degree of arbitrariness and constrainedness were adapted to the setting of abductive logic programming by Caroprese et al. [5].

In this paper we showed how the problem of computing constrained explanations for abductive theories of a specific form can be cast as an application of ASP via a direct rewriting of a theory into a disjunctive logic program. This is an important first step towards computing constrained explanations for arbitrary abductive theories. The work opens several avenues for future research. First, it is important to extend the proposal to the other classes of abductive theories identified by Caroprese et al. [5] and then, to the general case. Second, the effectiveness of the rewriting proposed in this paper, as well as rewritings that might exist for other classes of abductive theories, has to be verified experimentally on realistic benchmarks where reasoning tasks involve abduction.

References

[1] K. Eshghi and R. A. Kowalski, "Abduction Compared with Negation by Failure," in *Proceedings of the 6th International Conference on Logic Programming*, G. Levi and M. Martelli, Eds. MIT Press, 1989, pp. 234–254.

[2] A. C. Kakas and P. Mancarella, "Generalized stable models: A semantics for abduction," in *Proceedings of the 9th European Conference on Artificial Intelligence, ECAI 1990*, L. Aiello, Ed. London/Boston: Pitman, 1990, pp. 385–391.

[3] A. C. Kakas, R. A. Kowalski, and F. Toni, "Abductive Logic Programming," *J. Log. Comput.*, vol. 2, no. 6, pp. 719–770, 1992.

[4] M. Denecker and A. C. Kakas, "Abduction in Logic Programming," in *Computational Logic: Logic Programming and Beyond, Essays in Honour of Robert A. Kowalski*, ser. Lecture Notes in Computer Science, A. C. Kakas and F. Sadri, Eds., vol. 2407. Springer, 2002, pp. 402–436.

[5] L. Caroprese, I. Trubitsyna, M. Truszczynski, and E. Zumpano, "A measure of arbitrariness in abductive explanations," *TPLP*, vol. 14, no. 4-5, pp. 665–679, 2014. [Online]. Available: http://dx.doi.org/10.1017/S1471068414000271

[6] T. Eiter, G. Gottlob, and N. Leone, "Abduction from Logic Programs: Semantics and Complexity." *Theor. Comput. Sci.*, vol. 189, no. 1-2, pp. 129–177, 1997.

[7] L. Caroprese and E. Zumpano, "Indefinite abductive explanations," *J. Appl. Non Class. Logics*, vol. 29, no. 3, pp. 233–254, 2019.

[8] M. Gelfond and V. Lifschitz, "The Stable Semantics for Logic Programs," in *ICLP/SLP*, R. A. Kowalski and K. A. Bowen, Eds. MIT Press, 1988, pp. 1070–1080.

[9] C. Baral and M. Gelfond, "Logic Programming and Knowledge Representation," *J. Log. Program.*, vol. 19/20, pp. 73–148, 1994.

[10] H. E. Pople, "On the Mechanization of Abductive Logic," in *Proceedings of the 3rd International Joint Conference on Artificial Intelligence, IJCAI 1973*, N. J. Nilsson, Ed. William Kaufmann, 1973, pp. 147–152.

[11] K. Inoue and C. Sakama, "Abductive Framework for Nonmonotonic Theory Change," in *Proceedings of the 14th International Joint Conference on Artificial Intelligence, IJCAI 95*. Morgan Kaufmann, 1995, pp. 204–210.

[12] M. Denecker and D. D. Schreye, "Representing Incomplete Knowledge in Abductive Logic Programming," *J. Log. Comput.*, vol. 5, no. 5, pp. 553–577, 1995.

[13] L. Console, L. Portinale, and D. T. Dupré, "Using Compiled Knowledge to Guide and Focus Abductive Diagnosis." *IEEE Trans. Knowl. Data Eng.*, vol. 8, no. 5, pp. 690–706, 1996.

[14] S. do Lago Pereira and L. N. de Barros, “Planning with Abduction: A Logical Framework to Explore Extensions to Classical Planning.” in *Proceedings of the 17th Brazilian Symposium on Artificial Intelligence, SBIA 2004*, ser. Lecture Notes in Computer Science, A. L. C. Bazzan and S. Labidi, Eds., vol. 3171. Springer, 2004, pp. 62–72.

[15] J. Balsa, V. Dahl, and J. G. P. Lopes, “Datalog Grammars for Abductive Syntactic Error Diagnosis and Repair,” in *Proceedings of the Natural Language Understanding and Logic Programming Workshop, Lisbon, 1995*, 1995, pp. 111–125.

[16] T. Eiter and K. Makino, “On computing all abductive explanations,” in *Proceedings of the Eighteenth National Conference on Artificial Intelligence and Fourteenth Conference on Innovative Applications of Artificial Intelligence, July 28 - August 1, 2002, Edmonton, Alberta, Canada*, R. Dechter, M. J. Kearns, and R. S. Sutton, Eds. AAAI Press / The MIT Press, 2002, pp. 62–67. [Online]. Available: http://www.aaai.org/Library/AAAI/2002/aaai02-010.php

[17] ——, “On computing all abductive explanations from a propositional horn theory,” *J. ACM*, vol. 54, no. 5, p. 24, 2007. [Online]. Available: https://doi.org/10.1145/1284320.1284323

[18] G. Greco, “Solving abduction by computing joint explanations,” *Ann. Math. Artif. Intell.*, vol. 50, no. 1-2, pp. 143–194, 2007. [Online]. Available: https://doi.org/10.1007/s10472-007-9069-y

[19] L. Caroprese, I. Trubitsyna, M. Truszczynski, and E. Zumpano, “The View-update Problem for Indefinite Databases,” in *Proceedings of the 13th European Conference on Logics in Artificial Intelligence, JELIA 2012*, ser. Lecture Notes in Computer Science, L. F. del Cerro, A. Herzig, and J. Mengin, Eds., vol. 7519. Springer, 2012, pp. 134–146.

[20] L. Caroprese, I. Trubitsyna, and E. Zumpano, “View updating through active integrity constraints,” in *Logic Programming, 23rd International Conference, ICLP 2007, Porto, Portugal, September 8-13, 2007, Proceedings*, ser. Lecture Notes in Computer Science, V. Dahl and I. Niemelä, Eds., vol. 4670. Springer, 2007, pp. 430–431.

Luciano Caroprese is currently a researcher at the Institute for high performance computing and networking (ICAR-CNR), Cosenza, Italy and cooperates with the DIMES Department of the University of Calabria. He received his Ph.D. in computer science from the University of Calabria in 2008. His area of research includes logic programming, deductive database, database integration, P2P systems, data analytics and machine learning. Contact him at l.caroprese@dimes.unical.it.

Ester Zumpano is an associate professor of computer engineering at the DIMES Department of the University of Calabria, Italy. She obtained her Ph.D. in computer and systems engineering in 2003. Her areas of research include health information systems, data integration, logic programming, view updating, distributed systems, artificial intelligence and database management. Contact her at e.zumpano@dimes.unical.it

Bart Bogaerts is an assistant professor at the Vrije Universiteit Brussel. He received his Ph.D. in computer science from Katholieke Universiteit Leuven in 2015. His research interests range from high-level representation languages to performance optimisations in SAT, from abstract, algebraical frameworks to unify semantics of logics to implementation of knowledge base systems, from applications of KR to integration of declarative problem solving paradigms. Contact him at bart.bogaerts@vub.be.