Disjunctive Explanations

Katsumi Inoue[1] and **Chiaki Sakama**[2]

[1] Department of Electrical and Electronics Engineering
Kobe University
Rokkodai, Nada, Kobe 657-8501, Japan
inoue@eedept.kobe-u.ac.jp

[2] Department of Computer and Communication Sciences
Wakayama University
Sakaedani, Wakayama 640-8510, Japan
sakama@sys.wakayama-u.ac.jp

Abstract. Abductive logic programming has been widely used to declaratively specify a variety of problems in AI including updates in data and knowledge bases, belief revision, diagnosis, causal theory, and default reasoning. One of the most significant issues in abductive logic programming is to develop a reasonable method for *knowledge assimilation*, which incorporates obtained explanations into the current knowledge base. This paper offers a solution to this problem by considering *disjunctive explanations* whenever multiple explanations exist. Disjunctive explanations are then to be assimilated into the knowledge base so that the assimilated program preserves all and only minimal answer sets from the collection of all possible updated programs. We describe a new form of abductive logic programming which deals with disjunctive explanations in the framework of *extended abduction*. The proposed framework can be well applied to view updates in disjunctive databases.

1 Introduction

The task of abduction is to infer explanations accounting for an observation. In general, we may encounter multiple explanations for the given observation. When there are multiple explanations of G, we observe that the disjunction of these explanations also accounts for G. In this paper, we formalize this idea by extending the notion of *explanation* to more general one than the traditional framework of *abductive logic programming* (ALP). Suppose that we are given the background knowledge K and a set of abducibles $\mathcal{A}$. Then, each set E of instances of elements from $\mathcal{A}$ satisfying that (i) $K \cup E \models G$ and (ii) $K \cup E$ is consistent, is called an *elementary explanation* in this paper. Then, any disjunction of elementary explanations is called an *explanation*. The reason why we use the term "explanation" for a disjunction of (elementary) explanations is that if $\{e_1\}$ and $\{e_2\}$ are (elementary) explanations of G then, in first-order logic or logic programming with the answer set semantics, $e = e_1 \vee e_2$ satisfies that (i) $K \cup \{e\} \models G$ and (ii) $K \cup \{e\}$ is consistent.

The use of disjunctive explanations is quite natural when the background knowledge K is represented in *disjunctive logic programs*. Also, disjunctive explanations are useful in various applications involving abduction. For example,

- *Weakest explanations*. In abduction, we usually seek for *least presumptive* or *weakest explanations*. Such an explanation is often called a *weakest sufficient condition* [22]. When $\{e_1\}$ and $\{e_2\}$ are minimal elementary explanations of G, where the minimality is defined in terms of the set inclusion relation, each explanation $\{e_i\}$ $(i = 1, 2)$ is most preferred in traditional formalizations of abduction because $\{e_i\}$ is weaker than any non-minimal explanation like $\{e_1, e_2\}$, i.e., $\{e_1, e_2\} \models e_i$. However, the disjunctive explanation $\{e_1 \vee e_2\}$ is much weaker, i.e., $\{e_i\} \models e_1 \vee e_2$. For another example, when $\{a, b\}$ and $\{c\}$ are the two minimal elementary explanations, $\{a \vee c, b \vee c\}$ is the weakest explanation because we see that $(a \wedge b) \vee c \equiv (a \vee c) \wedge (b \vee c)$.
- *Skeptical reasoning and minimization*. In query answering from *circumscription* [11], we often need disjunctive explanations. For example, if both $\neg ab(a)$ and $\neg ab(b)$ credulously explain g and the clause $ab(a) \vee ab(b)$ can be entailed from the background theory, then the disjunction $\neg ab(a) \vee \neg ab(b)$ skeptically explains g. A minimization principle with disjunctive explanations is also employed in abduction from *causal theories* [20].
- *Negative (anti-)explanation and contraction of hypotheses*. In *extended abduction* [14], we may want to remove abducible facts from the background theory. For example, suppose that the program is given as:

$$\begin{aligned} g &\leftarrow not\, p, \\ p &\leftarrow a, \\ p &\leftarrow b, \\ a; b &\ , \end{aligned}$$

  and the abducibles are given as $\{a, b\}$. Then, to explain g, it is necessary to remove the disjunction $a; b$ from the program. However, the previous framework of extended abduction [14, 13] cannot do that, because only instances of elements from the abducibles can be manipulated. Here, removing $\{a\}$ or $\{b\}$ or $\{a, b\}$ cannot be successful because neither a nor b is in the program.
- *Knowledge base update*. Adapting alternative solutions for an update request to the background theory usually results in multiple alternative new states. The disjunction of these solutions offers a solution representing every possible change in a single state [5, 6, 25]. This technique reduces the size of knowledge bases through a sequence of updates and keeps only one current knowledge base at a time.

The last application—knowledge base update—is particularly important when we want to assimilate explanations into our current knowledge base. While *knowledge assimilation* is one of the most significant problems in ALP [19, 17], not much work has been reported so far. This paper offers a solution to this problem by assimilating disjunctive explanations into a knowledge base. We also introduce disjunctive explanations into the framework of *extended abduction* [14], where both addition and removal of hypotheses are allowed to explain or unexplain an observation. When there are multiple preferred explanations involving removal of hypotheses, assimilating them into one knowledge base is much more difficult than in the case of normal abduction which only adds hypotheses.

It is known that extended abduction can be used to formalize various update problems in AI and databases [14, 16, 26]. That is, an *insertion/deletion* of a fact G into/from a database is accomplished by a minimal *explanation/anti-explanation* of G. Then, the notion of disjunctive explanations in this paper can also be applied to update problems in databases. In particular, the *view update* problem in *disjunctive databases*, i.e., databases possibly containing disjunctions which represent indefinite or uncertain information, can also be realized within the proposed framework. When we build a database in real-life situations, a database is likely to include such disjunctive facts. Developing an update technique in disjunctive databases is therefore important from practical viewpoints. However, disjunctive databases are more expressive than Datalog [4], and view updates in disjunctive databases are more difficult than the case of Datalog. In fact, there are few studies on the subject of updating disjunctive databases and many problems have been left open. Hence, with our proposed framework, we can make advances in studies of view updates in disjunctive databases.

The rest of this paper is organized as follows. Section 2 reviews a framework of disjunctive logic programs and its answer set semantics. Section 3 introduces the abductive framework considering disjunctive explanations. Section 4 extends our disjunctive abduction to extended abduction which allows removal of abducibles from programs. Section 5 discusses related issues, and Section 6 is a summary. Due to the lack of space, we omit the proofs of theorems in this paper.

2 Disjunctive Programs

A knowledge base or database is represented in an *extended disjunctive program* (EDP) [9], or simply called a *program*, which consists of a finite number of *rules* of the form:

$$L_1; \cdots ; L_l \leftarrow L_{l+1}, \ldots, L_m, \, not\, L_{m+1}, \ldots, not\, L_n \tag{1}$$

where each L_i is a literal ($n \geq m \geq l \geq 0$), and *not* is *negation as failure* (NAF). The symbol ; represents a disjunction and is often written also as $\vee$. A rule with variables stands for the set of its ground instances. We assume that function symbols never appear in a program, which implies that a number of the ground instances of a variable is finite.[3] The left-hand side of the rule is the *head*, and the right-hand side is the *body*. A rule with the empty head is an *integrity constraint*. Any rule with the empty body $H \leftarrow$ is called a *fact* and is also written as H without the symbol $\leftarrow$.

Any program K is divided into two parts, $K = \mathcal{I}(K) \cup \mathcal{F}(K)$, where $\mathcal{I}(K) \cap \mathcal{F}(K) = \emptyset$, and $\mathcal{I}(K)$ (resp. $\mathcal{F}(K)$) denotes the set of non-fact rules (resp. facts)

[3] This assumption is necessary only for later use in representing explanation closures of an observation in first-order logic (Definition 3.4).

in K. When we consider a database written as a program, $\mathcal{I}(K)$ (resp. $\mathcal{F}(K)$) represents an *intensional database* (resp. *extensional database*).

We can consider more general form of programs allowing *nested expressions* [21]. See [21] for the definition of answer sets for such nested programs.[4] An EDP is called an *extended logic program* (ELP) if it contains no disjunction ($l \leq 1$), and an ELP is called a *normal logic program* (NLP) if every L_i is an atom.

The semantics of a program is given by its *answer sets*. First, let K be an EDP without NAF (i.e., $m = n$) and $S \subseteq \mathcal{L}$, where $\mathcal{L}$ is the set of all ground literals in the language of K. Then, S is an *answer set* of K if S is a minimal set satisfying the conditions:

1. For each ground rule $L_1; \cdots; L_l \leftarrow L_{l+1}, \ldots, L_m$ from K, $\{L_{l+1}, \ldots, L_m\} \subseteq S$ implies $\{L_1, \ldots, L_l\} \cap S \neq \emptyset$;
2. If S contains a pair of complementary literals L and $\neg L$, then $S = \mathcal{L}$.

Second, given *any* EDP K (with NAF) and $S \subseteq \mathcal{L}$, consider the EDP (without NAF) K^S obtained as follows: a rule $L_1; \cdots; L_l \leftarrow L_{l+1}, \ldots, L_m$ is in K^S if there is a ground rule of the form (1) from K such that $\{L_{m+1}, \ldots, L_n\} \cap S = \emptyset$. Then, S is an *answer set* of K if S is an answer set of K^S. An answer set is *consistent* if it is not $\mathcal{L}$. A program is *consistent* if it has a consistent answer set. Note that every answer set S of any EDP is minimal [9], that is, no other answer set S' of K satisfies that $S' \subset S$. The set of all answer sets of K is written as $\mathcal{AS}(K)$. For a literal L, we write $K \models L$ if $L \in S$ for every $S \in \mathcal{AS}(K)$.

3 Disjunctions in Normal Abduction

An *abductive program* is a pair $\langle K, \mathcal{A} \rangle$, where both K and $\mathcal{A}$ are EDPs. Each element of $\mathcal{A}$ and its any instance is called an *abducible*. When a rule is an abducible, it is called an *abducible rule*. Such an abducible rule can be associated with a unique literal called the *name* [12]. Then, with this naming technique, we can always assume in this paper that the abducibles $\mathcal{A}$ of an abductive program $\langle K, \mathcal{A} \rangle$ is a set of literals. Moreover, we assume without loss of generality that, any rule from K having an abducible in its head is always a fact consisting of abducibles only.[5] In abduction, we are given an *observation* G to be explained or unexplained. Without loss of generality, such an observation is assumed to be a non-abducible ground literal [15].

We firstly consider *normal abduction*, and later in Section 4 extend our framework by considering *extended abduction* [14].

[4] Nested expressions are necessary in this paper only because we will later consider the answer sets of a program containing DNF formulas called explanation closures (Theorem 3.4).

[5] A similar assumption is usually used in literature, e.g., [17]. If there is a fact containing both an abducible a and a non-abducible or there is a rule containing an abducible a in its head and a non-empty body, then such an abducible a is made a non-abducible by introducing a rule $a \leftarrow a'$ with a new abducible a' and then replacing a with a' in every fact consisting abducibles only.

Definition 3.1 Let $\langle K, \mathcal{A} \rangle$ be an abductive program and G an observation. A set E is an *elementary explanation of G (wrt $\langle K, \mathcal{A} \rangle$)* if

1. E is a set of ground instances of elements from $\mathcal{A}$,
2. $K \cup E \models G$, and
3. $K \cup E$ is consistent.

Note here that we use the term "elementary explanation" instead of just calling "explanation". The latter term is reserved for the next definition.

Definition 3.2 Any disjunction of elementary explanations of G is called a *(disjunctive) explanation* of G.

By definition, elementary explanations are also explanations. Disjunctive explanations deserve to be called "explanations" as the next proposition holds.

Proposition 3.1 *Let E be a (disjunctive) explanation of G wrt $\langle K, \mathcal{A} \rangle$. Then, $K \cup E \models G$ and $K \cup E$ is consistent.*

We provide an entailment relationship between programs/explanations as follows. Let R and R' be sets of *formulas with nested expressions* [21]. We write $R \models R'$ if for any $S \in \mathcal{AS}(R)$, there exists $S' \in \mathcal{AS}(R')$ such that $S' \subseteq S$. In this case, we say that R' is *weaker than* R. For example, $\{a, b\} \models \{a\} \models \{a; b\}$. We also say that R and R' are *equivalent* if $\mathcal{AS}(R) = \mathcal{AS}(R')$.

Definition 3.3 An (elementary/disjunctive) explanation E of G is *minimal* (or *weakest*) if for any (elementary/disjunctive) explanation E' of G, $E \models E'$ implies $E' \models E$.

Note that we assumed that the set of abducibles $\mathcal{A}$ consists of literals only. Then, for elementary explanations E and E', the relation $E \models E'$ is equivalent to $E' \subseteq E$. Hence, E is a minimal elementary explanation of G iff no other explanation of G is a proper subset of E.

We can also define an alternative ordering between explanations. Given an abductive program $\langle K, \mathcal{A} \rangle$, we say that an explanation E of G is *less presumptive* than an explanation E' of G if $K \cup E' \models K \cup E$. A *least presumptive explanation* is then defined as a minimal element in the less presumptive relation. We also say that E and E' are *equivalent relative to* K if $\mathcal{AS}(K \cup E) = \mathcal{AS}(K \cup E')$.

Definition 3.4 Let $\mathcal{ME}(G)$ be the set of minimal elementary explanations of G. The *explanation closure of G (wrt $\langle K, \mathcal{A} \rangle$)* is the disjunctive explanation:

$$\bigvee_{E \in \mathcal{ME}(G)} E\,.$$

The explanation closure gives the least presumptive explanation for the observation. To verify this fact, we consider an alternative formalization of abduction with the enlarged hypothesis space which consists of *disjunctive hypotheses.*

Given an abductive program $\langle K, \mathcal{A} \rangle$, the *enlarged abducible set*, written $\mathcal{D}(\mathcal{A})$, consists of every disjunction of abducibles from $\mathcal{A}$. Then, we can define an abductive program $\langle K, \mathcal{D}(\mathcal{A}) \rangle$, in which we can abduce any disjunction of abducibles to explain an observation. Of course, we can also define elementary and disjunctive explanations for the abductive program $\langle K, \mathcal{D}(\mathcal{A}) \rangle$. However, weakest elementary explanations wrt $\langle K, \mathcal{D}(\mathcal{A}) \rangle$ may contain redundant abducibles as disjuncts. For instance, when K is a program consisting of two rules:

$$\begin{aligned} p &\leftarrow a, \\ &\leftarrow b, \end{aligned}$$

and $\mathcal{A} = \{a, b\}$, as p's explanations, $\{a; b\}$ is weaker than $\{a\}$. To adopt $\{a\}$ as a preferred explanation of p, we need the notion of least presumptive explanations. In this case, $\{a\}$ and $\{a; b\}$ are equivalent relative to K.

Theorem 3.2 *If a formula F is the explanation closure of G wrt $\langle K, \mathcal{A} \rangle$, then F is equivalent (relative to K) to a least presumptive elementary explanation of G wrt $\langle K, \mathcal{D}(\mathcal{A}) \rangle$. Conversely, if E is a least presumptive elementary explanation of G wrt $\langle K, \mathcal{D}(\mathcal{A}) \rangle$, then E is equivalent (relative to K) to the explanation closure of G wrt $\langle K, \mathcal{A} \rangle$.*

Corollary 3.3 *The least presumptive elementary explanation of G wrt $\langle K, \mathcal{D}(\mathcal{A}) \rangle$ is unique up to the equivalence relation relative to K, and is equivalent to the explanation closure of G wrt $\langle K, \mathcal{A} \rangle$.*

Example 3.1 Let K be the program:

$$\begin{aligned} p; \neg q &\leftarrow a,\ b, \\ p &\leftarrow r,\ b, \\ q &\leftarrow c,\ not\, r, \\ r &\leftarrow d,\ not\, q. \end{aligned}$$

Also let the abducibles be $\mathcal{A} = \{a, b, c, d\}$. Then, the minimal elementary explanations of p wrt $\langle K, \mathcal{A} \rangle$ is: $\mathcal{ME}(p) = \{\{a, b, c\}, \{b, d\}\}$. The explanation closure of p is thus

$$F = (a, b, c); (b, d).$$

On the other hand, the least presumptive elementary explanation of p wrt $\langle K, \mathcal{D}(\mathcal{A}) \rangle$ is given by

$$E = \{a; d,\quad b,\quad c; d\}.$$

In fact, $\mathcal{AS}(K \cup E) = \mathcal{AS}(K \cup \{F\}) = \{\{a, b, c, p, q\}, \{b, d, p, r\}\}$.

The next theorem states that the explanation closure F of G wrt $\langle K, \mathcal{A} \rangle$ exactly reflects all the possible *minimal changes* from the original program K with the minimal elementary explanations $\mathcal{ME}(G)$ wrt $\langle K, \mathcal{A} \rangle$. With this property, we can say that all possible explanations are *assimilated* into the current program so that the resulting program $K \cup \{F\}$ is uniquely determined. Note here that F is a disjunction of conjunctions of abducibles, that is, a DNF formula. If necessary, we can convert F into an equivalent CNF formula (by Theorem 3.2) which is in the form of a program. The merit of introduction of explanation closures is that we can just stay in the traditional abductive framework where the abducibles are given as literals and hence it is not necessary to consider the enlarged abducible set for computing weakest explanations.

In the following, for a set $\mathcal{S}$ of sets of literals, we denote the set of minimal elements in $\mathcal{S}$ as $\mu\,\mathcal{S}$, i.e., $\mu\,\mathcal{S} = \{\, I \in \mathcal{S} \mid \text{there is no } J \in \mathcal{S} \text{ such that } J \subset I \,\}$.

Theorem 3.4 *Let F be the explanation closure of G wrt $\langle\, K, \mathcal{A}\,\rangle$, and $\mathcal{ME}(G)$ be the set of minimal elementary explanations of G wrt $\langle\, K, \mathcal{A}\,\rangle$. Then,*

$$\mathcal{AS}(K \cup \{F\}) = \mu \bigcup_{E \in \mathcal{ME}(G)} \mathcal{AS}(K \cup E).$$

Note in Theorem 3.4 that the program augmented with the explanation closure $K \cup \{F\}$ preserves *all and only minimal answer sets* from the collection of programs with individual minimal elementary explanations. In other words, non-minimal answer sets produced by the minimal elementary explanations together with K are lost in $\mathcal{AS}(K \cup \{F\})$. This is because the program $K \cup \{F\}$ is an EDP, of which any answer set is minimal. For example, when the program K is

$$\begin{aligned} a; b\ &, \\ p &\leftarrow b, \\ p &\leftarrow c, \end{aligned}$$

and $\mathcal{A} = \{a, b, c\}$ is the abducibles, we have $\mathcal{ME}(p) = \{b, c\}$. Then,

$$\mathcal{AS}(K \cup \{b\}) \cup \mathcal{AS}(K \cup \{c\}) = \{\{b, p\}, \{a, c, p\}, \{b, c, p\}\}.$$

On the other hand,

$$\mathcal{AS}(K \cup \{b; c\}) = \{\{b, p\}, \{a, c, p\}\}.$$

When we consider the skeptical entailment, non-minimal answer sets are not useful and eliminating them does not change the consequences that are true in all answer sets.

4 Disjunctions in Extended Abduction

In this section, we extend the notion of disjunctive explanations to allow for removal of abducible disjunctions from programs.

We firstly give a definition for *extended abduction* [14, 16, 26, 13]. The following definition is based on [13].

Definition 4.1 Let $\langle K, \mathcal{A} \rangle$ be an abductive program.

1. A pair (P, N) is a *scenario for* $\langle K, \mathcal{A} \rangle$ if P and N are sets of ground instances of elements from $\mathcal{A}$ and $(K \setminus N) \cup P$ is a consistent program.
2. Let G be a ground literal.
   (a) A pair (P, N) is an *elementary explanation of* G *(wrt* $\langle K, \mathcal{A} \rangle$*)* if (P, N) is a scenario for $\langle K, \mathcal{A} \rangle$ such that $(K \setminus N) \cup P \models G$.
   (b) A pair (P, N) is an *elementary anti-explanation of* G *(wrt* $\langle K, \mathcal{A} \rangle$*)* if (P, N) is a scenario for $\langle K, \mathcal{A} \rangle$ such that $(K \setminus N) \cup P \not\models G$.
   (c) An elementary (anti-)explanation (P, N) of G is *minimal* if for any elementary (anti-)explanation (P', N') of G, $P' \subseteq P$ and $N' \subseteq N$ imply $P' = P$ and $N' = N$.

Thus, to explain or unexplain observations, extended abduction not only introduces hypotheses to a program but also removes them from it. On the other hand, abduction in Definition 3.1 is called *normal abduction*, which only introduces hypotheses to explain observations, and is a special case of extended abduction. That is, E is an explanation of G wrt $\langle K, \mathcal{A} \rangle$ (under normal abduction) iff $(E, \emptyset)$ is an explanation of G wrt $\langle K, \mathcal{A} \rangle$ (under extended abduction).

4.1 Problem in Combining Removed Hypotheses

It is not obvious to extend the notion of elementary (anti-)explanations in extended abduction to take disjunctions of multiple (anti-)explanations. The difficulty lies in the following question: when there are more than one way to remove hypotheses in order to (un)explain an observation, how can we construct a combined (anti-)explanations so that the resulting program reflects the semantics for every possible minimal change of the current program? We illustrate this difficulty with the following example.

Example 4.1 [10, Example 3.4] [6] Let K be the program

$$\begin{aligned} p &\leftarrow a,\ b, \\ p &\leftarrow e, \\ p &\leftarrow q,\ c, \\ q &\leftarrow a,\ d, \\ a &\ , \\ b; d &\ , \\ b; e &\ . \end{aligned}$$

Suppose that the abducibles are $\mathcal{A} = \{a, b, c, d, e\}$. The unique minimal elementary anti-explanations of p wrt $\langle K, \mathcal{A} \rangle$ is

$$(P_1, N_1) = (\emptyset, \{a\}).$$

[6] Example 4.1 was originally described in the context of *view updates* of disjunctive databases in [10]. Here, we modified it for the use in extended abduction.

On the other hand, there are two minimal elementary anti-explanations of p wrt $\langle\, K, \mathcal{D}(\mathcal{A})\,\rangle$: one is (P_1, N_1), and the other is

$$(P_2, N_2) = (\emptyset, \{b; e\}).$$

To express these two changes in one state, Grant *et al.* [10] actually construct the two programs by reflecting these two anti-explanations on the fact part $\mathcal{F}(K)$:

$$K_1 = \mathcal{I}(K) \cup \{\, b; d,\ \ b; e\,\},$$
$$K_2 = \mathcal{I}(K) \cup \{\, a,\ \ b; d\,\}.$$

Then, [10] takes the disjunction of these fact parts, i.e., $\mathcal{F}(K_1) \vee \mathcal{F}(K_2)$, and converting the resulting DNF formula into CNF, yielding

$$((b \vee d) \wedge (b \vee e)) \vee (a \wedge (b \vee d)) = (b \vee d) \wedge (a \vee b \vee e).$$

That is, the new program is computed as

$$K' = \mathcal{I}(K) \cup \{\, b; d,\ \ a; b; e\,\}.$$

By computing the difference between K and K', an anti-explanation of p would be expressed as

$$(P', N') = (\{\, a; b; e\,\},\ \{\, a,\ b; e\,\}).$$

Unless we follow this expensive procedure, it is difficult to compose the last scenario (P', N') directly from the minimal elementary anti-explanations, (P_1, N_1) and (P_2, N_2), of p wrt $\langle\, K, \mathcal{D}(\mathcal{A})\,\rangle$. Moreover, it is impossible to construct (P', N') only from the unique minimal elementary explanation (P_1, N_1) of p wrt $\langle\, K, \mathcal{A}\,\rangle$.

From the above example, one may expect that two (anti-)explanations, (P_1, N_1) and (P_2, N_2), can be combined by constructing a new (anti-)explanation:

$$(\{P_1 \vee P_2,\ N_1 \vee N_2\},\ N_1 \cup N_2).$$

Unfortunately, this is not the case as the next example shows.

Example 4.2 Let K be the program

$$\begin{aligned} &p \leftarrow a,\ not\, b,\\ &p \leftarrow a,\ not\, c,\\ &b,\\ &c, \end{aligned}$$

and the abducibles be $\mathcal{A} = \{a, b, c\}$. The two minimal elementary explanations of p is $(\{a\}, \{b\})$ and $(\{a\}, \{c\})$. Combining these two in the above way results in $(P, N) = (\{\, a,\ b; c\,\},\ \{\, b,\ c\,\})$. However, this scenario cannot be an explanation of p because $(K \setminus N) \cup P \not\models p$.

4.2 From Extended Abduction to Normal Abduction

From the discussion in Section 4.1, we had better consider an alternative way to combine multiple (anti-)explanations in extended abduction. In [13], extended abduction is shown to be reduced to normal abduction. Here, we use this method to translate removal of abducibles from programs to addition of abducibles to programs. Recall that, without loss of generality, the set of abducibles $\mathcal{A}$ can be assumed to be a set of literals and there is no rule which has a non-empty body and a head containing abducible literals. Under this assumption, the translation ν shown in [13] is simplified as follows. For addition of an abducible literal, we do not have to give it a name and leave it as it is. For removal of an abducible literal a, we give a name to a through NAF by $not\ del(a)$. Then, deletion of an abducible a is realized by addition of $del(a)$ to the program.

For an abductive program $\langle K, \mathcal{A} \rangle$, the program $\nu(K, \mathcal{A}) = \langle \nu(K), \nu(\mathcal{A}) \rangle$ is defined as follows.

$$\nu(K) = (K \setminus \mathcal{A}) \cup \{ a \leftarrow not\ del(a) \mid a \in K \cap \mathcal{A} \},$$
$$\nu(\mathcal{A}) = \mathcal{A} \cup \{ del(a) \mid a \in K \cap \mathcal{A} \}.$$

Theorem 4.1 [13, Theorem 1] (P, N) is a minimal elementary explanation of G wrt $\langle K, \mathcal{A} \rangle$ under extended abduction iff E is a minimal elementary explanation of G wrt $\nu(K, \mathcal{A})$ under normal abduction, where $P = \{a \mid a \in E \cap \mathcal{A}\}$ and $N = \{a \mid del(a) \in E\}$.

The above theorem presents that all minimal elementary explanations are computable by normal abduction from $\nu(K, \mathcal{A})$. For anti-explanations, the next theorem shows that $\nu(K, \mathcal{A})$ preserves every minimal elementary anti-explanation of $\langle K, \mathcal{A} \rangle$ in the form of a scenario $(E, \emptyset)$. Namely, we do not have to consider removal of hypotheses in a scenario. Then, to compute these anti-explanations, we can utilize the relationship between explanations and anti-explanations (see [13, Theorem 2]).

Theorem 4.2 *(P, N) is a minimal elementary anti-explanation of G wrt $\langle K, \mathcal{A} \rangle$ iff $(E, \emptyset)$ is a minimal anti-explanation of G wrt $\nu(K, \mathcal{A})$, where $P = \{a \mid a \in E \cap \mathcal{A}\}$ and $N = \{a \mid del(a) \in E\}$.*

4.3 Disjunctive (Anti-)Explanations

Now, we are ready to compose disjunctive explanations for extended abduction. Firstly, we extend Definition 4.1 for extended abduction by allowing removal of disjunctive hypotheses from a program.

Definition 4.2 Let $\langle K, \mathcal{A} \rangle$ be an abductive program, G a ground literal.

1. A pair (P, N) is a *d-scenario for* $\langle K, \mathcal{A} \rangle$ if P is a set of ground instances of elements from $\mathcal{A}$ and N is a set of ground instances of elements from $\mathcal{D}(\mathcal{A})$ such that $(K \setminus N) \cup P$ is a consistent.

2. A d-scenario (P, N) is an *elementary d-explanation of* G *(wrt* $\langle K, \mathcal{A} \rangle$*)* if $(K \setminus N) \cup P \models G$.
3. A d-scenario (P, N) is an *elementary d-anti-explanation of* G *(wrt* $\langle K, \mathcal{A} \rangle$*)* if $(K \setminus N) \cup P \not\models G$.
4. An elementary d-(anti-)explanation (P, N) of G is *minimal* if for any elementary d-(anti-)explanation (P', N') of G, $P \models P'$ and $N \models N'$ imply $P' \models P$ and $N' \models N$.

In the above definition, we allow removal of disjunctive hypotheses from the enlarged abducible set $\mathcal{D}(\mathcal{A})$, but addition of hypotheses is allowed only from the literal abducibles $\mathcal{A}$. This asymmetry is due to our intention that hypotheses to be added should be made disjunctive just in the same way as normal abduction although hypotheses to be removed could only be translated into normal abduction through NAF of the form $not\, del(\_)$. Note also that the minimality of d-(anti-)explanations is now defined through the entailment relationship.

For translating abducible removal into abducible addition, we slightly modify the mapping ν for preserving minimal elementary (anti-)explanations, and consider the mapping ν^d as follows. For an abductive program $\langle K, \mathcal{A} \rangle$, the program $\nu^d(K, \mathcal{A}) = \langle \nu^d(K), \nu^d(\mathcal{A}) \rangle$ is defined as follows.

$$\nu^d(K) = (K \setminus \mathcal{D}(\mathcal{A})) \cup \{ a \leftarrow not\, del(a) \mid a \in K \cap \mathcal{D}(\mathcal{A}) \},$$
$$\nu^d(\mathcal{A}) = \mathcal{A} \cup \{ del(a) \mid a \in K \cap \mathcal{D}(\mathcal{A}) \}.$$

Note that the difference between ν and ν^d is that the naming technique is applied to the enlarged abducible set $\mathcal{D}(\mathcal{A})$ instead of the original abducibles $\mathcal{A}$ only. The new abducible set $\nu^d(\mathcal{A})$ is, however, defined with $\mathcal{A}$ without considering disjunctive hypotheses. This is because we do not have to consider any removal of hypotheses for $\nu^d(K, \mathcal{A})$ so that we can define the notions of *(disjunctive) explanations*, *minimal explanations*, and *explanation closures* in the same way as Definitions 3.2, 3.3, and 3.4 for normal abduction. Similarly, we can define the closure formula for anti-explanations as follows.

Definition 4.3 The *anti-explanation closure of* G *(wrt* $\langle K, \mathcal{A} \rangle$*)* is the disjunctive explanation:

$$\bigvee_{(E, \emptyset) \in \mathcal{MEA}^{\nu}(G)} E,$$

where $\mathcal{MEA}^{\nu}(G)$ is the set of all minimal elementary anti-explanations of G wrt $\nu^d(K, \mathcal{A})$.

The following theorems show that the translation ν^d preserves the minimal answer sets from the program augmented with any minimal elementary d(-anti)-explanation. Here, for a program K containing literals of the form $del(\_)$, we will write:

$$\mathcal{AS}^{-del}(K) = \mu \{ S \cap \mathcal{L}_K \mid S \in \mathcal{AS}(K) \},$$

where $\mathcal{L}_K$ denotes the set of literals in the language of K not containing any literal of the form $del(\_)$. Note that we need to select the minimal elements from the right hand side. This is because eliminating all literals of the form $del(\_)$ from each answer set may produce a literal set that properly includes others.

Theorem 4.3 *Let F be the explanation closure of G wrt $\langle K, \mathcal{A} \rangle$, and $\mathcal{ME}^d(G)$ be the set of minimal elementary d-explanations of G wrt $\langle K, \mathcal{A} \rangle$. Then,*

$$\mathcal{AS}^{-del}(\nu^d(K) \cup \{F\}) = \mu \bigcup_{(P,N) \in \mathcal{ME}^d(G)} \mathcal{AS}((K \setminus N) \cup P).$$

Theorem 4.4 *Let H be the anti-explanation closure of G wrt $\langle K, \mathcal{A} \rangle$, and $\mathcal{MEA}^d(G)$ be the set of minimal elementary d-anti-explanations of G wrt $\langle K, \mathcal{A} \rangle$. Then,*

$$\mathcal{AS}^{-del}(\nu^d(K) \cup \{H\}) = \mu \bigcup_{(P,N) \in \mathcal{MEA}^d(G)} \mathcal{AS}((K \setminus N) \cup P).$$

Example 4.3 (cont. from Example 4.1) The fact part $\mathcal{I}(K) = K \cap \mathcal{D}(\mathcal{A}) = \{ a,\ b; d,\ b; e \}$ is translated into

$$\begin{aligned} a &\leftarrow not\, del(a), \\ b; d &\leftarrow not\, del(b; d), \\ b; e &\leftarrow not\, del(b; e). \end{aligned}$$

The two minimal elementary anti-explanations of p wrt $\nu^d(K, \mathcal{A})$ are $(\{del(a)\}, \emptyset)$ and $(\{del(b; e)\}, \emptyset)$, which respectively correspond to the two d-anti-explanations of p wrt $\langle K, \mathcal{A} \rangle$, $(P_1, N_1) = (\emptyset, \{a\})$ and $(P_2, N_2) = (\emptyset, \{b; e\})$. Then, the anti-explanation closure of p is $H = del(a); del(b; e)$. Assimilating this formula into the program, we obtain the new program $K' = \nu^d(K) \cup \{del(a); del(b; e)\}$. Then,

$$\mathcal{AS}^{-del}(K') = \{\{b\}, \{d, e, p\}, \{a, d, q\}\}.$$

Example 4.4 (cont. from Example 4.2) The fact part $\mathcal{F}(K)$ is translated into

$$\begin{aligned} b &\leftarrow not\, del(b), \\ c &\leftarrow not\, del(c). \end{aligned}$$

The two minimal elementary explanations of p wrt $\nu^d(K, \mathcal{A})$ are $\{a, del(b)\}$ and $\{a, del(c)\}$, which respectively correspond to $(\{a\}, \{b\})$ and $(\{a\}, \{c\})$. Then, the explanation closure is $F = (a, del(b)); (a, del(c))$. By converting F into CNF, the minimal explanation of p wrt $\nu^d(K, \mathcal{A})$ is obtained as $E = \{ a,\ del(b); del(c) \}$. Then,

$$\mathcal{AS}^{-del}(\nu^d(K) \cup E) = \{\{a, b, p\}, \{a, c, p\}\}.$$

5 Related Work

1. Disjunctive explanations. The idea of taking a disjunction of multiple explanations has appeared at times in the literature of computing abduction, although no previous work has formally investigated the effect of such disjunctive explanations in depth. Helft *et al.* [11] define an explanation as a disjunction of elementary explanations in abduction from first-order theories for answering queries in *circumscription.* Konolige [20] defines a *cautious explanation* as a disjunction of all preferred explanations, and uses it to relate consistency-based explanations with abductive explanations in propositional causal theories. Lin [22] provides a method to compute *weakest sufficient conditions* for propositional theories, in which he constructs the disjunction of elementary explanations obtained from prime implicates. In ALP, disjunctions of elementary explanations are sometimes obtained in computing abduction through Clark completion [3, 8, 23]. Such procedures are designed for computing normal abduction from hierarchical or acyclic NLPs. Inoue and Sakama [16] extend this completion method to compute extended abduction. We can use these procedures to compute explanation closures directly in some restricted classes of logic programs.

2. View updates in disjunctive databases. Although there are some studies on updating incomplete information in relational databases [1], only a few works [10, 7] focused on updating disjunctive databases. Grant *et al.* [10] translate view updates into a set of disjunctive facts based on expansion of an SLD-tree, so that updates are achieved by inserting/deleting these disjunctive facts to/from a database. Their method is correct for stratified programs, but cannot achieve an insertion of p into a non-stratified EDP K shown in Example 3.1. Fernández *et al.* [7] realize view updates in a wide class of EDPs through construction of minimal models that satisfy an update request. In their algorithm, however, computation is done on all possible models of the Herbrand base, and how to compute disjunctive solutions directly from changes of facts was an *open problem* in the class of EDPs. We solved this problem by translating extended abduction to normal abduction without computing all possible models. Furthermore, updates are performed without using abduction in [10, 7]. Hence, the notion of disjunctive (anti-)explanations in abduction does not appear in these work.

For non-disjunctive deductive databases, abductive frameworks have been used to realize view updates. Bry [2] translates abduction into a disjunctive program and database updates are realized by bottom-up computation on a meta-program specifying an update procedure. Kakas and Mancarella [18] characterize view updates through abduction in deductive databases. The procedures in [18, 2] are based on normal abduction and do not consider extended abduction.

3. Knowledge assimilation with abduction. Not much work has been reported to assimilate obtained multiple explanations into the current knowledge base. Kakas and Mancarella [19] discussed two ways for handling the problem of multiple explanations. One is to generate all consistent scenarios accounting for an observation and work with all of them simultaneously. They suggest to use an ATMS for this purpose. The other way is to generate one preferred explanation at a time according to some priority. Since such a choice of explanation could be wrong in the subsequent observations, they suggest the use of a belief revision mechanism through a Doyle-style TMS.

Our proposal somewhat differs from Kakas and Mancarella's two methods. Our method is similar to the spirit suggested by Fagin *et al.* [5], which defines the result of assimilation or updates to be the disjunction of all the possible theories with minimal change. This method presents a semantically consistent picture of theory changes. Rossi and Naqvi [25] optimize this approach by taking the disjunction of updated extensional databases instead of composing the disjunction of the whole databases with intensional ones. Grant *et al.* [10] follow the same line on view updates in disjunctive databases. An interesting alternative approach is also suggested by Fagin *et al.* [6], in which multiple alternative theories called "flocks" are kept as they are.

6 Summary

This paper has presented a method to construct the weakest explanations and anti-explanations in normal and extended abduction. For normal abduction, we formally established the effect of disjunctive explanations, in which all and only minimal answer sets are preserved for the minimal elementary explanations. We also presented that the explanation closure is equivalent to a least presumptive explanation consisting of disjunctive hypotheses. These results imply a practical merit that computing least presumptive explanations wrt $\langle K, \mathcal{D}(\mathcal{A}) \rangle$ can easily be realized by traditional abductive procedures [18, 3, 15, 8, 17, 16] for $\langle K, \mathcal{A} \rangle$ or corresponding *answer set programming* [24] which simulates normal abduction. That is, the minimal elementary explanations are firstly computed by these procedures, then the disjunction of them is just composed. We have also applied these results to extended abduction, and proposed a method to combine multiple solutions that involve removal of hypotheses.

The notion of disjunctive explanations is quite useful in various applications, and our method has shed some light on the problem of knowledge assimilation. In particular, considering view updates in disjunctive databases is generally difficult in the presence of disjunctive information. Our solution in this paper correctly achieves view updates in a large class of disjunctive databases.

References

1. S. Abiteboul. Updates, a new frontier. In: *Proceedings of the 2nd International Conference on Database Theory, Lecture Notes in Computer Science*, 326, Springer, pages 1–18, 1988.
2. F. Bry. Intensional updates: abduction via deduction. In: *Proceedings of ICLP '90*, pages 561–575, MIT Press, 1990.
3. L. Console, D. Theseider Dupré and P. Torasso. On the relationship between abduction and deduction. *Journal of Logic and Computation*, 1:661–690, 1991.
4. T. Eiter, G. Gottlob, and H. Mannila. Adding disjunction to Datalog. In: *Proc. 13th ACM SIGACT-SIGMOD-SIGART Symp. on Principles of Database Systems*, pages 267–278, 1994.
5. R. Fagin, J. D. Ullman, and M. Y. Vardi. On the semantics of updates in databases (preliminary report). In: *Proceedings of the 2nd ACM SIGACT-SIGMOD Symposium on Principles of Database Systems*, pages 352–365, 1983.
6. R. Fagin, G. M. Kuper, J. D. Ullman, and M. Y. Vardi. Updating logical databases. In: *Advances in Computing Research*, Volume 3, pages 1–18, JAI Press, 1986.

7. J. Fernández, J. Grant and J. Minker. Model theoretic approach to view updates in deductive databases. *Journal of Automated Reasoning*, 17:171–197, 1996.
8. T. H. Fung and R. Kowalski. The iff procedure for abductive logic programming. *Journal of Logic Programming*, 33:151–165, 1997.
9. M. Gelfond and V. Lifschitz. Classical negation in logic programs and disjunctive databases. *New Generation Computing*, 9:365–385, 1991.
10. J. Grant, J. Horty, J. Lobo and J. Minker. View updates in stratified disjunctive databases. *Journal of Automated Reasoning*, 11:249–267, 1993.
11. N. Helft, K. Inoue, and D. Poole. Query answering in circumscription. In: *Proceedings of IJCAI-91*, pages 426–431, Morgan Kaufmann, 1991.
12. K. Inoue. Hypothetical reasoning in logic programs. *Journal of Logic Programming*, 18(3):191-227, 1994.
13. K. Inoue. A simple characterization of extended abduction. In *Proceedings of the 1st International Conference on Computational Logic, Lecture Notes in Artificial Intelligence*, 1861, pages 718–732, Springer, 2000.
14. K. Inoue and C. Sakama. Abductive framework for nonmonotonic theory change. In: *Proceedings of IJCAI-95*, pages 204–210, Morgan Kaufmann, 1995.
15. K. Inoue and C. Sakama. A fixpoint characterization of abductive logic programs. *Journal of Logic Programming*, 27(2):107–136, 1996.
16. K. Inoue and C. Sakama. Computing extended abduction through transaction programs. *Annals of Mathematics and Artificial Intelligence*, 25(3,4):339-367, 1999.
17. A. C. Kakas, R. A. Kowalski and F. Toni. The role of abduction in logic programming. In: D. M. Gabbay, C. J. Hogger and J. A. Robinson (eds.), *Handbook of Logic in Artificial Intelligence and Logic Programming*, volume 5, pages 235–324, Oxford University Press, 1998.
18. A. C. Kakas and P. Mancarella. Database updates through abduction. In: *Proceedings of the 16th International Conference on Very Large Databases*, pages 650–661, Morgan Kaufmann, 1990.
19. A. C. Kakas and P. Mancarella. Knowledge assimilation and abduction. In: J. P. Martins and M. Reinfrank (eds.), *Truth Maintenance Systems, Lecture Notes in Artificial Intelligence*, 515, 54–70, Springer, 1991.
20. K. Konolige. Abduction versus closure in causal theories. *Artificial Intelligence*, 53:255–272, 1992.
21. V. Lifschitz, L. R. Tang, and H. Turner. Nested expressions in logic programs. *Annals of Mathematics and Artificial Intelligence*, 25:369–389, 1999.
22. F. Lin. On strongest necessary and weakest sufficient conditions. In: *Proceedings of the 7th International Conference on Principles of Knowledge Representation and Reasoning*, pages 167–175, Morgan Kaufmann, 2000.
23. F. Lin and J.-H. You. Abductive logic programming: a new definition and an abductive procedure based on rewriting. In: *Proceedings of IJCAI-01*, pages 655–661, Morgan Kaufmann, 2001.
24. V. W. Marek, and M. Truszczyński. Stable models and an alternative logic programming paradigm. In: K. R. Apt *et al.*, editors, *The Logic Programming Paradigm—A 25 Year Perspective*, pages 375–398, Springer, 1999.
25. F. Rossi and S. A. Naqvi. Contributions to the view update problem. In: *Proceedings of ICLP '89*, pages 398–415, MIT Press, 1989.
26. C. Sakama and K. Inoue. Updating extended logic programs through abduction. In: *Proceedings of the 5th International Conference on Logic Programming and Nonmonotonic Reasoning, Lecture Notes in Artificial Intelligence*, 1730, pages 147-161, Springer, 1999.