The Effect of Partial Deduction in Abductive Reasoning*

Chiaki Sakama†
Advanced Software Technology and
Mechatronics Research Institute of Kyoto
17 Chudoji Minami-machi, Shimogyo, Kyoto 600 Japan
sakama@astem.or.jp

Katsumi Inoue
Department of Information and Computer Sciences
Toyohashi University of Technology
Tempaku-cho, Toyohashi 441 Japan
inoue@tutics.tut.ac.jp

Abstract

Partial deduction is known as an optimization technique in logic programming. In the context of abductive logic programming, however, we present in this paper that normal partial deduction does not preserve explanations for abductive reasoning. Then we provide an alternative method of partial deduction, called *abductive partial deduction*, which is shown to preserve the meanings of abductive logic programs. A method of *partial abduction* is also introduced as an optimization for abductive reasoning in logic programs.

1 Introduction

Partial deduction [Kom92] is an optimization technique in logic programming, which performs deduction on a part of a program while retaining the meaning of the original program. Partial deduction is used in various extensions of logic programming, and is known to preserve the semantics of normal logic programs [TS84, LS91, Seki91, Seki93] and disjunctive logic programs [SS94, BD94].

Abductive logic programming [KKT92] is one of the extensions of logic programming, which realizes a mechanism of abductive reasoning in AI. Recent studies have widely investigated theoretical aspects of abductive logic

*In *Proc. 12th International Conference on Logic Programming* (ICLP'95), MIT Press, pp. 383-397, 1995.

†Address after April 1995: Department of Computer and Communication Sciences, Wakayama University, 930 Sakaedani, Wakayama 640 Japan.

programming, then optimizing abductive logic programs is becoming an important issue from practical viewpoints.

In abductive logic programming, a primary goal is computing explanations for a given observation. In particular, selecting the "best" explanations among many candidate explanations is important. Considering partial deduction in abductive logic programs, our special concern is the preservation of such explanations. However, since the inference mechanism in abductive logic programming is different from standard logic programming, it is not clear whether partial deduction techniques are directly applicable to abductive logic programs.

In this paper, we investigate the effect of partial deduction in abductive logic programs. We first show that normal partial deduction does not preserve the meanings of abductive logic programs. Then we propose an alternative method of partial deduction, called *abductive partial deduction*, which preserves explanations in abductive reasoning. Some variants of abductive partial deduction are introduced to preserve best explanations. We finally apply the technique of abductive partial deduction to *partial abduction*, which optimizes abductive reasoning in logic programs.

The rest of this paper is organized as follows. In Section 2, we review abductive framework and abductive logic programming. In Section 3, we present the problem of normal partial deduction in abductive logic programs and propose new partial deduction techniques. Section 4 addresses a method of partial abduction. Section 5 discusses related issues and Section 6 summarizes the paper.

2 Abductive Framework

Abduction reasons from observation to explanation, which is formally presented as follows.

Given a background theory T and an *observation* O, *abduction* is defined as an inference of an *explanation* E of O such that

$$T \cup E \models O \text{ where } T \cup E \text{ is consistent.} \quad (1)$$

Abductive logic programming [KKT92] is a form of such abductive framework in which T is given as a logic program.

A *normal logic program* is a finite set of clauses of the form:

$$A \leftarrow B_1 \wedge \ldots \wedge B_m \wedge not\, B_{m+1} \wedge \ldots \wedge not\, B_n \;\; (n \geq m \geq 0) \quad (2)$$

where A and B_i's are atoms and *not* is the negation-as-failure operator. The left-hand side of the clause (2) is the *head*, and the right-hand side is the *body*. A clause with the empty head (i.e., $A = false$) is an *integrity constraint*. A program containing no *not* is a *Horn logic program*, and a Horn logic program without integrity constraints is a *definite logic program*. We often use the Greek letter Γ to denote a conjunction (possibly *true*) in the body. Since we are concerned with the semantic aspect of abductive logic programs, we consider ground programs throughout the paper.

An *abductive logic program* is defined as a pair $\langle P, \mathcal{A} \rangle$, where P is a normal logic program and $\mathcal{A}$ is a set of atoms called *abducibles*. An abductive logic program is called an *abductive Horn program* if P is a Horn logic program, and especially called an *abductive definite program* if P is a definite logic program. We often write an abductive logic program just as P when the set of abducibles $\mathcal{A}$ is clear from the context.

A declarative semantics of abductive logic programs is given by the notion of *belief sets* [IS93]. First, given a normal logic program P and its interpretation I, I is a *stable model* [GL88] of P if I coincides with the least Herbrand model of the Horn logic program P^I defined as

$$P^I = \{ A \leftarrow B_1 \wedge \ldots \wedge B_m \mid \text{there is a ground clause of the form (2) from } P \text{ such that } \{B_{m+1}, \ldots, B_n\} \cap I = \emptyset \}.$$

A program P is *consistent* if it has at least one stable model. Next, given an abductive logic program $\langle P, \mathcal{A} \rangle$, an interpretation I is a *belief set* of P (wrt E) if I is a stable model of the normal logic program $P \cup E$ where $E \subseteq \mathcal{A}$.[1] An abductive logic program has multiple belief sets in general.

In an abductive logic program, abductive reasoning is characterized as follows. Let $\langle P, \mathcal{A} \rangle$ be an abductive logic program and O a ground atom which represents an *observation*.[2] Then, a set $E \subseteq \mathcal{A}$ is an *explanation* of O in P iff O is true in every belief set of P wrt E. In this sense, we consider the entailment relation in (1) under the belief set semantics.[3]

In abductive reasoning, computation of the *best* explanations is particularly important, since there might be many candidate explanations which can account for an observation from the relation (1). A pre-specified set of abducibles $\mathcal{A}$ is one of such conditions to restrict candidate hypotheses. Further criteria for choosing explanations are proposed in the literature. Among them, we consider the following ones in this paper.

(a) *minimal* explanation [Pop73, CP86, Poo89, etc.]
An explanation E is *minimal* iff there is no other explanation F such that $F \subset E$.

(b) *least specific* [Sti89] or *least presumptive* [Poo89] explanation
An explanation E is *less specific* than an explanation F in P (written $E \leq F$) iff $P \cup F \models E$.[4] An explanation E is *least specific* in P iff there is no explanation F in P such that $F < E$.

[1] In [KM90], belief sets are called *generalized stable models*.

[2] Without loss of generality, O is assumed to be a non-abducible ground atom [IS93].

[3] We can consider an alternative definition of explanations such that E is an explanation of O iff O is true in *some* belief set of P wrt E. Employing either definition is not important in this paper.

[4] Here, E is identified with the conjunction of each abducible included in E.

(c) *most specific* [Sti89] or *basic* [CP86] explanation

An explanation E is *more specific* than an explanation F in P (written $F \leq E$) iff $P \cup E \models F$. An explanation E is *most specific* in P iff E is a minimal set satisfying the condition that there is no explanation F in P such that $E < F$ and $E \not\subset F$.

Here, $\leq$ is a pre-order relation and $E < F$ iff $E \leq F$ and $F \not\leq E$.

The condition (a) is usually imposed to avoid introducing unnecessary assumptions. On the other hand, the criterion (b) or (c) is chosen depending on applications. For instance, (b) is appropriate in natural language processing, and (c) is useful in diagnostic tasks [Sti89]. Note that the definition of the most specific explanation implies that any explanation F such that $E < F$ contains E as a proper subset. The most specific explanation E is one of such minimal sets satisfying the condition.

Example 2.1 ([Poo89]) Consider the program:

$$sore\_leg \leftarrow broken\_leg, \quad (3)$$

$$broken\_leg \leftarrow broken\_tibia, \quad (4)$$

where $\mathcal{A} = \{broken\_leg, broken\_tibia\}$. Given the observation $O = sore\_leg$, $\{broken\_leg\}$ and $\{broken\_tibia\}$ are the two minimal explanations; $\{broken\_leg\}$ is the least specific explanation; and $\{broken\_tibia\}$ is the most specific explanation.

Also note that the least/most specific explanations are not necessarily minimal explanations.

Example 2.2 Let $P = \{\ o \leftarrow a \wedge b,\ \ b \leftarrow a\ \}$ where $\mathcal{A} = \{a, b\}$ and $\{a\} \leq \{a, b\} \leq \{a\}$. Then, both $\{a\}$ and $\{a, b\}$ are the least specific explanations of o, while $\{a, b\}$ is not a minimal explanation. On the other hand, let $P = \{\ o \leftarrow a,\ \ a \leftarrow c,\ \ c \leftarrow a \wedge b\ \}$ where $\mathcal{A} = \{a, b, c\}$ and $\{a\} < \{c\} < \{a, b\}$. Then $\{a, b\}$ is a most specific explanation of o, while $\{a\}$ is a minimal explanation.

In this paper, when we use the term "best explanations", we mean either of the above three explanations.

3 Abductive Partial Deduction

3.1 Normal Partial Deduction does not Preserve Best Explanations

We first examine whether partial deduction preserves explanations, especially best ones in abductive reasoning.

For a normal logic program P, partial deduction is defined as unfolding between clauses as follows.

Given a clause C from P,

$$C:\ H \leftarrow A \wedge \Gamma\,,$$

suppose that $C_1, \ldots, C_k$ are all of the clauses in P such that each of which has the atom A in its head:

$$C_i:\ A \leftarrow \Gamma_i \quad (1 \leq i \leq k)\,.$$

Then *normal partial deduction* of P (with respect to C on A) is defined as the program $\pi^N_{\{C;A\}}(P)$ (called a *residual program*) such that

$$\pi^N_{\{C;A\}}(P) = (P \setminus \{C\}) \cup \{C'_1, \ldots, C'_k\}$$

where each C'_i is defined as

$$C'_i:\ H \leftarrow \Gamma \wedge \Gamma_i\,.$$

When we simply say normal partial deduction of P (written $\pi^N(P)$), it means normal partial deduction of P with respect to any clause on any atom. Note here that when there is no C_i having the atom A in its head, the clause C is just removed in the residual program $\pi^N_{\{C;A\}}(P)$.

It is known that normal partial deduction preserves the least Herbrand model semantics of definite logic programs [TS84], the perfect model semantics of stratified logic programs [Seki91], and the stable/well-founded semantics of normal logic programs [Seki90, Seki93].

Our concern here is whether normal partial deduction also preserves the meanings of abductive logic programs. Unfortunately, the next example shows that normal partial deduction does not preserve belief sets in general.

Example 3.1 In the program of Example 2.1, the set $I = \{sore\_leg,\ broken\_leg\}$ becomes a belief set of P with respect to the assumption $\{broken\_leg\}$. On the other hand, performing partial deduction on *broken_leg* in the clause (3) generates the residual program:

$$\begin{aligned} sore\_leg &\leftarrow broken\_tibia, \\ broken\_leg &\leftarrow broken\_tibia, \end{aligned}$$

in which the above set I is not a belief set anymore.

The above example shows that normal partial deduction does not preserve (best) explanations either. For instance, in the original program, *sore_leg* has two minimal explanations, $\{broken\_leg\}$ and $\{broken\_tibia\}$. However, $\{broken\_tibia\}$ is the unique minimal explanation in the residual program. Moreover, the least specific explanation $\{broken\_leg\}$ is lost in the residual program. On the other hand, if we perform partial deduction on *broken_tibia* in the clause (4) of the original program, the residual program contains the single clause:

$$sore\_leg \quad \leftarrow \quad broken\_leg,$$

and the most specific explanation $\{broken\_tibia\}$ is lost.

The problem is explained as follows. In standard logic programming, the meaning of a program is given by the set of facts derived from the program and normal partial deduction preserves such facts in general. In abductive logic programming, however, the meaning of a program is given by the set of "conditional consequences". That is, new facts are possibly derived in a program by assuming intermediate atoms as hypotheses. In such a situation, causal relationships between atoms are also important and normal partial deduction often loses such relations as presented in the above example.

Thus, to preserve the meanings of abductive logic programs we may need some mechanism for reserving intermediate atoms during partial deduction. To this effect, we introduce *abductive partial deduction* which includes such mechanism.

3.2 Abductive Partial Deduction = Partial Deduction + Reservation

Definition 3.1 Let $\langle P, \mathcal{A} \rangle$ be an abductive logic program and C a clause from P of the form:

$$C:\; H \leftarrow A \wedge \Gamma\,. \tag{5}$$

Suppose that $C_1, \ldots, C_k$ are all of the clauses in P such that each of which has the atom A in its head:

$$C_i:\; A \leftarrow \Gamma_i \quad (1 \leq i \leq k)\,. \tag{6}$$

Then *abductive partial deduction* of P (with respect to C on A) is defined as the *residual program* $\pi^{A}_{\{C;A\}}(P)$ such that

$$\pi^{A}_{\{C;A\}}(P) = \begin{cases} P \cup \{C'_1, \ldots, C'_k\}, & \text{if } A \text{ is an abducible;} \\ (P \setminus \{C\}) \cup \{C'_1, \ldots, C'_k\}, & \text{otherwise,} \end{cases}$$

where each C'_i is a clause of the form:

$$C'_i:\; H \leftarrow \Gamma \wedge \Gamma_i\,. \tag{7}$$

By $\pi^{A}(P)$, we mean abductive partial deduction of P with respect to any clause on any atom.

The idea of abductive partial deduction is that when partial deduction is performed on abducibles, abductive partial deduction retains the original clause C together with the unfolded clauses C'_i. With this mechanism, abductive partial deduction *reserves* intermediate atoms which could be used as assumptions.

Now we first show that abductive partial deduction preserves belief sets of abductive logic programs.

Lemma 3.1 ([Seki90, SS94]) Let P be a normal logic program. Then I is a stable model of P iff I is a stable model of $\pi^N(P)$. $\Box$

Theorem 3.2 Let $\langle P, \mathcal{A} \rangle$ be an abductive logic program. Then I is a belief set of P iff I is a belief set of $\pi^A(P)$.

Proof: Let I be a belief set of P. Then I is a stable model of $P \cup E$ for some $E \subseteq \mathcal{A}$. Let us consider normal partial deduction of $P \cup E$ with respect to any clause C of the form (5) on any atom A. If A is not an abducible, $\pi^N_{\{C;A\}}(P \cup E) = \pi^A_{\{C;A\}}(P \cup E) = \pi^A_{\{C;A\}}(P) \cup E$. By Lemma 3.1, I is a stable model of $\pi^N_{\{C;A\}}(P \cup E)$, then so is $\pi^A_{\{C;A\}}(P) \cup E$. Hence, I is a belief set of $\pi^A_{\{C;A\}}(P)$. Else if A is an abducible, consider a tautological clause $C' = A \leftarrow A$ and $P' = P \cup \{C'\}$. Clearly, P' does not change the meaning of P. Then, I is a stable model of $\pi^N_{\{C;A\}}(P \cup E)$ iff I is a stable model of $\pi^N_{\{C;A\}}(P' \cup E)$. On the other hand, $\pi^N_{\{C;A\}}(P' \cup E) = \pi^A_{\{C;A\}}(P \cup E) \cup \{C'\}$ holds, hence I is a stable model of $\pi^A_{\{C;A\}}(P \cup E) \cup \{C'\} = \pi^A_{\{C;A\}}(P \cup E)$. Next we show that I is also a stable model $\pi^A_{\{C;A\}}(P) \cup E$. The difference between $\pi^A_{\{C;A\}}(P \cup E)$ and $\pi^A_{\{C;A\}}(P) \cup E$ is that if A is included in E, the clause $H \leftarrow \Gamma$ is generated from C and $A \leftarrow$ by unfolding in $\pi^A_{\{C;A\}}(P \cup E)$. In this case, however, the clause is also derived in $\pi^A_{\{C;A\}}(P) \cup E$ from C and $A \in E$. Thus, I is a stable model of $\pi^A_{\{C;A\}}(P \cup E)$ iff I is a stable model of $\pi^A_{\{C;A\}}(P) \cup E$. Hence, I is a belief set of $\pi^A_{\{C;A\}}(P)$.

The converse is shown in the same manner. $\Box$

Corollary 3.3 Normal partial deduction preserves belief sets if unfolding is performed on non-abducible atoms. $\Box$

Thus abductive partial deduction preserves the meanings of abductive logic programs by reserving abducibles when unfolding. Reserving abducibles is a sufficient condition but not always necessary.

Example 3.2 Let $P = \{\ o \leftarrow a \wedge p,\ \ p \leftarrow q\ \}$ where $\mathcal{A} = \{a\}$. Then, all P, $\pi^N(P)$, and $\pi^A(P)$ have the belief sets $\emptyset$ and $\{a\}$.

The proof of Theorem 3.2 shows an alternative characterization of abductive partial deduction.

Corollary 3.4 Let $\langle P, \mathcal{A} \rangle$ be an abductive logic program. Then, $\pi^A(P) = \pi^N(P')$ where $P' = P \cup \{\ A \leftarrow A \mid A \in \mathcal{A}\ \}$. $\Box$

That is, to compute abductive partial deduction we can use normal partial deduction in a program containing tautological clauses for each abducible. For computing explanations, Theorem 3.2 implies that abductive partial deduction preserves explanations, especially the best ones.

Corollary 3.5 For any observation O, E is an explanation of O in P iff E is an explanation of O in $\pi^A(P)$. □

Example 3.3 Consider again the program of Example 2.1. Then abductive partial deduction with respect to the clause (3) on *broken_leg* generates the residual program:

$$\begin{array}{rcl} sore\_leg & \leftarrow & broken\_leg, \\ sore\_leg & \leftarrow & broken\_tibia, \\ broken\_leg & \leftarrow & broken\_tibia, \end{array}$$

in which the minimal explanations $\{broken\_leg\}$ and $\{broken\_tibia\}$, the least specific explanation $\{broken\_leg\}$, and the most specific explanation $\{broken\_tibia\}$, are preserved. On the other hand, if we perform abductive partial deduction in the original program with respect to the clause (4) on *broken_tibia*, the residual program coincides with the original program and each explanation is also unchanged.

3.3 Further Optimization

In this section, we consider variants of abductive partial deduction in *abductive Horn programs*. Abductive Horn programs are used as background theories for abduction in many studies. To compute best explanations in abductive Horn programs, we can further optimize abductive partial deduction presented in the preceding section. Then we first consider optimizing abductive Horn programs for the most specific explanations.

Let us introduce a couple of notations. The *dependency graph* of a Horn program P is a directed graph in which each node presents a ground atom and there is a directed edge from A to B (we say A *depends* on B) iff there is a ground clause from P such that A appears in the head and B appears in the body of the clause. A program is said *acyclic with respect to abducibles* if the dependency graph contains no directed cycle that goes through two abducibles. An abducible A is called *terminal* if A depends on no other abducibles.

To compute most specific explanations in abductive Horn programs, we modify abductive partial deduction of Definition 3.1 as follows:

$$\pi^{MSE}_{\{C;A\}}(P) = \begin{cases} P, & \text{if } A \text{ is a terminal abducible;} \\ (P \setminus \{C\}) \cup \{C'_1, \ldots, C'_k\}, & \text{otherwise.} \end{cases}$$

The notion of $\pi^{MSE}(P)$ is correspondingly defined as before.

The above definition presents that a program is unchanged when partial deduction is performed on terminal abducibles, while unfolding is done as usual when partial deduction is performed on atoms other than terminal abducibles. We call such a variant as *abductive partial deduction for MSE*.

A program P is called *separable* if P contains no clause such that the head has an abducible and the body includes non-abducibles. In a separable program, an abducible can be defined only by abducibles. Thus, structural knowledge about the relations between abducibles is completely aside from the non-abducible atoms, which explains the name. Most of abductive programs in the literature are of this form. Now we have the following result.

Theorem 3.6 Let P be a separable abductive definite program and O an observation. Then, for any most specific explanation E of O in P, there is an explanation F of O in $\pi^{MSE}(P)$ such that $E \leq F$. Also, any explanation F of O in $\pi^{MSE}(P)$ is an explanation of O in P.

Proof: Let E be a most specific explanation of O in P. If partial deduction is performed on non-abducible atoms, $\pi^{MSE}(P)$ reduces to normal partial deduction and the result follows by Corollary 3.3. Else if partial deduction is performed on terminal abducibles, $\pi^{MSE}(P) = P$ and the result also holds. Otherwise, consider the case that partial deduction is performed on a non-terminal abducible A. Since P is separable, there is a clause C_i of the form (6) such that Γ_i consists of abducibles. Then, put $F = E \setminus \{A\} \cup \Gamma_i$. Since E is an explanation of O in P, F is an explanation of O in $\pi^{MSE}(P)$. Also, $P \cup E \models O$ implies $P \cup F \models O$, hence F is an explanation of O in P. By the construction of F, $P \cup F \models E$ holds, therefore $E \leq F$.

Moreover, if F is an explanation of O in $\pi^{MSE}(P)$, $\pi^{MSE}(P) \cup F \models O$ holds. In this case, $\pi^{MSE}(P)$ just reduces a deduction step between O and F in P, hence $P \cup F \models O$ also holds. Since $P \cup F$ is consistent, F is an explanation of O in P. □

Example 3.4 Let $P = \{\ o \leftarrow a,\quad a \leftarrow b\ \}$ where $\mathcal{A} = \{a, b\}$ and b is terminal. Then o has the most specific explanation $\{b\}$, and $\pi^{MSE}_{\{o \leftarrow a;a\}}(P) = \{\ o \leftarrow b,\ \ a \leftarrow b\ \}$ has the same most specific explanation $\{b\}$. On the other hand, let $P = \{\ o \leftarrow a,\ \ a \leftarrow b,\ \ b \leftarrow a\ \}$ with $\mathcal{A} = \{a, b\}$, where neither a nor b is terminal. Then, both $\{a\}$ and $\{b\}$ are the most specific explanations of o. In this case, $\pi^{MSE}_{\{a \leftarrow b;b\}}(P) = \{\ o \leftarrow a,\ \ a \leftarrow a,\ \ b \leftarrow a\ \}$ has the most specific explanation $\{a\}$, where $\{b\} \leq \{a\}$ holds.

Thus $\pi^{MSE}(P)$ approximates P in the sense that the residual program does not necessarily preserve the most specific explanation E which is minimal, but has an explanation F more specific than E in the original program. In particular, if a program is acyclic with respect to abducibles, the following strong result holds.

Theorem 3.7 Let P be a separable abductive definite program acyclic wrt abducibles. Then, for any observation O, E is a most specific explanation of O in P iff E is a most specific explanation of O in $\pi^{MSE}(P)$.

Proof: We first show that any most specific explanation E of O in P consists of terminal abducibles. If an abducible $A \in E$ is not terminal, there is a clause C_i of the form (6) such that Γ_i consists of abducibles. Then, put $F = E \setminus \{A\} \cup \Gamma_i$. In this case, F is an explanation of O in P and $E \leq F$ holds. On the other hand, P is acyclic wrt abducibles, then $F \not\leq E$, thereby $E < F$. Since E is a most specific explanation, $E < F$ implies $E \subset F$. Then $A \in E$ implies $A \in F$, so $A \in \Gamma_i$, which contradicts the fact that P is acyclic wrt abducibles. Therefore, any most specific explanation E of O in P consists of terminal abducibles. Since $\pi^{MSE}(P)$ preserves terminal abducibles, the result follows. □

Note that when a program is not separable or contains integrity constraints, reserving terminal abducibles is not enough.

Example 3.5 Consider the non-separable program $P = \{\ o \leftarrow a,\ \ a \leftarrow b \wedge p,\ \ p \leftarrow q\ \}$ with $\mathcal{A} = \{a, b\}$, where $\{a\}$ is the most specific explanation of o. Then, reserving only the terminal abducible b, the residual program $\pi^{MSE}_{\{o \leftarrow a;a\}}(P) = \{\ o \leftarrow b \wedge p,\ \ a \leftarrow b \wedge p,\ \ p \leftarrow q\ \}$ has no explanation of o. On the other hand, consider the program containing integrity constraints $P = \{\ o \leftarrow a,\ \ a \leftarrow b,\ \ \leftarrow b\ \}$ where $\mathcal{A} = \{a, b\}$ and b is terminal. Then $\{a\}$ is the most specific explanation of o in P, while the residual program $\pi^{MSE}_{\{o \leftarrow a;a\}}(P) = \{\ o \leftarrow b,\ \ a \leftarrow b,\ \ \leftarrow b\ \}$ has no explanation of o.

Next we consider a variant for the least specific explanations.

To compute the least specific explanations in abductive Horn programs, we can simplify residual programs as follows. *Abductive partial deduction for LSE* is defined as

$$\pi^{LSE}_{\{C;A\}}(P) = \begin{cases} P \setminus \{C_1, \ldots, C_k\}, & \text{if } A \text{ is an abducible;} \\ (P \setminus \{C\}) \cup \{C'_1, \ldots, C'_k\}, & \text{otherwise,} \end{cases}$$

where C_i and C'_j are the same as Definition 3.1, and $\pi^{LSE}(P)$ is defined as before.

The above definition presents that when partial deduction is performed on an abducible A, each clause C_i having A in its head can be eliminated in the residual program as far as the least specific explanations are concerned. Such a clause elimination technique is called *cut* in [Ino92]. Then the following results hold.

Lemma 3.8 Let P be an abductive Horn program. Then, $E \leq F$ and $E \not< F$ imply $P \models E \Leftrightarrow F$.

Proof: By definition, $E \not< F$ iff $E \not\leq F$ or $F \leq E$. Since $E \leq F$ by assumption, we show that $E \leq F$ and $F \leq E$ imply $P \models E \Leftrightarrow F$. By each definition, it holds that $P \cup F \models E$ and $P \cup E \models F$, which imply $P \models F \Rightarrow E$ and $P \models E \Rightarrow F$. Therefore, $P \models E \Leftrightarrow F$. □

Theorem 3.9 Let P be an abductive Horn program and O an observation. Then, for any least specific explanation E of O in P, there is an explanation F of O in $\pi^{LSE}(P)$ such that $P \models F \Leftrightarrow E$.

Proof: Let E be a least specific explanation of O in P. If partial deduction is performed on non-abducible atoms, $\pi^{LSE}(P)$ reduces to normal partial deduction and the result follows by Corollary 3.3. Otherwise, consider the case that partial deduction is performed on an abducible A in (5) and any clause (6) is eliminated. If $P \cup E \not\models \Gamma_i$, elimination of (6) does not affect the construction of the explanation E. Moreover, if $P \cup E$ is consistent, so is $\pi^{LSE}(P) \cup E$. Hence, E is also an explanation of O in $\pi^{LSE}(P)$, and the result holds. Else if $P \cup E \models \Gamma_i$, put $F = E \setminus \Gamma_i \cup \{A\}$. In this case, $P \cup E \models O$ implies $\pi^{LSE}(P) \cup F \models O$. Also, the consistency of $P \cup E$ implies the consistency of $\pi^{LSE}(P) \cup F$. Thus, F is an explanation of O in $\pi^{LSE}(P)$. Moreover, $P \cup E \models O$ implies $P \cup F \models O$, hence F is an explanation of O in P. By the construction of F, $P \cup E \models F$ holds, thereby $F \leq E$. Since E is least specific, $F \not< E$ holds. Therefore, $P \models F \Leftrightarrow E$ by Lemma 3.8. □

Example 3.6 Let $P = \{\ o \leftarrow a,\ \ a \leftarrow b\ \}$ where $\mathcal{A} = \{a, b\}$. Then o has the least specific explanation $\{a\}$, and $\pi^{LSE}_{\{o \leftarrow a; a\}}(P) = \{\ o \leftarrow a\ \}$ has the same least specific explanation $\{a\}$. On the other hand, let $P = \{\ o \leftarrow a,\ \ a \leftarrow\ \}$ where $\mathcal{A} = \{a\}$. Then, both $\{a\}$ and $\emptyset$ (i.e., *true*) are the least specific explanations of o. In this case, $\pi^{LSE}_{\{o \leftarrow a; a\}}(P) = \{\ o \leftarrow a\ \}$ has the least specific explanation $\{a\}$, where $P \models a \Leftrightarrow true$ holds.

Thus $\pi^{LSE}(P)$ approximates P in the sense that for any least specific explanation E in P, the residual program has a least specific explanation F which is logically equivalent to E in P.

For an abductive definite program, the following result also holds.

Theorem 3.10 Let P be an abductive definite program and O an observation. Then any explanation F of O in $\pi^{LSE}(P)$ is also an explanation of O in P.

Proof: If F is an explanation of O in $\pi^{LSE}(P)$, $\pi^{LSE}(P) \cup F \models O$ holds. In this case, adding eliminated clauses C_i to $\pi^{LSE}(P)$ does not affect the derivation of O from $\pi^{LSE}(P) \cup F$. Then, after such addition, $\pi^{LSE}(P)$ is just normal partial deduction on non-abducibles, hence $P \cup F \models O$ also holds. In an abductive definite program, the consistency of $\pi^{LSE}(P) \cup F$ implies the consistency of $P \cup F$, hence the result follows. □

Note that $\pi^{LSE}(P)$ may produce the least specific explanation which is not least specific in P. Moreover, in the presence of integrity constraints a produced explanation may not be an explanation in P.

Example 3.7 Let $P = \{\ o \leftarrow a,\ \ o \leftarrow b,\ \ a \leftarrow b\ \}$ where $\mathcal{A} = \{a, b\}$. Then, $\pi^{LSE}_{\{o \leftarrow a; a\}}(P) = \{\ o \leftarrow a,\ \ o \leftarrow b\ \}$ has the least specific explanations $\{a\}$ and $\{b\}$, while $\{b\}$ is not least specific in P. On the other hand, let $P = \{\ o \leftarrow a,\ \ b \leftarrow a,\ \ \leftarrow b,\ \ p \leftarrow b\ \}$ with $\mathcal{A} = \{a, b\}$. Then, $\pi^{LSE}_{\{p \leftarrow b; b\}}(P) = \{\ o \leftarrow a,\ \ \leftarrow b,\ \ p \leftarrow b\ \}$ has the explanation $\{a\}$ of o, while it is not an explanation in P because $P \cup \{a\}$ is inconsistent.

4 Partial Abduction

In this section, we provide a method of *partial abduction* by using abductive partial deduction presented in the preceding section. Partial abduction optimizes abductive reasoning by specializing a program with respect to a given observation, which is defined as follows.

Definition 4.1 Let P be an abductive logic program and O an observation. Let us define

$$\begin{aligned} PA^0_{O,P} &= P\,, \\ PA^{i+1}_{O,P} &= \bigcup_{O \leftarrow \Gamma \,\in\, PA^i_{O,P}} \Big(\bigcup_{B_j \in \Gamma} \pi^{A}_{\{O \leftarrow \Gamma; B_j\}}(PA^i_{O,P}) \Big)\,. \end{aligned}$$

Then we say that any $PA^i_{O,P}$ $(i > 0)$ is obtained by *partial abduction*.

In the above definition, $\bigcup_{B_j \in \Gamma} \pi^{A}_{\{O \leftarrow \Gamma; B_j\}}(PA^i_{O,P})$ means the result of abductive partial deduction, which is performed iteratively for any atom B_j in Γ. Then $PA^{i+1}_{O,P}$ is defined as abductive partial deduction performed for every clause from $PA^i_{O,P}$ containing O in the head.

Partial abduction is defined as an iterative application of abductive partial deduction, hence the next result follows from Corollary 3.5.

Theorem 4.1 Let P be an abductive logic program and O an observation. Then, O has an explanation E in P iff O has an explanation E in $PA^i_{O,P}$. $\Box$

Partial abduction optimizes abductive logic programs in the sense that $PA^i_{O,P}$ reduces inference steps from an observation to explanations. In particular, in a propositional program the iterative computation of $PA^i_{O,P}$ reaches the fixpoint $PA^n_{O,P} = PA^{n+1}_{O,P}$ (denoted by $PA_{O,P}$), and the following result holds.[5]

[5] A clause of the form $A \leftarrow B \wedge B \wedge \Gamma$ in a program is identified with $A \leftarrow B \wedge \Gamma$, so that infinite unfolding like $p \leftarrow q \wedge p,\ \ p \leftarrow q \wedge q \wedge p, \ldots$ never happens.

Theorem 4.2 Let P be an abductive Horn program and O an observation. Then, O has an explanation E in P iff $O \leftarrow E$ is in $PA_{O,P}$ and $PA_{O,P} \cup E$ is consistent.

Proof: Partial abduction reduces intermediate non-abducible atoms between the observation and an explanation, then $P \cup E \models O$ iff $O \leftarrow E$ is in $PA_{O,P}$. Also, when $P \cup E$ is consistent, $PA_{O,P} \cup E$ is consistent, and vice versa. Hence the result follows. □

Example 4.1 Let P be the program:

$$\begin{array}{rcl} \textit{wet-shoes} & \leftarrow & \textit{wet-grass}, \\ \textit{wet-grass} & \leftarrow & \textit{rained}, \\ \textit{wet-grass} & \leftarrow & \textit{sprinkler-on}, \end{array}$$

where $\mathcal{A} = \{\, \textit{wet-grass}, \textit{rained}, \textit{sprinkler-on} \,\}$. Then, given the observation $O = \textit{wet-shoes}$, $PA_{O,P}$ includes

$$\begin{array}{rcl} \textit{wet-shoes} & \leftarrow & \textit{wet-grass}, \\ \textit{wet-shoes} & \leftarrow & \textit{rained}, \\ \textit{wet-shoes} & \leftarrow & \textit{sprinkler-on}. \end{array}$$

In non-Horn abductive logic programs, an explanation E of O in P does not necessarily imply the existence of $O \leftarrow E$ in $PA_{O,P}$.

Example 4.2 Let $P = \{\ o \leftarrow \textit{not}\, p,\ \ p \leftarrow \textit{not}\, a\ \}$ where $\mathcal{A} = \{a\}$. Then, $\{a\}$ is the explanation of o, but $PA_{O,P} = P$ does not contain $o \leftarrow a$.

If we use π^{MSE} or π^{LSE} instead of π^{A} in Definition 4.1, we can define corresponding partial abduction in abductive Horn programs. Partial abduction is also realized by goal-oriented partial evaluation like [LS91] by translating abductive partial deduction into normal partial deduction by Corollary 3.4.

5 Discussion

Partial deduction is usually used for program optimization in logic programming. However, in Section 3 we argued that due to the nature of abductive reasoning, special attention should be paid for applying partial deduction to abductive logic programs. The point is that abduction is a form of causal reasoning between causes and effects, and normal partial deduction often loses such relationships in a program. This observation also suggests that we should be careful to use normal partial deduction where causality plays an important role in the meaning of a program. For instance, let us consider the program:

$$\begin{array}{rcl} \textit{pass\_exam} & \leftarrow & \textit{study\_hard}, \\ \textit{study\_hard} & \leftarrow & \textit{bad\_score}. \end{array}$$

If we perform partial deduction in the first clause of the program, it produces

$$\begin{aligned} pass\_exam &\leftarrow bad\_score, \\ study\_hard &\leftarrow bad\_score, \end{aligned}$$

in which the resultant clause is somewhat meaningless if we read it as a declarative sentence. Moreover, since the causal knowledge "if one studies hard, she/he passes the exam" is lost, introducing the fact *study_hard* never implies *pass_exam* in the residual program. In such situations, abductive partial deduction is also useful to preserve causalities by reserving appropriate intermediate knowledge.

Partial deduction is also discussed in the context of abductive reasoning in [Hop92]. In the paper, Hoppe argues structural similarities between partial deduction and Poole's *Theorist* procedure. According to his analysis, partial deduction is regarded as a special case of Theorist, and incremental nonmonotonic partial deduction is required to realize Theorist on a partial evaluator. However, he never considers the effect of partial deduction in abductive reasoning nor discusses the issue of explanation preservation in general.

To optimize abductive reasoning, Reiter and de Kleer [RK87] propose the *clause management system* (CMS), which generalizes de Kleer's ATMS and realizes efficient search for abductive reasoning in propositional theories. In the CMS, given a theory Σ, every clause C (*prime implicate*) satisfying (a) $\Sigma \models C$ and (b) $\Sigma \not\models C'$ for any proper subset C' of C, is stored. In contrast to ours, the CMS provides a global optimization technique which is different from our partial deduction technique. Moreover, due to the global nature, the CMS is more likely to produce an exponentially huge number of prime implicates when used as a compilation technique.

Poole [Poo93] introduces a mechanism to compute *partial explanations*, which is similar to our partial abduction. He computes partial explanations by an SLD-like top-down procedure together with the best-first strategy. However, the best explanations computed by his procedure are based on probabilities and are different from the most specific or least specific explanations in this paper. Furthermore, it is restricted to acyclic Horn programs having no clause with an abducible head.

6 Summary

This paper investigated the effect of partial deduction in abductive reasoning. We first showed that normal partial deduction does not preserve the meaning of abductive logic programs. Then, we introduced abductive partial deduction which preserves belief sets and (best) explanations in abductive logic programs. We also presented some variants of abductive partial deduction and introduced partial abduction to optimize abductive reasoning in logic programming.

The usefulness of abductive reasoning is now well-recognized in various AI problems, and abductive logic programming is a promising technique to realize it. Then optimization of abductive logic programs is an important research issue, and the techniques presented in this paper contribute as a step towards the goal. The results of this paper are also directly applicable to abductive logic programs containing classical negation. Future research includes the treatment of programs containing variables or disjunctions.

References

[BD94] Brass, S. and Dix, J., A Disjunctive Semantics Based on Unfolding and Bottom-up Evaluation, *Proc. IFIP Workshop on Disjunctive Logic Programming and Disjunctive Databases*, 1994.

[CP86] Cox, P. T. and Pietrzykowski, T., Causes for Events: Their Computation and Applications, *Proc. 8th Int. Conf. on Automated Deduction*, LNCS 230, 608-621, 1986.

[GL88] Gelfond, M. and Lifschitz, V., The Stable Model Semantics for Logic Programming, *Proc. ICLP'88*, 1070-1080.

[Hop92] Hoppe, T., On the Relationship between Partial Deduction and Abductive Reasoning, *Proc. ECAI-92*, 154-158.

[Ino92] Inoue, K. Linear Resolution for Consequence Finding, *Artificial Intelligence* 56, 301-353, 1992.

[IS93] Inoue, K. and Sakama, C., Transforming Abductive Logic Programs to Disjunctive Programs, *Proc. ICLP'93*, 335-353.

[KM90] Kakas, A. C. and Mancarella, P., Generalized Stable Models: A Semantics for Abduction, *Proc. ECAI-90*, 385-391.

[KKT92] Kakas, A. C., Kowalski, R. A. and Toni, F., Abductive Logic Programming, *J. Logic and Computation* 2(6), 719-770, 1992.

[Kom92] Komorowski, J., An Introduction to Partial Deduction, *Proc. 3rd Int. Workshop on Meta-programming in Logic*, LNCS 649, 49-69, 1992.

[LS91] Lloyd, J. W. and Shepherdson, J. C., Partial Evaluation in Logic Programming, *J. Logic Programming* 11, 217-242, 1991.

[Poo89] Poole, D., Explanation and Prediction: An Architecture for Default and Abductive Reasoning, *Computational Intelligence* 5, 97-110, 1989.

[Poo93] Poole, D., Logic Programming, Abduction and Probability, *New Generation Computing* 11, 377-400, 1993.

[Pop73] Pople, Jr, H. E., On the Mechanization of Abductive Logic, *Proc. IJCAI-73*, 147-152.

[RK87] Reiter, R. and de Kleer, J., Foundations of Assumption-based Truth Maintenance Systems: Preliminary Report, *Proc. AAAI-87*, 183-188.

[SS94] Sakama, C. and Seki, H., Partial Deduction of Disjunctive Logic Programs: A Declarative Approach, *Proc. 4th Int. Workshop on Logic Program Synthesis and Transformation*, LNCS, 1994.

[Seki90] Seki, H., A Comparative Study of the Well-Founded and the Stable Model Semantics: Transformation's Viewpoint, *Proc. NACLP Post-conf. Workshop on Logic Programming and Nonmonotonic Reasoning*, 1990.

[Seki91] Seki, H., Unfold/Fold Transformation of Stratified Programs, *Theoretical Computer Science* 86, 107-139, 1991.

[Seki93] Seki, H., Unfold/Fold Transformation of General Logic Programs for the Well-Founded Semantics, *J. Logic Programming* 16, 5-23, 1993.

[Sti89] Stickel, M. E., Rationale and Methods for Abductive Reasoning in Natural-Language Interpretation, *Proc. Int. Scientific Symposium on Natural Language and Logic*, LNAI 459, 233-252, 1989.

[TS84] Tamaki, H. and Sato, T., Unfold/Fold Transformation of Logic Programs, *Proc. ICLP'84*, 127-138.