Abducing Priorities to Derive Intended Conclusions

Katsumi Inoue
Dept. Electrical and Electronics Engineering
Kobe University
Rokkodai, Nada, Kobe 657-8501, Japan
inoue@eedept.kobe-u.ac.jp

Chiaki Sakama
Dept. Computer and Communication Sciences
Wakayama University
Sakaedani, Wakayama 640-8510, Japan
sakama@sys.wakayama-u.ac.jp

Abstract

We introduce a framework for finding preference information to derive desired conclusions in nonmonotonic reasoning. A new abductive framework called *preference abduction* enables us to infer an appropriate set of priorities to explain the given observation skeptically, thereby resolving the multiple extension problem in the answer set semantics for extended logic programs. Preference abduction is also combined with a usual form of abduction in abductive logic programming, and has applications such as specification of rule preference in legal reasoning and preference view update. The issue of learning abducibles and priorities is also discussed, in which abduction to a particular cause is equivalent to abduction to preference.

1 Introduction

In commonsense reasoning, it is important to represent and reason about preference in order to reduce nondeterminism due to incomplete knowledge. To represent such knowledge about preference, it is required that priorities among commonsense knowledge are to be found out. For example, to get the desired result of the *Yale shooting problem* [Hanks and McDermott, 1987], an adequate priority should be expressed according to our commonsense. The essence of this problem can be represented by the following *extended logic program*, P, where $ab1$ and $ab2$ are abnormality predicates, $loaded_i$ and $alive_j$ denote that the gun is loaded at the time T_i and the turkey is alive at the time T_j, respectively. We also assume that some unknown action *wait* is done at T_0, and that the *shoot* action, which causes the turkey dead whenever the gun is loaded, is done at T_1.

$$
\begin{aligned}
P:\quad & loaded_1 \leftarrow loaded_0,\ not\, ab1, && (1)\\
& alive_2 \leftarrow alive_1,\ not\, ab2, && (2)\\
& \neg alive_2 \leftarrow loaded_1, &&\\
& \neg loaded_1 \leftarrow alive_2, &&\\
& ab1 \leftarrow loaded_0,\ \neg loaded_1, &&\\
& ab2 \leftarrow loaded_1, &&\\
& loaded_0 \leftarrow\ , &&\\
& alive_1 \leftarrow\ . &&
\end{aligned}
$$

Here, *not* denotes negation as failure, and (1) and (2) represent the inertia of actions. Without any priority information, we do not know which default of (1) or (2) should take precedence. Then, to the contrary of our intention that $\neg alive_2$ should be inferred, neither $alive_2$ nor $\neg alive_2$ is decided from the above program. In fact, there are two answer sets of P, one including $loaded_1, ab2, \neg alive_2$ (intended), and the other containing $alive_2, \neg loaded_1, ab1$.

Historically, the Yale shooting problem revealed the so called *multiple extension problem*. In this case, we should decide which $ab1$ or $ab2$ must have a higher priority for minimization. Using *prioritized circumscription* [Lifschitz, 1985], for example, the criterion can be manually given for the Yale shooting problem that $ab1$ should be minimized with a higher priority than $ab2$ in order to derive $\neg alive_2$. Other than circumscription, recent development in the field of logic programming and nonmonotonic reasoning has provided a number of mechanisms for freely specifying preference on multiple extensions in default reasoning. Such prioritized reasoning systems include *prioritized default logics* and *prioritized logic programs* [Brewka, 1994; Baader and Hollunder, 1995; Dimopoulos and Kakas, 1995; Sakama and Inoue, 1996; Brewka and Eiter, 1998]. For the Yale shooting problem, we would like to prefer the default (1) to (2), and then a higher priority is given to (1) in these frameworks.

Although the Yale shooting problem is so simple that we can find the proper priority manually, it becomes more complicated and difficult to find priorities among many complex default knowledge in the real world's commonsense reasoning. Hence, a framework and a method for automatic finding of such priority information are highly required.

In this paper, we provide a framework for finding priorities as a part of a *prioritized logic program* [Sakama and Inoue, 1996] in order to derive an intended conclusion as a theorem of the logic program. To this end, we introduce the notion of *preference abduction*, which infers a sufficient priority relation to make the intended conclusion hold. This inference is in fact a form of abduction, i.e., abduction of meta-knowledge which is preference in this case. We further provide an integrated framework of abduction, in which both literals and priorities can be abduced. Using such an abductive framework, we can infer *skeptical explanations* of an observation even when only credulous explanations are obtained due to non-determinism of a given abductive program.

There are many applications of nonmonotonic reasoning that require to find out priorities among conflicting rules. For example, in the legal domain, priorities among the conflicting laws are often required for disputants to derive their desired conclusion, which give them the advantage in the argumentation of a court. The proposal in this paper enables us to derive a desired conclusion by abducing appropriate priorities in such cases. An interesting application is *preference view*, which transfers a given priority relation among observations into a priority relation among base abducible literals.

This paper is organized as follows. Section 2 introduces the theoretical background in this paper. Section 3 provides the framework for preference abduction. Section 4 goes on elaborating on preference abduction. Section 5 discusses related work, and Section 6 is the conclusion. Due to the lack of space, we will omit proofs of propositions and theorems in this paper.

2 Background

2.1 Extended Logic Programs

An *extended logic program* (ELP) [Gelfond and Lifschitz, 1990] is a set of rules of the form

$$L_0 \leftarrow L_1,\, \ldots,\, L_m,\, not\, L_{m+1},\, \ldots,\, not\, L_n \qquad (3)$$

where L_i's ($0 \le i \le n$; $n \ge m$) are literals. Here, the left-hand side L_0 is called the *head* of the rule (3), and the right-hand side is called the *body* of the rule. The head is possibly empty. A rule with an empty body is called a *fact*, and each fact $L \leftarrow$ is identified with the literal L. Two kinds of negation appear in a program: *not* is the *negation as failure* (NAF) operator, and $\neg$ is *classical negation*. Intuitively, the rule (3) can be read as: if $L_1, \ldots, L_m$ are believed and $L_{m+1}, \ldots, L_n$ are not believed then L_0 is believed.

The semantics of an ELP P is given by the *answer set semantics* [Gelfond and Lifschitz, 1990], which is defined by the following two steps. Let $\mathcal{L}_P$ be the set of all ground literals in the language of P, and let $S \subseteq \mathcal{L}_P$. First, let P be a *not*-free ELP (i.e., for each rule $m = n$). Then, S is an *answer set* of P if S is a minimal set satisfying the conditions:

1. For each ground rule $L_0 \leftarrow L_1, \ldots, L_m$ from P, $\{L_1, \ldots, L_m\} \subseteq S$ implies $L_0 \in S$.
2. If S contains a pair of complementary literals L and $\neg L$, then $S = \mathcal{L}_P$.

Second, let P be any ELP and $S \subseteq \mathcal{L}_P$. Then, define a *not*-free ELP P^S as follows: a rule

$$L_0 \leftarrow L_1,\, \ldots,\, L_m$$

is in P^S iff there is a ground rule of the form (3) from P such that $\{ L_{m+1}, \ldots, L_n \} \cap S = \emptyset$. For P^S, its answer sets have already been defined. Then, S is an *answer set* of P if S is an answer set of P^S.

The class of ELPs is a subset of Reiter's *default logic* [Gelfond and Lifschitz, 1990]. An answer set of an ELP P is *consistent* if it is not $\mathcal{L}_P$. P is *consistent* if it has a consistent answer set. An ELP P *(skeptically) entails* a literal L, written as $P \models L$, if L is included in every answer set of P. On the other hand, P *credulously infers* L if L is included in an answer set of P.

2.2 Abductive Logic Programs

An *abductive (extended) logic program* (ALP) is a pair $\langle P, \Gamma \rangle$, where P is an ELP and Γ is a set of literals from the language of P. The set Γ is identified with the set of ground instances from Γ, and each literal in Γ is called an *abducible*. The model-theoretic semantics for ALPs is given in [Inoue and Sakama, 1996]. A set $S \subseteq \mathcal{L}_P$ is called a *belief set* of $\langle P, \Gamma \rangle$ if S is a consistent answer set of $P \cup A$ for some $A \subseteq \Gamma$. Note that belief sets reduce to consistent answer sets when $\Gamma = \emptyset$.

Let O be a ground literal called an *observation*. $A \subseteq \Gamma$ is an *skeptical explanation* of O (wrt $\langle P, \Gamma \rangle$) if $P \cup A \models O$ and $P \cup A$ is consistent. On the other hand, $A \subseteq \Gamma$ is a *credulous explanation* of O (wrt $\langle P, \Gamma \rangle$) if there is a belief set S of $\langle P, \Gamma \rangle$ such that $O \in S$ and S is a consistent answer set of $P \cup A$. A skeptical/credulous explanation A of O is *minimal* if no $A' \subset A$ is a skeptical/credulous explanation of O.

In an ALP $\langle P, \Gamma \rangle$, each abducible in Γ is a literal. Often however, we would like to introduce rules of the form (3) with $n \ge m \ge 1$ in Γ. Such a rule, called an *abducible rule*, intuitively means that if the rule is abduced then it is used for inference together with background knowledge P. This extended abductive framework is introduced in [Inoue, 1994] as a *knowledge system*. Any knowledge system $\langle P, \Gamma \rangle$, where both P and Γ are ELPs, can be translated into an ALP $\langle P', \Gamma' \rangle$ where Γ' is a set of literals [Inoue, 1994]: For each abducible rule R in Γ, a new *naming* atom δ_R is associated with R, and let

$$\begin{aligned} P' &= P \cup \{ (H \leftarrow B, \delta_R) \mid R = (H \leftarrow B) \in \Gamma \}, \\ \Gamma' &= \{ \delta_R \mid R \in \Gamma \}. \end{aligned}$$

2.3 Prioritized Logic Programs

A reflexive and transitive relation $\preceq$ is defined on $\mathcal{L}_P$. Each $e_1 \preceq e_2$ is called a *priority*, and we say e_2 *has a priority over* e_1. We write $e_1 \prec e_2$ if $e_1 \preceq e_2$ and $e_2 \not\preceq e_1$. When $\mathbf{x}$ and $\mathbf{y}$ are tuples of variables, $e_1(\mathbf{x}) \preceq e_2(\mathbf{y})$ stands for any priority $e_1(\mathbf{s}) \preceq e_2(\mathbf{t})$ for any instances $\mathbf{s}$ of $\mathbf{x}$ and $\mathbf{t}$ of $\mathbf{y}$.

A *prioritized (extended) logic program* (PLP) by [Sakama and Inoue, 1996] is given as a pair (P, Φ), where P is an ELP and Φ is a set of priorities on $\mathcal{L}_P$.[1] The

[1] In [Sakama and Inoue, 1996], a PLP (P, Φ) is defined with a *general extended disjunctive program* P, which allows NAF and disjunctions in heads of rules, and Φ may contain NAF formulas. Here, we consider a subset of their PLPs.

declarative semantics of PLP is defined using the answer sets. Given a PLP (P, Φ), suppose that S_1 and S_2 are two distinct answer sets of P. Then, S_2 is *preferable to* S_1, written as $S_1 \preceq S_2$, if for some element $e_2 \in S_2 \setminus S_1$, (i) there is an element $e_1 \in S_1 \setminus S_2$ such that $e_1 \preceq e_2$, and (ii) there is no element $e_3 \in S_1 \setminus S_2$ such that $e_2 \prec e_3$. Here, the relation $\preceq$ on answer sets is also defined as reflexive and transitive. Note that the condition (ii) is automatically satisfied if there is no priority chained on more than two different elements (i.e., $e_1 \preceq e_2 \preceq e_3$ implies either $e_1 = e_2$ or $e_2 = e_3$). An answer set S of P is called a *preferred answer set* (or *p-answer set*, for short) of P (wrt Φ) if $S \preceq S'$ implies $S' \preceq S$ for any answer set S' of P.

By definition, (P, Φ) has a p-answer set if P has a finite number of answer sets. In particular, the p-answer sets of (P, Φ) coincide with the answer sets of P when $\Phi = \emptyset$. It is also clear that if a program P has the unique answer set, it also becomes the unique p-answer set of (P, Φ) for any Φ. We say (P, Φ) *entails* a literal L, written as $P \models_\Phi L$, if L is included in every preferred answer set of P.

Using PLPs, we can represent preference knowledge naturally, and it is helpful to reduce non-determinism in logic programming. Moreover, various forms of commonsense reasoning such as (prioritized) minimal abduction, (prioritized) default reasoning, and prioritized circumscription can be realized in terms of PLP. In particular, the mapping from prioritized circumscription of any clause set to a PLP is given in [Sakama and Inoue, 1996], which much extends the previous translation into a stratified logic program by Gelfond and Lifschitz [1988].

3 Preference Abduction

In this section, we introduce *preference abduction*, which is the process of abducing priorities to explain given observations.

3.1 Basic Framework

Given an ELP P and a literal O, we first consider the case that O is credulously inferred by P but is not skeptically entailed by P. In this case, there exists a multiple extension problem, that is, both answer sets containing O and answer sets not containing O coexist. Let AS^+ be the set of answer sets containing O, and AS^- the set of answer sets not containing O. A direct way to prefer answer sets containing O is to construct priorities between answer sets in AS^+ and AS^-, so that some subset of AS^+ are made the set of preferred answer sets of P. However, there are many ways to associate priorities between AS^+ and AS^-. Hence, we assume the existence of some set Ψ of pre-specified candidate hypotheses for priorities in the following abductive framework.

Definition 3.1 Let P be a consistent ELP, and O a literal. Suppose that Ψ is a set of candidate priorities on $\mathcal{L}_P$. A set ψ of priorities is a *(skeptical) explanation* of O (wrt $\langle P, \Psi \rangle$) if

1. $\psi \subseteq \Psi$, and
2. $P \models_\psi O$.

Also, ψ is a *minimal explanation* of O if no $\psi' \subset \psi$ is an explanation of O.

Given a pair $\langle P, \Psi \rangle$ for preference abduction, let S_1 and S_2 be two distinct answer sets of P. Then, in order to find priorities ψ from Ψ such that $S_1 \preceq S_2$ holds, one should select a literal $e_1 \in S_1 \setminus S_2$ and another literal $e_2 \in S_2 \setminus S_1$ such that (i) $e_1 \preceq e_2$ and (ii) for any literal $e_3 \in S_1 \setminus S_2$, $e_2 \not\prec e_3$, i.e., $e_2 \preceq e_3$ implies $e_3 \preceq e_2$.

Example 3.1 Suppose that the ELP P is given as

$$p \leftarrow \mathit{not}\, q, \quad q \leftarrow \mathit{not}\, p, \quad o \leftarrow p, \quad \neg o \leftarrow q,$$

and $\Psi = \{p \preceq q,\ q \preceq p\}$. There are two answer sets of P: $S1 = \{p, o\}$ and $S2 = \{q, \neg o\}$. Suppose we want to find an explanation of o. Abducing the priority $q \preceq p$, we get the relation $S2 \preceq S1$, hence $P \models_{\{q \preceq p\}} o$.

Example 3.2 (Yale shooting) Consider the ELP P introduced in Section 1. The candidate hypotheses for this problem can be supplied as $\Psi = \{ab1 \preceq ab2,\ ab2 \preceq ab1\}$. Then, $\{ab1 \preceq ab2\}$ is the explanation of $\neg alive_2$. This abduced priority corresponds to our commonsense that the abnormality wrt the *shoot* action should be stronger than that wrt the *wait* action.

3.2 Combining with Credulous Abduction

When an observation O cannot be credulously inferred by P, the basic framework in Section 3.1 cannot give a sufficient explanation of O. In such a case, we can combine preference abduction with ordinary abduction in Section 2.1 so that O gets a skeptical explanation. An extended abductive framework is given as follows.

Definition 3.2 A *preference abduction framework* is a triple $\langle P, \Gamma, \Psi \rangle$, where P is an ELP, $\Gamma \subseteq \mathcal{L}_P$ is a set of abducibles, and Ψ is a set of candidate priorities on $\mathcal{L}_P$. A pair (A, ψ) is a *(skeptical) explanation* of a literal O (wrt $\langle P, \Gamma, \Psi \rangle$) if

1. $A \subseteq \Gamma$,
2. $\psi \subseteq \Psi$,
3. $P \cup A$ is consistent, and
4. $P \cup A \models_\psi O$.

Also, (A, ψ) is a *minimal explanation* of O if for any explanation (A', ψ') of O, $A' \subseteq A$ and $\psi' \subseteq \psi$ imply $A' = A$ and $\psi' = \psi$.

Note that the basic framework of preference abduction in Section 3.1 is a special case of Definition 3.2, where $\Gamma = \emptyset$. Moreover, the traditional ALP framework in Section 2.2 is also a special case, where $\Psi = \emptyset$. We can also consider an abductive framework $\langle P, \Gamma, \Psi \rangle$ in which Γ is a set of abducible rules and Ψ includes priorities on such abducible rules. In that case, a naming technique similar to the one in Section 2.2 can be applied to abducible rules, and then priorities among rules are translated into priorities among rule names, thereby reducing such an abductive framework to that in Definition 3.2.

Example 3.3 Let us consider the abductive framework $\langle P, \Gamma, \Psi \rangle$, where

$$\begin{aligned} P: \quad & p \leftarrow a,\, not\, q, \\ & q \leftarrow not\, p, \\ & \neg q \leftarrow b, \\ \Gamma: \quad & a, \;\; b, \\ \Psi: \quad & p \preceq q, \;\; q \preceq p. \end{aligned}$$

There are four belief sets of $\langle P, \Gamma \rangle$: $S1 = \{q\}$, $S2 = \{a, p\}$, $S3 = \{a, q\}$, $S4 = \{a, b, p, \neg q\}$. Then, both $E1 = (\{a\}, \{q \preceq p\})$ and $E2 = (\{a, b\}, \emptyset)$ are the minimal explanations of p. In fact, $P \cup \{a\} \models_{\{q \preceq p\}} p$ and $P \cup \{a, b\} \models p$. For $E1$ the p-answer set of $P \cup \{a\}$ is $S2$, while $S4$ is the unique answer set of $P \cup \{a, b\}$ for $E2$.

Note in the above example that explanations are obtained either from credulous explanations with abduced priorities or from skeptical explanations wrt the given ELP. Hence, a naive procedure to compute preference abduction is as follows.

Procedure 3.1 $\underline{PrefAbd(P, \Gamma, \Psi, O, E)}$
Input: a preference abduction framework $\langle P, \Gamma, \Psi \rangle$,
a literal O (observation).
Output: a skeptical explanation E of O wrt $\langle P, \Gamma, \Psi \rangle$.

1. Compute a credulous explanation A of O wrt $\langle P, \Gamma \rangle$;
2. If A is a skeptical explanation of O wrt $\langle P, \Gamma \rangle$, then return $E = (A, \emptyset)$;
3. Otherwise, compute the answer sets of $P \cup A$;
   AS^+ := the set of answer sets containing O;
   AS^- := the set of answer sets not containing O;
4. Find priorities $\psi \subseteq \Psi$ and $P\text{-}AS \subseteq AS^+$ such that $T \preceq S$ for any $S \in P\text{-}AS$ and any $T \in AS^-$;
   Then, ψ is an explanation of O wrt $\langle P \cup A, \Psi \rangle$;
   Return $E = (A, \psi)$.

In Procedure 3.1, computing credulous explanations of O at Step 1 can be realized by existing abductive procedures such as [Kakas and Mancarella, 1990; Inoue and Sakama, 1996]. At Step 2, each credulous explanation A is checked to see whether it is skeptical or not. This test can easily be realized by checking the consistency of $P \cup A \cup \{\leftarrow O\}$. At Step 3, the answer sets of $P \cup A$ or belief sets of $\langle P, \Gamma \rangle$ can be computed by some bottom-up procedures, e.g., [Inoue and Sakama, 1996]. At Step 4, it can be shown that: if ψ is an explanation of O wrt $\langle P \cup A, \Psi \rangle$, then (A, ψ) is an explanation of O wrt $\langle P, \Gamma, \Psi \rangle$. Hence, the next theorem holds.

Theorem 3.1 *Procedure* $PrefAbd(P, \Gamma, \Psi, O, E)$ *is sound. That is, if it terminates, its output E is a skeptical explanation of O wrt $\langle P, \Gamma, \Psi \rangle$.*

The completeness of Procedure 3.1 holds for a ground ELP P and finite Γ and Ψ if we assume (i) the existence of an abductive procedure that is complete for computing credulous explanations at Step 1, and (ii) the exhaustive search for finding ψ at Step 4. However, since many existing abductive procedures are designed to compute credulous explanations, it is more difficult to compute skeptical explanations directly. In fact, the skeptical explanation $\{a, b\}$ of p wrt $\langle P, \Gamma \rangle$ in Example 3.3 cannot be obtained by top-down abductive procedures in general. In this sense, to compute skeptical explanations of an observation, it is easier to compute credulous explanations first, then priorities are added to make explanations skeptical as in Procedure 3.1.

4 Finding Further Preference

In Section 3.2, we considered an abductive framework $\langle P, \Gamma, \Psi \rangle$ in which Γ and Ψ are pre-specified. However, such candidate hypotheses are often insufficiently given so that we cannot explain an observation skeptically.

Example 4.1 (legal reasoning [Kowalski and Toni, 1996]) Suppose that the ELP P is given as:

$$inherits(x, y) \leftarrow beneficiary(x, y),\, not\, \neg inherits(x, y), \quad (4)$$

$$\neg inherits(x, y) \leftarrow murder(x, y),\, not\, inherits(x, y). \quad (5)$$

$$beneficiary(a, b) \leftarrow, \quad murder(c, d) \leftarrow,$$
$$beneficiary(j, h) \leftarrow, \quad murder(j, h) \leftarrow,$$

Rule (4) indicates that a person inherits an estate if he/she is the beneficiary of a valid will and it cannot be shown that the person does not inherit it. Rule (5) says that a person usually does not inherit an estate if he/she murders the owner of the estate. The program P has two answer sets, one containing $inherits(j, h)$ and the other $\neg inherits(j, h)$. Given the observation $\neg inherits(j, h)$, we cannot get any explanation wrt $\langle P, \emptyset, \emptyset \rangle$.

In this section, we consider a method to generate new abducibles for obtaining further preference.

4.1 Generating New Abducibles

A method to discover new abducibles is considered in [Inoue and Haneda, 1999], where abducibles are newly invented in learning ALPs. Here, we modify their method by associating priorities with new abducibles.

Firstly, notice that rules (4) and (5) in Example 4.1 are the source of non-determinism in the program. Then, these *non-deterministic rules* are converted into abducible rules. Without loss of generality, we assume that such rules in an ELP P are of the form:[2]

$$\begin{array}{lcl} \alpha & \leftarrow & B_1,\, not\, \beta\,, \\ \beta & \leftarrow & B_2,\, not\, \alpha\,, \end{array} \quad (6)$$

where α and β are literals and both B_1 and B_2 are conjunctions of literals and NAF formulas. Here, we assume that neither α nor β appears in the head or the body of any rule other than (6) in P. Now, let N_1 be a pair of rules of the form (6), and $P_1 = P \setminus N_1$. Also, let E_1 be the set of ground instances of α and β that are entailed by P. Then, Γ_1 is obtained by converting N_1 into abducible rules:

$$\begin{array}{lcl} \alpha & \leftarrow & B_1\,, \\ \beta & \leftarrow & B_2\,. \end{array} \quad (7)$$

[2] Using the unfolding operation, we can get pairs of rules of the form (6) which cause non-determinism in P.

Next, using the translation in Section 2.2, each abducible rule R of the form (7) in Γ_1 can be named with a new atom δ_R, and put $P_2 = P_1 \cup \{ (H \leftarrow B, \delta_R) \mid R = (H \leftarrow B) \in \Gamma_1 \}$ and $\Gamma_2 = \{\delta_R \mid R \in \Gamma_1\}$. Then, compute the set E_2 of instances of new abducibles in Γ_2 such that $P_2 \cup E_2 \models e$ for every $e \in E_1$. This identification of E_2 from E_1 is easy, and it is used to assure that the literals in E_1 can also be entailed by the new program. We now obtain the ALP $\langle P', \Gamma' \rangle = \langle P_2 \cup E_2, \Gamma_2 \rangle$.

Proposition 4.1 *Let P be an ELP, and $\langle P', \Gamma' \rangle$ the ALP constructed as above. Then, for every consistent answer set S of P, there is a belief set S' of $\langle P', \Gamma' \rangle$ such that $S = S' \setminus \Gamma'$.*[3]

Example 4.2 (cont. from Example 4.1)
Let N_1 be the last two rules (4,5) in P, $P_1 = P \setminus N_1$, and $E_1 = \{ \mathit{inherits}(a, b), \neg\mathit{inherits}(c, d) \}$. Then, non-deterministic rules N_1 are converted into abducible rules:

$$\begin{aligned}\Gamma_1 : \quad & \mathit{inherits}(x, y) \leftarrow \mathit{beneficiary}(x, y), \\ & \neg\mathit{inherits}(x, y) \leftarrow \mathit{murder}(x, y).\end{aligned}$$

By naming these abducible rules with $\delta_{\mathit{inherit}}(x, y)$ and $\delta_{\neg\mathit{inherit}}(x, y)$, the ALP $\langle P', \Gamma' \rangle$ is obtained as:

$$\begin{aligned}P' = \{ \quad & \mathit{inherits}(x, y) \leftarrow \mathit{beneficiary}(x, y),\ \delta_{\mathit{inherit}}(x, y), \\ & \neg\mathit{inherits}(x, y) \leftarrow \mathit{murder}(x, y),\ \delta_{\neg\mathit{inherit}}(x, y), \\ & \delta_{\mathit{inherit}}(a, b) \leftarrow, \quad \delta_{\neg\mathit{inherit}}(c, d) \leftarrow \quad \} \cup P_1, \\ \Gamma' = \{ \quad & \delta_{\mathit{inherit}}(x, y), \ \ \delta_{\neg\mathit{inherit}}(x, y) \ \ \}.\end{aligned}$$

In the ALP, $\mathit{inherits}(j, h)$ is concluded by abducing $\delta_{\mathit{inherit}}(j, h)$, while $\neg\mathit{inherits}(j, h)$ is skeptically explained by $\delta_{\neg\mathit{inherit}}(j, h)$.

4.2 From Abducibles to Priorities

So far, we have not yet introduced new priorities into the process of finding new abducibles. That is, appropriate literals are just abduced to explain the observation. *Such a right selection of hypotheses in abduction can be considered as our preference of some particular causes over others.* With this regard, we can acquire new preference information from abductive programs as follows.

Suppose that δ_α and δ_β are the naming atoms for a pair of abducible rules of the form (7). Then, the ALP $\langle P', \Gamma' \rangle$ in Section 4.1 can be further translated into the semantically equivalent ELP P^* by replacing each pair of abducibles δ_α and δ_β in Γ' with the pair of rules:

$$\begin{array}{lcl}\delta_\alpha & \leftarrow & \mathit{not}\, \delta_\beta\,, \\ \delta_\beta & \leftarrow & \mathit{not}\, \delta_\alpha\,.\end{array} \tag{8}$$

This time, we have the following relationship between the answer sets of the original ELP P and those of P^*.

Proposition 4.2 *Let P be a consistent ELP, and P^* the ELP constructed as above. Then, for every answer set S of P, there is an answer set S^* of P^* such that $S = S^* \setminus \Gamma'$, where Γ' is the same as in Proposition 4.1. Conversely, for every answer set S^* of P^*, there is an answer set S of P such that $S = S^* \setminus \Gamma'$.*

Using the above new P^*, it is easy to associate priorities on the newly introduced abducibles. Once the ALP $\langle P', \Gamma' \rangle$ is constructed, we can just consider the abductive framework $\langle P^*, \Psi^* \rangle$, where Ψ^* is the candidate priorities on Γ'. In this way, abduction to particular causes and preference abduction are made transferable into each other.

Theorem 4.3 *Let $\langle P', \Gamma' \rangle$ and $\langle P^*, \Psi^* \rangle$ be the same as in the above discussion, and O be an observation. Then, there is a skeptical explanation of O wrt $\langle P', \Gamma' \rangle$ iff there is a skeptical explanation of O wrt $\langle P^*, \Psi^* \rangle$.*

Example 4.3 (cont. from Example 4.2)
The ALP $\langle P', \Gamma' \rangle$ constructed in Example 4.2 can be now translated into the ELP $P^* = P' \cup P_{\Gamma'}$ where $P_{\Gamma'}$ is given as:

$$\begin{aligned}\delta_{\mathit{inherit}}(x, y) &\leftarrow \mathit{not}\, \delta_{\neg\mathit{inherit}}(x, y), \\ \delta_{\neg\mathit{inherit}}(x, y) &\leftarrow \mathit{not}\, \delta_{\mathit{inherit}}(x, y).\end{aligned}$$

Then, P^* has two answer sets, one containing $\delta_{\mathit{inherit}}(j, h)$ and $\mathit{inherits}(j, h)$, and the other containing $\delta_{\neg\mathit{inherit}}(j, h)$ and $\neg\mathit{inherits}(j, h)$.

Now, let us consider the preference abduction $\langle P^*, \Psi^* \rangle$, where Ψ^* contains the candidate priorities:

$$\begin{aligned}\delta_{\mathit{inherit}}(x, y) &\preceq \delta_{\neg\mathit{inherit}}(x, y), \\ \delta_{\neg\mathit{inherit}}(x, y) &\preceq \delta_{\mathit{inherit}}(x, y).\end{aligned}$$

Given the observation $\neg\mathit{inherits}(j, h)$, we have the explanation $\{ \delta_{\mathit{inherit}}(j, h) \preceq \delta_{\neg\mathit{inherit}}(j, h) \}$.

4.3 Preference View Updates

The advantage of the above translation into preference abduction is that priorities do not have to be given on the target observations but are given on the source hypotheses. In this sense, we call such inference to preference *preference view updates*, which are analogous to the notion of *view updates* in deductive databases. In preference view updates, the priority request on given observations $O_1 \preceq O_2$ is translated into priorities on their causes $\psi \subseteq \Psi^*$. A typical application of this kind is abduction to *rule preference* in the legal domain.

Example 4.4 (cont. from Example 4.3)
Suppose that one rather prefers the conclusion $\mathit{inherits}(j, h)$ to the opposite $\neg\mathit{inherits}(j, h)$. This preference view:

$$\neg\mathit{inherits}(j, h) \preceq \mathit{inherits}(j, h)$$

can be translated into the priority between the hypotheses:

$$\delta_{\neg\mathit{inherit}}(j, h) \preceq \delta_{\mathit{inherit}}(j, h).$$

This last relation indicates that she/he should prefer the rule (4) to the other rule (5) in her/his argument.

It should be noted that, as view updates in databases can be characterized through abduction [Kakas and Mancarella, 1990], our formulation of preference view updates are also based on preference abduction.

[3] The converse of Proposition 4.1 does not hold. In Example 4.2, the ALP $\langle P', \Gamma' \rangle$ has a belief set containing neither $\mathit{inherits}(j, h)$ nor $\neg\mathit{inherits}(j, h)$, which is not an answer set of P.

5 Related Work

As far as the authors know, there are very few work on abducing priorities to derive desired conclusion. Zhang and Foo [1998] associate priorities to resolve conflicts between rules in updating ELPs. Their framework can be regarded as a kind of preference abduction to be applied to theory updates. In general, preference information is helpful to resolve contradiction in a program. Priorities on defaults specify the guideline that some defaults are to be kept but some are discarded in restoring the consistency. Wakaki *et al.* [1998] present a method of finding priorities as a part of the circumscription policy to be used in prioritized circumscription. In their method, priorities are selected from the set of all possible orderings on minimized predicates. On the other hand, our method can discover new priorities on literals for PLPs as shown in Section 4.

There are a lot of recent work on introducing priorities into abductive and nonmonotonic reasoning. Eiter and Gottlob [1995] discuss the computational complexity of a form of *prioritized abduction*, whose prioritization is similar to that of prioritized circumscription. Sakama and Inoue [1996] propose a different kind of prioritied abduction in the context PLPs, in which priorities are used to select desired abducibles from multiple explanations. Priorities have also been used to represent preference between conflicting default rules in PLPs and prioritied default logics [Brewka, 1994; Baader and Hollunder, 1995; Dimopoulos and Kakas, 1995; Sakama and Inoue, 1996; Brewka and Eiter, 1998]. None of these work, however, discusses how to find an appropriate set of priorities to derive desired conclusions.

Brewka [1994] argues the importance of the ability of using defaults that reason about preference between other defaults. Our preference abduction would also be extended by introducing such dynamic preference into not only deduction but abduction on PLPs, but the issue is not addressed in this paper.

6 Conclusion

This paper introduced a novel framework for finding preference to derive intended conclusions in nonmonotonic reasoning. Preference abduction is not only an extension of traditional ALPs, but much extends the reasoning ability of PLPs. Applications of preference abduction include the resolution of the multiple extension problem, skeptical abduction, preference view updates, and abduction to rule preference in legal reasoning.

In this paper, we also presented an interesting fact that abduction to a particular cause and abduction to preference are sometimes transferable to each other. This line of research would extend the applicability of existing frameworks for ALPs as computational tools for prioritied default reasoning. The design of a more sophisticated algorithm to compute preference abduction also remains to be explored.

References

[Baader and Hollunder, 1995] F. Baader and B. Hollunder. Priorities on defaults with prerequisites, and their application in treating specificity in terminological default logic. *J. Automated Reasoning*, 15(1):41–68, 1995.

[Brewka, 1994] G. Brewka. Reasoning about priorities in default logic. In: *Proc. AAAI-94*, pages 940–945.

[Brewka and Eiter, 1998] G. Brewka and T. Eiter. Preferred answer sets for extended logic programs. In: *Proc. KR '98*, pages 86–97, 1998.

[Dimopoulos and Kakas, 1995] Y. Dimopoulos and A. C. Kakas. Logic programming without negation as failure. In: *Proc. Int'l Logic Programming Symp. '95*, pages 369–383, 1995.

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

[Gelfond and Lifschitz, 1988] M. Gelfond and V. Lifschitz. Compiling circumscriptive theories into logic programs. In: *Proc. AAAI-88*, pages 455–459, 1988.

[Gelfond and Lifschitz, 1990] M. Gelfond and V. Lifschitz. Logic programs with classical negation. In: *Proc. 7th Int'l Conf. on Logic Programming*, pages 579–597, 1990.

[Hanks and McDermott, 1987] S. Hanks and D. McDermott. Nonmonotonic logic and temporal projection. *Artificial Intelligence*, 33:379–412, 1987.

[Inoue, 1994] K. Inoue. Hypothetical reasoning in logic programs. *J. Logic Programming*, 18(3):191-227, 1994.

[Inoue and Haneda, 1999] K. Inoue and H. Haneda. Learning abductive and nonmonotonic logic programs. In: P. Flach and A. Kakas, editors. *Abductive and Inductive Reasoning—Essays on their Relation and Integration*, pages 241–262, Kluwer Academic, 1999.

[Inoue and Sakama, 1996] K. Inoue and C. Sakama. A fixpoint characterization of abductive logic programs. *J. Logic Programming*, 27(2):107–136, 1996.

[Kakas and Mancarella, 1990] A. C. Kakas and P. Mancarella. Database updates through abduction. In: *Proc. 16th Int'l Conf. Very Large Databases*, pages 650–661, 1990.

[Kowalski and Toni, 1996] R. A. Kowalski and F. Toni. Abstract argumentation. *Artificial Intelligence and Law*, 4:275–296, 1996.

[Lifschitz, 1985] V. Lifschitz. Computing circumscription. In: *Proc. IJCAI-85*, pages 121–127, 1985.

[Sakama and Inoue, 1996] C. Sakama and K. Inoue. Representing priorities in logic programs. In: *Proc. 1996 Joint Int'l Conf. and Symp. on Logic Programming*, pages 82–96, 1996.

[Wakaki *et al.*, 1998] T. Wakaki, K. Satoh, K. Nitta and S. Sakurai. Finding priorities of circumscription policy as a skeptical explanation in abduction. *J. IEICE Trans. Information and Systems*, E-81D(10), 1998.

[Zhang and Foo, 1998] Y. Zhang and N. Y. Foo. Updating logic programs. In: *Proc. 13th European Conf. on Artificial Intelligence*, pages 403-407, 1998.