# On Abductive Equivalence

Katsumi Inoue
National Institute of Informatics
2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo 101-8430, Japan
ki@nii.ac.jp

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

February 25, 2005

**Abstract**

We consider the problem of identifying equivalence of two knowledge bases which are capable of abductive reasoning. Here, a knowledge base is written in either first-order logic or nonmonotonic logic programming. In this work, we will give two definitions of abductive equivalence. The first one, *explainable equivalence*, requires that two abductive programs have the same explainability for any observation. Another one, *explanatory equivalence*, guarantees that any observation has exactly the same explanations in each abductive framework. Explanatory equivalence is a stronger notion than explainable equivalence, and in fact, the former implies the latter. In first-order abduction, explainable equivalence can be verified by the notion of extensional equivalence in default theories. In nonmonotonic logic programs, explanatory equivalence can be checked by means of the notion of relative strong equivalence. We also discuss how the two notions of abductive equivalence can be applied to extended abduction, where abducibles can not only be added to a program but also be removed from the program to explain an observation.

## 1 Introduction

Nowadays, abduction is used in many AI applications, including diagnosis, design, updates, and discovery. Abduction is an important paradigm for problem solving, and is incorporated in programming technologies, i.e., *abductive logic programming* (ALP) [14; 2]. Automated abduction is also studied in the literature as an extension of deductive methods or a part of inductive systems [5; 8], and its computational properties have also been studied [25; 3; 4].

In this work, we are concerned with such computational issues on abductive reasoning. Despite being a problem-solving paradigm, ALP has a lot of issues which have not been fully understood yet. In particular, there are no concrete methods for (a) evaluation of abductive power in ALP, (b) measurement of efficiency in abductive reasoning, (c) semantically correct simplification and optimization, (d) debugging and verification in ALP, and (e) standardization in ALP. Since all these topics are important for any programming paradigm, the lack of them is a serious drawback of ALP. Then, it can be recognized that all the above issues are related to different notions of identification or *equivalence* in ALP. In particular, the item (c) is related to understanding the semantics of ALP with respect to modularity and *contexts*.

The notion of equivalence between two knowledge bases is also one of the most important problems in knowledge representation based on logic. For example, one axiom set $A_1$ represents the specification of a device or a program and the other formula set $A_2$ is a result of the design of a hardware/software system. Then, we should check whether $A_2$ is equivalent to $A_1$, intending the verification of the design. Similarly, the notion of equivalence in logic programming has recently become important. Because a logic program is used to represent knowledge of a problem domain [1], we often have to consider whether two logic programs $P_1$ and $P_2$ represent the same knowledge. For example, one logic program $P_1$ may be viewed as a specification of knowledge in some domain, and another representation $P_2$ may be expected to be a compact form of $P_1$ which can easily be computed.

Abduction can be formalized in various logics [15; 5]. Then, we can consider several notions of equivalence in several logics for abduction. In this paper, we will give two definitions of abductive equivalence in two logical frameworks for abduction. Two logics we consider here are *first-order logic* (FOL) and *abductive logic programming* (ALP). The first abductive equivalence, called *explainable equivalence*, requires that two abductive programs have the same explainability for any observation. Another one, *explanatory equivalence*, guarantees that any observation has exactly the same explanations in each abductive framework. Explanatory equivalence is stronger than explainable equivalence, and in fact, the former implies the latter.

In this paper, we characterize these two notions of abductive equivalence in terms of other well-known concepts in AI and logic programming. In abduction in first-order logic, we will see that explainable equivalence can be verified by the notion of *equivalence* in *default logic* [21], which is defined for the families of *extensions* of two default theories. On the other hand, abductive equivalence in ALP is more complicated than in the case of FOL due to the nonmonotonicity in logic programs. In fact, equivalence between two abductive logic programs has little been discussed in the literature except that effects of some program transformation techniques in ALP are analyzed in [23]. In this work, by means of the notion of *strong equivalence* [18; 16] and its relativized extension [17; 13; 27], we will show that explanatory equivalence can be checked in ALP.

Finally, we also discuss how the two notions of abductive equivalence can be applied to *extended abduction* [9; 12], where hypotheses can not only be added to a program but also be removed from the program to explain an observation. In extended abduction, abductive equivalence can be characterized by the notion of *update equivalence* [13].

The rest of this paper is organized as follows. Section 2 presents two definitions for abductive equivalence. Section 3 considers first-order logic as the representation language, while Section 4 considers nonmonotonic logic programming for ALP. Section 5 extends the equivalence results in ALP to extended abduction. Section 6 gives concluding remarks.

## 2 Abductive Equivalence

We start with a question as to when two abductive frameworks are equivalent. As far as the authors know, there is no answer for such a question in the literature of ALP. Moreover, no such a concept can be found in philosophy, either. It is conceivable that there must be several aspects on this question. When can we consider that an *explanation* $E$ is equivalent to another explanation $F$ for an observation? When can we say that an *observation* $G$ is equivalent to another observation $H$ in an abductive framework? In what circumstances, can we say that *abduction by* person $A$ is equivalent to abduction by person $B$? When can we regard that *abduction with* knowledge $P$ is equivalent to abduction with knowledge $Q$?

There are also many parameters which should be considered important in defining equivalence notions in abductive frameworks. In the *world*, both background knowledge and observations are surely essential. In an *agent* who performs abduction, on the other hand, her abductive power must depend on her *logic* (language, syntax, semantics) of background knowledge, observations and hypotheses. Moreover, the quality of abduction is relevant to other parameters such as axioms, inference procedures, logics of explanations, and criteria of best explanations. If we would take all such parameters into account, the task of defining the equivalence notion might become combinatorial and too complex.

In the following, we thus consider a rather simple framework for our problem while we try to hold the essence of equivalence notions as much as possible. First, logic, background knowledge and hypotheses are put as input parameters in each abductive framework. Second, a logic of explanations is taken into account in a definition, but its diversity is reflected in different notions of abductive equivalence.

The following definition of abductive frameworks is a standard one [15; 25; 3; 4]. As a notation, $\Sigma \models_L F$ means that a formula $F$ is derived from a set $\Sigma$ of formulas in a logic $L$.

**Definition 2.1** Let $B$ and $H$ be sets of formulas in some underlying logic $L$. An *abductive framework* is defined as a triple $(L, B, H)$, where $B$ is called *background knowledge* and each element of $H$ is called a *candidate hypothesis*.

**Definition 2.2** Let $(L, B, H)$ be an abductive framework, and $O$ a formula in $L$, and $E$ a formula belonging to $H$. We define that $E$ is an *explanation* of an *observation* $O$ in $(L, B, H)$ if $B \cup E \models_L O$ and $B \cup E$ is consistent in $L$. We say that $O$ is *explainable* in $(L, B, H)$ if it has an explanation in $(L, B, H)$.

**Remark.** Definition 2.2 requires that each explanation $E$ must be consistent with the background knowledge $B$ in the logic $L$. This condition is sometimes too strong in realistic cases, and can be weaken if the logic $L$ is *paraconsistent*.

In the next two sections, we consider two logics for abduction both of which are popular formalisms in AI: first-order logic (FOL) (Section 3), and logic programming with negation as failure (ALP) (Section 4).

We now give two definitions for abductive equivalence. We assume that the underlying logic $L$ is common when two abductive theories are compared.

**Definition 2.3** Two abductive frameworks $(L, B_1, H_1)$ and $(L, B_2, H_2)$ are *explainably equivalent* if, for any observation $O$, there is an explanation of $O$ in $(L, B_1, H_1)$ iff there is an explanation of $O$ in $(L, B_2, H_2)$.

Explainable equivalence requires that two abductive frameworks have the same *explainability* for any observation. Explainable equivalence may reflect a situation that two theories have different knowledge to derive the same goals.

**Definition 2.4** Two abductive frameworks $(L, B_1, H_1)$ and $(L, B_2, H_2)$ are *explanatorily equivalent* if, for any observation $O$, $E$ is an explanation of $O$ in $(L, B_1, H_1)$ iff $E$ is an explanation of $O$ in $(L, B_2, H_2)$.

Explanatory equivalence assures that two abductive frameworks have the same *explanation power* for any observation. Explanatory equivalence is stronger than explainable equivalence. In fact, the former implies the latter. The two notions coincide if $H_1 = H_2 = \emptyset$.

**Proposition 2.1** *If abductive frameworks $(L, B_1, H_1)$ and $(L, B_2, H_2)$ are explanatorily equivalent, then they are explainably equivalent.*

**Proposition 2.2** *Two abductive frameworks $(L, B_1, \emptyset)$ and $(L, B_2, \emptyset)$ are explainably equivalent iff they are explanatorily equivalent.*

For explanatory equivalence, we can assume that the hypotheses $H$ are common in two abductive frameworks in Definition 2.4, as the following property holds.

**Proposition 2.3** *Suppose that $A_1 = (L, B_1, H_1)$ and $A_2 = (L, B_2, H_2)$ are abductive frameworks. If $A_1$ and $A_2$ are explanatorily equivalent, then $H_1' = H_2'$, where $H_i' = \{h \in H_i \mid B_i \cup \{h\}$ is consistent in $L\}$ for $i = 1, 2$.*

**Proof.** Assume that $H_1' \setminus H_2' \neq \emptyset$. Then, for a formula $\varphi \in H_1' \setminus H_2'$, $\{\varphi\}$ is an explanation of $\varphi$ in $A_1$ because $B_1 \cup \{\varphi\}$ is consistent in $L$. However, $\{\varphi\}$ is not an explanation of $\varphi$ in $A_2$. Hence, $A_1$ and $A_2$ are not explanatorily equivalent. □

Note in Proposition 2.3 that any hypothesis $h$ in $H_i \setminus H_i'$ cannot be added without violating the consistency of $B_i \cup \{h\}$ in $L$. Thus, $H_i'$ is the set of hypotheses that can be actually used in explanations of some formulas.

**Example 2.1** Suppose two abductive frameworks, $A_1 = (\text{FOL}, \{a \rightarrow p\}, \{a, b\})$ and $A_2 = (\text{FOL}, \{b \rightarrow p\}, \{a, b\})$. Then, $A_1$ and $A_2$ are explainably equivalent, but are not explanatorily equivalent. On the other hand, $A_3 = (\text{FOL}, \{a \rightarrow p\}, \{b\})$ and $A_4 = (\text{FOL}, \{b \rightarrow p\}, \{b\})$ are neither explainably equivalent nor explanatorily equivalent.

## 3 Abduction in First-order Logic

Abduction is used in many AI applications, and classical first-order logic (FOL) is most often used as the underlying logic for abduction [20; 15; 3; 25; 8]. When the underlying logic $L$ is FOL, the relation $\models_L$ becomes the usual entailment relation $\models$. A first-order formula $f$ is *closed* if $f$ contains no free variables. A *ground instance* of a first-order formula $f$ is a formula obtained by replacing every variable in $f$ with a term containing no variables. In first-order abduction, explanations are usually defined as a set of ground instances from hypotheses as follows [20; 8].

**Definition 3.1** Suppose an abductive framework $(\text{FOL}, B, H)$, where both the background knowledge $B$ and the hypotheses $H$ are sets of first-order formulas. Given a closed formula $O$ as an observation, a set $E$ of ground instances of elements of $H$ is an *explanation* of $O$ in $(\text{FOL}, B, H)$ if $\Sigma \cup E \models G$ and $\Sigma \cup E$ is consistent.

In the following, $Th(\Sigma)$ denotes the set of logical consequences of a set $\Sigma$ of first-order formulas. That is, $Th(\Sigma) = \{F \mid \Sigma \models F\}$. The next definition is originally given for *default logic* by Reiter [21].

**Definition 3.2** [22; 20] Let $B$ and $H$ be sets of first-order formulas. An *extension of $B$ with respect to $H$* is $Th(B \cup S)$ where $S$ is a maximal subset of ground instances of elements from $H$ such that $B \cup S$ is consistent.

When an abductive framework $(\text{FOL}, B, H)$ is given, we can associate a Reiter's *default theory* $\Delta = (D, B)$ where $D$ is the set of *prerequisite-free normal defaults* $\{\frac{:d}{d} \mid d \in H\}$ such that there is a one-to-one correspondence between the *extensions* of $\Delta$ (which are defined in[21]) and the extensions of $B$ with respect to $H$ [20]. Using the notion of extensions in Definition 3.2, explainable equivalence can be characterized in first-order abduction.

**Theorem 3.1** *Two abductive frameworks $(\text{FOL}, B_1, H_1)$ and $(\text{FOL}, B_2, H_2)$ are explainably equivalent iff the extensions of $B_1$ with respect to $H_1$ coincide with the extensions of $B_2$ with respect to $H_2$.*

**Proof.** First, we claim that the union of the extensions of $B$ with respect to $H$ are exactly the set of formulas explainable in $(\text{FOL}, B, H)$. To see this, we can use a well-known theorem [20; 25] that a formula $O$ can be explained in $(\text{FOL}, B, H)$ iff there is a consistent extension $X$ of $B$ with respect to $H$ such that $X$ contains $O$. Thus, the set of all explainable formulas are precisely those formulas contained in at least one extension of $B$ with respect to $H$.

Now, let $A_1 = (\text{FOL}, B_1, H_1)$ and $A_2 = (\text{FOL}, B_2, H_2)$ be two abductive frameworks. Suppose that the extensions of $B_1$ with respect to $H_1$ coincide with those of $B_2$ with respect to $H_2$. By the above claim, the set of formulas explainable in $A_1$ is equal to the set of formulas explainable in $A_2$. This means that $A_1$ and $A_2$ are explainably equivalent.

Conversely, assume that there is an extension $X_2$ of $B_2$ with respect to $H_2$ which is not an extension of $B_1$ with respect to $H_1$. Let $F_{X_2}$ be a first-order formula which is logically equivalent to $X_2$. Such a formula actually exists because $X_2 = Th(B_2 \cup S)$ holds for some maximally consistent subset $S$ of $H_2$, and hence $X_2$ is logically equivalent to $\bigwedge_{f \in B_2} f \wedge \bigwedge_{g \in S} g$. Since $X_2$ is consistent, $F_{X_2}$ is consistent too. Then, $F_{X_2}$ is explainable in $A_2$ because $S$ is an explanation of $F_{X_2}$.

Now, if $F_{X_2}$ is not explainable in $A_1$, then obviously $A_1$ and $A_2$ are not explainably equivalent. Hence, there is an explanation of $F_{X_2}$ in $A_1$. Then, there is an extension $X_1$ of $B_1$ with respect to $H_1$ which contains $F_{X_2}$. Since $X_2$ is not an extension of $B_1$ with respect to $H_1$, $X_1 \neq X_2$ holds. Then, $X_2 \subset X_1$. Let $F_{X_1}$ be a formula which is logically equivalent to $X_1$. By the same argument as above, $F_{X_1}$ is explainable in $A_1$. However, this $F_{X_1}$ cannot be explained in $A_2$. This is because, if $F_{X_1}$ were explained in $A_2$, there must be an extension $X_2'$ of $B_2$ with respect to $H_2$ such that $X_2 \subset X_2'$, which is impossible because any extension is *orthogonal* to another extension in a default theory [21]. In any case, $A_1$ and $A_2$ are not explainably equivalent. □

In [19], Reiter's default theories $\Delta_1 = (D_1, B_1)$ and $\Delta_2 = (D_2, B_2)$ are said to be *equivalent* if the extensions of $\Delta_1$ are the same as the extensions of $\Delta_2$. Using this notation, explainable equivalence in first-order abduction can also be represented as follows.

**Corollary 3.2** *Two abductive frameworks* $(\text{FOL}, B_1, H_1)$ *and* $(\text{FOL}, B_2, H_2)$ *are explainably equivalent iff the default theories* $(D_1, B_1)$ *and* $(D_2, B_2)$ *are equivalent where* $D_i = \{\frac{:d}{d} \mid d \in H_i\}$ *for* $i = 1, 2$.

**Example 3.1** Suppose two abductive frameworks, $A_1 = (\text{FOL}, B_1, H_1)$ and $A_2 = (\text{FOL}, B_2, H_2)$, where

$$
\begin{aligned}
B_1 &= \{a \rightarrow p,\ \ b \rightarrow \neg p\}, \\
H_1 &= \{a,\ b,\ a \equiv c,\ b \equiv d,\ p \equiv q\}, \\
B_2 &= \{c \rightarrow q,\ \ d \rightarrow \neg q\}, \text{ and} \\
H_2 &= \{c,\ d,\ a \equiv c,\ b \equiv d,\ p \equiv q\}.
\end{aligned}
$$

Then, $A_1$ and $A_2$ are explainably equivalent. In fact, the two extensions of $B_1$ with respect to $H_1$ are $Th(B_1 \cup (H_1 \setminus \{b\})) = Th(\{a, \neg b, c, \neg d, p, q\})$ and $Th(B_1 \cup (H_1 \setminus \{a\})) = Th(\{\neg a, b, \neg c, d, \neg p, \neg q\})$, which are respectively equivalent to the two extensions of $B_2$ with respect to $H_2$, $Th(B_2 \cup (H_2 \setminus \{d\}))$ and $Th(B_2 \cup (H_2 \setminus \{c\}))$.

Logical equivalence of background theories implies explainable equivalence when the hypotheses are common.

**Corollary 3.3** *If* $B_1 \equiv B_2$*, then abductive frameworks* $(\text{FOL}, B_1, H)$ *and* $(\text{FOL}, B_2, H)$ *are explainably equivalent. However, the converse does not hold.*

**Proof.** If $B_1 \equiv B_2$, then any extension of $B_1$ with respect to $H$ is an extension of $B_2$ with respect to $H$ and vice versa. The converse does not hold as Example 2.1 shows. □

It is interesting to see that we can transform any abductive framework to an explainably equivalent abductive framework whose background theory is empty. The next property is also derived by the representation theory for default logic [19].

**Corollary 3.4** *For any abductive framework* $(\text{FOL}, B, H)$*, there is an abductive framework* $(\text{FOL}, \emptyset, H')$ *which is explainably equivalent to* $(\text{FOL}, B, H)$.

**Proof.** Put $H' = \{h \wedge \varphi \mid h \in H\} \cup \{\varphi\}$, where $\varphi = \bigwedge_{f \in B} f$. Then, it holds for any $O$ that, $B \cup E \models O$ iff $E' \models O$ where $E \subseteq H$ and $E' = \{h \wedge \varphi \mid h \in E\} \cup \{\varphi\} \subseteq H'$. □

An abductive framework $(L, B, H)$ is called $(B, H)$-*compatible* if $B \cup H$ is consistent. Explainable equivalence can be easily verified for $(B, H)$-compatible frameworks.

**Corollary 3.5** *Let* $(\text{FOL}, B_1, H_1)$ *and* $(\text{FOL}, B_2, H_2)$ *be* $(B_i, H_i)$*-compatible abductive frameworks for* $i = 1, 2$*. Then,* $(\text{FOL}, B_1, H_1)$ *and* $(\text{FOL}, B_2, H_2)$ *are explainably equivalent iff* $B_1 \cup H_1 \equiv B_2 \cup H_2$.

**Proof.** For any $(B, H)$-compatible abductive framework $(\text{FOL}, B, H)$, we have that $B \cup H$ is consistent. Then, $Th(B \cup H)$ is the unique extension of $B$ with respect to $H$. By Theorem 3.1, $(\text{FOL}, B_1, H_1)$ and $(\text{FOL}, B_2, H_2)$ are explainably equivalent iff $Th(B_1 \cup H_1) = Th(B_2 \cup H_2)$. Hence, the corollary holds. □

An abductive framework $(\text{FOL}, B, \mathcal{L})$ is called *assumption-free* where $\mathcal{L}$ is the set of all literals in the underlying language. It is known that the complexity of finding explanations in assumption-free abductive frameworks is not harder than that in assumption-based frameworks [25]. Explainable equivalence in the assumption-free case can also be simply characterized as follows.

**Corollary 3.6** *Abductive frameworks* $(\text{FOL}, B_1, \mathcal{L})$ *and* $(\text{FOL}, B_2, \mathcal{L})$ *are explainably equivalent iff* $B_1 \equiv B_2$.

**Proof.** For an assumption-free abductive framework $(\text{FOL}, B, \mathcal{L})$, each extension of $B$ with respect to $\mathcal{L}$ is logically equivalent to a model of $B$. Hence, explainable equivalence implies that the models of $B_1$ coincide with the models of $B_2$, and vice versa. □

For explanatory equivalence in first-order abduction, logical equivalence of background theories is necessary and sufficient.

**Theorem 3.7** *Two abductive frameworks* $(\text{FOL}, B_1, H)$ *and* $(\text{FOL}, B_2, H)$ *are explanatorily equivalent iff* $B_1 \equiv B_2$.

**Proof.** If $B_1 \equiv B_2$, then for any $E$ and any $O$, it holds that, $B_1 \cup E \models O$ iff $B_2 \cup E \models O$, and that, $B_1 \cup E$ is consistent iff $B_2 \cup E$ is consistent. Hence, $(\text{FOL}, B_1, H)$ and $(\text{FOL}, B_2, H)$ are explanatorily equivalent.

Conversely, suppose that $(\text{FOL}, B_1, H)$ and $(\text{FOL}, B_2, H)$ are explanatorily equivalent. Then, for any formula $O$ and any $E$ from $H$, it holds that $B_1 \cup E \models O$ iff $B_2 \cup E \models O$. Then, for any $E$, we have $Th(B_1 \cup E) = Th(B_2 \cup E)$. That is, $B_1 \cup E \equiv B_2 \cup E$ holds for any $E$. This implies $B_1 \equiv B_2$ when $E = \emptyset$. □

# 4 Abductive Logic Programming

Abductive logic programming (ALP) is another popular formalization of abduction in AI [14; 2; 4]. Background knowledge in ALP is called a *logic program*, and the candidate hypotheses are given as literals called *abducibles*. The most significant difference between abduction in FOL and ALP is that ALP allows the nonmonotonic *negation-as-failure* operator $not$ in background knowledge. In abduction, addition of hypotheses may invalidate explanations of some observations if the background theory is nonmonotonic.

Recall that a *(logic) program* is a set of rules of the form

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

where each $L_i$ is a literal $(n \geq m \geq l \geq k \geq 0)$, and $not$ is *negation as failure* (NAF). The symbol ; represents a disjunction. The left-hand side of the rule is called the *head*, and the right-hand side is called the *body*. A rule with variables stands for the set of its ground instances. Intuitively, the rule in the above form can be read as follows: if all $L_{l+1}, \ldots , L_m$ are believed and all $L_{m+1}, \ldots , L_n$ are disbelieved then either some $L_i$ $(1 \leq i \leq k)$ should be believed or some $L_j$ $(k + 1 \leq j \leq l)$ should be disbelieved.

In this paper, the semantics of a logic program is given by its *answer sets* [6; 1; 11], while another semantics can be considered as well in ALP [14; 4]. Intuitively speaking, each answer set represents a set of literals corresponding to beliefs which can be built by a rational reasoner on the basis of a program [1]. The answer sets for a program are defined in the following two steps [6; 11]. First, let $P$ be a program without NAF (i.e., $k = l$ and $m = n$) and $S \subseteq \mathcal{L}$, where $\mathcal{L}$ is the set of all ground literals in the language of $P$. Then, $S$ is an *answer set* of $P$ if $S$ is a minimal set satisfying the conditions:

1. $S$ satisfies every rule in $P$, that is, for any ground rule of the form $L_1; \cdots ; L_l \leftarrow L_{l+1}, \ldots , L_m$ from $P$, if $\{L_{l+1}, \ldots , L_m\} \subseteq S$ then $\{L_1, \ldots , L_l\} \cap S \neq \emptyset$;

2. If $S$ contains a pair of complementary literals $L$ and $\neg L$, then $S = \mathcal{L}$.

Second, given *any* program $P$ (with NAF) and $S \subseteq \mathcal{L}$, consider the program (without NAF) $P^S$ obtained as follows: a rule $L_1; \cdots ; L_k \leftarrow L_{l+1}, \ldots , L_m$ is in $P^S$ if there is a ground rule of the form

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

from $P$ such that $\{L_{k+1}, \ldots , L_l\} \subseteq S$ and $\{L_{m+1}, \ldots , L_n\} \cap S = \emptyset$. Then, $S$ is an *answer set* of $P$ if $S$ is an answer set of $P^S$. An answer set is *consistent* if it is not $\mathcal{L}$. A program is *consistent* if it has a consistent answer set. A program has none, one, or multiple answer sets in general. A typical program which has no answer set is $\{p \leftarrow not\, p\}$. Problem solving by representing knowledge as a logic program and then computing its answer sets is called *answer set programming* (ASP). In ASP, alternative belief sets of a reasoner are represented by multiple answer sets of a program.

**Definition 4.1** An *abductive (logic) program* is defined as a pair $\langle P, \mathcal{A}\rangle$, where $P$ is a logic program and $\mathcal{A}$ is a set of literals called *abducibles*. Instead of using the notation

$(\text{ALP}, P, \mathcal{A})$, we also use $\langle P, \mathcal{A}\rangle$ to represent an abductive framework whose underlying logic is ALP.

**Definition 4.2** Let $\langle P, \mathcal{A}\rangle$ be an abductive program, and $G$ a conjunction of ground literals called *observations*. A set $E \subseteq \mathcal{A}$ is a *(credulous) explanation* of $G$ in $\langle P, \mathcal{A}\rangle$ if every ground literal in $G$ is true in a consistent answer set of $P \cup E$.

Note that both abducibles and observations are restricted to ground literals in ALP. However, it is known for this framework that rules can be allowed in abducibles and that observations can contain NAF formulas as well as literals [10]. We assume that the set of observations includes the special atom $\top$, which represents the empty conjunction of observations. Note that $\top$ is always true in any set of ground literals. Definition 4.2 can also be represented in a different way as follows [10]. A *belief set (with respect to E)* of an abductive program $\langle P, \mathcal{A}\rangle$ is a consistent answer set of a logic program $P \cup E$ where $E \subseteq \mathcal{A}$. Then, $E \subseteq \mathcal{A}$ is an explanation of $G$ if $G$ is true in a belief set of $\langle P, \mathcal{A}\rangle$ with respect to $E$.

**Remark.** In Definition 4.2, explanations are defined in a *credulous* way. Another, *skeptical* notion for explanations is defined as $E \subseteq \mathcal{A}$ such that $G$ is true in all consistent answer sets of $P \cup E$. Abductive equivalence relative to skeptical explanations can also be defined in a similar way, but characterization of such notions needs different formalizations. For instance, instead of taking the union of belief sets in the equation of Theorem 4.1, skeptical consequences are computed by taking the intersection of them.

According to Section 2, we consider two types of abductive equivalence for ALP.

**Definition 4.3** Abductive programs $\langle P_1, \mathcal{A}_1\rangle$ and $\langle P_2, \mathcal{A}_2\rangle$ are *explainably equivalent* if, for any ground literal $G$, $G$ is explainable in $\langle P_1, \mathcal{A}_1\rangle$ iff $G$ is explainable in $\langle P_1, \mathcal{A}_2\rangle$.

**Definition 4.4** Abductive programs $\langle P_1, \mathcal{A}\rangle$ and $\langle P_2, \mathcal{A}\rangle$ are *explanatorily equivalent* if, for any conjunction of ground literals $G$, $E$ is an explanation of $G$ in $\langle P_1, \mathcal{A}\rangle$ iff $E$ is an explanation of $G$ in $\langle P_2, \mathcal{A}\rangle$.

Explainable equivalence in ALP guarantees the same explainability for any ground literal as a *single observation*, but it does not matter how each observation is explained. Hence, we do not have to care about whether multiple observations can be *jointly* explained by a common explanation. On the other hand, explanatory equivalence in ALP guarantees that, any *conjunction* (or *set*) *of observations* has exactly the same credulous explanations. Hence, explanatory equivalence implies that any set of abducibles $E \subseteq \mathcal{A}$ should explain the same set of observations in each abductive program. Again, explanatory equivalence implies explainable equivalence.

We now show that explainable equivalence in ALP can be checked by comparing the belief sets of two abductive programs. Because there exist several methods to compute belief sets using ASP [7; 24; 10; 11], checking explainable equivalence is also possible using such methods. In the following, we denote the set of all belief sets of $\langle P, \mathcal{A}\rangle$ as $BS(P, \mathcal{A})$.

**Theorem 4.1** *Abductive programs $\langle P_1, \mathcal{A}_1 \rangle$ and $\langle P_2, \mathcal{A}_2 \rangle$ are explainably equivalent iff*

$$\bigcup_{S \in BS(P_1, \mathcal{A}_1)} S = \bigcup_{S \in BS(P_2, \mathcal{A}_2)} S.$$

**Proof.** Recall that $E \subseteq \mathcal{A}$ is an explanation of a ground literal $G$ iff $G$ is true in a belief set of $\langle P, \mathcal{A} \rangle$ with respect to $E$. Then, the set of all explainable literals are precisely those literals contained in some belief sets of $\langle P, \mathcal{A} \rangle$ with respect to some $E$. Hence, the union of the belief sets of $\langle P, \mathcal{A} \rangle$ are exactly the set of literals explainable in $\langle P, \mathcal{A} \rangle$. Therefore, two abductive programs are explainably equivalent iff the unions of the belief sets of two abductive programs coincide. $\square$

The next corollary gives a sufficient condition.

**Corollary 4.2** *Abductive programs $\langle P_1, \mathcal{A}_1 \rangle$ and $\langle P_2, \mathcal{A}_2 \rangle$ are explainably equivalent if $BS(P_1, \mathcal{A}_1) = BS(P_2, \mathcal{A}_2)$.*

In some case of $(B, H)$-compatible problems, explanatory equivalence can be easily verified. Here, a logic program is *definite* if every its rule is NAF-free and has exactly one atom in the head and only atoms in the body. A definite program has a unique answer set that is equivalent to its *least model*. An abductive program $\langle P, \mathcal{A} \rangle$ is called *definite* if $P$ is a definite logic program and $\mathcal{A}$ is a set of atoms.

**Corollary 4.3** *Suppose that $\langle P_1, \mathcal{A}_1 \rangle$ and $\langle P_2, \mathcal{A}_2 \rangle$ are definite abductive programs. Then, $\langle P_1, \mathcal{A}_1 \rangle$ and $\langle P_2, \mathcal{A}_2 \rangle$ are explainably equivalent if the least model of $P_1 \cup \mathcal{A}_1$ coincides with that of $P_2 \cup \mathcal{A}_2$.*

**Example 4.1** Given the common set of abducibles $\mathcal{A} = \{a, b\}$ and three logic programs:

$$\begin{array}{rcl} P_1 & = & \{\, p \leftarrow a, \;\; q \leftarrow b \,\}, \\ P_2 & = & \{\, p \leftarrow b, \;\; q \leftarrow a \,\}, \\ P_3 & = & \{\, p \leftarrow \,, \;\; q \leftarrow a, \quad \leftarrow a, b\}, \end{array}$$

the three abductive programs $\langle P_i, \mathcal{A} \rangle$ (for $i = 1, 2, 3$) are all explainably equivalent, but none of them are explanatorily equivalent. In particular, the least model of $P_1 \cup \mathcal{A}$ is $\{p, q, a, b\}$, which is identical to that of $P_2 \cup \mathcal{A}$. $P_3$ is not definite because of the third rule, but $\langle P_3, \mathcal{A} \rangle$ has three belief sets: $\{p\}, \{p, q, a\}, \{p, b\}$, the union of which is equal to that of $\langle P_i, \mathcal{A} \rangle$ for $i = 1, 2$.

Explanatory equivalence in ALP, on the other hand, requires a more semantical notion of logic programming. Note that explanatory equivalence of $\langle P_1, \mathcal{A} \rangle$ and $\langle P_2, \mathcal{A} \rangle$ implies $BS(P_1, \mathcal{A}) = BS(P_2, \mathcal{A})$, but the converse does not hold.

**Example 4.2** Suppose $P_1 = \{a \leftarrow \,, \;\; p \leftarrow a\}$, $P_2 = \{a \leftarrow \mathit{not}\, a, \;\; p \leftarrow a\}$ and $\mathcal{A} = \{a\}$. Then, $BS(P_1, \mathcal{A}) = BS(P_2, \mathcal{A}) = \{\{a, p\}\}$. However, $\emptyset$ is an explanation of $p$, $a$ and $\top$ in $\langle P_1, \mathcal{A} \rangle$, but is not an explanation of them in $\langle P_2, \mathcal{A} \rangle$. In fact, $P_2$ alone has no answer set although $P_2 \cup \{a\}$ has the answer set $\{a, p\}$.

To characterize explanatory equivalence precisely, we need to utilize the concept of equivalence in logic programming and ASP. There are several notions for equivalence in logic programming, and *weak equivalence* and *strong equivalence* are most well known. We say that two programs are *weakly equivalent* if they simply agree with their answer sets. The notion of weak equivalence is similar to that of *logical equivalence* in FOL and other classical logics. Given two abductive programs $\langle P_1, \mathcal{A}\rangle$ and $\langle P_2, \mathcal{A}\rangle$, weak equivalence of $P_1$ and $P_2$ is not a sufficient condition for explanatory equivalence of them, and is not even a sufficient condition for explainable equivalence. However, weak equivalence is meaningful when the abducibles are empty.

**Proposition 4.4** *Abductive programs $\langle P_1, \emptyset\rangle$ and $\langle P_2, \emptyset\rangle$ are explanatorily equivalent iff $P_1$ and $P_2$ are weakly equivalent.*

On the other hand, *strong equivalence* [18; 16] is a more context-sensitive notion for equivalence of logic programs. Two logic programs $P_1$ and $P_2$ are said to be *strongly equivalent* if for any additional logic program $R$, $P_1 \cup R$ and $P_2 \cup R$ have the same answer sets. Obviously, strong equivalence implies weak equivalence (when $R = \emptyset$). When we allow NAF in logic programs, weak equivalence is too fragile as a criterion. For example, $\{\, p \leftarrow not\, a\}$ and $\{\, p \leftarrow \,\}$ are weakly equivalent with the same unique answer set $\{p\}$, but are not strongly equivalent because the addition of $a$ to both results in the withdrawal of $p$ in the former only. In [16], it is argued that strong equivalence can be used to simplify a part of a logic program without looking at the other part. For example, $\{\, p \leftarrow p \,\}$ and $\emptyset$ are strongly equivalent, so that the rule in the former can always be eliminated from any program.

For many applications, however, strong equivalence is too strong, and often we can restrict the language for additional programs $R$ to some subset $\mathcal{R}$ of the whole language of programs. Then, two programs $P_1$ and $P_2$ are said to be *strongly equivalent with respect to* $\mathcal{R}$ if $P_1 \cup R$ and $P_2 \cup R$ have the same answer sets for any program $R \subseteq \mathcal{R}$ [13]. Such restriction of $\mathcal{R}$ is practicably interesting because knowledge bases are usually divided into invariable and variable parts such that only variable parts are changed in updates. The equivalence notion with such restriction is called *relative strong equivalence* [17; 13; 27]. Using this notion, explanatory equivalence can be characterized as follows.

**Theorem 4.5** *Two abductive programs $\langle P_1, \mathcal{A}\rangle$ and $\langle P_2, \mathcal{A}\rangle$ are explanatorily equivalent iff $P_1$ and $P_2$ are strongly equivalent with respect to $\mathcal{A}$.*

**Proof.** Suppose that $\langle P_1, \mathcal{A}\rangle$ and $\langle P_2, \mathcal{A}\rangle$ are explanatorily equivalent. Then, for any conjunction $G$ of literals and any $E \subseteq \mathcal{A}$, it holds that, $E$ is an explanation of $G$ in $\langle P_1, \mathcal{A}\rangle$ iff $E$ is an explanation of $G$ in $\langle P_2, \mathcal{A}\rangle$. The latter equivalence then implies that, for any $G$ and any $E$, we have that, $G$ is true in a belief set of $\langle P_1, \mathcal{A}\rangle$ with respect to $E$ iff $G$ is true in a belief set of $\langle P_2, \mathcal{A}\rangle$ with respect to $E$. Then, for any $G$ and any $E$, $G$ is true in an answer set of $P_1 \cup E$ iff $G$ is true in an answer set of $P_2 \cup E$. That is, for any $E$ and any set $S$ of literals, $S$ is an answer set of $P_1 \cup E$ iff $S$ is an answer set of $P_2 \cup E$. Hence, $P_1$ and $P_2$ are strongly equivalent with respect to $\mathcal{A}$. The converse direction can also be proved by tracing the above proof backward. $\square$

**Example 4.3** Given the common set of abducibles $\mathcal{A} = \{a, b\}$, consider three programs

$$\begin{aligned} P_1 &= \{\, p \leftarrow a,\;\; a \leftarrow b \,\}, \\ P_2 &= \{\, p \leftarrow a,\;\; p \leftarrow b,\;\; a \leftarrow b \,\}, \\ P_3 &= \{\, p \leftarrow b,\;\; a \leftarrow b \,\}. \end{aligned}$$

Then, the three abductive programs $\langle P_i, \mathcal{A} \rangle$ (for $i = 1, 2, 3$) are explainably equivalent. Although $\langle P_1, \mathcal{A} \rangle$ is explanatorily equivalent to $\langle P_2, \mathcal{A} \rangle$, it is not to $\langle P_3, \mathcal{A} \rangle$ [23]. In fact, $P_1$ and $P_2$ are strongly equivalent with respect to $\mathcal{A}$, while $P_1$ and $P_3$ are not because the addition of $a$ derives $p$ in $P_1$ but this is not the case in $P_3$. This example shows that unfold/fold transformation [26] does not preserve explanatory equivalence in ALP [23] even when $P_1$ and $P_2$ are definite.

# 5 Abduction with Removal of Hypotheses

The two notions of abductive equivalence in Section 4 can be applied to *extended abduction* [9; 12] in ALP, in which abducibles can not only be added to a program but also be removed from the program to explain an observation. Extended abduction is defined by Inoue and Sakama[9] in autoepistemic logic for formalizing dynamics of abductive theories, and is then incorporated in ALP [12]. The intuition behind extended abduction is that, when the underlying logic is nonmonotonic, removal of some formulas makes other formulas become true. Hence, explanations are caused not only by addition of new hypotheses but also by deletion of old hypotheses.

To characterize abductive equivalence in extended abduction, we need to extend both the definition of belief sets and the notion of relative strong equivalence by taking removals of literals and rules into account.

**Definition 5.1** Let $\langle P, \mathcal{A} \rangle$ be an abductive program, and $G$ a conjunction of ground literals. A pair $(E, F)$ where $E, F \subseteq \mathcal{A}$ is a *(credulous) explanation* of $G$ (in $\langle P, \mathcal{A} \rangle$) if $G$ is true in some consistent answer set of $(P \setminus F) \cup E$.

The notion of *normal* abduction, which has been discussed in Section 4, can be defined as the task of finding explanations with $F = \emptyset$ in extended abduction.

**Remark.** In extended abduction, Inoue and Sakama also define the notion of *anti-explanations* as follows [9; 12]. A pair $(E, F) \in 2^{\mathcal{A}} \times 2^{\mathcal{A}}$ is an *anti-explanation* of $G$ if $G$ is not explainable in $(P \setminus F) \cup E$. The notion of anti-explanations is useful when there are negative observations which should not exist in the world. Because explanations are defined in a credulous way in Section 4, anti-explanations are defined in a skeptical way: $(E, F)$ is an anti-explanation of $G$ if $G$ is not true in any consistent answer set of $(P \setminus F) \cup E$. Abductive equivalence relative to anti-explanations can also be defined in a way similar to that for explanations in this section, but characterization of such notions needs different formalizations.

We now define the notion of abductive equivalence for extended abduction. This can be done as straightforward extensions of Definitions 4.3 and 4.4.

**Definition 5.2** Abductive programs $\langle P_1, \mathcal{A}_1 \rangle$ and $\langle P_2, \mathcal{A}_2 \rangle$ are *explainably equivalent in extended abduction* if, for any ground literal $G$, $G$ is explainable in $\langle P_1, \mathcal{A}_1 \rangle$ iff $G$ is explainable in $\langle P_1, \mathcal{A}_2 \rangle$.

**Definition 5.3** Abductive programs $\langle P_1, \mathcal{A} \rangle$ and $\langle P_2, \mathcal{A} \rangle$ are *explanatorily equivalent in extended abduction* if, for any conjunction of ground literals $G$, $(E, F)$ is an explanation of $G$ in $\langle P_1, \mathcal{A} \rangle$ iff $(E, F)$ is an explanation of $G$ in $\langle P_2, \mathcal{A} \rangle$.

The notion of belief sets also needs to be extended by taking removal of abducibles into account. That is, a *belief set (with respect to* $(E, F)$*)* of $\langle P, \mathcal{A} \rangle$ is a consistent answer set of $(P \setminus F) \cup E$ where $E, F \subseteq \mathcal{A}$. Then, a pair $(E, F)$ is an explanation of $G$ iff $G$ is true in a belief set of $\langle P, \mathcal{A} \rangle$ with respect to $(E, F)$.

Like Theorem 4.1, explainable equivalence in extended abduction can be checked by comparing the unions of the belief sets of two abductive programs.

**Example 5.1** Suppose the set of abducibles $\mathcal{A} = \{a\}$ and two logic programs:

$$\begin{aligned} P_1 &= \{\, p \leftarrow not\, a, \;\; a \leftarrow \,\}, \\ P_2 &= \{\, p \leftarrow a \,\}, \end{aligned}$$

the abductive programs $A_1 = \langle P_1, \mathcal{A} \rangle$ and $A_2 = \langle P_2, \mathcal{A} \rangle$ are explainably equivalent in extended abduction. In fact, $p$ has the unique explanation $(\emptyset, \{a\})$ in $A_1$ and the explanations $(\{a\}, \emptyset)$ and $(\{a\}, \{a\})$ in $A_2$, and $a$ has the explanations $(\emptyset, \emptyset)$, $(\{a\}, \emptyset)$ and $(\{a\}, \{a\})$ in $A_1$ and the explanations $(\{a\}, \emptyset)$ and $(\{a\}, \{a\})$ in $A_2$. Obviously, $A_1$ and $A_2$ are not explanatorily equivalent in extended abduction.

To characterize explanatory equivalence in extended abduction, we use the equivalence criterion called *update equivalence* [13]. Given two sets of rules $\mathcal{Q}$ and $\mathcal{R}$, two logic programs $P_1$ and $P_2$ are said to be *update equivalent with respect to* $(\mathcal{Q}, \mathcal{R})$ if $(P_1 \setminus Q) \cup R$ and $(P_2 \setminus Q) \cup R$ have the same answer sets for any two logic programs $Q \subseteq \mathcal{Q}$ and $R \subseteq \mathcal{R}$. Here, two parameters $\mathcal{Q}$ and $\mathcal{R}$ correspond to the languages for deletion and addition, respectively. Update equivalence is suitable for taking program updates into account when two logic programs are compared. Clearly, the notion of strong equivalence is a special case of update equivalence where $\mathcal{Q}$ is empty and $\mathcal{R}$ is the set of all rules in the language. The notion of update equivalence is strong enough to capture explanatory equivalence as the next theorem shows.

**Theorem 5.1** *Two abductive programs $\langle P_1, \mathcal{A} \rangle$ and $\langle P_2, \mathcal{A} \rangle$ are explanatorily equivalent in extended abduction iff $P_1$ and $P_2$ are update equivalent with respect to $(\mathcal{A}, \mathcal{A})$.*

**Proof.** Suppose that $\langle P_1, \mathcal{A} \rangle$ and $\langle P_2, \mathcal{A} \rangle$ are explanatorily equivalent in extended abduction. Then, for any conjunction $G$ of literals and any $E, F \subseteq \mathcal{A}$, it holds that, $(E, F)$ is an explanation of $G$ in $\langle P_1, \mathcal{A} \rangle$ iff $(E, F)$ is an explanation of $G$ in $\langle P_2, \mathcal{A} \rangle$. The latter equivalence then implies that, for any $G$ and any $(E, F)$, we have that, $G$ is true in a belief set of $\langle P_1, \mathcal{A} \rangle$ with respect to $(E, F)$ iff $G$ is true in a belief set of $\langle P_2, \mathcal{A} \rangle$ with respect to $(E, F)$. Then, for any $G$ and any $(E, F)$, $G$ is true in an answer set of $(P_1 \setminus F) \cup E$ iff $G$ is true in an answer set of $(P_2 \setminus F) \cup E$. That is, for any $(E, F)$ and any set $S$ of literals, $S$ is an answer set of $(P_1 \setminus F) \cup E$ iff $S$ is an answer set of $(P_2 \setminus F) \cup E$. Hence, $P_1$ and $P_2$ are update equivalent with respect to $(\mathcal{A}, \mathcal{A})$. The converse direction can also be proved by tracing the above proof backward. $\Box$

**Example 5.2** Suppose that two programs $P_1$ and $P_2$ are given as

$$\begin{aligned} P_1 &= \{\, p \leftarrow a, q, \quad q \leftarrow not\, b, \quad b \leftarrow \,\}, \\ P_2 &= \{\, p \leftarrow a, not\, b, \quad q \leftarrow not\, b, \quad b \leftarrow \,\}. \end{aligned}$$

Let $\mathcal{A}_1 = \{a, b\}$ and $\mathcal{A}_2 = \{a, b, p, q\}$. Then, $\langle P_1, \mathcal{A}_1\rangle$ and $\langle P_2, \mathcal{A}_1\rangle$ are explanatorily equivalent, while $\langle P_1, \mathcal{A}_2\rangle$ and $\langle P_2, \mathcal{A}_2\rangle$ are not explanatorily equivalent. In fact, $P_1$ and $P_2$ are update equivalent with respect to $(\mathcal{A}_1, \mathcal{A}_1)$, but are not with respect to $(\mathcal{A}_2, \mathcal{A}_2)$. For the latter claim, we see that the answer sets of $P_1 \cup \{a, q\}$ are $\{\{a, b, p, q\}\}$, which are not the same as the answer sets of $P_2 \cup \{a, q\}$, i.e., $\{\{a, b, q\}\}$.

# 6 Discussion

We have introduced the notion of abductive equivalence in this paper. We have considered two definitions of abductive equivalence in two logics. Two important differences between FOL and ALP as the underlying logics are that (1) explainability is considered for all formulas in FOL while only literals are considered as observations in ALP, and that (2) nonmonotonicity by NAF appears in ALP while this is not the case in FOL. Intuitively, the restriction of observations to literals in ALP gives more chances for two abductive programs to be equivalent, but the existence of nonmonotonicity in ALP makes comparison of abductive programs more complicated.

We have observed that logical equivalence of background theories in FOL or weak equivalence of logic programs does not simply imply abductive equivalence except for some very simple cases. That is why we need to characterize abductive equivalence in terms of other known concepts in classical or nonmonotonic logics. Having such characterizations in this paper, the next target will be to develop transformation techniques which preserve abductive equivalence.

We have considered a rather simple framework for abductive equivalence. In future work, further parameters should be considered in defining abductive equivalence. For example, we can consider another underlying logic for background theories, hypotheses and observations. The criteria of *best explanations* are also important. It is easy to show that explanatory equivalence implies coincidence of the *minimal* explanations. However, the converse does not hold.

Another future topic is to define the concept of *generality/specificity* or *strength/weakness* for abductive frameworks. These concepts are useful for comparing two abductive frameworks, and generality is related to induction too. It might be natural for such relations to be *anti-symmetric*, that is, two abductive frameworks are explainably/explanatorily *equivalent* iff one is both stronger and weaker than another at the same time. Once such a notion is formalized, suppose we know that an abductive program $\langle P_1, \mathcal{A}_1\rangle$ is weaker than another abductive program $\langle P_2, \mathcal{A}_2\rangle$. This means, for example, that there is a literal $G$ which cannot be explained in the former but can be in the latter. Then, we expect that $P_1$ may have less knowledge than $P_2$ or $\mathcal{A}_1$ may have less hypotheses than $\mathcal{A}_2$. However, the situation is more complicated in nonmonotonic programs. Hence, relationships between amounts of background theories and hypotheses should be important in these concepts for abductive frameworks.

Abduction has been used in the process of *scientific discovery*. We should update our theory in accordance with situation change and discovery of surprising facts. The notions of equivalence and generality are always important in evaluating such scientific processes. We hope that our work can serve a basis for the theory of abductive change.

# References


[1] Baral, C. and Gelfond, M., 1994, Logic programming and knowledge representation, *J. Logic Programming*, 19/20:73–148.

[2] Denecker, M. and Kakas, A., 2002, Abductive logic programming, in: *Computational Logic: Logic Programming and Beyond—Essays in Honour of Robert A. Kowalski, Part I*, A.C. Kakas and F. Sadri, eds, LNAI 2407, Springer, pp.402–436.

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

[4] Eiter, T., Gottlob, G., and Leone, L., 1998, Abduction from logic programs: semantics and complexity, *Theoretical Computer Science*, 189:129–177.

[5] Flach, P.A. and Kakas, A.C., eds, 2000, *Abduction and Induction—Essays on their Relation and Integration*, Kluwer Academic.

[6] Gelfond, M. and Lifschitz, L., 1991, Classical negation in logic programs and disjunctive databases, *New Generation Computing*, 9:365–385.

[7] Inoue, K., 1991, Extended logic programs with default assumptions, in: *Proc. 8th International Conference on Logic Programming*, MIT Press, pp.490–504.

[8] Inoue, K., 2002, Automated abduction, in: *Computational Logic: Logic Programming and Beyond—Essays in Honour of Robert A. Kowalski, Part II*, A.C. Kakas and F. Sadri, eds, LNAI 2408, Springer, pp.311–341.

[9] Inoue, K. and Sakama, C., 1995, Abductive framework for nonmonotonic theory change, In: *Proc. IJCAI-95*, Morgan Kaufmann, pp.204–210.

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

[11] Inoue, K. and Sakama, C., 1998, Negation as failure in the head, *J. Logic Programming*, 35:39–78.

[12] Inoue, K. and Sakama, C., 1999, Computing extended abduction through transaction programs, *Annals of Mathematics and Artificial Intelligence*, 25:339-367.

[13] Inoue, K. and Sakama, C., 2004, Equivalence of logic programs under updates, in: *Proc. 9th European Conference on Logics in Artificial Intelligence*, LNAI 3229, Springer, pp.174–186.

[14] Kakas, A.C., Kowalski, R.A., and Toni, F., 1998, The role of abduction in logic programming, in: *Handbook of Logic in Artificial Intelligence and Logic Programming, Volume 5*, D.M. Gabbay, C.J. Hogger, and J.A. Robinson, eds, Oxford University Press, pp.235–324.

[15] Levesque, H.J., 1989, A knowledge-level account of abduction (preliminary version), in: *Proc. IJCAI-89*, Morgan Kaufmann, pp.1061–1067.

[16] Lifschitz, V., Pearce, D., and Valverde, A., 2001, Strongly equivalent logic programs, *ACM Transactions on Computational Logic*, 2:526–541.

[17] Lin, F., 2002, Reducing strong equivalence of logic programs to entailment in classical propositional logic, in: *Proc. 8th International Conference on Principles of Knowledge Representation and Reasoning*, Morgan Kaufmann, pp.170–176.

[18] Maher, M.J., 1988, Equivalence of logic programs, in: *Foundations of Deductive Databases and Logic Programming*, J. Minker, ed., Morgan Kaufmann, pp.627–658.

[19] Marek, V.W., Truer, J., and Truszczyński, M., 1997, Representation theory for default logic, *Annals of Mathematics and Artificial Intelligence*, 21:343–358.

[20] Poole, D., 1988, A logical framework for default reasoning, *Artificial Intelligence*, 36:27–47.

[21] Reiter, R., 1980, A logic for default reasoning, *Artificial Intelligence*, 13:81–132.

[22] Reiter, R., 1987, A theory of diagnosis from first principles, *Artificial Intelligence*, 32:571–95.

[23] Sakama, C. and Inoue, K., 1995, The effect of partial deduction in abductive reasoning, in: *Proc. 12th International Conference on Logic Programming*, MIT Press, pp.383–397.

[24] Satoh, K. and Iwayama, N., 1991, Computing abduction by using the TMS, in: *Proc. 8th International Conference on Logic Programming*, MIT Press, pp.505–518.

[25] Selman, B. and Levesque, H.J., 1996, Support set selection for abductive and default reasoning, *Artificial Intelligence*, 82:259–272.

[26] Tamaki, H. and Sato, T., 1984, Unfold/fold transformation of logic programs, in: *Proc. 2nd International Conference on Logic Programming*, MIT Press, pp.127–138.

[27] Woltran, S., 2004, Characterizations for relativized notions of equivalence in answer set programming, in: *Proc. 9th European Conference on Logics in Artificial Intelligence*, LNAI 3229, Springer, pp.161–173.