Journal of Applied Logic 7 (2009) 318–328

Contents lists available at ScienceDirect

Journal of Applied Logic

www.elsevier.com/locate/jal

Equivalence issues in abduction and induction

Chiaki Sakama [a,*], Katsumi Inoue [b]

[a] Department of Computer and Communication Sciences, Wakayama University, Sakaedani, Wakayama 640-8510, Japan
[b] National Institute of Informatics, 2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo 101-8430, Japan

ARTICLE INFO

Article history:
Received 18 December 2007
Received in revised form 18 December 2007
Accepted 19 October 2008
Available online 6 November 2008

Keywords:
Abduction
Induction
Equivalence

ABSTRACT

This paper discusses several equivalence issues in abduction and induction. Three different problems: equivalence of theories, equivalence of explanations, and equivalence of observations are considered in the context of first-order logic and nonmonotonic logic programming. Necessary and sufficient conditions for those problems, and computational complexity results are provided. These equivalence measures provide methods for comparing different abductive or inductive theories, and also state conditions for optimizing those theories in program development.

1. Introduction

Consider a multiagent society where individual agents have their own knowledge bases. To solve problems cooperatively, agents must share their information in the society. It is likely, however, that the same information is represented in different ways by each agent. To understand information contents and to identify different information sources, the notion of equivalence relation between theories is important. The equivalence relation between theories is also utilized in program development. Given a specification of a problem, a programmer transforms it into an executable program which would be further optimized to increase efficiency. In every step, a program is requested to be semantically equivalent to the original specification.

There is a number of ways for identifying different logical theories. In classical logic, two first-order theories are equivalent if they have the same logical consequences. In logic programming, two logic programs are equivalent if they have the same semantics [13]; and a stronger notion of equivalence is used under the name of *strong equivalence* [12] or *update equivalence* [8]. These equivalence relations compare capabilities of deductive reasoning between theories. On the other hand, considering intelligent agents that can perform commonsense reasoning, it is necessary to have a framework of comparing capabilities of *non-deductive* reasoning like abduction and induction. This motivates the studies by Inoue and Sakama [9,10,18] which introduce several criteria for identifying abductive or inductive theories.

Abduction and induction have analogous inference mechanisms: they both produce hypotheses to explain observations using background theories [4]. There are at least three parameters in this task: theories, explanations, and observations. Several equivalence issues are then considered:

Equivalence of theories: Two abductive (or inductive) theories are considered equivalent if they produce the same explanations for any observation. This equivalence measure is useful for comparing "information contents" of different theories.

* Corresponding author.
E-mail addresses: sakama@sys.wakayama-u.ac.jp (C. Sakama), ki@nii.ac.jp (K. Inoue).

1570-8683/$ – see front matter
doi:10.1016/j.jal.2008.10.006

Equivalence of explanations: Two explanations are considered equivalent if they account for the same observation under a given abductive (or inductive) theory. This equivalence measure is useful for comparing "explanatory power" of different explanations.

Equivalence of observations: Two observations are considered equivalent if they have the same explanations under a given abductive (or inductive) theory. This equivalence measure is useful for comparing "evidential power" of different observations.

The conditions for those equivalence issues generally depend on the logic on which abduction or induction is based. Moreover, those conditions differ among individual abduction (or induction) algorithms. Inoue and Sakama [9,10] study the problem of equivalence of abductive theories under first-order logic and *abductive logic programming* (ALP). Pearce et al. [16] characterize a part of the problem in the context of equilibrium logic. Sakama and Inoue [18] study the corresponding problem in induction and compare conditions for different algorithms in *inductive logic programming* (ILP). On the other hand, equivalence issues with respect to explanations or observations have not been studied so far.

These equivalence measures provide methods for identifying abductive or inductive capabilities of different theories. Moreover, those measures are meaningful and important in the following aspects.

(1) From the viewpoint of program development, if a theory T_1 is transformed to another syntactically different T_2, equivalence of theories guarantees identification of results of abduction (or induction) from each theory. This provides guidelines for optimizing background theories as well as candidate hypotheses in abduction and induction.
(2) It may happen that some algorithm may produce different explanations for the same observation from two theories due to its incorrectness or incompleteness. If two equivalent theories produce different hypotheses in face of some common observations, it indicates that the algorithm is incomplete or incorrect. Thus, equivalence of theories would be used for testing and verifying correctness and completeness of an algorithm.
(3) In system diagnoses, several explanations exist for system failures. If two different disorders of components explain the same system failure, those components may cause the failure interactively. Thus, equivalence of explanations would be used for identifying interrelation of possible causes.
(4) If different observations turn out to be equivalent, they are different appearance of the same phenomenon. For instance, when a person has a sore throat and another person sniffles, a doctor would diagnose both of them as suffering from hay fever. By identifying different symptoms, the same prescription is applied in diagnoses.

Thus, equivalence in abduction and induction is useful not only for identifying different theories, but for program development, verification, and diagnoses.

The purpose of this paper is to discuss those equivalence issues for abduction and induction, and to investigate formal properties. We first review the results of [9–11,18] on the equivalence of abductive (or inductive) theories in Section 2. We then investigate the remaining two problems: equivalence of explanations in Section 3 and equivalence of observations in Section 4. We provide results under two logics, first-order logic and (nonmonotonic) logic programming, which are the two most popular logics used in the literature of abduction and induction. Section 5 discusses the results of this paper, and Section 6 summarizes the paper.

2. Equivalence of abductive theories

2.1. Abductive equivalence in first-order logic

In this section, we first consider the case that the underlying logic is *first-order logic*. The first-order language consists of an alphabet and all formulas defined over it. The definition is the standard one in the literature [3].

As stated in Section 1 we capture both abduction and induction as a process of hypothesis generation given the background theory and observations. To understand two inference mechanisms in a unified framework, we define abduction in a general setting.

Definition 2.1 *(abductive theory).* An *abductive* (*first-order*) *theory* is defined as a pair (B, H) where B and H are sets of first-order formulas, respectively representing the *background theory* and a *candidate hypothesis*. An abductive theory is called *propositional* if both B and H are finite propositional theories.

Let O be any consistent formula representing an *observation*. Then, a set $E \subseteq H$ is an *explanation* of O if

- $B \cup E \models O$, and
- $B \cup E$ is consistent.

An explanation is called *ground* if it contains no variable.

Note that the above definition also characterizes *induction*. A typical induction problem is: given a finite set G of observations and the background theory B, find a hypothesis E such that $B \cup E \models G$ where $B \cup E$ is consistent.[1] Let us compare the definitions of abduction and induction. First, abduction considers a single observation, while induction supposes multiple observations. Since a finite set G of observations is represented as a single formula $O = \bigwedge_{g \in G} g$, abduction is also used for explaining multiple observations in general. Second, abduction selects explanations from a candidate hypothesis H, while induction does not always have such a hypothesis set in advance. This situation in induction is represented by an abductive theory (B, H) by putting $H = \mathcal{F}$ with the set $\mathcal{F}$ of all first-order formulas in the language. Third, abduction is often distinguished from induction by the form of hypotheses: abduction computes hypothetical *facts* for explaining observations, while induction computes hypothetical *rules* for that purpose. To fill the gap, we provided a candidate hypothesis H as a set of first-order formulas. Fourth, induction often considers no background theory. In this situation, we can just put $B = \emptyset$ in an abductive theory.

Thus, from the mathematical viewpoint, there is no essential difference between abduction and induction in Definition 2.1. With this reason, we discuss equivalence issues in abduction hereafter, but similar results hold for induction as well.

Note that in induction problems *negative observations* are often considered as well as positive ones. For any negative observation G, the condition $B \cup E \not\models G$ is requested for any explanation E. To handle negative observations, the notion of *anti-explanations* in the context of *extended abduction* can be used [6]. Using extended abduction, equivalence problems in this paper are extended to handle negative observations as well. For simplicity reasons, we handle positive observations only in this paper. Equivalence of abductive theories in extended abduction is discussed in [10].

To study equivalence issues in abductive logic, Inoue and Sakama [9] introduce two different frameworks of abductive equivalence.

Definition 2.2 *(explainable equivalence).* Two abductive theories (B_1, H_1) and (B_2, H_2) are *explainably equivalent* if, for any observation O, there is an explanation of O in (B_1, H_1) iff there is an explanation of O in (B_2, H_2).

Definition 2.3 *(explanatory equivalence).* Two abductive theories (B_1, H_1) and (B_2, H_2) are *explanatorily equivalent* if, for any observation O, there is an explanation E_1 of O in (B_1, H_1) iff there is an explanation E_2 of O in (B_2, H_2) such that $E_1 \equiv E_2$.

Explainable equivalence requires that two abductive theories have the same *explainability* for any observation. By contrast, explanatory equivalence assures that two abductive theories have the same *explanation contents* for any observation. Explanatory equivalence is stronger than explainable equivalence and the former implies the latter.

Example 2.1. Consider two abductive theories (B_1, H_1) and (B_2, H_2) such that

B_1: *grass_is_wet* ⊃ *shoes_are_wet*,
rained_last_night ⊃ *grass_is_wet*,
sprinkler_was_on ⊃ *grass_is_wet*,
rained_last_night.
H_1: *sprinkler_was_on*.
B_2: *grass_is_wet* ⊃ *shoes_are_wet*,
rained_last_night ⊃ *grass_is_wet*,
sprinkler_was_on ⊃ *grass_is_wet*.
H_2: *rained_last_night*, *sprinkler_was_on*.

Then, (B_1, H_1) and (B_2, H_2) are explainably equivalent, but not explanatory equivalent. That is, every observation explainable in (B_1, H_1) is also explainable in (B_2, H_2), and vice versa. On the other hand, the observation $O =$ *shoes_are_wet* has the explanation $E = \emptyset$ in (B_1, H_1), but E does not explain O in (B_2, H_2).

Thus, two equivalence relations compare explanation power of abductive theories in different ways. Inoue and Sakama [9] provide necessary and sufficient conditions for each equivalence relation. In the following, $Th(\Sigma)$ denotes the set of logical consequences of a set Σ of first-order formulas.

Definition 2.4 *(extension).* Let (B, H) be an abductive theory. An *extension* of (B, H) is defined as $Th(B \cup S)$ where S is a maximal subset of H such that $B \cup S$ is consistent. The set of all extensions of (B, H) is denoted by $Ext(B, H)$.

Theorem 2.1. *(See [9].) Let (B_1, H_1) and (B_2, H_2) be two abductive theories. Then,*

(1) *(B_1, H_1) and (B_2, H_2) are explainably equivalent iff $Ext(B_1, H_1) = Ext(B_2, H_2)$.*

[1] This type of induction is called *explanatory induction* [4].

(2) *(B_1, H_1) and (B_2, H_2) are explanatorily equivalent iff $B_1 \equiv B_2$ and $H'_1 = H'_2$ where $H'_i = \{h \in H_i \mid B_i \cup \{h\}$ is consistent$\}$ for $i = 1, 2$.*

Since any element in $H_i \setminus H'_i$ is of no use for explaining observations, two abductive theories are assumed to have a common hypothesis set H for explanatory equivalence [9].

The next theorem states the computational complexity of each equivalence problem.

Theorem 2.2. *(See [9].) The following complexity results holds with respect to the size of background theories and candidate hypotheses.*

(1) *Deciding explainable equivalence of two propositional abductive theories is Π_2^P-complete.*
(2) *Deciding explanatory equivalence of two propositional abductive theories is coNP-complete.*

2.2. Abductive logic programming

Next, we consider the case that the underlying logic is *abductive logic programming* (ALP) [1]. In contrast to first-order logic, in ALP the background theory and a hypothesis are given as *nonmonotonic logic programs* in general. We first review definitions of basic notions.

A logic program considered in this paper is the class of *general extended disjunctive program* (GEDP) [7], which is a set of rules of the form:

$$L_1; \cdots; L_k; not L_{k+1}; \cdots; not L_l$$
$$\leftarrow L_{l+1}, \ldots, L_m, not L_{m+1}, \ldots, not L_n \quad (n \geqslant m \geqslant l \geqslant k \geqslant 0)$$

where each L_i is a positive/negative literal, i.e., A or $\neg A$ for an atom A, and *not* is *negation as failure*. *notL* is called an *NAF-literal*. The left-hand side of $\leftarrow$ is the *head*, and the right-hand side is the *body*. A semicolon ";" in the head represents disjunction, and a comma "," in the body represents conjunction. The rule is read as: 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 $(1 \leqslant i \leqslant k)$ is believed or some L_j $(k+1 \leqslant j \leqslant l)$ is disbelieved.

For each rule r of the above form, $head^+(r)$, $head^-(r)$, $body^+(r)$ and $body^-(r)$ denote the sets of literals $\{L_1, \ldots, L_k\}$, $\{L_{k+1}, \ldots, L_l\}$, $\{L_{l+1}, \ldots, L_m\}$, and $\{L_{m+1}, \ldots, L_n\}$, respectively. Also, $not\_head^-(r)$ and $not\_body^-(r)$ denote the sets of NAF-literals $\{not L_{k+1}, \ldots, not L_l\}$ and $\{not L_{m+1}, \ldots, not L_n\}$, respectively. A disjunction or conjunction of (NAF-)literals in a rule is identified with its corresponding sets of (NAF-)literals. A rule r is often written as

$$head^+(r); not\_head^-(r) \leftarrow body^+(r), not\_body^-(r)$$

or $head(r) \leftarrow body(r)$ where $head(r) = head^+(r) \cup not\_head^-(r)$ and $body(r) = body^+(r) \cup not\_body^-(r)$. A rule $L \leftarrow$ is identified with the literal L. A program P is *basic* if $head^-(r) = body^-(r) = \emptyset$ for every rule r in P. A program P is an *extended disjunctive program* (EDP) if $head^-(r) = \emptyset$ for every rule r in P. A program, rule, or literal is *ground* if it contains no variable. The domain of a program is given as the *Herbrand universe*, the set of all ground terms in the language. A program P with variables is a shorthand of its *ground instantiation*, denoted as $Ground(P)$, the (possibly infinite) set of ground rules obtained from P by substituting variables in P by elements of its Herbrand universe in every possible way.

The semantics of a GEDP is given by the *answer set semantics* [5,7]. Let Lit be the set of all ground literals in the language of a program. Consider a program P and a set of literals $S \subseteq Lit$. Then, the *reduct* P^S is the program which contains the ground rule $head^+(r) \leftarrow body^+(r)$ iff there is a rule r in $Ground(P)$ such that $head^-(r) \setminus S = \emptyset$ and $body^-(r) \cap S = \emptyset$. Given a basic program P, let S be a set of ground literals satisfying the conditions:

(1) S satisfies every rule in P, that is, for any ground rule $head^+(r) \leftarrow body^+(r)$ in $Ground(P)$, $body^+(r) \subseteq S$ implies $head^+(r) \cap S \neq \emptyset$; and
(2) if S contains a pair of complementary literals L and $\neg L$, then $S = Lit$.

An *answer set* of a basic program P is a minimal set S satisfying the above two conditions. Given a GEDP P and a set S of literals, S is an *answer set* of P if S is an answer set of P^S. A program has none, one, or multiple answer sets in general. An answer set is *consistent* if it is not Lit. A program is *consistent* if it has a consistent answer set; otherwise it is inconsistent. The set of all answer sets of a program P is denoted by $AS(P)$.

Two programs P_1 and P_2 are *equivalent* if $AS(P_1) = AS(P_2)$.

A literal L is a consequence of *skeptical reasoning* (resp. *credulous reasoning*) in a program P if L is included in every (resp. some) answer set of P. The set of consequences of skeptical reasoning (resp. credulous reasoning) in P is denoted as $skp(P)$ (resp. $crd(P)$). For a consistent program P, it holds that

$$skp(P) = \bigcap_{S \in AS(P)} S \quad \text{and} \quad crd(P) = \bigcup_{S \in AS(P)} S.$$

When P is inconsistent, it holds that $skp(P) = crd(P) = Lit$ if $AS(P) = \{Lit\}$; and $skp(P) = Lit$ and $crd(P) = \emptyset$ if $AS(P) = \emptyset$. Clearly, $skp(P) \subseteq crd(P)$ holds for any consistent program P.

Example 2.2. Consider two programs:

$$\begin{aligned} P_1: \quad & p \leftarrow not\ q, \\ & q \leftarrow not\ p. \\ P_2: \quad & p \leftarrow q, \\ & q; not\ q \leftarrow \end{aligned}$$

where $AS(P_1) = \{\{p\}, \{q\}\}$ and $AS(P_2) = \{\emptyset, \{p, q\}\}$. Then, $skp(P_1) = skp(P_2) = \emptyset$ and $crd(P_1) = crd(P_2) = \{p, q\}$.

Answer sets have the following property.

Proposition 2.3. *(See [5].) If a program P is an EDP, $AS(P)$ is an antichain, that is, no element $S \in AS(P)$ is a proper subset of another element $T \in AS(P)$.*

The above proposition does not hold for GEDPs in general (see Example 2.2).
The next proposition is used later.

Proposition 2.4. *Let P be a basic program. For any $L \in Lit$, $L \in skp(P)$ iff $skp(P) = skp(P \cup \{L\})$.*

Proof. Suppose that P is consistent and $L \in skp(P)$. Then, L is included in any answer set S of P. Since P is a basic program, S is a minimal set satisfying P. As $L \in S$, S is a minimal set satisfying $P \cup \{L\}$. Thus, S is an answer set of $P \cup \{L\}$. On the other hand, any answer set T of $P \cup \{L\}$ contains L. As T is a minimal set satisfying $P \cup \{L\}$, T satisfies P. If T is not minimal, there is a minimal set $T' \subset T$ satisfying P. Then, T' is an answer set of P, and becomes an answer set of $P \cup \{L\}$. This contradicts the antichain property of Proposition 2.3. Thus, T is a minimal set satisfying P, so T is an answer set of P. Hence, $AS(P) = AS(P \cup \{L\})$, thereby $skp(P) = skp(P \cup \{L\})$. Conversely, suppose $skp(P) = skp(P \cup \{L\})$. Clearly $L \in skp(P \cup \{L\})$, thereby $L \in skp(P)$.

Next, suppose that P is inconsistent. By the definition, $skp(P) = Lit$. When $AS(P) = \{Lit\}$, any set S satisfying every rule in P contains a pair of complementary literals. Then, any set S satisfying every rule in $P \cup \{L\}$ also contains a pair of complementary literals. Thus, $AS(P \cup \{L\}) = \{Lit\}$ and $skp(P \cup \{L\}) = Lit$. When $AS(P) = \emptyset$, no set satisfies every rule in P. Since P is basic, there is also no set satisfying every rule in $P \cup \{L\}$. Thus, $AS(P \cup \{L\}) = \emptyset$ and $skp(P \cup \{L\}) = Lit$. Hence, the result holds. □

Definition 2.5 *(abductive program).* An *abductive program* is defined as a pair $\langle P, \mathcal{A} \rangle$ where P and $\mathcal{A}$ are GEDPs representing the background theory and a candidate hypothesis, respectively. In particular, an abductive program $\langle P, \mathcal{A} \rangle$ is called an *abductive EDP* if both P and $\mathcal{A}$ are EDPs. An abductive program $\langle P, \mathcal{A} \rangle$ is called an *abductive definite program* if both P and $\mathcal{A}$ are definite logic programs, i.e., sets of definite Horn clauses. Any instance of an element in $\mathcal{A}$ is called an *abducible*. An abductive program $\langle P, \mathcal{A} \rangle$ is called *propositional* if both P and $\mathcal{A}$ are finite ground programs.

In the literature of abductive logic programming, abducibles are usually restricted to (ground) literals. Any abductive program $\langle P, \mathcal{A} \rangle$ which contains rules in $\mathcal{A}$ is transformed to a semantically equivalent abductive program in which abducibles contain only literals. Given an abductive program $\langle P, \mathcal{A} \rangle$, let

$$P' = P \cup \bigl\{head(r) \leftarrow body(r), A_r \mid r \in \mathcal{A}\bigr\},$$

$$\mathcal{A}' = \{A_r \mid r \in \mathcal{A}\},$$

where A_r is a newly introduced atom (called the *name* of r) uniquely associated with each rule r in $\mathcal{A}$. With this setting, for any observation in the language of $\langle P, \mathcal{A} \rangle$, there is a 1–1 correspondence between explanations in $\langle P, \mathcal{A} \rangle$ and those in $\langle P', \mathcal{A}' \rangle$.[2]

Definition 2.6 *(belief sets).* Let $\langle P, \mathcal{A} \rangle$ be an abductive program. For any $E \subseteq \mathcal{A}$, a consistent answer set S of $P \cup E$ is called a *belief set* of $\langle P, \mathcal{A} \rangle$ (with respect to E). A belief set is often denoted as S_E when S is a belief set with respect to E. The set of all belief sets of $\langle P, \mathcal{A} \rangle$ is represented as $BS(P, \mathcal{A})$.

Note that belief sets coincide with answer sets when $\mathcal{A} = \emptyset$. That is, $BS(P, \emptyset) = AS(P)$.

We define an *observation* O as a conjunction of ground literals. O is identified with the set of ground literals included in it. We assume that O is consistent, i.e., $L \in O$ implies $\neg L \notin O$ for any $L \in Lit$.

[2] The *naming* technique is introduced by Poole [15] in the context of default reasoning.

Definition 2.7 *(credulous and skeptical explanation).* Let $\langle P, \mathcal{A} \rangle$ be an abductive program and O an observation. A set $E \subseteq \mathcal{A}$ is a *credulous explanation* of O in $\langle P, \mathcal{A} \rangle$ if $O \subseteq S_E$ for some belief set S_E of $\langle P, \mathcal{A} \rangle$. A set $E \subseteq \mathcal{A}$ is a *skeptical explanation* of O in $\langle P, \mathcal{A} \rangle$ if $O \subseteq S_E$ for every belief set S_E of $\langle P, \mathcal{A} \rangle$.

A credulous or skeptical explanation is called *ground* if it contains no variable. Any skeptical explanation is also a credulous explanation, but not vice versa. Abduction for credulous explanations (resp. skeptical explanations) is also called *credulous abduction* (resp. *skeptical abduction*).

Example 2.3. Let $\langle P, \mathcal{A} \rangle$ be an abductive program such that

$$\begin{aligned} P:\ & \mathit{watch\text{-}TV}\,;\, \mathit{sleeping} \leftarrow \mathit{holiday}, \mathit{not}\ \mathit{busy}, \\ & \mathit{working} \leftarrow \mathit{holiday},\ \mathit{busy}, \\ & \mathit{holiday} \leftarrow . \\ \mathcal{A}:\ & \mathit{busy}. \end{aligned}$$

The observation $O_1 = \mathit{watch\text{-}TV}$ has the empty set $E_1 = \emptyset$ as the credulous explanation, but has no skeptical explanation. The observation $O_2 = \mathit{working}$ has the credulous and skeptical explanation $E_2 = \{\mathit{busy}\}$.

Explainable and explanatory equivalence relations are introduced in the context of abductive logic programming.

Definition 2.8 *(explainable and explanatory equivalence in ALP).* Two abductive programs $\langle P_1, \mathcal{A}_1 \rangle$ and $\langle P_2, \mathcal{A}_2 \rangle$ are *explainably equivalent* in credulous (resp. skeptical) abduction if, for any observation O, there is a credulous (resp. skeptical) explanation of O in $\langle P_1, \mathcal{A}_1 \rangle$ iff there is a credulous (resp. skeptical) explanation of O in $\langle P_2, \mathcal{A}_2 \rangle$. Two abductive programs $\langle P_1, \mathcal{A}_1 \rangle$ and $\langle P_2, \mathcal{A}_2 \rangle$ are *explanatorily equivalent* in credulous (resp. skeptical) abduction if, for any observation O, there is a credulous (resp. skeptical) explanation E of O in $\langle P_1, \mathcal{A}_1 \rangle$ iff there is a credulous (resp. skeptical) explanation E of O in $\langle P_2, \mathcal{A}_2 \rangle$.

In [9,11] necessary and sufficient conditions for explainable and explanatory equivalence in credulous abduction are given. We review those results below, together with new results.[3]

Theorem 2.5. *Let $\langle P_1, \mathcal{A}_1 \rangle$ and $\langle P_2, \mathcal{A}_2 \rangle$ be two abductive programs, and $\mathcal{C}_i = \{E \subseteq \mathcal{A}_i \mid P_i \cup E$ is consistent$\}$ for $i = 1, 2$. Then, the following results hold.*

(1) *$\langle P_1, \mathcal{A}_1 \rangle$ and $\langle P_2, \mathcal{A}_2 \rangle$ are explainably equivalent in credulous abduction iff*

$$\max\bigl(BS(P_1, \mathcal{A}_1)\bigr) = \max(BS(P_2, \mathcal{A}_2))$$

where $\max(X) = \{x \in X \mid \neg\exists y \in X$ *such that* $x \subset y\}$.

(2) *If $\langle P_1, \mathcal{A}_1 \rangle$ and $\langle P_2, \mathcal{A}_2 \rangle$ are explainably equivalent in skeptical abduction, then*

$$\bigcup_{E \subseteq \mathcal{A}_1} \Bigl(\bigcap_{S \in AS(P_1 \cup E)} S \Bigr) = \bigcup_{F \subseteq \mathcal{A}_2} \Bigl(\bigcap_{T \in AS(P_2 \cup F)} T \Bigr).$$

(3) *$\langle P_1, \mathcal{A}_1 \rangle$ and $\langle P_2, \mathcal{A}_2 \rangle$ are explanatorily equivalent in credulous abduction iff $\mathcal{C}_1 = \mathcal{C}_2$ and for any $E \subseteq \mathcal{C}_1$,*

$$\max\bigl(AS(P_1 \cup E)\bigr) = \max\bigl(AS(P_2 \cup E)\bigr).$$

(4) *$\langle P_1, \mathcal{A}_1 \rangle$ and $\langle P_2, \mathcal{A}_2 \rangle$ are explanatorily equivalent in skeptical abduction iff $\mathcal{C}_1 = \mathcal{C}_2$ and for any $E \subseteq \mathcal{C}_1$,*

$$\bigcap_{S \in AS(P_1 \cup E)} S = \bigcap_{T \in AS(P_2 \cup E)} T.$$

Proof. The results of (1) and (3) are due to [11]. Here we show (2) and (4).

(2) Put $\Omega_1 = \bigcup_{E \subseteq \mathcal{A}_1} (\bigcap_{S \in AS(P_1 \cup E)} S)$ and $\Omega_2 = \bigcup_{F \subseteq \mathcal{A}_2} (\bigcap_{T \in AS(P_2 \cup F)} T)$. Suppose that $\Omega_1 \neq \Omega_2$. In this case, there is a literal L in $(\Omega_1 \setminus \Omega_2) \cup (\Omega_2 \setminus \Omega_1)$. Without loss of generality, let $L \in \Omega_1 \setminus \Omega_2$. Then, $L \in \bigcap_{S \in AS(P_1 \cup E)} S$ for some $E \subseteq \mathcal{A}_1$, but $L \notin \bigcap_{T \in AS(P_2 \cup F)} T$ for any $F \subseteq \mathcal{A}_2$. Then, $O = L$ is explainable in P_1 but unexplainable in P_2. This contradicts the assumption that $\langle P_1, \mathcal{A}_1 \rangle$ and $\langle P_2, \mathcal{A}_2 \rangle$ are explainably equivalent. Hence, the result holds.

(4) $\langle P_1, \mathcal{A}_1 \rangle$ and $\langle P_2, \mathcal{A}_2 \rangle$ are explanatorily equivalent in skeptical abduction $\Leftrightarrow$ for any observation O, E is a skeptical explanation of O in $\langle P_1, \mathcal{A}_1 \rangle$ iff E is a skeptical explanation of O in $\langle P_2, \mathcal{A}_2 \rangle$ $\Leftrightarrow$ for any observation O, O is included in

[3] In [9] conditions are given for observations that consist of a single literal. The result is generalized in [11] to observations that are conjunctions of literals.

any consistent answer set of $P_1 \cup E$ with $E \subseteq \mathcal{A}_1$ iff O is included in any consistent answer set of $P_2 \cup E$ with $E \subseteq \mathcal{A}_2 \Leftrightarrow \mathcal{C}_1 = \mathcal{C}_2$ and for any $E \subseteq \mathcal{C}_1$, $\bigcap_{S \in AS(P_1 \cup E)} S = \bigcap_{T \in AS(P_2 \cup E)} T$. $\square$

Theorem 2.5(2) provides a necessary condition for explainable equivalence in skeptical abduction. Since explanatory equivalence implies explainable equivalence, the condition of (4) is a sufficient condition for explainable equivalence in skeptical abduction.

Theorem 2.6. *The following complexity results hold with respect to the size of background theories and candidate hypotheses.*

(1) *Deciding explainable equivalence of two propositional abductive programs is Π_2^P-hard in both credulous and skeptical abduction.*
(2) *Deciding explanatory equivalence of two propositional abductive programs is Π_2^P-hard in both credulous and skeptical abduction.*

Proof. (1) The problem contains a special case that a program P is an EDP and $\mathcal{A}$ is empty. In this case, $\max(BS(P, \mathcal{A})) = AS(P)$, so the problem in credulous abduction reduces to deciding the equivalence relation $AS(P_1) = AS(P_2)$ of two EDPs (by Theorem 2.5(1)). The task is Π_P^2-complete [14], hence the result holds. To see the result in skeptical abduction, the problem contains a special case that P is a basic program and $\mathcal{A}$ is empty. In this case, the problem reduces to deciding equivalence of skeptical consequences between two background programs (by Theorem 2.5(2)(4)). By Proposition 2.4, skeptical reasoning of a literal in a basic program can be transformed to the problem of deciding $skp(P_1) = skp(P_2)$ in $O(1)$. Because deciding whether a literal is a skeptical consequence of a basic program is Π_2^P-complete [2], the decision problem of $skp(P_1) = skp(P_2)$ is Π_2^P-hard. Hence, the result holds. (2) Consider again a special case that P is an EDP and $\mathcal{A}$ is empty. In this case, $\max(AS(P \cup E)) = AS(P)$, so the problem in credulous abduction reduces to deciding the equivalence relation $AS(P_1) = AS(P_2)$ of two EDPs (by Theorem 2.5(3)). The task is Π_P^2-complete and the result holds. The proof of skeptical abduction is similar to the proof of (1). $\square$

3. Equivalence of explanations

Next we turn to the problem of equivalence of explanations.

Definition 3.1 *(equivalent explanation).* Let (B, H) be an abductive theory. For any observation O, suppose that E_1 is an explanation of O in (B, H) iff E_2 is an explanation of O in (B, H). In this case, E_1 and E_2 are *equivalent explanations*.

The notion of equivalent explanations provides a method for identifying different explanations which are abduced for an arbitrary observation in a background theory. The next result holds for first-order abduction.

Theorem 3.1. *Let (B, H) be an abductive theory. Then, two explanations E_1 and E_2 are equivalent iff $B \cup E_1 \equiv B \cup E_2$.*

Proof. E_1 and E_2 are equivalent iff $B \cup E_1 \models O \Leftrightarrow B \cup E_2 \models O$ for any formula O iff $B \cup E_1 \equiv B \cup E_2$. $\square$

The above theorem presents that different formulas E_1 and E_2 can become an equivalent explanation depending on the context of B.

Example 3.1. Given $B = \{p \supset q, q \supset p, p \wedge q \supset r\}$ and $H = \{p, q\}$, $E_1 = \{p\}$, $E_2 = \{q\}$, and $E_3 = \{p, q\}$ are all equivalent explanations.

In the context of abductive logic programming, the notion of equivalent explanations is defined for credulous and skeptical abduction.

Definition 3.2 *(equivalent explanation in ALP).* Let $\langle P, \mathcal{A} \rangle$ be an abductive program. For any observation O, suppose that E_1 is a credulous (resp. skeptical) explanation of O in $\langle P, \mathcal{A} \rangle$ iff E_2 is a credulous (resp. skeptical) explanation of O in $\langle P, \mathcal{A} \rangle$. In this case, E_1 and E_2 are *equivalent explanations* in credulous (resp. skeptical) abduction.

Proposition 3.2. *Let $\langle P, \mathcal{A} \rangle$ be an abductive program.*

(1) *If two explanations E_1 and E_2 are equivalent in credulous abduction, $crd(P \cup E_1) = crd(P \cup E_2)$.*
(2) *Two explanations E_1 and E_2 are equivalent in skeptical abduction iff $skp(P \cup E_1) = skp(P \cup E_2)$.*

Proof. (1) If two explanations E_1 and E_2 are equivalent in credulous abduction, for any observation O, $O \subseteq S$ for some consistent answer set S of $P \cup E_1$ iff $O \subseteq T$ for some consistent answer set T of $P \cup E_2$. Put $O = S$. Then, for any consistent answer set S of $P \cup E_1$, there is a consistent answer set T of $P \cup E_2$ such that $S \subseteq T$. Likewise, putting $O = T$, for any consistent answer set T of $P \cup E_2$, there is a consistent answer set S of $P \cup E_1$ such that $T \subseteq S$. Hence, $crd(P \cup E_1) = crd(P \cup E_2)$ holds. (2) If two explanations E_1 and E_2 are equivalent in skeptical abduction, for any observation O, $O \subseteq S$ for every consistent answer set S of $P \cup E_1$ iff $O \subseteq T$ for every consistent answer set T of $P \cup E_2$. Hence, $skp(P \cup E_1) = skp(P \cup E_2)$. Conversely, if $skp(P \cup E_1) = skp(P \cup E_2)$, for any observation O, O is included in every answer set S of $P \cup E_1$ iff O is included in every answer set S of $P \cup E_2$. Hence, the result follows. □

The converse of Proposition 3.2(1) does not hold in general.

Example 3.2. Let $\langle P, \mathcal{A} \rangle$ be an abductive program such that

$$\begin{aligned} P:\quad & p \leftarrow r, not\ q, \\ & q \leftarrow s, not\ p, \\ & p \leftarrow r, s, \\ & q \leftarrow r, s. \\ \mathcal{A}:\quad & r; s \leftarrow, \\ & r \leftarrow, \\ & s \leftarrow. \end{aligned}$$

For $E_1 = \{r; s \leftarrow\}$ and $E_2 = \{r \leftarrow, s \leftarrow\}$, it holds that $crd(P \cup E_1) = crd(P \cup E_2) = \{p, q, r, s\}$. However, E_1 and E_2 are not equivalent in credulous abduction, since $O = r, s$ is explained by E_2 but not by E_1.

The next theorem characterizes equivalence of explanations.

Theorem 3.3. *Let $\langle P, \mathcal{A} \rangle$ be an abductive program. Then, two explanations E_1 and E_2 are equivalent (in both credulous and skeptical abduction) if $AS(P \cup E_1) = AS(P \cup E_2)$ where $P \cup E_1$ and $P \cup E_2$ are consistent. The only-if part also holds for credulous abduction if $\langle P, \mathcal{A} \rangle$ is an abductive EDP.*

Proof. The if part is obvious for both skeptical and credulous abduction. We show the only-if part for credulous abduction. Let P and $\mathcal{A}$ be EDPs. Suppose that $AS(P \cup E_1) \neq AS(P \cup E_2)$ and there is a consistent answer set S such that $S \in AS(P \cup E_1) \setminus AS(P \cup E_2)$. If $S \not\subset T_i$ for any $T_i \in AS(P \cup E_2)$, there is a literal $L \in S \setminus T_i$ for any T_i. Let U be a finite set such that $U \subseteq \bigcup_i (S \setminus T_i)$. Then, $U \subseteq S$ but $U \not\subseteq T_i$ for any T_i. Thus, E_1 explains U but E_2 does not. This contradicts the equivalence assumption of E_1 and E_2. So, $S \subset T$ holds for some $T \in AS(P \cup E_2)$ (†). Since $AS(P \cup E_1)$ is an antichain, $T \notin AS(P \cup E_1)$. Then, $T \in AS(P \cup E_2) \setminus AS(P \cup E_1)$. Repeating the same argument as above, it is shown that $T \subset S'$ holds for some $S' \in AS(P \cup E_1)$. By (†), $S \subset S'$ holds for two answer sets S and S' of $P \cup E_1$. But this is impossible, since $AS(P \cup E_1)$ is an antichain (Proposition 2.3). □

The only-if part of Theorem 3.3 does not hold for abductive GEDPs in general.

Example 3.3. Let $\langle P, \mathcal{A} \rangle$ be an abductive program such that

$$\begin{aligned} P:\quad & p; not\ p \leftarrow . \\ \mathcal{A}:\quad & p. \end{aligned}$$

Then, $E_1 = \emptyset$ and $E_2 = \{p\}$ are equivalent in credulous abduction, but $AS(P \cup E_1) = \{\emptyset, \{p\}\}$ and $AS(P \cup E_2) = \{\{p\}\}$ are different.

In skeptical abduction, the condition $AS(P \cup E_1) = AS(P \cup E_2)$ is not necessary for the equivalence of explanations. For instance, in the abductive program $\langle P, \mathcal{A} \rangle$ where $P = \emptyset$ and $\mathcal{A} = \{p; q \leftarrow, r; s \leftarrow\}$, two skeptical explanations $E_1 = \{p; q \leftarrow\}$ and $E_2 = \{r; s \leftarrow\}$ are equivalent but $AS(P \cup E_1) \neq AS(P \cup E_2)$.

Theorem 3.4. *The following complexity results hold with respect to the size of background theories and explanations.*

(1) *Deciding equivalence of two explanations in a propositional abductive theory is coNP-complete.*
(2) *Deciding equivalence of two explanations in a propositional abductive program is Π_2^P-hard in credulous abduction. The decision problem is Π_2^P-complete for propositional abductive EDPs.*
(3) *Deciding equivalence of two explanations in a propositional abductive program is Π_2^P-hard in skeptical abduction.*
(4) *Deciding equivalence of two explanations in a propositional abductive definite program can be done in polynomial time.*

Proof. (1) As deciding logical equivalence of two propositional theories is coNP-complete, the result holds by Theorem 3.1. (2) When $\langle P, \mathcal{A} \rangle$ is an abductive EDP, $AS(P \cup E_1) = AS(P \cup E_2)$ becomes the necessary and sufficient condition (Theorem 3.3). Since testing the equivalence of two (propositional) programs $P \cup E_1$ and $P \cup E_2$ is Π_2^P-complete [14], the result holds.

When $\langle P, \mathcal{A} \rangle$ is an abductive program, the decision problem contains the case that both P and $\mathcal{A}$ are EDPs. Hence, the problem is Π_2^P-hard. (3) By Proposition3.2(2), the relation $skp(P \cup E_1) = skp(P \cup E_2)$ is necessary and sufficient. The task of deciding the equivalence of skeptical consequences is Π_2^P-hard (Theorem 2.6). (4) In definite programs, credulous and skeptical explanations coincide. Then, two ground explanations are equivalent if $P \cup E_1$ and $P \cup E_2$ have the same least model. Checking the equivalence of two definite programs is done in polynomial time, and the result holds. □

4. Equivalence of observations

We finally consider the problem of equivalence of observations.

Definition 4.1 *(equivalent observation).* Given an abductive theory (B, H), two observations O_1 and O_2 are *equivalent* if, for any $E \subseteq H$, E is an explanation of O_1 in (B, H) iff E is an explanation of O_2 in (B, H).

The notion of equivalent observations provides a method for identifying different evidences.

Theorem 4.1. *Let (B, H) be an abductive theory. Then, two observations O_1 and O_2 are equivalent iff $B \models O_1 \equiv O_2$.*

Proof. O_1 and O_2 are equivalent iff $B \cup E \models O_1 \Leftrightarrow B \cup E \models O_2$ for any $E \subseteq H$ such that $B \cup E$ is consistent iff $B \cup E \models O_1 \equiv O_2$ for any $E \subseteq H$ such that $B \cup E$ is consistent. Putting $E = \emptyset$, $B \models O_1 \equiv O_2$. When $B \models O_1 \equiv O_2$, $B \cup E \models O_1 \equiv O_2$ holds for any $E \subseteq H$ such that $B \cup E$ is consistent. Hence, the result holds. □

The result shows that equivalence of observations depends on the background theory but is independent of a candidate hypothesis.

Example 4.1. Given $(B_1, H_1) = (\{p \supset q\}, \{p, q\})$, $O_1 = p$ and $O_2 = p \wedge q$ are equivalent. On the other hand, O_1 and O_2 are not equivalent in $(B_2, H_2) = (\{p \vee q\}, \{p, q\})$.

In the above example, the equivalence of O_1 and O_2 implies that the additional evidence q in O_2 has no effect on constructing explanations in (B_1, H_1).

An equivalence relation between observations is defined for abductive logic programming in both credulous and skeptical abduction.

Definition 4.2 *(equivalent observation in ALP).* Given an abductive program $\langle P, \mathcal{A} \rangle$, two observations O_1 and O_2 are *equivalent* in credulous (resp. skeptical) abduction if, for any $E \subseteq \mathcal{A}$, E is a credulous (resp. skeptical) explanation of O_1 in $\langle P, \mathcal{A} \rangle$ iff E is a credulous (resp. skeptical) explanation of O_2 in $\langle P, \mathcal{A} \rangle$.

For equivalence of observations, however, we have no simple characterization.

Theorem 4.2. *Let $\langle P, \mathcal{A} \rangle$ be an abductive program.*

(1) *Two observations O_1 and O_2 are equivalent in credulous abduction iff for any $E \subseteq \mathcal{A}$ such that $P \cup E$ is consistent, if $O_1 \subseteq S$ for some $S \in AS(P \cup E)$, $O_2 \subseteq T$ for some $T \in AS(P \cup E)$, and vice versa.*
(2) *Two observations O_1 and O_2 are equivalent in skeptical abduction iff for any $E \subseteq \mathcal{A}$ such that $P \cup E$ is consistent, if $O_1 \subseteq S$ for every $S \in AS(P \cup E)$, $O_2 \subseteq S$ for every $S \in AS(P \cup E)$, and vice versa.*

Proof. In credulous (resp. skeptical) abduction, O_1 and O_2 are equivalent iff for any $E \subseteq \mathcal{A}$ such that $P \cup E$ is consistent, the fact that O_1 is included in some (resp. every) consistent answer set of $P \cup E$ implies that O_2 is included in some (resp. every) consistent answer set of $P \cup E$, and vice versa. Hence, the result holds. □

Example 4.2. Let $\langle P, \mathcal{A} \rangle$ be an abductive program such that

$$\begin{aligned} P:\quad & win \leftarrow lottery, not \neg win, \\ & \neg win \leftarrow lottery,\ not\ win. \\ \mathcal{A}:\quad & lottery. \end{aligned}$$

Then, $O_1 = win$ and $O_2 = \neg win$ are equivalent in both credulous and skeptical abduction.

In the above example, the equivalence of O_1 and O_2 presents a situation that *win* or $\neg$*win* could equally happen on the same ground *lottery*.

For complexity, we have the next results.

Theorem 4.3. *The following complexity results hold with respect to the size of background theories and candidate hypotheses.*

(1) *Deciding equivalence of two observations in a propositional abductive theory is coNP-complete.*
(2) *Deciding equivalence of two observations in a propositional abductive program is Σ_2^P-hard in credulous abduction and Π_2^P-hard in skeptical abduction.*

Proof. (1) The problem is equivalent to checking unsatisfiability of $B \cup \{O_1 \not\equiv O_2\}$ (by Theorem 4.1), which is a task of coNP-complete. (2) The problem contains the case that the abducibles are empty. Given two observations O_1 and O_2, consider the following two rules: $G_1 \leftarrow O_1$ and $G_2 \leftarrow O_2$, where G_1 and G_2 are new atoms appearing nowhere in $\langle P, \emptyset \rangle$. Let P' be the program which is obtained from P by adding these rules. Then, O_1 and O_2 are equivalent in credulous abduction iff $G_1 \in S$ for some consistent $S \in AS(P')$ implies $G_2 \in T$ for some consistent $T \in AS(P')$, and vice versa. The task of checking whether a literal is included in some answer set of a program is Σ_2^P-complete [7]. Hence, the Σ_2^P-hardness result holds. In case of skeptical abduction, O_1 and O_2 are equivalent iff $G_1 \in S \Leftrightarrow G_2 \in S$ for any consistent $S \in AS(P')$. The task of checking whether a literal is included in every answer set of a program is Π_2^P-complete [7]. Hence, the Π_2^P-hardness result holds. □

5. Discussion

In this section, we compare conditions in different equivalence issues. First, recall the problem of equivalence of abductive theories. In first-order abduction, explainable equivalence requires the equivalence of extensions, while explanatory equivalence requires the logical equivalence of two abductive theories (Theorem 2.1). Explanatory equivalence is stronger than explainable equivalence, while the task of deciding explanatory equivalence is not harder than the task of deciding explainable equivalence (Theorem 2.2). In abductive logic programming, explainable and explanatory equivalence is checked by comparing maximal elements of belief sets or answer sets of programs in credulous abduction. On the other hand, skeptical abduction requires comparison of intersections of all belief sets (Theorem 2.5). Since belief sets are computable using proof procedures of answer set programming, checking these equivalence relations is done with existing answer set solvers. Computational cost of checking each equivalence is generally expensive (Theorem 2.6). As a special case, however, explainable equivalence of two *definite* abductive programs, which has the background theory and a candidate hypothesis as definite logic programs, can be decided in polynomial time [9]. This would be good news for existing ILP systems in which background theories are usually given as definite logic programs.

Second, consider the problem of equivalence of explanations. In first-order abduction, the problem is identical to judging logical equivalence of two theories (Theorem 3.1). In abductive logic programming, the problem is identical to testing the equivalence of two programs for abductive EDPs (Theorem 3.3). Checking equivalence of explanations in abductive logic programming is generally harder than first-order abduction (Theorem 3.4). In the problem of equivalence of observations, it is identical to testing the logical equivalence of observations under an abductive theory in first-order abduction (Theorem 4.1). In abductive logic programming, comparison of skeptical (or credulous) consequences is requested (Theorem 4.2). It is observed that abductive logic programming is again harder than first-order abduction in general (Theorem 4.3). The complexity results are summarized in Table 1.

Comparing first-order abduction and abductive logic programming, logical equivalence characterizes each problem in first-order abduction. In abductive logic programming, on the other hand, different types of equivalence notions are used in different problems. What makes comparison of abductive programs more complicated is *nonmonotonicity* in abductive logic programming, which also makes computational task of equivalence testing harder than first-order abduction in general.

The results of this paper also have an important implication in program development in abductive and inductive logic programming. For instance, it is known that *partial deduction* in logic programming does not preserve explanations in abductive logic programming [17]. Consider the abductive program $\langle P, \mathcal{A} \rangle$:

$$\begin{aligned} P:\quad & shoes\_are\_wet \leftarrow grass\_is\_wet, \\ & grass\_is\_wet \leftarrow rained\_last\_night, \\ & grass\_is\_wet \leftarrow sprinkler\_was\_on. \\ \mathcal{A}:\quad & grass\_is\_wet. \end{aligned}$$

Table 1
Computational complexity.

Logic	Abductive theories (explainable/explanatory)	Explanations	Observations
FOL (propositional)	Π_2^P-complete/coNP-complete	coNP-complete	coNP-complete
ALP (credulous)	Π_2^P-hard	Π_2^P-hard[a]	Σ_2^P-hard
(skeptical)	Π_2^P-hard	Π_2^P-hard	Π_2^P-hard

[a] Completeness holds for (propositional) EDPs.

In $\langle P, \mathcal{A} \rangle$, the observation $O = shoes\_are\_wet$ has the explanation $E = \{grass\_is\_wet\}$. Performing partial deduction on the atom *grass_is_wet*, however, P becomes

$$\begin{aligned} P'\colon\ & shoes\_are\_wet \leftarrow rained\_last\_night, \\ & shoes\_are\_wet \leftarrow sprinkler\_was\_on, \\ & grass\_is\_wet \leftarrow rained\_last\_night, \\ & grass\_is\_wet \leftarrow sprinkler\_was\_on. \end{aligned}$$

As a result, in the abductive program $\langle P', \mathcal{A} \rangle$, the observation $O = shoes\_are\_wet$ has no explanation. Thus, both explainable and explanatory equivalences are not preserved by partial deduction. This example illustrates that basic program transformations widely used in program development in logic programming are not always applicable to optimize background theories in ALP/ILP. If applied, explanations of abduction and induction may change in general. The notion of explainable and explanatory equivalence provides a condition which any reasonable optimization in ALP/ILP should satisfy. The paper [17] introduces some program transformations that preserve abductive explanations, so that they preserve explainable and explanatory equivalence. Program development and optimization issues for abductive or inductive theories have been less explored and are to be further investigated.

6. Conclusion

In this paper, we studied different types of equivalence relations in abduction and induction: explainable and explanatory equivalence of abductive theories, equivalence of explanations, and equivalence of observations. In each case, necessary and sufficient conditions for equivalence as well as computational complexity results were investigated, under both classical logic and nonmonotonic logic programming. These results shed light on the equivalence issue in non-deductive reasoning and are applied to a general hypothetico-deductive framework.

A recent study [11] introduces methods for comparing explanation power of different abductive theories—one is comparing explainability for observations, and the other is comparing explanation contents for observations. Those two measures are represented by generality relations over abductive theories. The generality relations are naturally related to the notion of abductive equivalence of [9,10]. Similar orderings over explanations or observations could be considered for comparing explanations or observations, and would be related to equivalence relations of this paper. Those topics are left for future research.

References

[1] M. Denecker, A.C. Kakas, Abductive logic programming, in: A.C. Kakas, F. Sadri (Eds.), Computational Logic: Logic Programming and Beyond—Essays in Honour of Robert A. Kowalski, Part I, in: Lecture Notes in Artificial Intelligence, vol. 2407, Springer-Verlag, 2002, pp. 402–436.
[2] T. Eiter, G. Gottlob, On the computational cost of disjunctive logic programming: propositional case, Annals of Mathematics and Artificial Intelligence 15 (1995) 289–323.
[3] H.B. Enderton, A Mathematical Introduction to Logic, Academic Press, 1972.
[4] P.A. Flach, A.C. Kakas (Eds.), Abduction and Induction—Essays on their Relation and Integration, Kluwer Academic, 2000.
[5] M. Gelfond, V. Lifschitz, Classical negation in logic programs and disjunctive databases, New Generation Computing 9 (1991) 365–385.
[6] K. Inoue, C. Sakama, Abductive framework for nonmonotonic theory change, in: Proceedings of the 14th International Joint Conference on Artificial Intelligence, Morgan Kaufmann, 1995, pp. 204–210.
[7] K. Inoue, C. Sakama, Negation as failure in the head, Journal of Logic Programming 35 (1998) 39–78.
[8] K. Inoue, C. Sakama, Equivalence of logic programs under updates, in: Proceedings of the 9th European Conference on Logics in Artificial Intelligence, in: Lecture Notes in Artificial Intelligence, vol. 3229, Springer-Verlag, 2004, pp. 174–186.
[9] K. Inoue, C. Sakama, Equivalence in abductive logic, in: Proceedings of the 19th International Joint Conference on Artificial Intelligence, 2005, pp. 472–477.
[10] K. Inoue, C. Sakama, On abductive equivalence, in: L. Magnani (Ed.), Model-Based Reasoning in Science and Engineering: Cognitive Science, Epistemology, Logic. Studies in Logic, College Publications, London, 2006, pp. 333–352.
[11] K. Inoue, C. Sakama, Comparing abductive theories, in: Proceedings of the 18th European Conference on Artificial Intelligence, IOS Press, 2008, pp. 35–39.
[12] V. Lifschitz, D. Pearce, A. Valverde, Strongly equivalent logic programs, ACM Transactions on Computational Logic 2 (2001) 526–541.
[13] M.J. Maher, Equivalence of logic programs, in: J. Minker (Ed.), Foundations of Deductive Databases and Logic Programming, Morgan Kaufmann, 1988, pp. 627–658.
[14] E. Oikarinen, T. Janhunen, Verifying the equivalence of logic programs in the disjunctive case, in: Proceedings of the 7th International Conference on Logic Programming and Nonmonotonic Reasoning, in: Lecture Notes in Artificial Intelligence, vol. 2923, Springer-Verlag, 2004, pp. 180–193.
[15] D. Poole, A logical framework for default reasoning, Artificial Intelligence 36 (1988) 27–47.
[16] D. Pearce, H. Tompits, S. Woltran, Relativised equivalence in equilibrium logic and its applications to prediction and explanation: preliminary report, in: Proceedings of the LPNMR'07 Workshop on Correspondence and Equivalence for Nonmonotonic Theories (CENT2007), Tempe, AZ, 2007, pp. 37–48.
[17] C. Sakama, K. Inoue, The effect of partial deduction in abductive reasoning, in: Proceedings of the 12th International Conference on Logic Programming, MIT Press, 1995, pp. 383–397.
[18] C. Sakama, K. Inoue, Inductive equivalence of logic programs, in: Proceedings of the 15th International Conference on Inductive Logic Programming, in: Lecture Notes in Artificial Intelligence, vol. 3625, Springer-Verlag, 2005, pp. 312–329.