Noname manuscript No.
(will be inserted by the editor)

Brave Induction: a logical framework for learning from incomplete information

Chiaki Sakama · Katsumi Inoue

Received: date / Accepted: date

Abstract This paper introduces a novel logical framework for concept-learning called *brave induction*. Brave induction uses brave inference for induction and is useful for learning from incomplete information. Brave induction is weaker than explanatory induction which is normally used in *inductive logic programming*, and is stronger than *learning from satisfiability*, a general setting of concept-learning in clausal logic. We first investigate formal properties of brave induction, then develop an algorithm for computing hypotheses in full clausal theories. Next we extend the framework to induction in *nonmonotonic logic programs*. We analyze computational complexity of decision problems for induction on propositional theories. Further, we provide problem solving by brave induction in systems biology, requirement engineering, and multiagent negotiation.

Keywords brave induction · inductive logic programming · nonmonotonic logic programming

1 Introduction

1.1 Explanatory Induction

Logical foundations for *induction* is one of the central topics in machine learning, and different theories of induction have been proposed in the literature (Plotkin (1970); Helft (1989); De Raedt and Lavrač (1993); Muggleton (1995); Inoue (2004); Sakama (2005), for instance). A typical induction task constructs hypotheses to explain an

C. Sakama
Department of Computer and Communication Sciences
Wakayama University, Sakaedani, Wakayama 640-8510 Japan
Tel.: +81-73-457-8128
Fax: +81-73-457-8128
E-mail: sakama@sys.wakayama-u.ac.jp

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

observation (or examples) using the background knowledge. More precisely, given a first-order theory B as the background knowledge and a formula O as an observation, a hypothesis H *covers* O under B if

$$B \wedge H \models O \tag{1}$$

where $B \wedge H$ is consistent. This style of induction is called *explanatory induction* (Flach 1996) or *learning from entailment* (De Raedt 1997). It is used as a normal setting in *inductive logic programming* (ILP) (Muggleton and De Raedt 1994; Nienhuys-Cheng and Wolf 1997) and is also used for induction from full clausal theories (Inoue 2004).

By the definition, explanatory induction requires that a possible solution H together with B logically entails O. In other words, O is true in *every* model of $B \wedge H$. This condition is often too strong for building possible hypotheses, however.

Example 1.1 Suppose that there are 30 students in a class, of which 20 are European, 7 are Asian, and 3 are American. The situation is represented by the background knowledge B and the observation O:

$B:\ \ student(1) \wedge \cdots \wedge student(30),$

$O:\ \ euro(1) \wedge \cdots \wedge euro(20) \wedge asia(21) \wedge \cdots \wedge asia(27) \wedge usa(28) \wedge \cdots \wedge usa(30)$

where each number represents individual students. In this case, the following clause, saying that every student is either European, Asian, or American, appears a good hypothesis:

$$H:\ \ euro(x) \vee asia(x) \vee usa(x)\ \leftarrow\ student(x)\,. \tag{2}$$

Unfortunately, however, H does not satisfy the relation $B \wedge H \models O$. In fact, $B \wedge H$ has many models in which O is not true. An instance of such a model is:

$$\{\, student(1), \ldots, student(30),\ euro(1), \ldots, euro(30)\,\}.$$

Explanatory induction in ILP has mainly been used for learning *Horn* theories. When the background knowledge B and a hypothesis H are Horn theories, the intersection of all models of $B \wedge H$ coincides with the unique minimal model (or the least model). The relation (1) then implies that O is true in the least model of $B \wedge H$. On the other hand, when B or H contains *indefinite* information, $B \wedge H$ becomes a *non-Horn* theory which has multiple minimal models in general. In this case, an observation O may be true in some minimal models of $B \wedge H$ but not every one. However, the relation (1) excludes a hypothesis H due to the existence of a (minimal) model in which O is not true. As a result, meaningful hypotheses might be unqualified as presented above.

1.2 Learning from Interpretation and Learning from Satisfiability

Explanatory induction is used for classifying observed data and predicting unseen phenomena. By contrast, *learning from interpretations* (LFI) (De Raedt 1997; De Raedt and Dehaspe 1997a) seeks regularities over observed data. In LFI observations are given as interpretations, and induction seeks hypotheses which are satisfied by observations expanded by the background knowledge. More precisely, a hypothesis H *covers* O under B in LFI iff H is true in the least model of $B \wedge O$.[1]

[1] This is a definition for a definite clause theory B (De Raedt and Dehaspe 1997a), and its extension to non-Horn theory is discussed in Section 6.2.

Example 1.2 Consider the background knowledge B and the observation O in Example 1.1, in which O is viewed as the interpretation containing atoms appearing in O. Then, the hypothesis H of (2) covers O under B in LFI. Suppose, on the other hand, that $B' = B \cup \{ student(31) \}$. Then, H does not cover O under B' in LFI as the least model of $B' \wedge O$ does not satisfy H.

In Example 1.2, when there is a student whose nationality is unknown, LFI does not infer the hypothesis H. This is because in LFI, observations are assumed to be completely specified. Thus, if complete knowledge of observations is unavailable, one should be cautious with this approach (De Raedt and Dehaspe 1997a).

Learning from satisfiability (LFS) (De Raedt 1997; De Raedt and Dehaspe 1997b) is used for concept-learning in face of incompletely specified observations. By the definition, a hypothesis H covers O under B in LFS iff $B \wedge H$ has a model satisfying O.

Example 1.3 Consider again the background knowledge B and the observation O in Example 1.1. Then, the hypothesis H of (2) covers O under B in LFS. In Example 1.2, H also covers O under B' in LFS.

Thus, learning from satisfiability can induce the hypothesis H of (2) under both B and B'. Due to its weak setting, however, the hypotheses space of LFS is generally huge. In fact, any theory H becomes a possible solution of LFS as far as it is consistent with $B \wedge O$. In Example 1.1, the following hypotheses:

$$
\begin{aligned}
H_1 &: \; euro(x) \vee student(x) \leftarrow, \\
H_2 &: \; student(x) \leftarrow euro(x) \wedge asia(x), \\
H_3 &: \; \leftarrow asia(x) \wedge usa(x),
\end{aligned}
$$

all become solutions of LFS.

1.3 Brave Inference and Cautious Inference

When $B \wedge H$ is a non-Horn theory, $B \wedge H$ has multiple minimal models in general. In this case, two different types of inferences, *brave inference* and *cautious inference*, are considered in *nonmonotonic logics* (McDermott 1982) and *disjunctive logic programs* (Eiter and Gottlob 1995). Under the *minimal model semantics*, a formula F is a consequence of brave inference in a theory T if F is true in some minimal model of T. By contrast, F is a consequence of cautious inference in T if F is true in every minimal model of T. Brave and cautious inferences are also applied to *abduction* in artificial intelligence. Given the background knowledge B, an observation O is *explained* under *brave* (resp. *cautious*) abduction if O is true in some (resp. every) minimal model of a consistent theory $B \wedge H$ (Eiter *et al.* 1997). Here, $H \subseteq A$ and A is a set of formulas representing candidate hypotheses (called *abducibles*).

Example 1.4 Suppose the background knowledge B and abducibles A:

$$
\begin{aligned}
B &: \; light\_off \leftarrow power\_failure, \\
&\quad\; light\_off \leftarrow broken\_bulb, \\
&\quad\; broken\_bulb \vee melted\_fuse \leftarrow high\_current. \\
A &: \; power\_failure, \\
&\quad\; high\_current.
\end{aligned}
$$

Given the observation $O = light\_off$, $E_1 = power\_failure$ is the unique (minimal) explanation in cautious abduction, while $E_2 = high\_current$ as well as E_1 are two (minimal) explanations in brave abduction.

Thus, brave inference is weaker than cautious inference, and it is especially useful for hypothetical reasoning as it can compute more hypotheses than cautious one.

1.4 Outline of the Paper

In this paper, we apply brave inference to induction. *Brave induction* can induce non-Horn clauses from a full clausal theory with incomplete observations. It is weaker that explanatory induction but stronger than learning from satisfiability, thus provides a reasonable compromise between the two frameworks. Using brave induction, the hypothesis (2) becomes a solution of both B of Example 1.1 and B' of Example 1.2. Brave induction is also defined for induction from *nonmonotonic* logic programs containing *default negation*. We show potential applications of brave induction for problem solving in systems biology, requirement engineering, and multiagent negotiation.

The rest of this paper is organized as follows. Section 2 introduces a logical framework of brave induction and develops a procedure for computing hypotheses. Section 3 extends the framework to induction from nonmonotonic logic programs. Section 4 analyzes computational complexity of brave induction on propositional theories. Section 5 addresses problem solving by brave induction. Section 6 discusses related issues, and Section 7 concludes the paper. This paper is a revised and extended version of (Sakama and Inoue 2008). Section 4 and Section 5 are new in this paper. Moreover, new considerations and additional arguments are added throughout the paper.

2 Brave Induction

2.1 Logical Framework

We first introduce a logical framework of induction considered in this paper. A *first-order language* $\mathcal{L}$ consists of an alphabet and all formulas defined over it. The definition is the standard one in the literature (Nienhuys-Cheng and Wolf 1997). For induction we use a *clausal language* which is a subset of $\mathcal{L}$.

A *clausal theory* (or simply a *theory*) is a finite set of *clauses* of the form:

$$A_1 \vee \cdots \vee A_m \vee \neg A_{m+1} \vee \cdots \vee \neg A_n \quad (n \geq m \geq 0)$$

where each A_i $(1 \leq i \leq n)$ is an atom. Any variable in a clause is assumed to be universally quantified at the front. A clause of the above form is also written as

$$A_1 \vee \cdots \vee A_m \leftarrow A_{m+1} \wedge \cdots \wedge A_n\,. \tag{3}$$

$A_1 \vee \cdots \vee A_m$ is the *head* of the clause, and $A_{m+1} \wedge \cdots \wedge A_n$ is the *body*. Given a clause C of the above form, $head(C)$ represents the set $\{ A_1, \ldots, A_m \}$ and $body(C)$ represents the set $\{ A_{m+1}, \ldots, A_n \}$. A clause C is often identified with the set of literals $\{ A_1, \ldots, A_m, \neg A_{m+1}, \ldots, \neg A_n \}$. A *positive clause* is a clause C with $body(C) = \emptyset$, and a *negative clause* is a clause C with $head(C) = \emptyset$. A *Horn clause* is a clause C with

$\mid head(C) \mid \leq 1$. A *Horn theory* is a finite set of Horn clauses. A theory is identified with the conjunction of the clauses in it. A theory, a clause or an atom is *ground* if it contains no variable. A *(ground) substitution* θ replaces variables $x_1, \ldots, x_k$ occurring in a clause C (resp. an atom A) to (ground) terms $t_1, \ldots, t_k$ in $C\theta$ (resp. $A\theta$). A clause C *subsumes* a clause D if there is a substitution θ such that $C\theta \subseteq D$. A *conjunctive normal form* (CNF) formula is a conjunction of disjunction of literals, and a *disjunctive normal form* (DNF) formula is a disjunction of conjunction of literals. A CNF formula or a DNF formula is *ground* if it contains no variable. A DNF formula $F = c_1 \vee \cdots \vee c_k$ is *irredundant* if $F \not\equiv F'$ for any $F' = c_1 \vee \cdots \vee c_{i-1} \vee c_{i+1} \vee \cdots \vee c_k$ $(1 \leq i \leq k)$. A conjunction C of ground atoms is identified with the set of ground atoms in C. Given a literal L, $pred(L)$ represents the predicate of L, and $term(L)$ and $const(L)$ represent the sets of terms and constants in L, respectively.

The *domain* of a theory T is given as the *Herbrand universe* HU and an *interpretation* of T is defined as a subset of the *Herbrand base* HB. An interpretation I *satisfies* a ground clause (3) if $\{ A_{m+1}, \ldots, A_n \} \subseteq I$ implies $\{ A_1, \ldots, A_m \} \cap I \neq \emptyset$. An interpretation I *satisfies* a theory T if I satisfies every clause in T. An interpretation I is a *model* of T if I satisfies T. $Mod(T)$ represents the set of all models of T. A model $M \in Mod(T)$ is *minimal* if $N \subseteq M$ implies $M \subseteq N$ for any $N \in Mod(T)$. The set of minimal models of T is written as $MM(T)$. A theory T *entails* a formula F (written as $T \models F$) if F is true in any $I \in Mod(T)$. A theory T is *consistent* if $Mod(T) \neq \emptyset$; otherwise, T is *inconsistent*.

Let B, O and H be all clausal theories, where B, O, and H are respectively called a *background knowledge*, an *observation*, and a *hypothesis*. Here, B, O, and H are sets of clauses, but each set is identified with the conjunction of clauses included in the set as usual. We assume that B, O and H have the same HU and HB. Given B and O, *explanatory induction* construct H to explain O under B. Formally, a hypothesis H *covers* O under B if

$$B \wedge H \models O \tag{4}$$

where $B \wedge H$ is consistent. H is called a *solution* of explanatory induction.

Two new frameworks of induction are introduced next.

Definition 2.1 (brave and cautious induction) Let B be the background knowledge and O an observation. A hypothesis H *covers* O under B in *brave induction* if $B \wedge H$ has a minimal model satisfying O. In this case, H is called a *solution* of brave induction.

By contrast, H *covers* O under B in *cautious induction* if $B \wedge H$ is consistent and every minimal model of $B \wedge H$ satisfies O. In this case, H is called a *solution* of cautious induction.

The names of brave and cautious induction are taken from brave and cautious inferences. Under the minimal model semantics, a formula F is a consequence of *cautious inference* in a theory T if it is true in every minimal model of T, while F is a consequence of *brave inference* in T if F is true in some minimal model of T.[2] When a theory contains indefinite or incomplete information, brave inference infers more results than cautious inference in general. Brave and cautious inferences have been used in different reasoning tasks of deduction and abduction in artificial intelligence. It is thereby natural to apply these inferences to induction from non-Horn theories containing indefinite or incomplete information.

[2] Brave and cautious inferences are also called *credulous* and *skeptical* inferences, respectively.

Relations between explanatory induction, cautious induction, and brave induction are as follows.

Proposition 2.1 *(relation among solutions) Let B be the background knowledge and O an observation.*

1. *If H is a solution of explanatory induction, H is a solution of cautious induction. The converse implication also holds when O is a set of positive clauses.*
2. *If H is a solution of cautious induction, H is a solution of brave induction. The converse implication also holds when $B \wedge H$ is a Horn theory.*

Proof (1) The if-part is obvious by the definition. To see the converse, if H is a solution of cautious induction, any positive clause C in O is satisfied in every minimal model of a consistent theory $B \wedge H$. Then, C is satisfied in every model of a consistent theory $B \wedge H$. (2) The if-part is obvious by the definition. When $B \wedge H$ is a Horn theory, it has a unique minimal model. Hence, the result holds. □

The converse implication of Proposition 2.1(1) does not hold in general when O contains a clause C with $body(C) \neq \emptyset$.

Example 2.1 Let $B = \{ p(a) \vee q(a) \}$ and $O = \{\neg p(a)\}$. Then, $H = \{q(a)\}$ is a solution of cautious induction, but H is not a solution of explanatory induction because $B \wedge H$ has a model in which $p(a)$ is true.

Proposition 2.2 *(existence of solutions) Let B be the background knowledge and O an observation. Then, the following four conditions are equivalent.*

1. *$B \wedge O$ is consistent.*
2. *Brave induction has a solution.*
3. *Cautious induction has a solution.*
4. *Explanatory induction has a solution.*

Proof We prove (1)⇔(2), but (1)⇔(3) and (1)⇔(4) are proved in the same manner. If brave induction has a solution H, $B \wedge H$ has a minimal model satisfying O. Then, $B \wedge O$ is consistent. To see the converse, suppose that brave induction has no solution. Then, for any H no minimal model of $B \wedge H$ satisfies O. This implies that for any H no minimal model exists for $B \wedge H \wedge O$, so $B \wedge H \wedge O$ is inconsistent for any H.[3] Putting $H = \emptyset$, $B \wedge O$ is inconsistent. Hence, the result holds. □

Corollary 2.3 *(necessary condition of solutions) Let B be the background knowledge and O an observation. If H is a solution of brave induction, $B \wedge H \wedge O$ is consistent. The same condition holds for cautious and explanatory induction.*

We later provide another necessary and sufficient condition for brave induction in Proposition 2.7.

Next we compare properties of brave and cautious induction. In what follows, B, O, and H represent the background knowledge, an observation, and a hypothesis, respectively.

Proposition 2.4 *(conjunction of solutions) The fact that both H_1 and H_2 are solutions of brave induction does not imply that $H_1 \wedge H_2$ is a solution of brave induction. This is also the case for cautious induction.*

The above property also holds for explanatory induction (De Raedt and Dehaspe 1997b).[4]

[3] Every consistent clausal theory has a minimal model (Bossu and Siegel 1985).

[4] This property is also called *nonmonotonicity* of induction.

Table 1 Comparison of Properties

	Brave Ind.	Cautious Ind.	Explanatory Ind.
Conjunction of solutions	×	×	×
Disjunction of solutions	○	○	○
Conjunction of observations	×	○	○

Example 2.2 Let $B = \{ p(a) \leftarrow \}$ and $O = \{ q(a) \vee r(a) \leftarrow, \quad \leftarrow q(a) \wedge r(a) \}$. Then, both $H_1 = \{ q(x) \leftarrow p(x) \}$ and $H_2 = \{ r(x) \leftarrow p(x) \}$ cover O under B in brave or cautious induction, but $H_1 \wedge H_2$ does not.

Proposition 2.5 *(disjunction of solutions) If H_1 and H_2 are solutions of brave induction, so is $H_1 \vee H_2$. This is also the case for cautious and explanatory induction.*

Proof By $B \wedge (H_1 \vee H_2) \equiv (B \wedge H_1) \vee (B \wedge H_2)$, if both $B \wedge H_1$ and $B \wedge H_2$ have a minimal model satisfying O, so does $B \wedge (H_1 \vee H_2)$. Moreover, if O is satisfied in every (minimal) model of $B \wedge H_1$ and O is satisfied in every (minimal) model of $B \wedge H_2$, O is also satisfied in every (minimal) model of $(B \wedge H_1) \vee (B \wedge H_2)$. □

Proposition 2.6 *(conjunctions of observations) The fact that H covers both O_1 and O_2 under B implies that H covers $O_1 \wedge O_2$ under B in cautious and explanatory induction. But this is not the case for brave induction.*

Proof If O_1 and O_2 are satisfied in every (minimal) model of $B \wedge H$, so does $O_1 \wedge O_2$. A counter-example for brave induction is shown in Example 2.3. □

Example 2.3 Let $B = \{ p(x) \vee q(x) \leftarrow r(x), \quad s(a) \leftarrow \}$, $O_1 = \{ p(a) \}$, and $O_2 = \{ q(a) \}$. Then, $H = \{ r(x) \leftarrow s(x) \}$ covers both O_1 and O_2 under B in brave induction, but H does not cover $O_1 \wedge O_2$ under B.

Proposition 2.6 provides a property that distinguishes brave induction from cautious and explanatory induction. This property implies that given a series of observations, brave induction is not adapted for performing incremental computation of candidate hypotheses in general. Such an incremental computation is done in brave induction for hypotheses that are also solutions of cautious or explanatory induction.

Table 1 summarizes comparison of properties between brave, cautious and explanatory induction. By the table, we can observe that cautious induction and explanatory induction share similar properties. In fact, the difference between cautious and explanatory induction arises only when O contains non-positive clause (Proposition 2.1(1)). In many induction tasks, however, observations are usually given as a set of positive clauses or ground facts. On the other hand, brave induction may have solutions different from those of cautious or explanatory induction when B or H contains disjunctive clauses (Proposition 2.1(2)). Then, we are interested in computing possible solutions of brave induction under the background knowledge containing indefinite information, or computing solutions containing disjunctive information.

When B has a minimal model satisfying O, O is inferred by brave inference from B. In this case, $H = true$ covers O, which is a trivial and uninteresting solution. The problem of our interest is the case in which B has no minimal model satisfying O. When a clausal theory B has no minimal model satisfying O, $\neg O$ is derived from B under the *generalized closed world assumption* (GCWA) (Minker 1982).

Example 2.4 Let $B = \{\, p(a) \vee q(b) \,\}$ be the background knowledge which has two minimal models $\{p(a)\}$ and $\{q(b)\}$. Then, $H = true$ covers the observation $O_1 = \{p(a)\}$ under B in brave induction, while H does not cover the observation $O_2 = \{p(b)\}$. In this case, the GCWA derives $\neg p(b)$ but does not derive $\neg p(a)$.

It is worth noting that explanatory induction in Horn theories assumes that the background Horn theory B has no minimal model satisfying O.[5] In this case, $\neg O$ is derived from B under the *closed world assumption* (CWA) (Reiter 1978). Thus, brave induction in non-Horn theories is considered a natural extension of explanatory induction in Horn theories. In the next subsection, we develop an algorithm for computing brave induction.

2.2 Computation

In this section, we develop an algorithm for computing brave induction. We first characterizes the brave induction problem.

Proposition 2.7 *(necessary and sufficient condition for brave induction) Let B be the background knowledge, H a hypothesis, and O an observation. Then, $B \wedge H$ has a minimal model satisfying O iff there is a disjunction F of ground atoms such that $B \wedge H \models O \vee F$ and $B \wedge H \not\models F$.*[6]

Proof ($\rightarrow$) Suppose that $B \wedge H$ has a minimal model M such that $M \models O$. Consider a disjunction F of ground atoms satisfying (i) $M \not\models F$ and (ii) $N \models F$ for any $N \in MM(B \wedge H)$ such that $N \not\models O$. Such F is constructed by picking up ground atoms from each $N \setminus M$. Then, $B \wedge H \models O \vee F$ holds. As $M \not\models F$, $B \wedge H \not\models F$.

($\leftarrow$) Suppose that $B \wedge H \models O \vee F$ holds for a disjunction F of ground atoms and $B \wedge H \not\models F$. If $B \wedge H$ has no minimal model satisfying O, $B \wedge H \models O \vee F$ implies $B \wedge H \models F$. This contradicts the assumption that $B \wedge H \not\models F$. □

Throughout the section, the following conditions are assumed on the syntax of observations and hypotheses.

1. An observation O is a finite set of ground atoms.
2. A hypothesis H is a finite clausal theory such that each clause has the non-empty head.

The first condition is assumed as the normal problem setting in ILP (Nienhuys-Cheng and Wolf 1997). The second condition is also natural with the following reason. When B has no minimal model satisfying O, we introduce H to B to get a minimal model satisfying O. However, introducing negative clauses to B has an effect of eliminating minimal models of B but does not contribute to obtaining a new minimal model. So the exclusion of negative clauses in H is not a strong restriction.

[5] The condition is called *prior necessity* (Muggleton and De Raedt 1994).

[6] Related results are shown in (Gelfond *et al.* 1989, Theorem 4.5) in the context of circumscription, and in (Inoue 2002, Corollary 3.5) in terms of abduction.

The procedure for computing brave induction consists of four steps.

Step 1: Computing ground hypotheses

By Proposition 2.7, a solution of brave induction is obtained by computing H satisfying

$$B \wedge H \models O \vee F \tag{5}$$

and

$$B \wedge H \not\models F\,. \tag{6}$$

By (5), it holds that

$$B \wedge \neg O \models \neg H \vee F. \tag{7}$$

$\neg H \vee F$ is thus obtained by deduction from $B \wedge \neg O$. This technique is *inverse entailment* that is originally proposed by Muggleton for induction in Horn theories (Muggleton 1995), and is later extended by Inoue to full clausal theories (Inoue 2004).

As H is a clausal theory, put

$$H = (\Sigma_1 \leftarrow \Gamma_1) \wedge \cdots \wedge (\Sigma_k \leftarrow \Gamma_k) \tag{8}$$

where Σ_i $(i = 1, \ldots, k)$ is a disjunction of atoms and Γ_i $(i = 1, \ldots, k)$ is a conjunction of atoms. It then becomes

$$\neg H = (\neg\Sigma_1 \wedge \Gamma_1) \vee \cdots \vee (\neg\Sigma_k \wedge \Gamma_k). \tag{9}$$

Since F is a disjunction of ground atoms, every formula $\neg H \vee F$ in (7) is a disjunctive normal form. From $B \wedge \neg O$, a number of DNF formulas could be deduced. Among them, we take DNF formulas obtained as follows.

Definition 2.2 (prime clause, prime CNF) A ground clause C is called a *prime clause* with respect to a theory T if $T \models C$ but $T \not\models C'$ for any $C' \subset C$. A *prime CNF* formulas with respect to T is a conjunction of prime clauses with respect to T.

First, compute prime CNF formulas with respect to $B \wedge \neg O$. Prime CNF formulas are computed by a system of *consequence-finding* such as (Inoue 1992). Second, given a prime CNF formula $c_1 \wedge \cdots \wedge c_k$, produce an irredundant DNF formula $d_1 \vee \cdots \vee d_l$ where d_i $(1 \leq i \leq l)$ contains a literal from each c_j $(1 \leq j \leq k)$. Then,

$$B \wedge \neg O \models d_1 \vee \cdots \vee d_l$$

holds, and we identify the DNF formula $\neg H \vee F$ of (7) with $d_1 \vee \cdots \vee d_l$. After deriving such a ground DNF formula, the next task is to separate $\neg H$ and F in $d_1 \vee \cdots \vee d_l$. This is simply done as follows. By the assumption, Σ_i in H is non-empty, so that $\neg H$ is a DNF formula in which each disjunct $\neg\Sigma_i \wedge \Gamma_i$ of (9) contains at least one negative literal. Thus, from the DNF formula $d_1 \vee \cdots \vee d_l$, $\neg H$ is extracted by selecting disjuncts containing negative literals. Hence, H is obtained as a ground clausal theory.

Step 2: Generalization

As H is a clausal theory containing no variable, we generalize H in the next step. For this purpose, we use Plotkin's *least generalization under subsumption (LGS)* (Plotkin 1970).

Definition 2.3 (comparable) Clauses $C_1, \ldots, C_k$ are *comparable* if there is a predicate appearing in every $head(C_1), \ldots, head(C_k)$.

Let H be a clausal theory obtained by Step 1. Then, H is partitioned into

$$H = H_1 \wedge \cdots \wedge H_n \tag{10}$$

where each H_i $(1 \leq i \leq n)$ is a conjunction of comparable clauses.[7] Next, the LGS of each H_i is computed and collected as

$$lgs(H) = lgs(H_1) \wedge \cdots \wedge lgs(H_n) \tag{11}$$

where $lgs(H_i)$ represents the result of LGS of H_i. Note that since each H_i is a set of comparable clauses, $lgs(H_i)$ is a clause with the non-empty head.

Step 3: Constructing a weak form of hypotheses

We next consider a method of constructing a weaker hypothesis for brave induction. For each clause $lgs(H_1), \ldots, lgs(H_n)$ of (11), take the *greatest specialization under implication (GSI)* (Nienhuys-Cheng and Wolf 1997). The GSI of any finite set of clauses exists and is computed by taking a disjunction as

$$gsi(lgs(H_1), \ldots, lgs(H_n)) = lgs(H_1) \vee \cdots \vee lgs(H_n). \tag{12}$$

By $lgs(H_i) \models gsi(lgs(H_1), \ldots, lgs(H_n))$ for $i = 1, \ldots, n$, the GSI (12) provides a formula which is weaker than each $lgs(H_i)$.

Step 4: Optimization

Hypotheses computed in the above steps generally contain clauses or atoms that are useless or have no direct connection to explaining the observation. In this step, hypotheses are optimized to extract meaningful information.

Definition 2.4 (isolated) Two atoms A_1 and A_2 are *linked* if $term(A_1) \cap term(A_2) \neq \emptyset$. Given a clause C with the non-empty body, an atom A is *isolated* in C if there is no atom $A'(\neq A)$ in C such that A' and A are linked.

Example 2.5 Given $C = (p(x) \leftarrow q(x,y), r(y), s(z))$, the atom $s(z)$ is isolated in C.

Optimization is done in two steps.

1. Remove any isolated atom from the body of any clause $lgs(H_i)$ $(1 \leq i \leq n)$.
2. Remove any clause $lgs(H_i)$ $(1 \leq i \leq n)$ that is subsumed by another clause $lgs(H_j)$ $(1 \leq j \leq n)$.

The result of such reduction on $lgs(H_i)$ is denoted by $lgs^*(H_i)$. When $B \wedge lgs(H_i)$ is consistent, the reduction is performed as far as $B \wedge lgs^*(H_i)$ is consistent. Finally, put

$$H^{\wedge} = \bigwedge_i lgs^*(H_i)$$

if $B \wedge \bigwedge_i lgs^*(H_i) \wedge \neg F$ is consistent for a disjunction F of ground atoms computed by Step 1. On the other hand, put

$$H^{\vee} = \bigvee_i lgs^*(H_i)$$

[7] A clause C could be included in different H_i and H_j $(i \neq j)$ if C is comparable to clauses in both H_i and H_j.

Procedure: BRAIN

Input : the background knowledge B and an observation O;
Output : hypotheses $H^{\wedge}$ and $H^{\vee}$.

Step 1 : Compute ground and irredundant DNF formulas $\neg H \vee F$ from $B \wedge \neg O$, and extract $\neg H$ from $\neg H \vee F$.
Step 2 : Compute the least generalization under subsumption $lgs(H)$.
Step 3 : Compute the greatest specialization under implication $gsi(lgs(H_1), \ldots, lgs(H_n))$.
Step 4 : Produce $lgs^*(H_i)$ by reduction, and compute $H^{\wedge}$ and $H^{\vee}$.

Fig. 1 An algorithm for brave induction

if there is a disjunction F' of ground atoms such that $B \wedge \bigvee_i lgs^*(H_i) \wedge \neg F'$ is consistent and $B \wedge \bigvee_i lgs^*(H_i) \wedge \neg O \wedge \neg F'$ is inconsistent.

The algorithm (called BRAIN) for computing hypotheses is summarized in Figure 1.[8]

Now we show that BRAIN computes a solution for brave induction.

Theorem 2.8 *(soundness) Let $H^{\wedge}$ and $H^{\vee}$ be clausal theories obtained by* BRAIN. *Then, $H^{\wedge}$ and $H^{\vee}$ are solutions of brave induction.*

Proof Step 1 computes a ground clausal theory H satisfying $B \wedge H \models O \vee F$. In Step 2, $lgs(H_i) \models H_i$ holds for each H_i of (10). Then, $lgs(H) \models H$ and $B \wedge lgs(H) \models B \wedge H$. So $B \wedge lgs(H) \models O \vee F$. In Step 4, it holds that $lgs^*(H_1) \wedge \cdots \wedge lgs^*(H_n) \models lgs(H_1) \wedge \cdots \wedge lgs(H_n)$. Then, $\bigwedge_i lgs^*(H_i) \models lgs(H)$ and $B \wedge \bigwedge_i lgs^*(H_i) \models B \wedge lgs(H)$. So $B \wedge \bigwedge_i lgs^*(H_i) \models O \vee F$, and $\bigwedge_i lgs^*(H_i)$ satisfies the relation (5). If $B \wedge \bigwedge_i lgs^*(H_i) \wedge \neg F$ is consistent for a disjunction F of ground atoms computed by Step 1, then $B \wedge \bigwedge_i lgs^*(H_i) \not\models F$ and $\bigwedge_i lgs^*(H_i)$ satisfies the relation (6). Hence, $B \wedge H^{\wedge}$ has a minimal model satisfying O (Proposition 2.7).

Next, suppose $gsi(lgs(H_1), \ldots, lgs(H_n))$ in Step 3 is optimized as $\bigvee_i lgs^*(H_i)$ in Step 4. If $B \wedge \bigvee_i lgs^*(H_i) \wedge \neg O \wedge \neg F'$ is inconsistent for a disjunction F' of ground atoms, then $B \wedge \bigvee_i lgs^*(H_i) \models O \vee F'$. Thus, $\bigvee_i lgs^*(H_i)$ satisfies the relation (5). Moreover, if $B \wedge \bigvee_i lgs^*(H_i) \wedge \neg F'$ is consistent, $B \wedge \bigwedge_i lgs^*(H_i) \not\models F'$. Then, $\bigwedge_i lgs^*(H_i)$ satisfies the relation (6). Hence, $B \wedge H^{\vee}$ has a minimal model satisfying O (Proposition 2.7). □

Note that BRAIN is not complete with respect to solutions of brave induction. This is because we reduce seemingly useless hypotheses in the optimization phase of Step 4. We do not consider the incompleteness of the algorithm is a serious flaw, however. This is because there may exist possibly infinite solutions for explaining observations in general, and it seems meaningless to guarantee the completeness for computing tons of useless hypotheses. We select candidate solutions to reduce the hypotheses space at the cost of giving up the completeness.

Example 2.6 Consider the background knowledge B and the observation O:

$$B: \ teacher(0) \wedge student(1) \wedge \cdots \wedge student(30),$$
$$O: \ euro(1) \wedge \cdots \wedge euro(20) \wedge asia(21) \wedge \cdots \wedge asia(27) \wedge usa(28) \wedge \cdots \wedge usa(30).$$

[8] BRAIN is named after BRAve INduction.

BRAIN computes candidate hypotheses as follows.

(Step 1) $B \wedge \neg O$ entails the prime clauses:

$$teacher(0),\ student(1),\ \cdots,\ student(30),$$
$$\neg euro(1) \vee \cdots \vee \neg euro(20) \vee \neg asia(21) \vee \cdots \vee \neg asia(27) \vee \neg usa(28) \vee \cdots \vee \neg usa(30).$$

Taking the conjunction of them, the prime CNF formula:

$$teacher(0) \wedge student(1) \wedge \cdots \wedge student(30)$$
$$\wedge(\neg euro(1) \vee \cdots \vee \neg euro(20) \vee \neg asia(21) \vee \cdots \vee \neg asia(27) \vee \neg usa(28) \vee \cdots \vee \neg usa(30)),$$

which is equivalent to $B \wedge \neg O$, is computed.

Next, an irredundant DNF formula $\neg H_1 \vee \neg H_2 \vee \neg H_3 \vee F$ is obtained where

$$\begin{aligned} \neg H_1 &= (B \wedge \neg\, euro(1)) \vee \cdots \vee (B \wedge \neg\, euro(20)),\\ \neg H_2 &= (B \wedge \neg\, asia(21)) \vee \cdots \vee (B \wedge \neg\, asia(27)),\\ \neg H_3 &= (B \wedge \neg\, usa(28)) \vee \cdots \vee (B \wedge \neg\, usa(30)),\\ F &= false. \end{aligned}$$

By this, ground hypotheses:

$$\begin{aligned} H_1 &= (\neg B \vee euro(1)) \wedge \cdots \wedge (\neg B \vee euro(20)),\\ H_2 &= (\neg B \vee asia(21)) \wedge \cdots \wedge (\neg B \vee asia(27)),\\ H_3 &= (\neg B \vee usa(28)) \wedge \cdots \wedge (\neg B \vee usa(30)) \end{aligned}$$

are obtained.

(Step 2) The LGS of each H_i becomes

$$\begin{aligned} lgs(H_1) &= \neg teacher(0) \vee \neg student(x) \vee euro(x),\\ lgs(H_2) &= \neg teacher(0) \vee \neg student(y) \vee asia(y),\\ lgs(H_3) &= \neg teacher(0) \vee \neg student(z) \vee usa(z). \end{aligned}$$

Then, $lgs(H) = lgs(H_1)\ \wedge\ lgs(H_2)\ \wedge\ lgs(H_3)$.

(Step 3) By each $lgs(H_i)$, the greatest specialization becomes

$$\begin{aligned} gsi(lgs(H_1), \ldots, lgs(H_n)) &= lgs(H_1)\ \vee\ lgs(H_2)\ \vee\ lgs(H_3)\\ &= \neg teacher(0) \vee \neg student(x) \vee euro(x) \vee asia(x) \vee usa(x). \end{aligned}$$

(Step 4) The atom $teacher(0)$ is isolated in each $lgs(H_i)$ $(i = 1, 2, 3)$, so it is removed from the body of each clause. Since $B \wedge \bigwedge_i\ lgs^*(H_i) \wedge \neg F$ is consistent, $H^{\wedge}$ becomes

$$(euro(x) \leftarrow student(x))\ \wedge\ (asia(x) \leftarrow student(x))\ \wedge\ (usa(x) \leftarrow student(x)).$$

On the other hand, for the disjunction F' of ground atoms:

$$\begin{aligned} F' = &\ (asia(1) \vee usa(1)) \vee \cdots \vee (asia(20) \vee usa(20))\\ &\vee (euro(21) \vee usa(21)) \vee \cdots \vee (euro(27) \vee usa(27))\\ &\vee (asia(28) \vee euro(28)) \vee \cdots \vee (asia(30) \vee euro(30)), \end{aligned}$$

$B \wedge \bigvee_i\ lgs^*(H_i) \wedge \neg F'$ is consistent and $B \wedge \bigvee_i\ lgs^*(H_i) \wedge \neg O \wedge \neg F'$ is inconsistent.
Then, $H^\vee$ becomes

$$euro(x) \vee asia(x) \vee usa(x) \leftarrow student(x).$$

As a result, $H^\wedge$ and $H^\vee$ become two solutions of brave induction.
Note that if there are negative clauses

$$\leftarrow euro(x) \wedge asia(x),$$
$$\leftarrow euro(x) \wedge usa(x),$$
$$\leftarrow asia(x) \wedge usa(x)$$

in B, $B \wedge \bigwedge_i\ lgs^*(H_i)$ is inconsistent. In this case, $H^\wedge$ is not a solution of brave induction, while $H^\vee$ is still a solution.

Solutions of cautious induction are computed as a special case of BRAIN.

Corollary 2.9 *(computing cautious induction) Let $H^\wedge$ be clausal theories obtained by* BRAIN*. If $F = false$ in Step 1, $H^\wedge$ is a solution of cautious induction.*

Proof When $F = false$, BRAIN computes H satisfying $B \wedge H \models O$ (5) and $B \wedge H \not\models false$ (6). Then, H is a solution of explanatory induction. Since O is a set of ground atoms, solutions of explanatory induction coincide with those of cautious induction (Proposition 2.1(1)). Hence, the result holds. □

In Example 2.6, $H^\wedge$ also becomes a solution of cautious induction.

3 Brave Induction in Nonmonotonic Logic Programming

As presented in Section 2, brave induction is useful for learning theories with indefinite or incomplete information. Incomplete information is also represented as *default* rule in logic programming. In this section, we consider brave induction in *nonmonotonic logic programs.*

3.1 Answer Set Programming

Answer set programming (ASP) (Lifschitz 2002) represents incomplete knowledge in a logic program and realizes nonmonotonic default reasoning. In ASP a logic program is described by an *extended disjunctive program* (EDP). An EDP (or simply a *program*) is a set of rules of the form:

$$L_1\,;\,\cdots\,;\,L_l \ \leftarrow\ L_{l+1}\,,\ldots,\,L_m,\,not\,L_{m+1}\,,\ldots,\,not\,L_n \qquad (13)$$

($n \geq m \geq l \geq 0$) where each L_i is a positive/negative literal, i.e., A or $\neg A$ for an atom A. *not* represents *default negation* or *negation as failure* (NAF). $not\,L$ is called an *NAF-literal.* Literals and NAF-literals are called *LP-literals.* The symbol ";" represents disjunction and "," represents conjunction. The rule (13) is read "if all $L_{l+1},\ldots,L_m$ are believed and all $L_{m+1},\ldots,L_n$ are disbelieved, then some of $L_1,\ldots,L_l$ is believed". The left-hand side of "←" is the *head*, and the right-hand side is the *body.* For each rule r of the form (13), $head(r)$, $body^+(r)$ and $body^-(r)$ denote the sets of literals

$\{L_1, \ldots, L_l\}$, $\{L_{l+1}, \ldots, L_m\}$, and $\{L_{m+1}, \ldots, L_n\}$, respectively. Also, $not\_body^-(r)$ denotes the set of NAF-literals $\{not\,L_{m+1}, \ldots, not\,L_n\}$. A disjunction of literals and a conjunction of (NAF-)literals in a rule are identified with its corresponding sets of literals.[9] A rule r is often written as

$$head(r) \leftarrow body^+(r), not\_body^-(r) \quad \text{or} \quad head(r) \leftarrow body(r)$$

where $body(r) = body^+(r) \cup not\_body^-(r)$. A rule r is *disjunctive* if $head(r)$ contains more than one literal. A rule r is a *constraint* if $head(r) = \emptyset$; and r is a *fact* if $body(r) = \emptyset$. A program is *NAF-free* if no rule contains NAF-literals. A rule r_1 *subsumes* a rule r_2 if $head(r_1)\theta \subseteq head(r_2)$ and $body(r_1)\theta \subseteq body(r_2)$ hold for some substitution θ. A program, rule, or literal is *ground* if it contains no variable. A program P with variables is a shorthand of its *ground instantiation* $Ground(P)$, the set of ground rules obtained from P by substituting variables in P by elements of its Herbrand universe in every possible way. Two literals L_1 and L_2 have the same *sign* if both L_1 and L_2 are positive literals (or negative literals). A set S of ground literals is *consistent* if $L \in S$ implies $\neg L \notin S$ for any literal L; otherwise, S is *contradictory*. A set S of literals *satisfies* a program P if $body^+(r) \subseteq S$ and $body^-(r) \cap S = \emptyset$ imply $head(r) \cap S \neq \emptyset$ for any rule r in $Ground(P)$.

The semantics of an EDP is defined by the *answer set semantics* (Gelfond and Lifschitz 1991). Let Lit be the set of all ground literals in the language of a program. Suppose 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 $body^-(r) \cap S = \emptyset$. Given an NAF-free EDP P, let S be a set of ground literals that is (i) *closed* under P, i.e., for every ground rule r in $Ground(P)$, $body(r) \subseteq S$ implies $head(r) \cap S \neq \emptyset$; and (ii) *logically closed*, i.e., it is either consistent or equal to Lit. An *answer set* of an NAF-free EDP P is a minimal set S satisfying both (i) and (ii). Given an EDP P and a set S of ground 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. The set of all answer sets of P is written as $AS(P)$. An answer set is *consistent* if it is not Lit. A program P is *consistent* if it has a consistent answer set; otherwise, P is *inconsistent*.

Example 3.1 The program

$$\begin{aligned} &tea\,;\, coffee \leftarrow, \\ &milk \leftarrow tea,\, not\ lemon, \\ &lemon \leftarrow tea,\, not\ milk, \\ &milk \leftarrow coffee \end{aligned}$$

has the three answer sets:

$$\{\,tea,\, milk\,\}, \quad \{\,tea,\, lemon\,\}, \quad \{\,coffee,\, milk\,\},$$

which represent possible options for drink.

[9] By this fact, any duplicated appearance of the same literal in a rule is ignored. That is, a disjunction $(L; L)$ is identified with L, a conjunction (L, L) or $(not\,L,\, not\,L)$ is identified with L or $not\,L$, respectively.

3.2 Brave Induction in ASP

In this section, we consider the following problem setting:

- the background knowledge B is given as an EDP,
- an observation O is given as a set of ground literals,
- a hypothesis H is a finite set of rules.

Then, brave induction in ASP is defined as follows.

Definition 3.1 (brave and cautious induction in ASP) Let B be the background knowledge and O an observation. A hypothesis H *covers* O under B in *brave induction* if $B \cup H$ has a consistent answer set S such that $O \subseteq S$. H is called a *solution* of brave induction. By contrast, H *covers* O under B in *cautious induction* if $B \cup H$ is consistent and $O \subseteq S$ for any consistent answer set S of $B \cup H$. H is called a *solution* of cautious induction. [10]

Proposition 3.1 *(sufficient condition for the existence of solutions) Brave induction has a solution if $B \cup O$ is consistent. This is also the case for cautious induction.*

Proof If $B \cup O$ is consistent, $B \cup O$ has a consistent answer set S. Then, S becomes a consistent answer set of $B^S \cup O$ and $O \subseteq S$. By putting $H = O$, any consistent answer set S of $B \cup H$ satisfies $O \subseteq S$. □

In contrast to Proposition 2.2, the condition of Proposition 3.1 is not a necessary one.

Example 3.2 Let $B = \{\, p \leftarrow q,\ not\, p \,\}$ and $O = \{q\}$. Then, $B \cup O$ has no answer set thereby inconsistent. But $H = \{\, q \leftarrow p, \quad p \leftarrow \}$ becomes a solution of brave and cautious induction, since $B \cup H$ has the answer set $\{\, p,\ q \,\}$ in which q is included.

Proposition 3.2 *(necessary condition for the existence of solutions) Brave induction has a solution only if $AS(B \cup O) \neq \{Lit\}$.*

Proof When $AS(B \cup O) = \{Lit\}$, two cases are considered. (i) When $AS(B) = \{Lit\}$, B^{Lit} has the answer set Lit. This means that the set of NAF-free rules in B is contradictory, so that there is no way to recover consistency by introducing any H to B. In this case, brave induction has no solution. (ii) Else if $AS(B) \neq \{Lit\}$ but $AS(B \cup O) = \{Lit\}$, the set of NAF-free rules in B is consistent but $B^{Lit} \cup O$ is contradictory. This means that there is a literal L such that $L \in O$ and $\neg L \in S$ for any minimal closed set S of $B^{Lit} \cup O$. Suppose that there is a solution H such that $B \cup H$ has a consistent answer set T satisfying $O \subseteq T$. In this case, T is a minimal closed set satisfying B^{Lit}, so T is an answer set of $B^{Lit} \cup T$. Since $\neg L \in S$ for any minimal closed set S of $B^{Lit} \cup O$ and $O \subseteq T$, $\neg L \in U$ for any minimal closed set U of $B^{Lit} \cup T$. Here, $T \subseteq U$ holds. As T is an answer set of $B^{Lit} \cup T$, T is a minimal closed set of $B^{Lit} \cup T$ and $\neg L \in T$. By $L \in O \subseteq T$, $L \in T$ so T is contradictory. This contradicts the assumption that T is a consistent answer set. Thus, $B \cup H$ cannot have a consistent answer set T satisfying $O \subseteq T$. Hence, brave induction has no solution. By (i) and (ii), if brave induction has a solution, $AS(B \cup O) \neq \{Lit\}$. □

[10] In nonmonotonic logic programming, logical connectives in classical logic are not used. So we write $B \cup H$ instead of $B \wedge H$.

By the proof of Proposition 3.2, it is observed that the same necessary condition holds for cautious induction.

Proposition 3.3 *(necessary condition of solutions) If H is a solution of brave induction, $B \cup H \cup O$ is consistent. This is also the case for cautious induction.*

Proof If H is a solution of brave induction, $B \cup H$ has a consistent answer set S satisfying $O \subseteq S$. In this case, $B^S \cup H^S$ has a consistent answer set S satisfying $O \subseteq S$. Then, S becomes an answer set of $B^S \cup H^S \cup O$, so S is an answer set of $B \cup H \cup O$. Hence, $B \cup H \cup O$ is consistent. The case for cautious induction is proved in a similar way. □

Proposition 3.4 *(relation between brave and cautious induction)*

1. *Brave induction has a solution iff cautious induction has a solution.*
2. *If H is a solution of cautious induction, H is a solution of brave induction. The converse implication also holds when $B \cup H$ contains neither disjunction nor NAF.*

Proof If brave induction has a solution H, $B \cup H \cup O$ is consistent (Proposition 3.3). Put $H' = H \cup O$. Then, every consistent answer set S of $B \cup H'$ satisfies $O \subseteq S$. Hence, cautious induction has a solution. Conversely, if cautious induction has a solution H, it is also a solution of brave induction. When $B \cup H$ contains neither disjunction nor NAF, $B \cup H$ has at most one answer set. In this case, brave induction coincides with cautious induction. □

Some properties of brave or cautious induction follows.[11]

Proposition 3.5 *(conjunction of solutions) The fact that both H_1 and H_2 are solutions of brave induction does not imply that $H_1 \cup H_2$ is a solution of brave induction. This is also the case for cautious induction.*

Proposition 3.6 *(conjunctions of observations) The fact that H covers both O_1 and O_2 under B implies that H covers $O_1 \cup O_2$ under B in cautious induction. But this is not the case for brave induction.*

Proof If $O_1 \subseteq S$ and $O_2 \subseteq S$ hold for any consistent answer set S of $B \cup H$, then $O_1 \cup O_2 \subseteq S$. A counter-example for the case of brave induction is Example 2.3. □

There are algorithms for computing cautious induction in ASP (Sakama 2005). In what follows, we develop a procedure for computing brave induction in ASP. As the case of clausal theories, the problem of our interest is the case when B has no answer set including O. It is worth noting that in case of clausal theories, the consistency of $B \wedge H$ implies the consistency of B, H, and O. On the other hand, in case of ASP, the consistency of $B \cup H$ implies the consistency of O, but it does not necessarily imply the consistency of B or H. In fact, an inconsistent program B having no answer set can recover consistency by introducing an appropriate H.

Example 3.3 Let $B = \{\, p \leftarrow not\, p \,\}$. Then, $H = \{p\}$ recovers the consistency in $B \cup H$.

[11] We do not address the property of disjunction of solutions, since it requires a definition of disjunction of ASP programs.

For a technical reason, however, we assume the consistency of B in the rest of this section. In case of brave induction from clausal theories, inverse entailment is used for computing hypotheses. However, it is known that inverse entailment in classical logic is not applied to nonmonotonic logic programs (Sakama 2000). We then consider another method for computing possible hypotheses.

Step 1: Computing ground hypotheses

We first introduce a notion used in this step.

Definition 3.2 (relevant) Let L_0 be a ground literal and S a set of ground literals. Then, $L_1 \in S$ is *relevant* to L_0 if either (i) $const(L_0) \cap const(L_1) \neq \emptyset$, or (ii) for some literal $L_2 \in S$, $const(L_1) \cap const(L_2) \neq \emptyset$ and L_2 is relevant to L_0. Otherwise, $L_1 \in S$ is *irrelevant* to L_0.

Given an observation O, let $\Theta = \{ L \mid L \in Lit \text{ and } pred(L) \text{ appears in } O \}$. Suppose that the background knowledge B has a consistent answer set S. Then, construct a finite and consistent set R of ground rules satisfying the following conditions. For any rule $r \in R$,

1. $head(r) \subseteq O$ and for any $L \in O$, there is a rule $r \in R$ such that $head(r) = \{L\}$.
2. $body^+(r) = \{ L \mid L \in S \text{ and } L \text{ is relevant to the literal in } head(r) \}$.
3. $body^-(r) = \{ L \mid L \in Lit \setminus (S \cup \Theta) \text{ and } L \text{ is relevant to the literal in } head(r)$ and appears in $Ground(P) \}$.

The third condition requires that no rule contains default negation of literals in $S \cup \Theta$. The reason is that if $body^-(r)$ contains literals from S, $body(r)$ may contain both L in $body^+(r)$ and $not\, L$ in $body^-(r)$, which makes the rule meaningless. Also, if $body^-(r)$ contains literals from Θ, r may contain a *negative loop* that would make a program inconsistent. By its construction, different hypotheses are constructed by different answer sets in general.

Step 2: Generalization

The notion of LGS is extended to rules containing default negation. It is done by syntactically viewing rules as "clauses". That is, identify disjunction ";" with the classical one "$\vee$, and any NAF-literal "$not\, p(t_1, \ldots, t_n)$" with a new atom "$not\_p(t_1, \ldots, t_n)$" with the predicate "$not\_p$". $\neg p$ is also considered a predicate "$\neg\_p$" and is considered a predicate different from p. With this setting, the notion of comparable set of rules is defined as Definition 2.3, and the LGS of a comparable set of rules is defined in the same manner as the one in clausal theories (Sakama 2001). The generalization phase is similar to the case of clausal theories. For the set R of rules obtained by the Step 1, R is partitioned as $R = R_1 \cup \cdots \cup R_n$ where each R_i $(1 \leq i \leq n)$ is a comparable set of ground rules. Then, the LGS of each R_i is computed and collected as

$$lgs(R) = \{ lgs(R_1), \ldots, lgs(R_n)\}.$$

Step 3: Constructing a weak form of hypotheses

To construct a weak form of hypotheses, we introduce the notion of cardinality constraint rules (Niemelä *et al.* 1999). A *cardinality constraint rule* is a rule of the form:

$$h\, \{L_1, \ldots, L_l\}\, k \;\leftarrow\; L_{l+1}, \ldots, L_m,\; not\, L_{m+1}, \ldots, not\, L_n \tag{14}$$

Procedure: BRAIN^{not}

Input : the background knowledge B and an observation O;
Output : hypotheses $H^{\wedge}$ and $H^{\vee}$.

Step 1 : Select an answer set S of B and construct a set R of rules.
Step 2 : Compute $lgs(R)$.
Step 3 : Compute $CCR(lgs(R_1), \ldots, lgs(R_n))$.
Step 4 : Produce $lgs^*(R_i)$ by reduction, and compute $H^{\wedge}$ and $H^{\vee}$.

Fig. 2 An algorithm for brave induction in ASP

where h and k are two integers such that $h \leq k$. The rule (14) means if the body holds then at least h and at most k literals in the head hold. This type of rules is useful for representing knowledge in ASP and is used in the *smodel* system (Niemelä *et al.* 1999). A program with this type of rules is translated into a semantically equivalent EDP.

Given $lgs(R_1), \ldots, lgs(R_n)$, we construct a cardinality constraint rule

$$1\,\Sigma\,k \leftarrow \Gamma \tag{15}$$

where $\Sigma = head(lgs(R_1)) \cup \cdots \cup head(lgs(R_n))$, $\Gamma = body(lgs(R_1)) \cup \cdots \cup body(lgs(R_n))$, and k is the number of literals in Σ, i.e., $k = \mid \Sigma \mid$. We write $CCR(lgs(R_1), \ldots, lgs(R_n))$ to represent a cardinality constraint rule (15) constructed by $lgs(R_1), \ldots, lgs(R_n)$.

Step 4: Optimization

Optimization is done in two steps. First, remove any isolated literal from the body of any rule $lgs(R_i)$ $(1 \leq i \leq n)$. Here, the notion of "isolated literal" in a rule is defined by replacing a clause with a rule, and an atom with a literal in Definition 2.4. Second, remove any rule $lgs(R_i)$ $(1 \leq i \leq n)$ that is subsumed by another rule $lgs(R_j)$ $(1 \leq i \leq n)$. Let $lgs^*(R_i)$ be the result of such reduction over $lgs(R_i)$. When $B \cup \{\, lgs(R_i) \,\}$ is consistent, the reduction is performed as far as $B \cup \{\, lgs^*(R_i) \,\}$ is consistent. Finally, put

$$H^{\wedge} = \bigcup_i \{\, lgs^*(R_i) \,\}$$

if $B \cup \bigcup_i \{\, lgs^*(R_i) \,\}$ is consistent. Also, put

$$H^{\vee} = \{\, CCR(lgs^*(R_1), \ldots, lgs^*(R_n)) \,\}$$

if $B \cup \{\, CCR(lgs^*(R_1), \ldots, lgs^*(R_n)) \,\}$ is consistent.

The algorithm of brave induction in ASP (called BRAIN^{not}) is sketched in Figure 2. In what follows, we show that BRAIN^{not} computes hypotheses for brave induction in ASP. We say that O is *independent* of B if every predicate in O appears nowhere in B.

Lemma 3.7 *Let B be the background knowledge and O an observation. If O is independent of B, $B \cup H^{\wedge}$ has an answer set U such that $O \subseteq U$.*

Proof Let S be a consistent answer set of B. For any rule r in R, $head(r) \leftarrow body^+(r)$ is in R^S. Here, $body^+(r) \subseteq S$, $head(r) \subseteq O$, and for any $L \in O$, $pred(L)$ appears nowhere in B. Put $T = S \cup \{\, L \mid L \in head(r) \text{ and } r \in R^S \,\}$. By $B^T \cup R^T = B^S \cup R^T$, T becomes a minimal closed set of $B^T \cup R^T = (B \cup R)^T$. Since O is independent of B, every predicate in $head(r)$ appears nowhere in B so that T is a consistent set of literals. As every literal in O appears in the head of some rule r in R, T is a consistent answer set of $B \cup R$ such that $O \subseteq T$. Next, we show that $B \cup lgs(R)$ has a consistent answer set such that $O \subseteq U$. Let $R = R_1 \cup \cdots \cup R_n$. By the definition, $lgs(R_i)\theta \subseteq r$ for any $r \in R_i$ $(1 \leq i \leq n)$ with some ground substitution θ. Then, for any rule $r \in R_i$, $body^-(r) \cap S = \emptyset$ implies $body^-(lgs(R_i)\theta) \cap S = \emptyset$. So $B^S \cup R^S \subseteq B^S \cup lgs(R)^S$. Since O is independent of B, $lgs(R)^S \setminus R^S$ is a set of NAF-free rules whose heads have predicates appearing nowhere in B. Put $V = \{ L \mid r \in lgs(R)^S \setminus R^S,\ head(r) = \{L\}$ and $body^+(r) \subseteq S \}$. Then, $B^{S \cup V} \cup lgs(R)^{S \cup V} = B^S \cup lgs(R)^S$ has a minimal closed set $U = S \cup V$. Since O is independent of B, $S \cup V$ is consistent. As any literal in O appears in the head of some ground instance of a rule in $B \cup lgs(R)$, U is a consistent answer set of $B \cup lgs(R)$ such that $O \subseteq U$. When $B \cup \bigcup_i \{ lgs^*(R_i) \}$ is consistent, U also becomes a consistent answer set of $B \cup H^{\wedge}$. Hence, the result follows. □

Lemma 3.8 *Let B be the background knowledge and O an observation. If O is independent of B, $B \cup H^{\vee}$ has an answer set U such that $O \subseteq U$.*

Proof Let $r = CCR(lgs(R_1), \ldots, lgs(R_n))$. For any ground instance of $r\theta$ satisfying $body^+(r\theta) \subseteq S$ and $body^-(r\theta) \cap S = \emptyset$ for some answer set S of B, a set T of literals is constructed in a way that a literal L is selected from the head of each $r\theta$ whenever $L \in O$. Since O is independent of B, $S \cup T$ is consistent. Then, $U = S \cup T$ becomes a consistent answer set of $B \cup \{ CCR(lgs(R_1), \ldots, lgs(R_n)) \}$ and $O \subseteq U$. When $B \cup \{ CCR(lgs^*(R_1), \ldots, lgs^*(R_n)) \}$ is consistent, U also becomes a consistent answer set of $B \cup H^{\vee}$. □

By Lemmas 3.7 and 3.8, we have the next result.

Theorem 3.9 *(soundness) Any hypothesis computed by* BRAIN^{not} *becomes a solution of brave induction.*

BRAIN^{not} is incomplete with respect to solutions of brave induction, since it reduces seemingly useless hypotheses in the optimization phase.

Example 3.4 There are two couples, Adam and Nancy, and Bob and Jane. They plan to go to either sea or mountain on this weekend. Each couple can select one of them, but a husband and a wife go to the same place. The situation is represented as the background knowledge B:

$$\begin{aligned}
& s(x) \leftarrow not\ m(x), \\
& m(x) \leftarrow not\ s(x), \\
& c(a,n) \leftarrow, \\
& c(b,j) \leftarrow, \\
& \leftarrow c(x,y), s(x), m(y), \\
& \leftarrow c(x,y), s(y), m(x)
\end{aligned}$$

where the predicates s, m and c mean *sea*, *mountain* and *couple*, respectively, and the constants a, n, b and j mean Adam, Nancy, Bob and Jane, respectively. B has four answer sets:

$$\begin{aligned}
S_1 &= \{ c(a,n),\ c(b,j),\ s(a),\ s(n),\ s(b),\ s(j) \}, \\
S_2 &= \{ c(a,n),\ c(b,j),\ s(a),\ s(n),\ m(b),\ m(j) \}, \\
S_3 &= \{ c(a,n),\ c(b,j),\ m(a),\ m(n),\ s(b),\ s(j) \}, \\
S_4 &= \{ c(a,n),\ c(b,j),\ m(a),\ m(n),\ m(b),\ m(j) \}.
\end{aligned}$$

Suppose the observation that Adam and Nancy are tanned, but Bob and Jane are not. It is represented as:

$$O = \{\, t(a),\ t(n),\ \neg t(b),\ \neg t(j)\,\}$$

where the predicate t mean *tanned*.

BRAIN^{not} constructs candidate hypotheses as follows.

(Step 1) First, an answer set of B, for instance S_2, is selected. A set R of rules is then constructed as:

$$\begin{aligned}
t(a) &\leftarrow c(a,n),\ s(a),\ s(n),\ not\ m(a),\ not\ m(n),\\
t(n) &\leftarrow c(a,n),\ s(a),\ s(n),\ not\ m(a),\ not\ m(n),\\
\neg t(b) &\leftarrow c(b,j),\ m(b),\ m(j),\ not\ s(b),\ not\ s(j),\\
\neg t(j) &\leftarrow c(b,j),\ m(b),\ m(j),\ not\ s(b),\ not\ s(j).
\end{aligned}$$

Note that the body of each rule contains literals that are relevant to the literal in the head.

(Step 2) Next, the $lgs(R)$ is constructed as

$$\begin{aligned}
t(x) &\leftarrow c(a,n),\ s(x),\ not\ m(x),\\
\neg t(y) &\leftarrow c(b,j),\ m(y),\ not\ s(y).
\end{aligned}$$

(Step 3) Third, from the rules in $lgs(R)$, the cardinality constraint rule is constructed as

$$1\{\, t(x)\,,\ \neg\, t(y)\,\}2 \leftarrow c(a,n),\ c(b,j),\ s(x),\ m(y),\ not\ m(x),\ not\ s(y).$$

(Step 4) Finally, isolated literals $c(a,n)$ and $c(b,j)$ are removed, and $H^{\wedge}$ and $H^{\vee}$ become

$$\begin{aligned}
H^{\wedge} &= \{\, t(x) \leftarrow s(x),\ not\ m(x),\quad \neg t(y) \leftarrow\ m(y),\ not\ s(y)\,\},\\
H^{\vee} &= \{\, 1\{\, t(x)\,,\ \neg\, t(y)\,\}2 \leftarrow s(x),\ m(y),\ not\ m(x),\ not\ s(y)\,\},
\end{aligned}$$

which are two solutions of brave induction.

4 Computational Complexity

In this section, we consider computational complexity of brave induction. Throughout the section, we assume that the background knowledge, hypotheses, and observations are are all represented over a finite propositional language.[12]

The following two decision problems are considered.

- **Existence**: Given the background knowledge B and an observation O, deciding whether O has a solution of brave induction under B.
- **Verification**: Given the background knowledge B and an observation O, deciding whether a given hypothesis H is a solution of brave induction under B.

[12] We view a program with variables as a shorthand of its ground instantiation. When the language contains no function symbol, the ground instantiation of a program is finite. In this case, the ground instantiated program is identified with a finite propositional program.

We consider these problems in the context of clausal theories (CT) and answer set programming (ASP), and compare complexity results between brave induction and cautious induction.

Theorem 4.1 *In clausal theories the following complexity results hold.*

1. *Deciding the existence of solutions in brave induction is NP-complete. The same complexity result holds for cautious induction.*
2. *Deciding whether a given hypothesis is a solution of brave induction is Σ_2^P-complete, while the corresponding problem in cautious induction is coNP-complete.*

Proof (1) By Proposition 2.2, brave induction has a solution H iff $B \wedge O$ is consistent. The task of deciding the consistency of $B \wedge O$ is NP-complete. Since the existence of solutions coincides in brave and cautious induction, the same complexity result holds for cautious induction.

(2) To show the result, we consider a complementary problem: a ground clausal theory $B \wedge H$ has no minimal model satisfying a conjunction O of ground atoms. This is a task of the *extended GCWA* and is known Π_2^P-complete (Eiter and Gottlob 1995), so that the verification problem is Σ_2^P-complete. In case of cautious induction, H becomes a solution iff $B \wedge H \models O$ iff $B \wedge H \wedge \neg O$ is inconsistent. Deciding the unsatisfiability of $B \wedge H \wedge \neg O$ is coNP-complete. □

To show complexity results in ASP, we introduce a program transformation.

Let r be a ground rule of the form

$$L_1 \,;\, \cdots \,;\, L_l \;\leftarrow\; L_{l+1}\,, \ldots,\, L_m,\, not\, L_{m+1}\,, \ldots,\, not\, L_n \tag{16}$$

($n \geq m \geq l \geq 0$). The propositional formula $\phi(r)$ associated with r is defined as

$$L'_{l+1} \,\wedge\, \cdots \,\wedge\, L'_m \,\wedge\, \neg\, L'_{m+1} \,\wedge\, \cdots \,\wedge\, \neg\, L'_n \supset L'_1 \,\vee\, \cdots \,\vee\, L'_l \tag{17}$$

where $L'_i = L_i$ if L_i is a positive literal, and $L'_j = \overline{A}$ if L_j is a negative literal $\neg A$ ($1 \leq i,\, j \leq n$). Thus, any negative literal $\neg A$ in r is transformed to a new *atom* $\overline{A}$, and default negation $not\, L$ is transformed to a negative literal $\neg\, L$ in $\phi(r)$. Given a ground EDP P, we define a propositional theory $\phi(P)$ as

1. for any $r \in P$, $\phi(r)$ is in $\phi(P)$.
2. for any positive literal $L \in Lit$, the following formula is in $\phi(P)$

$$L \wedge \overline{L} \supset false. \tag{18}$$

Given a set S of literals, $\overline{S}$ is the set of atoms which is obtained from S by replacing every negative literal $\neg L$ with the corresponding atom $\overline{L}$. Conversely, given a set M of atoms, $M^{\neg}$ is the set of literals which is obtained from M by replacing every atom $\overline{L}$ with the corresponding negative literal $\neg L$.

Proposition 4.2 *If an EDP P is consistent, so is $\phi(P)$.*

Proof If P is consistent, there is a consistent minimal set S of literals such that $\{L_{l+1}\,, \ldots,\, L_m\} \subseteq S$ and $\{L_{m+1}\,, \ldots,\, L_n\} \cap S = \emptyset$ imply $\{L_1\,, \ldots,\, L_l\} \cap S \neq \emptyset$ for any rule (16) in P. In this case, $\overline{S}$ satisfies both (17) and (18). Hence, $\phi(P)$ is consistent. □

Table 2 Computational Complexity

	Brave Induction		Cautious Induction	
Language	CT	ASP	CT	ASP
Existence	NP	NP	NP	NP
Verification	Σ_2^P	Σ_2^P	coNP	Π_2^P

Theorem 4.3 *In ASP the following complexity results hold.*

1. *Deciding the existence of solutions in brave induction is NP-complete. The same complexity result holds for cautious induction.*
2. *Deciding whether a given hypothesis is a solution of brave induction is* Σ_2^P*-complete, while the corresponding problem in cautious induction is* Π_2^P*-complete.*

Proof (1) Given an EDP B and a set O of ground literals, we show that brave induction has a solution iff the propositional theory $\phi(B \cup O)$ is consistent. Suppose that $\phi(B \cup O)$ is consistent. Then, $\phi(B \cup O)$ has a model M satisfying the following three conditions: (i) $\overline{O} \subseteq M$, (ii) for any formula (17) in $\phi(B)$, $\{L'_{l+1}, \ldots, L'_m\} \subseteq M$ and $\{L'_{m+1}, \ldots, L'_n\} \cap M = \emptyset$ imply $\{L'_1, \ldots, L'_l\} \cap M \neq \emptyset$, and (iii) for any atom L, $\{L, \overline{L}\} \not\subseteq M$ by (18). In this case, $M^{\neg}$ is a consistent set of literals satisfying $B \cup O$. Then, $M^{\neg}$ is a consistent minimal closed set satisfying $B^{M^{\neg}} \cup M^{\neg}$. Thus, $M^{\neg}$ is an answer set of $B^{M^{\neg}} \cup M^{\neg}$ and $O \subseteq M^{\neg}$. Putting $H = M^{\neg}$, H is a solution of brave induction. Conversely, suppose that brave induction has a solution. Then, for any solution H of brave induction, $B \cup H \cup O$ is consistent (Proposition 3.3). In this case, $\phi(B \cup H \cup O)$ is consistent (Proposition 4.2), so that $\phi(B \cup O)$ is consistent. Hence, brave induction has a solution iff $\phi(B \cup O)$ is consistent. Since deciding the consistency of the propositional theory $\phi(B \cup O)$ is NP-complete, the result holds. By Proposition 3.4, brave induction has a solution iff cautious induction has a solution. Hence, the same complexity result holds for cautious induction.

(2) Deciding whether some (resp. every) consistent answer set S of $B \cup H$ satisfies $O \subseteq S$ is Σ_2^P-complete (resp. Π_2^P-complete) (Eiter and Gottlob 1995). Hence, the result holds. □

The complexity results are summarized in Table 2. In the table, every entry represents completeness for the respective class. These complexity results show that brave and cautious induction are in the same complexity class for checking the existence of solutions. It is worth noting that extending the language from CT to ASP does not lead to a complexity increase in this problem. On the other hand, for the task of solution verification, the complexity of brave induction is one level higher in the polynomial hierarchy than that of cautious induction in CT. By contrast, the complexities of brave induction and cautious induction are at the same level of the polynomial hierarchy in ASP.

5 Problem Solving by Brave Induction

5.1 Systems Biology

In this section, we show the use of brave induction for inference of *master reactions* from biochemical networks in *systems biology*. It is a crucial feature of flux distributions

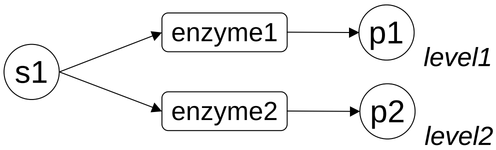

Fig. 3 Two reactions with one substrate and one product

that metabolic reactions with fluxes spanning several orders of magnitude coexist under the same conditions (Almaas *et al.* 2004). Although most metabolic reactions have low fluxes, the overall behavior of metabolism is dominated by several reactions with very high fluxes. In (Yamamoto *et al.* 2009), the states of each enzyme reaction is simply divided into two kinds, activated and non-activated, in order to analyze which chemical reactions have high fluxes. This analysis is helpful to solve the differential equations associated with reactions by ignoring non-activated reactions with low fluxes. We apply brave induction to hypothesis-finding in biochemical networks. Consider a simple chemical reaction represented in Figure 3.[13]

In Figure 3, the two reactions involve the same substrate s_1 and are catalyzed by the enzymes e_1 and e_2, then lead to the products p_1 and p_2, respectively. These two reactions are represented by the formulas:

$$R = \{\, reaction(e_1, s_1, p_1),\ \ reaction(e_2, s_1, p_2)\,\}.$$

Assume that the levels of concentration of compounds are classified into five, $l_1, \ldots, l_5$,[14] and that the concentration levels of the products p_1 and p_2 are as follows:

$$O = \{\, concentration(p_1, l_1),\ \ concentration(p_2, l_2)\,\}.$$

Next, suppose that the two enzymes e_1 and e_2 are of the same type t:

$$E = \{\, class(e_1, t),\ \ class(e_2, t)\,\}.$$

Let the background knowledge be $B = R \cup E$. Then, the next hypothesis H becomes a solution of brave induction.

$$\begin{aligned} H = \{\ & concentration(Y, l_1) \vee concentration(Y, l_2) \\ & \leftarrow reaction(Enzyme, X, Y) \wedge class(Enzyme, t)\ \}. \end{aligned}$$

The above H cannot be induced by explanatory nor cautious induction. The hypothesis H represents that an enzyme reaction of the type t leads to a product with low-level concentration that is either at level l_1 or l_2. In other words, an enzyme of the type t is non-activated or is inhibited by some reason. Note that, if the concentration level of p_1

[13] This example is given by Yoshitaka Yamamoto.

[14] On logical representation of concentration change, we here subdivide the levels into five or more instead of qualitative two-valued expression like "up" and "down" in (Yamamoto *et al.* 2009). With this refinement, more precise representation becomes possible. This is advised by Andrei Doncescu.

is the same as that of p_2, ordinary systems of explanatory induction can also induce an appropriate hypothesis. However, when we consider multiple levels of concentration, the hypothesis by brave induction is considered more useful.

In general, brave induction can induce a causal rule which combines multiple states as alternative effects. To infer master reactions correctly from biochemical pathways, it is necessary to set the background knowledge appropriately. This task often involves abduction (Tamaddoni-Nezhad *et al* 2006; Yamamoto *et al.* 2009), but causal rules given in the background knowledge are often incomplete. Brave induction can thus be useful to complete missing causal rules in these applications.

5.2 Requirements Engineering

Requirements engineering involves the elicitation of high-level stakeholder goals that are described by scenarios of desirable and undesirable system behavior. Alrajeh *et al.* (2007) introduce an ILP framework for inferring requirements from a set of scenarios and incomplete requirements specification. Scenarios represent examples of desirable and undesirable system behavior over time, while the requirements specification captures the initial but incomplete background knowledge of the envisioned system and its environment. The task is then to complete the specification by learning a set of missing requirements that cover all of the desirable scenarios but none of the undesired ones. Formally, the problem is specified as follows:

Given: a requirement specification $Spec$, a set Des of desirable scenarios, and a set Und of undesirable scenarios
Find: a set Pre of event precondition axioms satisfying the conditions:
- $Spec \cup Pre \models_M \neg P_u$ for any $P_u \in Und$,
- $Spec \cup Pre \not\models_M \neg P_d$ for any $P_d \in Des$,

where $\models_M$ means an entailment relation under an LTL[15] model M.

Any set of event precondition axioms that satisfy these two properties is said to be a correct extension of a requirements specification with respect to the given scenarios. The specification and scenarios are represented by *event calculus normal logic programs.* They compute Pre satisfying that $Spec \cup Pre$ has a *stable model* M such that every element in Und is false M and every element in Des is consistent with M. Incidentally, their program transformation produces a normal logic program $Spec \cup Pre$ which has a single stable model, but it is inherently a problem of brave induction.

5.3 Multiagent Negotiation

Negotiation is a process of reaching agreement between different agents. In a typical one-to-one negotiation, an agent makes a proposal on his/her request and the opponent agent decides whether it is acceptable or not. If a proposal is unacceptable as it is, an agent seeks conditions to accept it by extending his/her current belief to accommodate another agent's request.

Sakama (2008) formulates the process of building conditions in terms of induction. Given the current belief B of an agent and a proposal G of another agent, B could

[15] Linear Temporal Logic

accept G under the condition H if:

$$B \cup H \models G$$

where $B \cup H$ is consistent. Here, H is a condition that bridges the gap between the current belief of an agent and the request made by another agent. Viewing G as an observation, the problem of finding H is considered a process of building a hypothesis to explain G under B.

When B contains multiple minimal models or answer sets, however, the relation $B \cup H \models G$ is strong. This is because an agent would have alternative options for a deal, and the cautious inference requires that the proposal G must be satisfied in every possible option. To relax the condition, Sakama uses brave induction for negotiation. That is, B could accept G under the condition H if $B \cup H$ has an answer set satisfying G.

Example 5.1 Consider negotiation between a buyer and a seller. A seller agent has the knowledge base B which consists of the following rules:

$$product(pc, \$1500) \leftarrow not\ \neg\, product(pc, \$1500), \tag{19}$$

$$\leftarrow product(pc, x),\ product(pc, y),\ x \neq y, \tag{20}$$

$$\neg\, product(pc, \$1500) \leftarrow product(pc, x), x < \$1500,\ pay\_cash, \tag{21}$$

$$pay\_cash\,;\ pay\_card\ \leftarrow . \tag{22}$$

Here, the rule (19) represents that the normal price of a PC is 1500 USD. The rule (20) represents a constraint that the same pc cannot have different prices at the same time. The rule (21) represents if discount is made by payment with cash, the normal price is withdrawn. The rule (22) represents two options for payment. With this setting, B has two answer sets:

$$S_1 = \{\, product(pc, \$1500),\ pay\_cash \,\},$$
$$S_2 = \{\, product(pc, \$1500),\ pay\_card \,\},$$

which represent the seller's initial belief.

Next, suppose that a buyer proposes

$$G :\ product(pc, \$1300)$$

to the seller. As G is included in no answer set of B, the seller cannot accept G as it is. The seller then seeks a condition H to accept G and induces the hypothesis

$$H :\ product(pc, \$1300) \leftarrow pay\_cash.$$

Now $B \cup H$ has two answer sets:

$$S_3 = \{\, product(pc, \$1300),\ \neg\, product(pc, \$1500),\ pay\_cash \,\},$$
$$S_4 = \{\, product(pc, \$1500),\ pay\_card \,\},$$

of which S_3 satisfies G. Thus, H covers G under B in brave induction. Based on H, the seller returns the condition

$$G' :\ pay\_cash$$

as a counter-proposal.

6 Discussion

In the previous sections, we mainly compared brave induction with explanatory or cautious induction. Here, we compare brave induction with other forms of induction.

6.1 Learning from Satisfiability

De Raedt and Dehaspe (De Raedt 1997; De Raedt and Dehaspe 1997b) introduce the framework of *learning from satisfiability* (LFS). Given the background knowledge B and an observation O, a hypothesis H covers O under B in LFS iff $B \wedge H \wedge O$ is consistent. In other words, H covers O under B in LFS iff $B \wedge H$ has a model satisfying O. As already argued in the introduction, LFS is weaker than brave induction.

Proposition 6.1 *If a hypothesis H covers O under B in brave induction, H covers O under B in LFS.*

Proof The result holds by Proposition 2.3. □

The converse implication of Proposition 6.1 does not hold in general. Since brave induction is weaker than both explanatory and cautious induction (Proposition 2.1), the following relation holds.

$$\begin{aligned}&\text{explanatory induction} < \text{cautious induction}\\ &\qquad < \text{brave induction} < \text{learning from satisfiability}\end{aligned}$$

where $X < Y$ means that any solution of X is also a solution of Y, but not vice versa.

Compared with brave induction, LFS does not require the minimality of models. So any theory H becomes a solution as far as it is consistent with $B \wedge O$. Due to its weak condition, the hypotheses space for LFS is generally huge, and additional language bias would be necessary for practical usage. Brave induction is considered as a strengthened version of LFS, that is, we imposed the condition of *minimality* on models of $B \wedge H$ satisfying O. It is known that LFS does not satisfy the property of "conjunction of solutions" in Section 2.1 (De Raedt and Dehaspe 1997b). The following example illustrates that LFS also does not satisfy the property of "conjunction of observations".

Example 6.1 Let $B = \{p(a)\}$, $O_1 = \{q(a)\}$ and $O_2 = \{r(a)\}$. Then, $H = \{\, q(a) \vee r(a) \leftarrow p(a), \quad \leftarrow q(a) \wedge r(a)\,\}$ covers both O_1 and O_2 under B in LFS, but H does not cover $O_1 \wedge O_2$.

Thus, as for the properties of Table 1 in Section 2.1, LFS is the same as brave induction. In (De Raedt and Dehaspe 1997b) the authors say:

> One open question for further research is how learning from satisfiability (which employs a monotonic logic) could be used for inducing nonmonotonic logic programs.

In Definition 3.1, brave induction is defined as inducing hypothesis H such that $B \cup H$ has a consistent answer set S satisfying $O \subseteq S$. The definition is considered a strengthened version of LFS in nonmonotonic logic programs.

6.2 Learning from Interpretations and Confirmatory Induction

As argued in the introduction, *learning from interpretations* (LFI) realizes a different style of induction. When the background knowledge B is given as a definite clause theory, a hypothesis H *covers* O under B in LFI iff H is satisfied in the least model of $B \wedge O$ (De Raedt and Dehaspe 1997a). On the other hand, when B is a full clausal theory, Helft (1989) distinguishes two types of induction.[16] *Strong generalization* is a set H of clauses which are satisfied in every minimal model of $B \wedge O$, while weak generalization is a set H of clauses which are satisfied in some minimal model of $B \wedge O$.[17]

Example 6.2 Consider the background knowledge B and an observation O,

$$B: \ amarican(John) \vee english(John),$$
$$O: \ speak\_english(John).$$

Then,

$$H_1: \ american(x) \leftarrow speak\_english(x),$$
$$H_2: \ english(x) \leftarrow speak\_english(x)$$

are two weak generalizations, while

$$H_3: \ american(x) \vee english(x) \leftarrow speak\_english(x)$$

is a strong generalization.

In Example 6.2, none of H_1, H_2, and H_3 becomes a solution of brave induction. Solutions by brave induction are, for instance,

$$H_4: \ speak\_english(x) \leftarrow american(x),$$
$$H_5: \ speak\_english(x) \leftarrow english(x),$$

which are also solutions of LFI. Note that H_4 or H_5 does not become a solution of cautious induction, while $H_4 \cup H_5$ becomes a solution of cautious induction.

On the other hand, if the fact $american(Mary)$ is added to B, H_4 and H_5 are still solutions of brave induction but H_4 is not a solution of LFI anymore. This is because H_4 is not satisfied in $B \cup \{american(Mary)\}$. Thus, brave induction is neither stronger nor weaker than LFI. Generally speaking, LFI does not explain why particular individuals are observed under the background knowledge. In fact, LFI does not distinguish between B and O. Moreover, LFI assumes that all observations are completely specified, so that it has no mechanism of predicting unseen phenomena. This is in contrast to brave or cautious induction which has a mechanism of prediction.

Confirmatory induction or *descriptive induction* (Lachiche 2000) also builds hypotheses that are satisfied by observations. Given the background knowledge B and an observation O such that $B \wedge O$ is consistent, a hypothesis H *covers* O under B in confirmatory induction iff $Comp(B \wedge O) \models H$ where $Comp$ represents Clark's predicate completion (Clark 1978).

[16] Helft's semantics is often called *nonmonotonic ILP*, but we reserve the term for induction from nonmonotonic logic programs. Helft's semantics is similar to LFI in spirit (De Raedt and Dehaspe 1997a), while it is also viewed as an instance of confirmatory induction (De Raedt and Lavrač 1993; Lachiche 2000).

[17] Helft imposes additional conditions on the satisfiability of H in a model of $B \wedge O$, but we neglect them to make discussion simple.

Example 6.3 Given B and O of Example 6.2, $Comp(B \wedge O)$ becomes

$$amarican(x) \Leftrightarrow x = John \wedge \neg english(John),$$
$$english(x) \Leftrightarrow x = John \wedge \neg american(John),$$
$$speak\_english(x) \Leftrightarrow x = John,$$

together with the Clark's equality axioms. Thus, $H_1, \ldots, H_5$ are all solutions of confirmatory induction.

Like LFI, brave induction is neither stronger nor weaker than confirmatory induction in general.

6.3 Induction in Nonmonotonic Logic Programs

Otero (2001) introduces a framework for learning positive/negative examples in normal logic programs.[18] He considers *induction from several sets of examples* such that: given a normal logic program P and several sets of examples $E_1, \ldots, E_n$ where $E_i = E_i^+ \cup E_i^-$ $(1 \leq i \leq n)$, H is a solution of induction if there is a *stable model* M_i of $P \cup H$ such that $M_i \models E_i^+$ and $M_i \not\models E_i^-$ for each E_i.

Example 6.4 (Otero 2001) Consider the normal logic program B:

$$p \leftarrow not\ q$$

which has the unique stable model $\{p\}$. Given examples $E_1 = \{p\}$ and $E_2 = \{q\}$, $H = \{\, q \leftarrow not\, p \,\}$ becomes a solution as $B \cup H$ has two stable models E_1 and E_2.

In Otero's setting, examples are given as multiple sets and each set of examples is examined to be satisfied by a stable model of a program. This is different from the problem setting of brave induction which examines satisfaction of a single set of examples in an answer set of a program. Otero's framework reduces to the definition of cautious induction for a single set of examples.

Induction in answer set programming is introduced by Sakama (2005), which builds new rules to cover positive examples and uncover negative examples. More precisely, given an extended logic program B[19] and a ground literal L^+ (resp. L^-) as a positive (resp. negative) example, it finds a set H of rules such that

$$B \cup H \models L^+ \quad \text{and} \quad B \cup H \not\models L^-$$

and $B \cup H$ is consistent. This definition provides a logical framework of cautious induction in ASP.

Ray (2008) develops a nonmonotonic ILP system, called XHAIL, which combines abduction and induction for building hypotheses. The background theory is given as a normal logic program, and its semantics is given by the stable model semantics. Given examples, XHAIL first computes explanations by *brave abduction*. Next, XHAIL construct ground rules as hypotheses by putting abductive explanations in heads of rules

[18] A normal logic program is a logic program in which a rule can contain default negation but contains neither negative literals nor disjunction.

[19] Extended logic programs are a subclass of extended disjunctive programs such that a program contains no disjunction in the heads of rules.

and putting deductive consequences of B in bodies of rules. In this phase, mode declaration specifies atoms appearing in heads and bodies of possible hypotheses. Finally, the ground hypotheses is generalized in the inductive phase. The resulting hypothesis becomes a solution of brave induction because it is constructed from explanations of brave abduction. In this sense, it is said that XHAIL realizes brave induction. However, the paper (Ray 2008) does not mention any motivation of brave induction apart from technical reasons, nor investigate any formal property of brave induction. In fact, temporal theories in event calculus provided as a case study in (Ray 2008) always have a single stable model, and the result of brave induction coincides with that of cautious induction in this case study.

6.4 Further Extensions and Issues

In brave induction $B \wedge H$ has a minimal model in which an observation O is satisfied. In this case, H covers the *positive* observation O under B. In ILP, on the other hand, *negative* observations as well as positive ones are also handled. Given a negative observation N, it is required that H *uncovers* N under B. This condition is logically represented as $B \wedge H \not\models N$. Definition 2.1 is extended to handle negative observations as follows.

Definition 6.1 Let B be the background knowledge, P a positive observation, and N a negative observation. A hypothesis H is a solution of brave induction if $B \wedge H$ has a minimal model M such that $M \models P$ and $M \not\models N$.

Note that we are interested in minimal models in which a positive observation P is true, so a negative observation N is requested to be false in those minimal models. By putting $O = P \wedge \neg N$, the above definition reduces to Definition 2.1 and negative observations are handled within the framework of this paper.[20]

Example 6.5 Consider the background knowledge B and an observation O,

$$\begin{aligned} B :\ & amarican(John) \vee english(John), \\ & american(Mary), \\ & french(Nicolas). \\ P :\ & speak\_english(John), \\ N :\ & speak\_english(Nicolas). \end{aligned}$$

Then, putting $O = P \wedge \neg N$,

$$\begin{aligned} H_1 :\ & speak\_english(x) \leftarrow american(x), \\ H_2 :\ & speak\_english(x) \leftarrow english(x) \end{aligned}$$

are two solutions of brave induction.

[20] Strictly speaking, $\neg N$ requires *Skolemization* when a clausal theory N contains variables. For detailed technique, see (Inoue 2004).

In this paper, we introduced induction algorithms which produce clauses or rules that define more than one predicate. The problem is known as *multiple predicate learning* (MPL) (De Raedt and Lavrač 1996). In MPL the order of learning different clauses affects the results of learning tasks and even the existence of solutions, especially in the presence of negative observations (De Raedt and Lavrač 1993). As discussed in Section 2.1, however, brave induction is not adapted for incremental learning in general. Given an observation O containing multiple predicates, BRAIN computes a candidate hypothesis H based on the relation $B \wedge \neg O \models \neg H \vee F$ at once. Those hypotheses are verified by checking the consistency of $B \wedge H \wedge \neg F$.

Brave induction proposed in this paper uses minimal models as a semantical basis. Due to its minimality, however, it often fails to induce useful hypotheses.

Example 6.6 John and Mary are students. John takes the courses of mathematics and physics, and Mary takes the courses of mathematics and chemistry. The situation is represented by the background knowledge B and the observation O:

$$B: \ student(John) \wedge student(Mary),$$
$$O: \ math(John) \wedge math(Mary) \wedge physics(John) \wedge chemistry(Mary).$$

In this case, the clause

$$H: \quad math(x) \vee physics(x) \vee chemistry(x) \ \leftarrow \ student(x)$$

is *not* a solution of brave induction.

The problem of Example 6.6 is that $B \wedge H$ does not allow any student to take more than one course. In other words, disjunction is interpreted *exclusively* under the minimal model semantics. To allow H as a solution for explaining O, a semantics which allows *inclusive* interpretations is necessary. A semantics which allows both exclusive and inclusive interpretations of disjunction in a logic program is known as the *possible model semantics* (Sakama and Inoue 1994). For instance, the disjunctive clause $p \vee q$ has three possibles models $\{p\}$, $\{q\}$, and $\{p, q\}$, of which $\{p\}$ and $\{q\}$ are minimal models. Thus, the possible model semantics considers *non-minimal* models as well as minimal ones. Brave induction under the possible model semantics is defined by replacing minimal models with possible models in Definition 2.1. In Example 6.6, $B \wedge H$ has the possible models:

$$\{\, math(John),\ math(Mary),\ physics(John),\ chemistry(Mary)\,\},$$

so that H becomes a solution of brave induction for explaining O under the possible model semantics. Recently, it is known that the possible model semantics characterizes the semantics of *cardinality constraint rules* in ASP (Marek *et al.* 2007).[21]

7 Conclusion

This paper introduced a logical framework of brave induction and developed algorithms in both full clausal theories and answer set programming. The utility of brave induction in problem solving was illustrated in systems biology, requirements engineering, and

[21] For cardinality constraint rules, see Section 3.2.

multiagent negotiation. Brave induction is different from the existing frameworks for induction, and provides an intermediate solution between learning from satisfiability and explanatory induction. Compared with existing frameworks, brave induction has an advantage for managing incompleteness which may arise in the background knowledge, hypotheses and observations.

Brave and cautious inferences are widely used for commonsense reasoning from incomplete knowledge. In hypothetical reasoning, two different types of abduction under brave and cautious inferences are used in the literature. Since abduction and induction are both hypothetical reasoning which extend the background knowledge to explain observations, brave induction proposed in this paper has a right place and serves as a natural extension of brave abduction.

There are several directions for future work. From a theoretical viewpoint, this paper considered the minimal model semantics in clausal theories. Such a minimal model is defined by minimizing all predicates, but there is a notion of (P, Z)-minimal models in *circumscription* (McCarthy 1980) in which only some selected predicates P are minimized and some Z can be varied. Circumscriptive induction has been proposed in (Inoue and Saito 2004) by unifying descriptive and explanatory induction, so brave induction would be considered in the context of circumscriptive induction. From a computational viewpoint, the BRAIN procedure introduced in this paper is naive and needs further optimization. In particular, the introduction of inductive bias is important in practical setting. Implementing an efficient procedure for brave induction and validating its effect in practical applications are left for future work.

Acknowledgements The authors thank Yoshitaka Yamamoto for his consideration on the application to systems biology.

References

Almaas, E., Kovács, B., Vicsek, T., Oltvai, Z. N. and Barabási, A. L. (2004). Global organization of metabolic fluxes in the bacterium Escherichia coli. *Nature*, 427, 839–843.

Alrajeh, D., Ray, O., Russo, A and Uchitel, S. (2007). Extracting requirements from scenarios with ILP. In: *Proceedings of the 16th International Conference on Inductive Logic Programming*, Lecture Notes in Artificial Intelligence, 4455, 64–78, Springer.

Bossu, G. and Siegel, P. (1985). Saturation, nonmonotonic reasoning and the closed-world assumption. *Artificial Intelligence*, 25, 13–63.

Clark, K. L. (1978). Negation as failure. In: H. Gallaire and J. Minker (Eds.), *Logic and Data Bases* (pp. 293–322), Plenum, New York.

De Raedt, L. and Lavrač, N. (1993). The many faces of inductive logic programming. In: *Methodologies for Intelligent Systems, 7th International Symposium*, Lecture Notes in Computer Science, 689, 435–449, Springer.

De Raedt, L. and Lavrač, N. (1996). Multiple predicate learning in two inductive logic programming setting. *Journal of the IGPL*, 4(2), 227–254.

De Raedt, L. (1997). Logical settings for concept-learning. *Artificial Intelligence*, 95, 187–201.

De Raedt, L. and Dehaspe, L. (1997). Clausal discovery. *Machine Learning*, 26(2-3), 99–146.

De Raedt, L. and Dehaspe, L. (1997). Learning from satisfiability. In: *Proceedings of the 9th Dutch Conference on Artificial Intelligence* (pp. 303–312).

Eiter, T. and Gottlob, G. (1995). On the computational cost of disjunctive logic programming: propositional case. *Annals of Mathematics and Artificial Intelligence*, 15, 289–323.

Eiter, T., Gottlob, G. and Leone, N. (1997). Abduction from logic programs: semantics and complexity. *Theoretical Computer Science*, 189, 129–177.

Flach, P. A. (1996). Rationality postulates for induction. In: *Proceedings of the 6th International Conference on Theoretical Aspects of Rationality and Knowledge* (pp. 267–281), Morgan Kaufmann.

Flach, P. A. and Kakas, A. C. (2000). Abductive and inductive reasoning: background and issues. In: P. A. Flach and A. C. Kakas (Eds.). *Abduction and Induction — Essays on their Relation and Integration* (pp. 1–27), Kluwer Academic.

Gelfond, M., Przymusinska, H. and Przymusinski, T. (1989). On the relationship between circumscription and negation as failure. *Artificial Intelligence*, 38, 75–94.

Gelfond, M. and Lifschitz, V. (1991). Classical negation in logic programs and disjunctive databases. *New Generation Computing*, 9, 365–385.

Helft, N. (1989). Induction as nonmonotonic inference. In: *Proceedings of the 1st International Conference on Principles of Knowledge Representation and Reasoning* (pp. 149–156), Morgan Kaufmann.

Inoue, K. (1992). Linear resolution for consequence finding. *Artificial Intelligence*, 56, 301–353.

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

Inoue, K. (2002). Automated abduction. In: *Computational Logic: Logic Programming and Beyond. Essays in Honour of Robert A. Kowalski, Part II*, Lecture Notes in Artificial Intelligence, 2408, 311–341, Springer.

Inoue, K. (2004). Induction as consequence finding. *Machine Learning*, 55, 109–135.

Inoue, K. and H. Saito. (2004). Circumscription policies for induction. In: *Proceedings of the 14th International Conference on Inductive Logic Programming*, Lecture Notes in Artificial Intelligence, 3194, 164–179, Springer.

Lachiche, N. (2000). Abduction and induction from a non-monotonic reasoning perspective. In: P. A. Flach and A. C. Kakas (Eds.). *Abduction and Induction — Essays on their Relation and Integration* (pp. 107–116), Kluwer Academic.

Lifschitz, V. (2002). Answer set programming and plan generation. *Artificial Intelligence*, 138, 39–54.

Marek, V. M., Niemelä, I. and Truszczyński, M. (2007). Logic programs with monotone abstract constraint atoms. *Theory and Practice of Logic Programming*, 8(2), 167–199.

McCarthy, J. (1980). Circumscription – a form of nonmonotonic reasoning, *Artificial Intelligence*, 13, 27–39.

McDermott, D. (1982). Nonmonotonic logic II: nonmonotonic modal theories. *Journal of the ACM*, 29, 33–57.

Minker, J. (1982). On indefinite data bases and the closed world assumption. In: *Proceedings of the 6th International Conference on Automated Deduction*, Lecture Notes in Computer Science, 138 (pp. 292–308), Springer.

Muggleton, S. and De Raedt, R. (1994). Inductive logic programming: theory and methods. *Journal of Logic Programming*, 19/20, 629–679.

Muggleton, S. (1995). Inverse entailment and Progol. *New Generation Computing*, 13, 245–286.

Niemelä, I., Simons, P. and Soininen, T. (1999). Stable model semantics of weighted constraint rules. In: *Proceedings of the 5th International Conference on Logic Programming and Nonmonotonic Reasoning*, Lecture Notes in Artificial Intelligence, 1730, 317–331, Springer.

Nienhuys-Cheng, S.-H. and De Wolf, R. (1997). *Foundations of inductive logic programming*, Lecture Notes in Artificial Intelligence, 1228, Springer.

Otero, R. P. (2001). Induction of stable models. In: *Proceedings of the 11th International Conference on Inductive Logic Programming*, Lecture Notes in Artificial Intelligence, 2157, 193–205, Springer.

Plotkin, G. D. (1970). A note on inductive generalization. In: B. Meltzer and D. Michie (Eds.), *Machine Intelligence*, vol. 5 (pp. 153–63), Edinburgh University Press.

Ray, O. (2008). Nonmonotonic abductive inductive learning. *Journal of Applied Logic*, in print.

Reiter, R. (1978). On closed world databases. In: H. Gallaire and J. Minker (Eds.), *Logic and Data Bases* (pp. 55–76), Plenum, New York.

Sakama, C. and Inoue, K. (1994). An alternative approach to the semantics of disjunctive logic programs and deductive databases, *Journal of Automated Reasoning*, 13(1), 145–172.

Sakama, C. (2000). Inverse entailment in nonmonotonic logic programs. In: *Proceedings of the 10th International Conference on Inductive Logic Programming*, Lecture Notes in Artificial Intelligence, 1866, 209–224, Springer.

Sakama, C. (2001). Nonmonotonic inductive logic programming. In: *Proceedings of the 6th International Conference on Logic Programming and Nonmonotonic Reasoning*, Lecture Notes in Artificial Intelligence, 2173, 62–80, Springer.

Sakama, C. (2005). Induction from answer sets in nonmonotonic logic programs. *ACM Transactions on Computational Logic*, 6(2), 203–231.

Sakama, C. (2008). Inductive negotiation in answer set programming. In: *Proceedings of the 6th International Workshop on Declarative Agent Languages and Technologies*, Lecture Notes in Artificial Intelligence 5397, 143–160, Springer.

Sakama, C. and Inoue, K. (2008). Brave induction. In: *Proceedings of the 18th International Conference on Inductive Logic Programming*, Lecture Notes in Artificial Intelligence, 5194, 261–278, Springer.

Tamaddoni-Nezhad, A., Chaleil, R., Kakas, A. C. and Muggleton, S. (2006). Application of abductive ILP to learning metabolic network inhibition from temporal data. *Machine Learning*, 65, 209–230.

Yamamoto, Y., Inoue, K, and Doncescu, A. (2009). Integrating abduction and induction in biological inference using CF-Induction. In: H. Lodhi and S. Muggleton (Eds.), *Elements of Computational Systems Biology*, Wiley Book Series on Bioinformatics, John Wiley and Sons, in print.