



A FIXPOINT CHARACTERIZATION OF ABDUCTIVE LOGIC PROGRAMS*

KATSUMI INOUE AND CHIAKI SAKAMA†

▷ A new fixpoint semantics for abductive logic programs is provided, in which the belief models of an abductive program are characterized as the fixpoint of a disjunctive program obtained by a suitable program transformation. In the transformation, both negative hypotheses through negation as failure and positive hypotheses from the abducibles are dealt with uniformly. The result is further generalized to a fixpoint semantics for abductive extended disjunctive programs. These characterizations allow us to have a parallel bottom-up model generation procedure for computing abductive explanations from any (range-restricted and function-free) normal, extended, and disjunctive programs with integrity constraints. ◁

1. INTRODUCTION

Abduction is an inference to explanation. Recently, abduction has been recognized as a very important form of reasoning for logic programming as well as various AI problems. In [8, 12, 16, 21], abduction is expressed as an extension of logic programming. Eshghi and Kowalski [8] give an abductive interpretation of negation as failure [3] in the class of normal logic programs, and show a 1–1 correspondence between the *stable models* [13] of a normal logic program and the extensions of

*This is a revised and extended version of the paper [19] which was presented at the Tenth International Conference on Logic Programming, Budapest, Hungary, June 21–25, 1993.

†Department of Computer and Communication Sciences, Wakayama University, 930 Sakaedani, Wakayama 640, Japan. E-mail: sakama@sys.wakayam-u.ac.jp.

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

Received August 1994; accepted July 1995.

its associated abductive framework. Their approach is extended by [7, 21], and a comprehensive survey is found in [23]. Kakas and Mancarella [21] propose a framework of *abductive logic programming*, which is defined as a triple $\langle P, \Gamma, \mathcal{I} \rangle$, where P is a normal logic program, Γ is a set of abducible predicates, and \mathcal{I} is a set of integrity constraints. Then, a canonical model of $\langle P, \Gamma, \mathcal{I} \rangle$ (called *generalized stable model* or *belief model*) is defined as a stable model of $P \cup E$ which satisfies \mathcal{I} , where E is any set of ground atoms with predicates from Γ . On the other hand, Gelfond [12] proposes an abductive framework with an *extended disjunctive program* [14] P that allows disjunctions in heads and classical negation along with negation as failure. Further, Inoue [16] proposes a general framework for hypothetical reasoning, called a *knowledge system*, by allowing any two extended logic programs as P and Γ , and shows that every knowledge system can be transformed into a semantically equivalent abductive logic programming framework.

In all of the above frameworks, abduction is defined as a pair of *background knowledge* $P \cup \mathcal{I}$ (the program with integrity constraints) and *candidate hypotheses* Γ . Then, an important question for abductive logic programming framework is how each abductive framework can be represented by a *single* program. Namely, we would like to express meta-level information of candidate hypotheses at the object level, thereby obtaining a program which exactly reflects the meaning of the original abductive framework. Such an expression bridges the gap between abductive and usual (nonabductive) logic programming, and is useful for the computational aspect of abduction since we can apply any proof procedure for usual logic programs to programs transformed from abductive frameworks. Moreover, these transformations shed light on the relationships between different extensions of logic programming (including abduction, disjunction, and negation as failure), and clarify the expressive power of each language. Several studies have been devoted in this direction. For instance, Console et al. [4] characterize abductive frameworks through the completed programs, and Inoue [16] transforms a knowledge system into a single extended logic program.

On the other hand, Inoue et al. [17] have proposed program transformation techniques which translate a program containing negation as failure into a semantically equivalent *positive disjunctive program*, i.e., disjunctive programs containing neither negation as failure nor classical negation. These transformations show that negation and disjunction in logic programming have close relations in knowledge representation. Moreover, such transformations provide a constructive definition of stable models of a normal logic program or *answer sets* of an extended disjunctive program, and enable us to realize a bottom-up procedure to compute them based on *model generation* techniques [11, 26]. This procedure is formally characterized by a *fixpoint semantics* for extended disjunctive programs [34].

In this paper, we generalize the program transformation techniques of [17] for nonabductive programs to deal with abductive frameworks. We introduce a new translation from an abductive logic program into a positive disjunctive program, and show that the belief models of an abductive program can be characterized by the fixpoint closure of the transformed disjunctive program. In the transformation, both negative hypotheses through negation as failure and positive hypotheses from the abducibles are dealt with *uniformly*. This fixpoint characterization is further extended to a fixpoint semantics for *abductive extended disjunctive programs*, i.e., abductive programs that permit classical negation as well as disjunctions. For a procedural aspect of our fixpoint semantics, we also show that a

model generation procedure for positive disjunctive programs can be used as a sound and complete procedure for computing belief models for function-free and range-restricted programs.

This paper is organized as follows. Section 2 defines a framework for abductive logic programming. In Section 3, we successively present fixpoint theories for positive disjunctive programs, normal logic programs, abductive Horn programs, and abductive normal logic programs. These fixpoint theories are further generalized to a fixpoint semantics for abductive extended disjunctive programs in Section 4. Section 5 presents a model generation procedure for computing belief models. Some comparisons between our fixpoint framework and previously proposed approaches are discussed in Section 6, and the paper is concluded in Section 7.

2. MODEL THEORY FOR ABDUCTIVE LOGIC PROGRAMS

There are several definitions of abduction [2, 4, 7, 8, 12, 15, 16, 21, 24, 29]. The semantics of abduction we use here is based on the framework of Kakas and Mancarella [21]. As stated in Section 1, their abductive framework is given as a triple $\langle P, \Gamma, \mathcal{I} \rangle$, where P is a normal logic program, Γ is a set of abducible predicates, and \mathcal{I} is a set of integrity constraints. Compared with abduction based on first-order logic by [15, 29], Kakas and Mancarella define a program P not as first-order formulas, but as a normal logic program with negation as failure. This definition covers a more general class of programs than Console et al.'s *object-level* abduction [4] that is defined for hierarchical logic programs (see Section 6.2.1). Two different definitions by Gelfond [12] and Inoue [16] are more general than that by [21] in the sense that they allow more extended classes of programs for P and Γ . We will revisit such an extension in Section 4.

We define an *abductive normal logic program*¹ as a pair $\langle P, \Gamma \rangle$, in a way slightly different from Kakas and Mancarella's framework. Instead of separating integrity constraints \mathcal{I} from a program, we include them in a program P and do not distinguish them from other clauses. The main reason for this treatment is that we would like to check the consistency not by an extra mechanism for integrity checking, but within closure computation defined in the subsequent sections. For this purpose, we first give the syntax and the stable model semantics of normal logic programs with integrity constraints.

Definition 2.1. A *normal logic program* is a finite set of clauses that are either of the form

$$H \leftarrow B_1 \wedge \cdots \wedge B_m \wedge \text{not } B_{m+1} \wedge \cdots \wedge \text{not } B_n \quad (1)$$

or of the form

$$\leftarrow B_1 \wedge \cdots \wedge B_m \wedge \text{not } B_{m+1} \wedge \cdots \wedge \text{not } B_n, \quad (2)$$

where $n \geq m \geq 0$, and H and B_i s are atoms. The left-hand (right-hand) side of \leftarrow is called the *head* (*body*) of the clause. Each clause of the form (2) is called an

¹Normal logic programs are often called *general* logic programs in the literature. Similarly, abductive normal logic programs are called *abductive general logic programs* in the previous paper [19].

integrity constraint. An integrity constraint is also called a *negative clause* if it does not contain *not*, i.e., $m = n$. A *Horn program* is a normal logic program not containing *not*. A *definite program* is a Horn program not containing negative clauses.

REMARK 2.2. In Definition 2.1, we allow in a program integrity constraints as clauses with empty heads, which are not explicitly defined as such in [13]. While Kakas and Mancarella [21] define integrity constraints \mathcal{I} as first-order formulas separated from a program P , every integrity constraint in the form of a first-order formula F can be first characterized as a clause without a head $\leftarrow \text{not } F$, then can be translated into clauses using the transformation of [25]. For instance, an integrity constraint $p \supset q$ can be expressed by $\leftarrow p \wedge \text{not } q$.²

In the semantics of a normal logic program, a clause containing variables stands for the set of its ground instances. An *interpretation* of a program P is defined as a subset of \mathcal{HB} , where \mathcal{HB} denotes the Herbrand base for the language of P . An interpretation I satisfies a ground Horn clause $H \leftarrow B_1 \wedge \cdots \wedge B_m$ if $\{B_1, \dots, B_m\} \subseteq I$ implies $H \in I$. In particular, I satisfies a ground negative clause $\leftarrow B_1 \wedge \cdots \wedge B_m$ if $\{B_1, \dots, B_m\} \not\subseteq I$. For a Horn program P , the smallest interpretation satisfying every ground clause from P is called the *least model* of P . Note that the least model does not necessarily exist in the presence of negative clauses.

Definition 2.3. Let P be a normal logic program, and I an interpretation. The *reduct* P^I of P by I is defined as follows: A clause $H \leftarrow B_1 \wedge \cdots \wedge B_m$ (resp. $\leftarrow B_1 \wedge \cdots \wedge B_m$) is in P^I iff there is a ground clause $H \leftarrow B_1 \wedge \cdots \wedge B_m \wedge \text{not } B_{m+1} \wedge \cdots \wedge \text{not } B_n$ (resp. $\leftarrow B_1 \wedge \cdots \wedge B_m \wedge \text{not } B_{m+1} \wedge \cdots \wedge \text{not } B_n$) from P such that $\{B_{m+1}, \dots, B_n\} \cap I = \emptyset$.

Then, I is a *stable model* [13] of P if I is the least model of P^I .

Now, we define abductive normal logic programs and their semantics.

Definition 2.4. An *abductive normal logic program* is a pair $\langle P, \Gamma \rangle$, where P is a normal logic program, and Γ is a set of atoms from the language of P . We identify Γ with the set of all ground instances from Γ , and call each atom in Γ an *abducible*. Note that $\Gamma \subseteq \mathcal{HB}$.

When I is an interpretation of P and $E = I \cap \Gamma$, we often write I as I_E by specifying the abducibles E contained in I .

When P is a Horn program, $\langle P, \Gamma \rangle$ is called an *abductive Horn program*.

REMARK 2.5. Definition 2.4 is an extension of the definition by Kakas and Mancarella [21] to allow any normal logic program (with integrity constraints) in P , while [21] requires that abducibles may not appear in heads of clauses. Furthermore, we consider abducible *atoms* instead of abducible *predicates*, so that it may be the case that some instances of an atom can be abducibles, while other instances with the same predicate can be nonabducibles. See Example 2.8.

²When we allow classical negation in programs (Section 4), we need more clauses for the translation. For example, to express a first-order integrity constraint $p \supset q$, we need an additional clause $\leftarrow \neg q \wedge \text{not } \neg p$.

Definition 2.6. Let $\langle P, \Gamma \rangle$ be an abductive normal logic program. An interpretation I is a *belief model* of $\langle P, \Gamma \rangle$ if it is a stable model of a normal logic program $P \cup E$ for some $E \subseteq \Gamma$.³

A belief model I is Γ -*minimal* if no belief model J satisfies that $J \cap \Gamma \subset I \cap \Gamma$.

Each belief model in the above definition reduces to a stable model of P when $\Gamma = \emptyset$. In this sense, a belief model is called a *generalized stable model* by Kakas and Mancarella [21].

By definition, if I_E is a belief model, then $E = I_E \cap \Gamma$ holds. Similarly, if I_E is a Γ -minimal belief model, then there is no belief model J_F such that $F \subset E$. Since we allow abducibles in heads of ground clauses, such an abducible appearing in the head of a clause may be implied by other abducibles (see Example 2.8 below). In this case, each belief model I_E can be uniquely associated with its “generating” abducibles E . A similar notation has been adopted by Preist and Eshghi [30].

Definition 2.7. Let $\langle P, \Gamma \rangle$ be an abductive normal logic program, and O a ground atom called an *observation*. A set $E \subseteq \Gamma$ is an *explanation* of O (with respect to $\langle P, \Gamma \rangle$) if there is a belief model I_E of $\langle P, \Gamma \rangle$ such that $O \in I_E$.

An explanation E of O is *minimal* if no $E' \subset E$ is an explanation of O .

Example 2.8. Consider an abductive Horn program $\langle P, \Gamma_1 \rangle$, where P consists of

$$\begin{aligned} \text{sore}(\text{leg}) &\leftarrow \text{broken}(\text{leg}), \\ \text{broken}(\text{leg}) &\leftarrow \text{broken}(\text{tibia}), \end{aligned}$$

and $\Gamma_1 = \{\text{broken}(x)\}$. Let $O = \text{sore}(\text{leg})$ be an observation. Then

$$E = \{\text{broken}(\text{leg})\}$$

is a minimal explanation of O with respect to $\langle P, \Gamma_1 \rangle$, and

$$E' = \{\text{broken}(\text{tibia}), \text{broken}(\text{leg})\}$$

is a (nonminimal) explanation of O . However,

$$E'' = \{\text{broken}(\text{tibia})\}$$

is not called here an explanation of O with respect to $\langle P, \Gamma_1 \rangle$ since there is no belief model $I_{E''}$ satisfying $E'' = I_{E''} \cap \Gamma_1$. In fact, $\text{broken}(\text{tibia})$ causes $\text{broken}(\text{leg})$, so that E'' can never be the generating hypotheses of any belief model. Thus, the definition of (minimal) explanations is purely model-theoretic. In this case, the unique minimal explanation E reflects the fact that the evidence of $\text{broken}(\text{leg})$ is more likely than that of $\text{broken}(\text{tibia})$.

If the abducibles are given as $\Gamma_2 = \{\text{broken}(\text{tibia})\}$, then the above E'' is the only (minimal) explanation of O with respect to $\langle P, \Gamma_2 \rangle$, while neither E nor E' is an explanation of O by definition.

REMARK 2.9. Without loss of generality, we will assume that an observation O is a *nonabducible ground atom*. This condition is not restrictive for the following reasons. First, if O is an abducible, all of its explanations trivially contain O .

³For each abducible $A \in \Gamma$, we identify the atom A with the clause $A \leftarrow$ in E .

Second, if $O(\mathbf{x})$ contains a tuple of free variables \mathbf{x} , then we can introduce a new proposition O and add a clause $O \leftarrow O(\mathbf{x})$ to the program P so that O is treated as an observation. Third, we can ask the system why some atoms O_1, \dots, O_m are observed and other atoms O_{m+1}, \dots, O_n are not observed by introducing a clause $O \leftarrow O_1 \wedge \dots \wedge O_m \wedge \text{not } O_{m+1} \wedge \dots \wedge \text{not } O_n$ and computing explanations of O .

Definition 2.7 gives a credulous reading of the relationship between an observation and its explanations. We can give an alternative, skeptical meaning to an explanation of O : that is, every stable model of $P \cup E$ contains O .

Lemma 2.10. *Let $\langle P, \Gamma \rangle$ be an abductive normal logic program, E a subset of Γ , and O an observation.*

- (a) *E is an explanation of O with respect to $\langle P, \Gamma \rangle$ iff I_E is a belief model of $\langle P \cup \{\leftarrow \text{not } O\}, \Gamma \rangle$.*
- (b) *E is a minimal explanation of O with respect to $\langle P, \Gamma \rangle$ iff I_E is a Γ -minimal belief model of $\langle P \cup \{\leftarrow \text{not } O\}, \Gamma \rangle$.*

PROOF. (a) Immediately follows from the observation that the addition of $\leftarrow \text{not } O$ to P imposes the integrity constraint that O should be derived. This result is also stated by Satoh and Iwayama [35].

(b) E is a minimal explanation of O with respect to $\langle P, \Gamma \rangle$

iff no $E' \subset E$ is an explanation of O with respect to $\langle P, \Gamma \rangle$

iff no belief model $I_{E'}$ of $\langle P, \Gamma \rangle$ in which O is true satisfies $E' \subset E$

iff no belief model $I_{E'}$ of $\langle P \cup \{\leftarrow \text{not } O\}, \Gamma \rangle$ satisfies $E' \subset E$

iff I_E is a Γ -minimal belief model of $\langle P \cup \{\leftarrow \text{not } O\}, \Gamma \rangle$. \square

Example 2.11. Consider an abductive normal logic program $\langle P, \Gamma \rangle$ where

$$\begin{aligned} P &= \{p \leftarrow r \wedge b \wedge \text{not } q, \\ &\quad q \leftarrow a, \\ &\quad r \leftarrow, \\ &\quad \leftarrow \text{not } p\} \end{aligned}$$

and

$$\Gamma = \{a, b\}.$$

The unique belief model of $\langle P, \Gamma \rangle$ is $I_E = \{r, p, b\}$. If we regard $\leftarrow \text{not } p$ as an observation, $E = I_E \cap \Gamma = \{b\}$ is the unique explanation of p . Note here that we cannot add a to E because if we would abduce $E' = \{a, b\}$, q would block to derive p and the integrity constraint could not be satisfied. Hence, abduction is nonmonotonic relative to the addition of abducibles.

3. FIXPOINT THEORY FOR ABDUCTIVE LOGIC PROGRAMS

This section presents a fixpoint semantics for abductive normal logic programs. First, we introduce (i) a fixpoint semantics for positive disjunctive programs, then (ii) a fixpoint semantics for normal logic programs using a transformation into positive disjunctive programs by [17]. Next, (iii) a fixpoint semantics for abductive Horn programs is given using another program transformation, then finally it is extended

to (iv) a fixpoint semantics for abductive normal logic programs by combining the transformations of (ii) and (iii).

3.1. Fixpoint Semantics for Positive Disjunctive Programs

A *positive disjunctive program* is a finite set of clauses of the form

$$H_1 \vee \cdots \vee H_l \leftarrow B_1 \wedge \cdots \wedge B_m \quad (l, m \geq 0) \quad (3)$$

where H_i s and B_j s are atoms. An interpretation I *satisfies* a ground clause of the form (3) if $\{B_1, \dots, B_m\} \subseteq I$ implies $H_i \in I$ for some i ($1 \leq i \leq l$). I is a *minimal model* of P if it is a minimal interpretation satisfying all ground clauses from P .

To characterize the nondeterministic behavior of a disjunctive program, we define the following \mathbf{T}_P operator which operates over the set of all sets of interpretations. A similar but slightly different operator has been given by Sakama and Inoue [34].

Definition 3.1. Let P be a positive disjunctive program, and \mathbf{I} a set of interpretations. Then the mapping $\mathbf{T}_P : 2^{2^{\mathcal{H}B}} \rightarrow 2^{2^{\mathcal{H}B}}$ is defined as

$$\mathbf{T}_P(\mathbf{I}) = \bigcup_{I \in \mathbf{I}} T_P(I),$$

where the mapping $T_P : 2^{\mathcal{H}B} \rightarrow 2^{2^{\mathcal{H}B}}$ is defined as

$$T_P(I) = \begin{cases} \emptyset, & \text{if } \{B_1, \dots, B_m\} \subseteq I \text{ for some ground negative clause} \\ & \leftarrow B_1 \wedge \cdots \wedge B_m \text{ from } P; \\ \{J \mid \text{for each ground clause } C_i : H_1 \vee \cdots \vee H_{l_i} \leftarrow B_1 \wedge \cdots \wedge B_{m_i} \\ & \text{from } P \text{ such that } \{B_1, \dots, B_{m_i}\} \subseteq I \\ & \text{and } \{H_1, \dots, H_{l_i}\} \cap I = \emptyset, \\ & J = I \cup \bigcup_{C_i} \{H_j\} \ (1 \leq j \leq l_i)\}, & \text{otherwise.} \end{cases}$$

In particular, $\mathbf{T}_P(\emptyset) = \emptyset$.

The intuitive reading of Definition 3.1 is as follows. If an interpretation I does not satisfy some ground negative clause, then $T_P(I) = \emptyset$. Else, if there is a ground nonnegative clause C_i that is not satisfied by I (i.e., I satisfies the body of C_i but does not satisfy the head of C_i), then I is expanded by adding each single disjunct from the heads of every such C_i .

Definition 3.2. The ordinal powers of \mathbf{T}_P are defined as follows.

$$\begin{aligned} \mathbf{T}_P \uparrow 0 &= \{\emptyset\}, \\ \mathbf{T}_P \uparrow n + 1 &= \mathbf{T}_P(\mathbf{T}_P \uparrow n), \\ \mathbf{T}_P \uparrow \omega &= \bigcup_{\alpha < \omega} \bigcap_{\alpha \leq n < \omega} \mathbf{T}_P \uparrow n, \end{aligned}$$

where n is a successor ordinal and ω is a limit ordinal.

The above definition means that at the limit ordinal ω , the closure retains interpretations which are persistent in the preceding computation. That is, for any interpretation I in $\mathbf{T}_P \uparrow \omega$, there is an ordinal α smaller than ω such that, for every n ($\alpha \leq n < \omega$), I is included in $\mathbf{T}_P \uparrow n$. This closure definition is also used in [34] for computing possible models of positive disjunctive programs.

Example 3.3. For the following program P , $\mathbf{T}_P \uparrow \omega$ is obtained as follows.

$$\begin{aligned}
 P &= \{p \vee q \leftarrow r, \\
 &\quad s \leftarrow r, \\
 &\quad r \leftarrow, \\
 &\quad \leftarrow q \wedge s\}, \\
 \mathbf{T}_P \uparrow 1 &= \{\{r\}\}, \\
 \mathbf{T}_P \uparrow 2 &= \{\{r, s, p\}, \{r, s, q\}\}, \\
 \mathbf{T}_P \uparrow 3 &= \{\{r, s, p\}\} \\
 &= \mathbf{T}_P \uparrow \omega.
 \end{aligned}$$

Theorem 3.4. Let P be a positive disjunctive program.

- (a) $\mathbf{T}_P \uparrow \omega$ is a fixpoint.
- (b) Each element in $\mathbf{T}_P \uparrow \omega$ is a model of P .
- (c) Let \mathcal{MM}_P be the set of all minimal models of P . Then

$$\mathcal{MM}_P = \min(\mathbf{T}_P \uparrow \omega),$$

where

$$\min(\mathbf{I}) = \{I \in \mathbf{I} \mid \text{there is no } J \in \mathbf{I} \text{ such that } J \subset I\}.$$

PROOF. (a) The same as the proof of [34, Theorem 2.10].

(b) For any $I \in \mathbf{T}_P \uparrow \omega$, I satisfies each ground negative clause from P , and for any ground clause $H_1 \vee \cdots \vee H_l \leftarrow B_1 \wedge \cdots \wedge B_m$ from P , $\{B_1, \dots, B_m\} \subseteq I$ implies $A_i \in I$ for some i ($1 \leq i \leq l$). Therefore, I is a model of P .

(c) Since $\mathcal{MM}_P \supseteq \min(\mathbf{T}_P \uparrow \omega)$ is clear from (b), we show the other inclusion. Let I be a minimal model of P . Then for each atom A in I , there is a ground clause $H_1 \vee \cdots \vee H_l \leftarrow B_1 \wedge \cdots \wedge B_m$ from P such that $\{B_1, \dots, B_m\} \subseteq I$ and $A = H_i$ for some i ($1 \leq i \leq l$). By the definition of fixpoint construction, I is contained in $\mathbf{T}_P \uparrow \omega$. Since each element in $\mathbf{T}_P \uparrow \omega$ is a model of P , I is a minimal element of $\mathbf{T}_P \uparrow \omega$. Hence, $I \in \min(\mathbf{T}_P \uparrow \omega)$. \square

Corollary 3.5. A positive disjunctive program P is inconsistent (i.e., has no model) iff $\mathbf{T}_P \uparrow \omega = \emptyset$.

Corollary 3.6. For any definite program P , $\mathbf{T}_P \uparrow \omega$ contains a unique element which is the least model of P .

By definition, the fixpoint $\mathbf{T}_P \uparrow \omega$ always exists for any positive disjunctive program P , and is uniquely determined for each P . We call it the *disjunctive fixpoint* of P . Theorem 3.4 (c) characterizes a fixpoint construction of the *minimal model semantics* [27] for positive disjunctive programs. On the other hand, since Corollary 3.5 can be used as a test for the consistency of a positive disjunctive program, the emptiness of disjunctive fixpoints accounts for the soundness and completeness of *model generation theorem provers* [11, 17, 26] with respect to the satisfiability of first-order theories (see Section 5). Furthermore, Corollary 3.6 says

that, for definite programs, our fixpoint construction reduces to van Emden and Kowalski's fixpoint semantics [37].

3.2. Fixpoint Semantics for Normal Logic Programs

To characterize the stable models of a normal logic program, Inoue et al. [17] have proposed a program transformation which transforms a normal logic program into a semantically equivalent *not*-free disjunctive program.

Definition 3.7 [17]. Let P be a normal logic program. Then, P^κ is the program obtained as follows.

1. For each clause $H \leftarrow B_1 \wedge \cdots \wedge B_m \wedge \text{not } B_{m+1} \wedge \cdots \wedge \text{not } B_n$ in P ,

$$(H \wedge \neg KB_{m+1} \wedge \cdots \wedge \neg KB_n) \vee KB_{m+1} \vee \cdots \vee KB_n \leftarrow B_1 \wedge \cdots \wedge B_m \quad (4)$$

is in P^κ . In particular, each integrity constraint becomes

$$KB_{m+1} \vee \cdots \vee KB_n \leftarrow B_1 \wedge \cdots \wedge B_m.$$

2. For each atom B in \mathcal{HB} , P^κ includes the negative clause

$$\leftarrow \neg KB \wedge B. \quad (5)$$

Here, KB (resp. $\neg KB$) is a new *atom* which denotes B is believed (resp. disbelieved). In the transformation (4), each $\text{not } B_i$ is rewritten in $\neg KB_i$ and shifted to the head of the clause. Moreover, since the head H becomes true when each $\neg KB_i$ in the body is true, the condition $\neg KB_{m+1} \wedge \cdots \wedge \neg KB_n$ is added to H . The integrity constraint (5) says that each atom B cannot be true and disbelieved at the same time.

An *interpretation* I^κ of the transformed program is now defined as a subset of the new Herbrand base:

$$\mathcal{HB}^\kappa = \mathcal{HB} \cup \{KB \mid B \in \mathcal{HB}\} \cup \{\neg KB \mid B \in \mathcal{HB}\}.$$

An atom in \mathcal{HB}^κ is called *objective* if it is in \mathcal{HB} , and the set of objective atoms in an interpretation I^κ is denoted as $\text{obj}(I^\kappa)$. Note here that we consider KB and $\neg KB$ not as new formulas in a suitable modal logic, but as newly introduced atoms in the new program. The meaning of $\neg KB$ is given by the formula (5), while that of KB imposes the following *canonical constraint*.

Definition 3.8. An interpretation I^κ is *canonical* if it satisfies the condition: for each ground atom A , if $KA \in I^\kappa$, then $A \in I^\kappa$. For a set \mathbf{I}^κ of interpretations, we write

$$\text{obj}_c(\mathbf{I}^\kappa) = \{\text{obj}(I^\kappa) \mid I^\kappa \in \mathbf{I}^\kappa \text{ and } I^\kappa \text{ is canonical}\}.$$

In [17], it is shown that the stable models of a program can be produced *constructively* from the transformed program. To characterize their result by using the disjunctive fixpoint of the transformed program, we have to deal with a program like P^κ in Definition 3.7, which allows a disjunction of conjunctions of atoms in the head of a clause. Semantically, such a clause can be decomposed into a set of clauses of the form (3). Formally, a clause of the form

$$(H_{1,1} \wedge \cdots \wedge H_{1,k_1}) \vee \cdots \vee (H_{l,1} \wedge \cdots \wedge H_{l,k_l}) \leftarrow B_1 \wedge \cdots \wedge B_m. \quad (6)$$

stands for the $k_1 \times k_2 \times \cdots \times k_l$ clauses

$$H_{1,i_1} \vee H_{2,i_2} \vee \cdots \vee H_{l,i_l} \leftarrow B_1 \wedge \cdots \wedge B_m, \quad (7)$$

for every $i_1 = 1, \dots, k_1, i_2 = 1, \dots, k_2, \dots, i_l = 1, \dots, k_l$. In this sense, we will regard a program consisting of clauses of the form (6) also as a *positive disjunctive program*. For example, the program P^κ translated from a normal logic program P by Definition 3.7 is a positive disjunctive program.

When a clause of the form (6) is processed, the mapping presented in Definition 3.1 can be obviously applied to the multiple clauses of the form (7) whose bodies are exactly the same. Instead of doing so, we here slightly modify the mapping to manipulate a disjunction of conjunctions of atoms in the head directly, so that the clause (6) can be dealt with very efficiently. Now, for a conjunction of atoms $F = H_1 \wedge \cdots \wedge H_k$, we denote the set of its conjuncts as $\text{conj}(F) = \{H_1, \dots, H_k\}$. Let P be a program consisting of clauses of the form (6), and I an interpretation. The mapping $T_P : 2^{\mathcal{HB}} \rightarrow 2^{2^{\mathcal{HB}}}$ in Definition 3.1 is now redefined as

$$T_P(I) = \begin{cases} \emptyset, & \text{if } \{B_1, \dots, B_m\} \subseteq I \text{ for some ground negative clause} \\ & \leftarrow B_1 \wedge \cdots \wedge B_m \text{ from } P; \\ \{J \mid \text{for each ground clause } C_i : F_i \vee \cdots \vee F_{l_i} \leftarrow B_1 \wedge \cdots \wedge B_{m_i} \\ & \text{from } P \text{ such that } \{B_1, \dots, B_{m_i}\} \subseteq I \text{ and} \\ & \text{conj}(F_j) \not\subseteq I \text{ for any } j = 1, \dots, l_i, \\ & J = I \cup \bigcup_{C_i} \text{conj}(F_j) \ (1 \leq j \leq l_i)\}, & \text{otherwise.} \end{cases}$$

Using this definition, the mapping \mathbf{T}_P and its disjunctive fixpoint are also defined in the same way as in Section 3.1, and those properties presented there still hold. In particular, $\mathcal{MM}_P = \min(\mathbf{T}_P \uparrow \omega)$ (Theorem 3.4 (c)) holds.

The following theorem presents the fixpoint characterization of the stable model semantics for normal logic programs.

Theorem 3.9 [17, 34]. *Let P be a normal logic program, P^κ its transformed form, and ST_P the set of all stable models of P . Then*

$$ST_P = \text{obj}_c(\mathbf{T}_{P^\kappa} \uparrow \omega).$$

In particular, P has no stable model iff $\text{obj}_c(\mathbf{T}_{P^\kappa} \uparrow \omega) = \emptyset$.

Example 3.10. Let P be the normal logic program consisting of the clauses

$$\begin{aligned} p &\leftarrow \text{not } q, \\ q &\leftarrow \text{not } p, \\ r &\leftarrow q, \\ r &\leftarrow \text{not } r. \end{aligned}$$

Then

$$\begin{aligned} P^\kappa &= \{(p \wedge \neg Kq) \vee Kq \leftarrow, \\ & (q \wedge \neg Kp) \vee Kp \leftarrow, \\ & r \leftarrow q, \\ & (r \wedge \neg Kr) \vee Kr \leftarrow\} \\ &\cup \{\leftarrow \neg KB \wedge B \mid B \in \mathcal{HB}\}. \end{aligned}$$

Now,

$$\mathbf{T}_{P^\kappa} \uparrow 0 = \{\emptyset\},$$

$$\begin{aligned} \mathbf{T}_{P^\kappa} \uparrow 1 = & \{\{p, \neg Kq, q, \neg Kp, r, \neg Kr\}, \{p, \neg Kq, q, \neg Kp, Kr\}, \{p, \neg Kq, Kp, r, \neg Kr\}, \\ & \{p, \neg Kp, Kp, Kr\}, \{Kq, q, \neg Kp, r, \neg Kr\}, \{Kq, q, \neg Kp, Kr\}, \\ & \{Kq, Kp, r, \neg Kr\}, \{Kq, Kp, Kr\}\}, \end{aligned}$$

$$\mathbf{T}_{P^\kappa} \uparrow 2 = \{\{p, \neg Kq, Kp, Kr\}, \{Kq, q, \neg Kp, Kr, r\}, \{Kq, Kp, Kr\}\},$$

$$\mathbf{T}_{P^\kappa} \uparrow 3 = \mathbf{T}_{P^\kappa} \uparrow 2 = \mathbf{T}_{P^\kappa} \uparrow \omega.$$

In $\mathbf{T}_{P^\kappa} \uparrow \omega$, only the second element $\{Kq, q, \neg Kp, Kr, r\}$ is canonical. Hence, $\text{obj}_c(\mathbf{T}_{P^\kappa} \uparrow \omega) = \{\{q, r\}\}$, and $\{q, r\}$ is the unique stable model of P .

3.3. Fixpoint Semantics for Abductive Horn Programs

The basic idea behind the transformation presented in the previous subsection (Definition 3.7) is that we hypothesize the *epistemic* statement about an atom B to evaluate the negation-as-failure formula *not* B . Namely, we assume that B *should not* (or *should*) hold at the fixpoint. The correctness of the *negative hypothesis* $\neg KB$ is checked through the integrity constraint $\leftarrow \neg KB \wedge B$ during the fixpoint construction, while for the *positive hypothesis* KB , its integrity checking is carried out by the *canonical constraint* that all the “assumed” literals are actually “derived” at the fixpoint (Definition 3.8).

Now, we move on to abduction. We first present a transformation of an abductive Horn program. Each abducible can also be treated as an epistemic hypothesis as in the previous transformation. Thus, we can assume that each abducible is either true or false at the fixpoint in order to explain the observation. The only difference between the epistemic hypotheses from abducibles and those from negation-as-failure formulas is that *the positive hypothesis* KA for each abducible A *should always satisfy the canonical constraint*. This is because we can abduce the truth of A whenever A should be true but is not deductively derived from the program. Then, a natural translation of abductive Horn programs is as follows.

Let $\langle P, \Gamma \rangle$ be an abductive Horn program. The program P_Γ^c is obtained from $\langle P, \Gamma \rangle$ by replacing each Horn clause in P

$$H \leftarrow B_1 \wedge \cdots \wedge B_m \wedge A_1 \wedge \cdots \wedge A_n \quad (m, n \geq 0),$$

where B_i s are nonabducibles and A_j s are abducibles, with

$$(H \wedge KA_1 \wedge \cdots \wedge KA_n) \vee \neg KA_1 \vee \cdots \vee \neg KA_n \leftarrow B_1 \wedge \cdots \wedge B_m, \quad (8)$$

and by adding two clauses

$$\leftarrow \neg KA \wedge A, \quad (9)$$

$$A \leftarrow KA, \quad (10)$$

for each abducible A in Γ .

We can see that the clause (8) transformed from an abductive Horn program and the clause (4) transformed from a normal logic program are *dual* in the sense that an abduced atom A is dealt with as a positive hypothesis KA , while a negation-as-failure formula *not* B is dealt with as a negative hypothesis $\neg KB$. Moreover,

the constraints (9) and (5) are exactly the same, and they are commonly used. Here, however, we have the additional clause (10) for each abducible A . Since this clause derives A whenever an interpretation contains the positive hypothesis KA , it makes every interpretation in $\mathbf{T}_{P_{\Gamma}^{\varepsilon}} \uparrow \omega$ satisfy the canonical constraint. In other words, for the positive hypothesis KA for each abducible A , *we do not need the canonical constraint*. The above transformation can thus be rewritten by omitting each clause (10) as follows.

Definition 3.11. Let $\langle P, \Gamma \rangle$ be an abductive Horn program. Then, P_{Γ}^{κ} is the program obtained as follows.

1. For each Horn clause in P

$$H \leftarrow B_1 \wedge \cdots \wedge B_m \wedge A_1 \wedge \cdots \wedge A_n \quad (m, n \geq 0), \quad (11)$$

where B_i s are nonabducibles and A_j s are abducibles,

$$(H \wedge A_1 \wedge \cdots \wedge A_n) \vee \neg KA_1 \vee \cdots \vee \neg KA_n \leftarrow B_1 \wedge \cdots \wedge B_m \quad (12)$$

is in P_{Γ}^{κ} . In particular, each integrity constraint becomes

$$\neg KA_1 \vee \cdots \vee \neg KA_n \leftarrow B_1 \wedge \cdots \wedge B_m.$$

2. For each abducible A in Γ , P_{Γ}^{κ} contains the negative clause

$$\leftarrow \neg KA \wedge A.$$

Note in the transformation (12) that each hypothesis A_j can be considered to be *skipped* instead of being resolved. In fact, this operation is a bottom-up counterpart of the ‘‘Skip & Cut’’ rule in SOL-S resolution [15] that is a top-down abductive procedure. In this way, each abduced atom can be added to an interpretation without imposing the condition that it should be derived. The next lemma shows that two transformations, P_{Γ}^{ε} and P_{Γ}^{κ} , are equivalent in the sense that both fixpoints are the same as far as the objective atoms are concerned. Hence, we will use the transformation P_{Γ}^{κ} for an abductive Horn program P in the rest of this paper.

Lemma 3.12. Let $\langle P, \Gamma \rangle$ be an abductive Horn program. Then, $\text{obj}_c(\mathbf{T}_{P_{\Gamma}^{\varepsilon}} \uparrow \omega) = \{\text{obj}(I^{\kappa}) \mid I^{\kappa} \in \mathbf{T}_{P_{\Gamma}^{\varepsilon}} \uparrow \omega\} = \{\text{obj}(I^{\kappa}) \mid I^{\kappa} \in \mathbf{T}_{P_{\Gamma}^{\kappa}} \uparrow \omega\}$.

PROOF. Straightforward from the above discussion. \square

REMARK 3.13. In the translation from (11) into (12) in Definition 3.11, when some instances of A_j ($1 \leq j \leq n$) are abducibles but some are not, P_{Γ}^{κ} includes the original clause (11) as well as the transformed clauses of the form (12) with those abducible instances. For example, suppose that $a(0)$ is an abducible but $a(s(0)), a(s(s(0))), \dots$ are not, and that the program P contains the clause $p(x) \leftarrow a(x)$. Then, P_{Γ}^{κ} contains both of the clauses

$$\begin{aligned} p(x) &\leftarrow a(x), \\ (p(0) \wedge a(0)) \vee \neg Ka(0) &\leftarrow . \end{aligned}$$

In this case, the first clause has the instance $p(0) \leftarrow a(0)$, which is unnecessary in the presence of the second clause. Although the precise translation of the first

clause might be

$$p(x) \leftarrow a(x) \wedge x \neq 0,$$

we can keep the original clause as it is. This is because the clause $p(0) \leftarrow a(0)$ is always satisfied by any interpretation in $\mathbf{T}_{P_{\Gamma}^{\kappa}} \uparrow \omega$.

The relationship between the belief models of P and the disjunctive fixpoint of the transformed program P_{Γ}^{κ} is given as follows.

Lemma 3.14. *Let $\langle P, \Gamma \rangle$ be an abductive Horn program.*

- (a) *For any $I^{\kappa} \in \mathbf{T}_{P_{\Gamma}^{\kappa}} \uparrow \omega$, $\text{obj}(I^{\kappa})$ is a belief model of $\langle P, \Gamma \rangle$.*
- (b) *For any belief model I_E of $\langle P, \Gamma \rangle$, there is a belief model $I_{E'}$ of $\langle P, \Gamma \rangle$ such that $E' \subseteq E$, $I_{E'} \setminus E' = I_E \setminus E$, and $I_{E'} = \text{obj}(I^{\kappa})$ for some $I^{\kappa} \in \mathbf{T}_{P_{\Gamma}^{\kappa}} \uparrow \omega$.*

PROOF. (a) Suppose that I^{κ} is an interpretation in $\mathbf{T}_{P_{\Gamma}^{\kappa}} \uparrow \omega$. Let $E = \text{obj}(I^{\kappa}) \cap \Gamma$, and P' the definite program obtained from P by removing every negative clause. By Corollary 3.6, $\mathbf{T}_{P' \cup E} \uparrow \omega$ contains the unique element I . Then, for each ground clause

$$H \leftarrow B_1 \wedge \cdots \wedge B_m \wedge A_1 \wedge \cdots \wedge A_n \quad (A_j\text{s are abducibles})$$

from P' , if $\{B_1, \dots, B_m\} \subseteq I$, then either $\{A_1, \dots, A_n, H\} \subseteq I$ or $A_j \notin I$ for some j ($1 \leq j \leq n$). Also, for the corresponding ground clause

$$(H \wedge A_1 \wedge \cdots \wedge A_n) \vee \neg KA_1 \vee \cdots \vee \neg KA_n \leftarrow B_1 \wedge \cdots \wedge B_m$$

from P_{Γ}^{κ} , if $\{B_1, \dots, B_m\} \subseteq I$, then either $\{A_1, \dots, A_n, H\} \subseteq I^{\kappa}$ or $\neg KA_j \in I^{\kappa}$ for some j ($1 \leq j \leq n$). Hence, $I = \text{obj}(I^{\kappa})$. Recall that I is the least model of $P' \cup E$. Now, suppose to the contrary that I is not the least model of $P \cup E$. Since $P \setminus P'$ is a set of negative clauses, $P \cup E$ has no model. Then, there is a ground negative clause

$$\leftarrow B_1 \wedge \cdots \wedge B_m \wedge A_1 \wedge \cdots \wedge A_n \quad (A_j\text{s are abducibles})$$

from P such that $\{B_1, \dots, B_m\} \subseteq I$ and $\{A_1, \dots, A_n\} \subseteq E$. In this case, there must be the corresponding ground clause

$$\neg KA_1 \vee \cdots \vee \neg KA_n \leftarrow B_1 \wedge \cdots \wedge B_m$$

from P_{Γ}^{κ} . Since I^{κ} is a model of P_{Γ}^{κ} by Theorem 3.4(b), $\{B_1, \dots, B_m\} \subseteq I^{\kappa}$ implies $\neg KA_i \in I^{\kappa}$ for some i ($1 \leq i \leq n$). But this is impossible because $\{A_1, \dots, A_n\} \subseteq E \subseteq I \subseteq I^{\kappa}$ and I^{κ} satisfies all the negative clauses of the form $\leftarrow \neg KA \wedge A$. Hence, I is the least model of $P \cup E$, and thus the stable model of $P \cup E$. By definition, I is a belief model of $\langle P, \Gamma \rangle$.

(b) Suppose that I_E is a belief model of $\langle P, \Gamma \rangle$. For any atom $H_i \in I_E \setminus E$, there is a ground clause

$$C_i: H_i \leftarrow B_1 \wedge \cdots \wedge B_{m_i} \wedge A_1 \wedge \cdots \wedge A_{n_i} \quad (A_j\text{s are abducibles})$$

from P such that $\{B_1, \dots, B_{m_i}\} \subseteq I_E \setminus E$ and $\{A_1, \dots, A_{n_i}\} \subseteq E$. Let

$$E' = \bigcup_{H_i \in I_E \setminus E} \{A_1, \dots, A_{n_i}\}.$$

For C_i , there is the corresponding ground clause

$$(H_i \wedge A_1 \wedge \cdots \wedge A_{n_i}) \vee \neg K A_1 \vee \cdots \vee \neg K A_{n_i} \leftarrow B_1 \wedge \cdots \wedge B_{m_i}$$

from P_Γ^κ . Therefore, if $\{B_1, \dots, B_{m_i}\} \subseteq J$ for some $J \in \mathbf{T}_{P_\Gamma^\kappa} \uparrow \alpha$ and some ordinal α , then there exists $J' \in \mathbf{T}_{P_\Gamma^\kappa} \uparrow \alpha + 1$ such that $J \cup \{H_i, A_1, \dots, A_{n_i}\} \subseteq J'$. Since $\{H_i, A_1, \dots, A_{n_i}\} \subseteq I_E$ and I_E is a stable model of $P \cup E$, J' satisfies each negative clause in P_Γ^κ and is not pruned away. Hence, there exists $I^\kappa \in \mathbf{T}_{P_\Gamma^\kappa} \uparrow \omega$ such that $E' = I^\kappa \cap \Gamma$. By (a), $I_{E'} = \text{obj}(I^\kappa)$ is a belief model of $\langle P, \Gamma \rangle$. It follows immediately that $E' \subseteq E$ and $I_{E'} \setminus E' = I_E \setminus E$. \square

Lemma 3.14 characterizes the belief model semantics for abductive Horn programs. Namely, part (a) shows that every interpretation obtained from $\mathbf{T}_{P_\Gamma^\kappa} \uparrow \omega$ is a belief model of $\langle P, \Gamma \rangle$. Conversely, part (b) shows that every Γ -minimal belief model of $\langle P, \Gamma \rangle$ can be obtained from $\mathbf{T}_{P_\Gamma^\kappa} \uparrow \omega$.

REMARK 3.15. The completeness result by Lemma 3.14 (b) does not guarantee that every belief model itself can be obtained from $\mathbf{T}_{P_\Gamma^\kappa} \uparrow \omega$. In particular, non- Γ -minimal belief models are not obtainable in general. For example, if

$$P = \{p \leftarrow q \wedge a\}$$

and $\Gamma = \{a\}$, then $\{a\}$ is a belief model of $\langle P, \Gamma \rangle$, but for

$$P_\Gamma^\kappa = \{(p \wedge a) \vee \neg K a \leftarrow q, \leftarrow \neg K a \wedge a\},$$

$\mathbf{T}_{P_\Gamma^\kappa} \uparrow \omega = \{\emptyset\}$. Thus, belief models not obtainable from the disjunctive fixpoint have the property that the introduction of abducibles has no effect on the status of other atoms. Since such belief models are of no use for explaining observations, this kind of incompleteness is not a drawback. In other words, all the meaningful belief models are obtained at the fixpoint.

Similarly to Lemma 3.14 (b), we have the following completeness result for the minimal explanations of any observation.

Lemma 3.16. Let $\langle P, \Gamma \rangle$ be an abductive Horn program, O an observation. If $E \subseteq \Gamma$ is an explanation of O , then there is an explanation E' of O such that $E' \subseteq E$ and $I_{E'} = \text{obj}(I^\kappa)$ for some $I^\kappa \in \mathbf{T}_{P_\Gamma^\kappa} \uparrow \omega$.

PROOF. Since E is an explanation of O , there is a belief model I_E of $\langle P, \Gamma \rangle$ satisfying O . By Lemma 3.14 (b), there is a belief model $I_{E'}$ of $\langle P, \Gamma \rangle$ such that $E' \subseteq E$, $I_{E'} \setminus E' = I_E \setminus E$, and $I_{E'} = \text{obj}(I^\kappa)$ for some $I^\kappa \in \mathbf{T}_{P_\Gamma^\kappa} \uparrow \omega$. Since O is in $I_E \setminus E$, it is also in $I_{E'} \setminus E'$. Hence, E' is an explanation of O . \square

Example 3.17 [29, 18]. Suppose that P consists of the clauses

$$\begin{aligned} \text{sneeze}(x) &\leftarrow \text{person}(x) \wedge \text{cold}(x), \\ \text{sneeze}(x) &\leftarrow \text{person}(x) \wedge \text{hay-fever}(x), \\ \text{person}(\text{Tom}) &\leftarrow, \\ &\leftarrow \text{person}(x) \wedge \text{cold}(x) \wedge \text{hay-fever}(x), \end{aligned}$$

and the abducibles are $\Gamma = \{cold(x), hay-fever(x)\}$. Then, the abductive Horn program $\langle P, \Gamma \rangle$ is transformed into the following positive disjunctive program P_F^κ :

$$\begin{aligned} & (cold(x) \wedge sneeze(x)) \vee \neg K cold(x) \leftarrow person(x), \\ & (hay-fever(x) \wedge sneeze(x)) \vee \neg K hay-fever(x) \leftarrow person(x), \\ & person(Tom) \leftarrow, \\ & \neg K cold(x) \vee \neg K hay-fever(x) \leftarrow person(x), \\ & \leftarrow \neg KA \wedge A \quad \text{for every abducible } A. \end{aligned}$$

Let $O = sneeze(Tom)$ be the observation. Then

$$\mathbf{T}_{P_F^\kappa} \uparrow \omega = \{M_1, M_2, M_3\},$$

where

$$\begin{aligned} M_1 &= \{person(Tom), cold(Tom), sneeze(Tom), \neg K hay-fever(Tom)\}, \\ M_2 &= \{person(Tom), \neg K cold(Tom), hay-fever(Tom), sneeze(Tom)\}, \\ M_3 &= \{person(Tom), \neg K cold(Tom), \neg K hay-fever(Tom)\}. \end{aligned}$$

By extracting the abducibles from M_1 and M_2 , we can get the two explanations of O , $E_1 = \{cold(Tom)\}$ and $E_2 = \{hay-fever(Tom)\}$.

3.4. Fixpoint Semantics for Abductive Normal Logic Programs

Now, we show a transformation of abductive normal logic programs by combining the two transformations shown in Sections 3.2 and 3.3. Each negation-as-failure formula *not* B for a nonabducible B is translated in the same way as Definition 3.7: it is split into $\neg KB$ and KB . On the other hand, when a negation-as-failure formula *not* A mentions an abducible A , it should be split into $\neg KA$ and A . This is because, for each abducible A , we can deal with it as if the axiom (10) $A \leftarrow KA$ is present.

Definition 3.18. Let $\langle P, \Gamma \rangle$ be an abductive normal logic program. Then, P_F^κ is the program obtained as follows.

1. For each clause

$$\begin{aligned} H \leftarrow & B_1 \wedge \cdots \wedge B_m \wedge A_1 \wedge \cdots \wedge A_n \\ & \wedge \text{not } B_{m+1} \wedge \cdots \wedge \text{not } B_s \wedge \text{not } A_{n+1} \wedge \cdots \wedge \text{not } A_t \end{aligned} \quad (13)$$

in P , where $s \geq m \geq 0$, $t \geq n \geq 0$, B_j s are nonabducibles, and A_k s are abducibles,

$$\begin{aligned} & \left(H \wedge \bigwedge_{i=1}^n A_i \wedge \bigwedge_{j=m+1}^s \neg KB_j \wedge \bigwedge_{k=n+1}^t \neg KA_k \right) \\ & \vee \bigvee_{i=1}^n \neg KA_i \vee \bigvee_{j=m+1}^s KB_j \vee \bigvee_{k=n+1}^t A_k \leftarrow B_1 \wedge \cdots \wedge B_m \end{aligned} \quad (14)$$

is in P_{Γ}^{κ} . In particular, each integrity constraint is transformed into

$$\begin{aligned} & \neg KA_1 \vee \cdots \vee \neg KA_n \vee KB_{m+1} \vee \cdots \vee KB_s \\ & \vee A_{n+1} \vee \cdots \vee A_t \leftarrow B_1 \wedge \cdots \wedge B_m. \end{aligned}$$

2. For each atom H in \mathcal{HB} , P_{Γ}^{κ} includes the negative clause

$$\leftarrow \neg KH \wedge H.$$

Notice that a transformed program P_{Γ}^{κ} in Definition 3.18 reduces to the program P^{κ} in Section 3.2 when Γ is empty, and reduces to the program P_{Γ}^{κ} in Section 3.3 when P is a Horn program.

Lemma 3.19. *Let $\langle P, \Gamma \rangle$ be an abductive normal logic program, and E a subset of Γ . Then, I_E is a belief model of $\langle P, \Gamma \rangle$ iff I_E is a belief model of $\langle P^{I_E}, \Gamma \rangle$.*

PROOF. I_E is a belief model of $\langle P, \Gamma \rangle$

- iff I_E is a stable model of $P \cup E$ and $E = I_E \cap \Gamma$
- iff I_E is the least (and stable) model of $P^{I_E} \cup E^{I_E}$ and $E = I_E \cap \Gamma$
- iff I_E is a belief model of $\langle P^{I_E}, \Gamma \rangle$ (because $E^{I_E} = E$). \square

Lemma 3.20. *Let $\langle P, \Gamma \rangle$ be an abductive normal logic program.*

- (a) *For any $I \in \text{obj}_c(\mathbf{T}_{P_{\Gamma}^{\kappa}} \uparrow \omega)$, I is a belief model of $\langle P, \Gamma \rangle$.*
- (b) *For any belief model I_E of $\langle P, \Gamma \rangle$, there exists a belief model $I_{E'}$ of $\langle P, \Gamma \rangle$ in $\text{obj}_c(\mathbf{T}_{P_{\Gamma}^{\kappa}} \uparrow \omega)$ such that $E' \subseteq E$ and $I_{E'} \setminus E' = I_E \setminus E$.*

PROOF. (a) Let $I^{\kappa} \in \mathbf{T}_{P_{\Gamma}^{\kappa}} \uparrow \omega$ such that I^{κ} is canonical, and $I_E = \text{obj}(I^{\kappa})$. We consider the abductive Horn program $\langle P^{I_E}, \Gamma \rangle$. Let J^{κ} be an interpretation of the program $(P^{I_E})_{\Gamma}^{\kappa}$ such that $\text{obj}(J^{\kappa}) = I_E$. We will show that such J^{κ} exists in $\mathbf{T}_{(P^{I_E})_{\Gamma}^{\kappa}} \uparrow \omega$. Now, for each ground clause of the form (14) from P_{Γ}^{κ} , if $\{B_1, \dots, B_m\} \subseteq I_E \setminus E$, then either of the following holds:

- (i) $\{H, A_1, \dots, A_n\} \subseteq I_E$ and $\{\neg KB_{m+1}, \dots, \neg KB_s, \neg KA_{n+1}, \dots, \neg KA_t\} \subseteq I^{\kappa}$. In this case, since I^{κ} is canonical, it holds that $\{B_{m+1}, \dots, B_s\} \cap I_E = \emptyset$ and $\{A_{n+1}, \dots, A_t\} \cap E = \emptyset$. Then, there is a ground clause of the form (12) from $(P^{I_E})_{\Gamma}^{\kappa}$. This clause is satisfied by J^{κ} because $\{B_1, \dots, B_m\} \subseteq J^{\kappa}$ implies $\{H, A_1, \dots, A_n\} \subseteq J^{\kappa}$.
- (ii) $\neg KA_i \in I^{\kappa}$ for some i ($1 \leq i \leq n$). In this case, there may or may not exist the corresponding ground clause of the form (12) from $(P^{I_E})_{\Gamma}^{\kappa}$. If it does exist, the clause is satisfied by J^{κ} because $\{B_1, \dots, B_m\} \subseteq J^{\kappa}$ implies $\neg KA_i \in J^{\kappa}$.
- (iii) $KB_j \in I^{\kappa}$ for some j ($m+1 \leq j \leq s$). In this case, since I^{κ} is canonical, $B_j \in I_E$. Then, there is no corresponding clause of the form (12) in $(P^{I_E})_{\Gamma}^{\kappa}$.
- (iv) $A_k \in E$ for some k ($n+1 \leq k \leq t$). In this case, there is no corresponding clause of the form (12) in $(P^{I_E})_{\Gamma}^{\kappa}$.

Hence, J^{κ} is a model of $(P^{I_E})_{\Gamma}^{\kappa}$. By the above four cases, J^{κ} is actually contained in $\mathbf{T}_{(P^{I_E})_{\Gamma}^{\kappa}} \uparrow \omega$. Then, I_E is a belief model of $\langle P^{I_E}, \Gamma \rangle$ by Lemma 3.14 (a). Hence, I_E is a belief model of $\langle P, \Gamma \rangle$ by Lemma 3.19.

(b) Suppose that I_E is a belief model of $\langle P, \Gamma \rangle$. By Lemma 3.19, I_E is a belief model of $\langle P^{I_E}, \Gamma \rangle$. Then, by Lemma 3.14 (b), there is a belief model $I_{E'}$ of $\langle P^{I_E}, \Gamma \rangle$ such that $E' \subseteq E$, $I_{E'} \setminus E' = I_E \setminus E$, and $I_{E'} = \text{obj}(I^\kappa)$ for some $I^\kappa \in \mathbf{T}_{(P^{I_E})_\Gamma^\kappa} \uparrow \omega$. Again, by Lemma 3.19, this $I_{E'}$ is a belief model of $\langle P, \Gamma \rangle$ such that $E' \subseteq E$ and $I_{E'} \setminus E' = I_E \setminus E$. Thus, it remains to verify that $I_{E'} \in \text{obj}_c(\mathbf{T}_{P_\Gamma^\kappa} \uparrow \omega)$. Since $I_{E'} = \text{obj}(I^\kappa)$ and $I^\kappa \in \mathbf{T}_{(P^{I_E})_\Gamma^\kappa} \uparrow \omega$, for each clause (12), if $\{B_1, \dots, B_m\} \subseteq I_{E'}$, then either: (i) $\{H, A_1, \dots, A_n\} \subseteq I_{E'}$ or (ii) $\neg KA_i \in I^\kappa$ for some i ($1 \leq i \leq n$). Now, suppose that a clause (12) in $(P^{I_E})_\Gamma^\kappa$ was translated from the clause (11) in P^{I_E} , and that the ground instance of (11) corresponds to each ground clause C of the form (13) in P . Consider the following three cases.

Case 1. $\{B_{m+1}, \dots, B_s\} \cap I_{E'} = \emptyset$ and $\{A_{n+1}, \dots, A_t\} \cap E = \emptyset$.

In this case, let $\Delta(C) = \{\neg KB_{m+1}, \dots, \neg KB_s, \neg KA_{n+1}, \dots, \neg KA_t\}$.

Case 2. $B_j \in I_{E'}$ for some j ($m+1 \leq j \leq s$).

In this case, let $\Delta(C) = \{KB_j\}$.

Case 3. $A_k \in I_{E'}$ for some k ($n+1 \leq k \leq t$).

In this case, let $\Delta(C) = \{A_k\}$.

In either of these three cases, $I_{E'} \cup \Delta(C)$ obviously satisfies the corresponding ground clause of the form (14) from P_Γ^κ . Now, let $J^\kappa = I_{E'} \cup \min\text{-}\Delta(P)$ where $\min\text{-}\Delta(P)$ is a minimal subset of $\bigcup_C \Delta(C)$ such that each KB_j in Case 2 or A_k in Case 3 above is chosen in a way that J^κ satisfies every ground clause of the form (14) from P_Γ^κ . Then, $J^\kappa \in \mathbf{T}_{P_\Gamma^\kappa} \uparrow \omega$, and it holds that $\text{obj}(J^\kappa) = I_{E'}$ and that $KH \in J^\kappa$ implies $H \in I_{E'}$ for any $H \in \mathcal{HB}$. Hence, $I_{E'} \in \text{obj}_c(\mathbf{T}_{P_\Gamma^\kappa} \uparrow \omega)$. \square

The next lemma is a generalization of the result of Lemma 3.16.

Lemma 3.21. *Let $\langle P, \Gamma \rangle$ be an abductive normal logic program, and O an observation. If $E \subseteq \Gamma$ is an explanation of O , then there is an explanation E' of O such that $E' \subseteq E$ and $I_{E'} \in \text{obj}_c(\mathbf{T}_{P_\Gamma^\kappa} \uparrow \omega)$.*

PROOF. This follows from the completeness result by Lemma 3.20 (b). \square

The next theorem characterizes the belief model semantics of an abductive normal logic program and the minimal explanations of an observation in terms of the disjunctive fixpoint of the transformed program. In the following, when \mathbf{I}^κ is a set of interpretations, we write

$$\min_\Gamma(\mathbf{I}^\kappa) = \{I_E \in \mathbf{I}^\kappa \mid \text{there is no } J_F \in \mathbf{I}^\kappa \text{ such that } F \subset E\}.$$

Theorem 3.22. *Let $\langle P, \Gamma \rangle$ be an abductive normal logic program.*

- (a) *Let $\min\text{-}\mathcal{BM}_{\langle P, \Gamma \rangle}$ be the set of all Γ -minimal belief models of $\langle P, \Gamma \rangle$. Then, $\min\text{-}\mathcal{BM}_{\langle P, \Gamma \rangle} = \min_\Gamma(\text{obj}_c(\mathbf{T}_{P_\Gamma^\kappa} \uparrow \omega))$.*
- (b) *Let E be a subset of Γ , and O an observation. Then, E is a minimal explanation of O with respect to $\langle P, \Gamma \rangle$ iff $I_E \in \min_\Gamma(\text{obj}_c(\mathbf{T}_{(P \cup \{\neg \text{not}\})_\Gamma^\kappa} \uparrow \omega))$.*

PROOF. (a) By Lemma 3.20 (b), it follows immediately that $\min\text{-}\mathcal{BM}_{\langle P, \Gamma \rangle} \subseteq \text{obj}_c(\mathbf{T}_{P_\Gamma^\kappa} \uparrow \omega)$, and hence $\min\text{-}\mathcal{BM}_{\langle P, \Gamma \rangle} \subseteq \min_\Gamma(\text{obj}_c(\mathbf{T}_{P_\Gamma^\kappa} \uparrow \omega))$ holds. On the other hand, by Lemma 3.20 (a), every $I_E \in \text{obj}_c(\mathbf{T}_{P_\Gamma^\kappa} \uparrow \omega)$ is a belief model of $\langle P, \Gamma \rangle$. If $I_E \in \min_\Gamma(\text{obj}_c(\mathbf{T}_{P_\Gamma^\kappa} \uparrow \omega))$ is not in $\min\text{-}\mathcal{BM}_{\langle P, \Gamma \rangle}$, then there is $I_{E'} \in \min\text{-}\mathcal{BM}_{\langle P, \Gamma \rangle}$ such that $E' \subset E$. However, by the above discussion, $I_{E'} \in \min_\Gamma(\text{obj}_c(\mathbf{T}_{P_\Gamma^\kappa} \uparrow \omega))$, contradiction. Therefore, the result follows.

(b) By Lemma 3.21, for every minimal explanation E of O , there is a belief model I_E of $\langle P, \Gamma \rangle$ in $\text{obj}_c(\mathbf{T}_{P_\Gamma^\kappa} \uparrow \omega)$ such that I_E satisfies O . Then, by Lemma 2.10 (b), $I_E \in \min\text{-}\mathcal{BM}_{\langle P \cup \{\leftarrow \text{not } O\}, \Gamma \rangle}$. By (a), $\min\text{-}\mathcal{BM}_{\langle P \cup \{\leftarrow \text{not } O\}, \Gamma \rangle}$ is given by $\min_\Gamma(\text{obj}_c(\mathbf{T}_{P \cup \{\leftarrow \text{not } O\}} \uparrow \omega))$. Hence, the result follows. \square

Example 3.23. (cont. from Example 2.11) The abductive normal logic program $\langle P, \Gamma \rangle$, where $P = \{p \leftarrow r \wedge b \wedge \text{not } q, q \leftarrow a, r \leftarrow, \leftarrow \text{not } p\}$ and $\Gamma = \{a, b\}$, is transformed into P_Γ^κ which contains

$$\begin{aligned} & (p \wedge b \wedge \neg Kq) \vee \neg Kb \vee Kq \leftarrow r, \\ & (q \wedge a) \vee \neg Ka \leftarrow, \\ & r \leftarrow, \\ & Kp \leftarrow, \end{aligned}$$

and $\leftarrow \neg KH \wedge H$ for every $H \in \mathcal{HB}$. Then, $\{r, p, b, \neg Kq, \neg Ka, Kp\}$ is the unique canonical set in $\mathbf{T}_{P_\Gamma^\kappa} \uparrow \omega$, and hence $\min\text{-}\mathcal{BM}_{\langle P, \Gamma \rangle} = \{\{r, p, b\}\}$.

4. ABDUCTIVE EXTENDED DISJUNCTIVE PROGRAMS

Gelfond [12] and Inoue [16] proposed more general frameworks for abduction than that by Kakas and Mancarella [21] by allowing *classical negation* and disjunctions in a program. These extended abductive frameworks are powerful enough to describe complex knowledge in such areas as diagnosis and reasoning about action. In this section, we consider a fixpoint theory for such extended classes of abductive programs by generalizing the results in the previous section.

4.1. Fixpoint Semantics for Extended Disjunctive Programs

An *extended disjunctive program* is a disjunctive program which contains classical negation (\neg) along with negation as failure (*not*) in the program [14], and is defined as a finite set of clauses of the form⁴

$$L_1 \vee \cdots \vee L_l \leftarrow L_{l+1} \wedge \cdots \wedge L_m \wedge \text{not } L_{m+1} \wedge \cdots \wedge \text{not } L_n \quad (15)$$

where $n \geq m \geq l \geq 0$ and each L_i is a positive or negative *literal*. We denote the set of all ground literals in the language as $\mathcal{L} = \mathcal{HB} \cup \{\neg B \mid B \in \mathcal{HB}\}$. An extended disjunctive program P is called an *extended logic program* if $l \leq 1$ for every clause (15) of P . An extended disjunctive program P reduces to a normal logic program (resp. positive disjunctive program) if for any clause (15) of P , $l \leq 1$ (resp. $m = n$) and every L_i is a positive literal.

The semantics of extended disjunctive programs is given by the notion of *answer sets* in the following two steps. First, let P be an extended disjunctive program

⁴Gelfond and Lifschitz [14] use the connective “|” instead of “ \vee ” to distinguish its meaning from the classical first-order logic. Here, we take the liberty of using the connective \vee .

without *not* (i.e., $m = n$ for any clause of P), and $S \subseteq \mathcal{L}$. Then, S is a *consistent answer set* of P iff S is a minimal set satisfying the conditions:

1. For each ground clause $L_1 \vee \dots \vee L_l \leftarrow L_{l+1} \wedge \dots \wedge L_m$ ($l \geq 1$) from P , if $\{L_{l+1}, \dots, L_m\} \subseteq S$, then $L_i \in S$ for some i ($1 \leq i \leq l$). In particular, for each ground integrity constraint $\leftarrow L_1 \wedge \dots \wedge L_m$ from P , it must be that $\{L_1, \dots, L_m\} \not\subseteq S$; and
2. S does not contain both B and $\neg B$ for any atom B .

Next, let P be any extended disjunctive program, and $S \subseteq \mathcal{L}$. The *reduct* P^S of P by S is defined as follows: A clause $L_1 \vee \dots \vee L_l \leftarrow L_{l+1} \wedge \dots \wedge L_m$ is in P^S iff there is a ground clause of the form (15) from P such that $\{L_{m+1}, \dots, L_n\} \cap S = \emptyset$. Then, S is a *consistent answer set* of P iff S is a consistent answer set of P^S .⁵

Since the answer set semantics of extended disjunctive programs is a direct extension of both the minimal model semantics of positive disjunctive programs and the stable model semantics of normal logic programs, the results presented in Sections 3.1 and 3.2 can be naturally extended. The extra condition we have to consider is the constraint that an atom B and its negation $\neg B$ cannot be in a consistent answer set at the same time.

Definition 4.1 [17]. Let P be an extended disjunctive program. The program P^κ is defined as follows.

1. For each clause (15) in P , P^κ contains the clause

$$(L_1 \wedge \neg \mathcal{K}L_{m+1} \wedge \dots \wedge \neg \mathcal{K}L_n) \vee \dots \vee (L_l \wedge \neg \mathcal{K}L_{m+1} \wedge \dots \wedge \neg \mathcal{K}L_n) \\ \vee \mathcal{K}L_{m+1} \vee \dots \vee \mathcal{K}L_n \leftarrow L_{l+1} \wedge \dots \wedge L_m. \quad (16)$$

2. For each literal L in \mathcal{L} , P^κ includes the negative clause

$$\leftarrow \neg \mathcal{K}L \wedge L. \quad (17)$$

3. For each atom B in $\mathcal{H}\mathcal{B}$, P^κ includes the negative clause

$$\leftarrow \neg B \wedge B. \quad (18)$$

Note in the above definition that the transformed program P^κ is a *positive disjunctive program*. This is because we regard each negative literal $\neg B$ as an *atom*, and then its meaning is given by the extra integrity constraint (18). In the following, the function obj_c defined in Definition 3.8 is extended to a collection of sets of literals in an obvious way.

Theorem 4.2 [17, 34]. Let P be an extended disjunctive program, and AS_P the set of all consistent answer sets of P . Then

$$AS_P = obj_c(\mathcal{M}\mathcal{M}_{P^\kappa}) = obj_c(\min(\mathbf{T}_{P^\kappa} \uparrow \omega)).$$

The above theorem says that the answer sets of an extended disjunctive program P are characterized in terms of the minimal models of P^κ . For extended

⁵In this paper, we do not consider the contradictory answer set \mathcal{L} that contains all literals since we are interested only in consistent theories augmented with abducibles.

logic programs, the result of Theorem 4.2 is further simplified so that we need not compute the minimal models of the disjunctive fixpoint (cf. Theorem 3.9).

Corollary 4.3 [34]. *Let P be an extended logic program. Then*

$$\mathcal{AS}_P = \text{obj}_c(\mathbf{T}_{P^\kappa} \uparrow \omega).$$

4.2. Fixpoint Semantics for Abductive Extended Disjunctive Programs

Now, we define abduction within extended disjunctive programs.

Definition 4.4. An *abductive extended disjunctive program* is a pair $\langle P, \Gamma \rangle$, where P is an extended disjunctive program and Γ is a set of literals from the language of P . The set Γ is identified with the set of ground instances from Γ , and each literal in Γ is called an *abducible*. Note that $\Gamma \subseteq \mathcal{L}$. When P is an extended logic program, $\langle P, \Gamma \rangle$ is called an *abductive extended logic program*.

For $S \subseteq \mathcal{L}$, we often write S as S_E when $E = S \cap \Gamma$.

A set of literals S is a *belief set* of $\langle P, \Gamma \rangle$ if it is a consistent answer set of an extended disjunctive program $P \cup E$ for some subset E of Γ . A belief set S_E is Γ -*minimal* if no belief set T_F satisfies that $F \subset E$.

Let O be a ground literal called an *observation*. $E \subseteq \Gamma$ is a (*minimal*) *explanation* of O if there is a (Γ -minimal) belief set S_E of $\langle P, \Gamma \rangle$ such that $O \in S_E$.

Note that the notion of belief sets reduces to that of belief models for abductive normal logic programs. The transformation for an abductive extended disjunctive program is defined in the same way as in Definition 3.18.

Definition 4.5. Let $\langle P, \Gamma \rangle$ be an abductive extended disjunctive program. Then, P_Γ^κ is obtained as follows.

1. For each clause in P of the form

$$\begin{aligned} H_1 \vee \cdots \vee H_l \leftarrow B_1 \wedge \cdots \wedge B_m \wedge A_1 \wedge \cdots \wedge A_n \\ \wedge \text{not } B_{m+1} \wedge \cdots \wedge \text{not } B_s \wedge \text{not } A_{n+1} \wedge \cdots \wedge \text{not } A_t \end{aligned} \quad (19)$$

where $l \geq 0$, $s \geq m \geq 0$, $t \geq n \geq 0$, H_i s are literals, B_j s are nonabducible literals, and A_k s are abducible literals, P_Γ^κ contains the clause

$$\begin{aligned} (H_1 \wedge PRE) \vee \cdots \vee (H_l \wedge PRE) \vee \neg KA_1 \vee \cdots \vee \neg KA_n \\ \vee KB_{m+1} \vee \cdots \vee KB_s \vee A_{n+1} \vee \cdots \vee A_t \leftarrow B_1 \wedge \cdots \wedge B_m \end{aligned} \quad (20)$$

where $PRE = A_1 \wedge \cdots \wedge A_n \wedge \neg KB_{m+1} \wedge \cdots \wedge \neg KB_s \wedge \neg KA_{n+1} \wedge \cdots \wedge \neg KA_t$.

2. P_Γ^κ includes the clauses

$$\begin{aligned} \leftarrow \neg KL \wedge L \quad \text{for each literal } L \in \mathcal{L}, \\ \leftarrow \neg H \wedge H \quad \text{for each atom } H \in \mathcal{HB}. \end{aligned}$$

It is easy to see that the transformed clause (20) for abductive extended disjunctive programs is a generalization of transformed clauses (4), (12), (14), and (16) for normal logic, abductive Horn, abductive normal logic, and extended disjunctive programs. Note, again, that the transformed program P_Γ^κ is a positive disjunctive program. Therefore, we can get its disjunctive fixpoint like abductive normal logic programs.

Lemma 4.6. Let $\langle P, \Gamma \rangle$ be an abductive extended disjunctive program.

- (a) For any $S \in \text{obj}_c(\min(\mathbf{T}_{P_F^\kappa} \uparrow \omega))$, S is a belief set of $\langle P, \Gamma \rangle$.
- (b) For any belief set S_E of $\langle P, \Gamma \rangle$, there exists a belief set $S_{E'}$ of $\langle P, \Gamma \rangle$ in $\text{obj}_c(\min(\mathbf{T}_{P_F^\kappa} \uparrow \omega))$ such that $E' \subseteq E$ and $S_{E'} \setminus E' = S_E \setminus E$.

PROOF. The proofs can be given in a similar way to the proof of Lemma 3.20, except that, according to the existence of disjunctions in P , each S^κ is taken from $\min(\mathbf{T}_{P_F^\kappa} \uparrow \omega)$ (as in Theorem 3.4 (c) and Theorem 4.2) instead of $\mathbf{T}_{P_F^\kappa} \uparrow \omega$. \square

The next theorem characterizes the belief set semantics of an abductive extended disjunctive program and the minimal explanations of an observation.

Theorem 4.7. Let $\langle P, \Gamma \rangle$ be an abductive extended disjunctive program.

- (a) Let $\min\text{-BS}_{\langle P, \Gamma \rangle}$ be the set of all Γ -minimal belief sets of $\langle P, \Gamma \rangle$. Then

$$\min\text{-BS}_{\langle P, \Gamma \rangle} = \min_{\Gamma}(\text{obj}_c(\mathcal{MM}_{P_F^\kappa})) = \min_{\Gamma}(\text{obj}_c(\min(\mathbf{T}_{P_F^\kappa} \uparrow \omega))).$$

- (b) Let E be a subset of Γ , and O an observation. Then, E is a minimal explanation of O with respect to $\langle P, \Gamma \rangle$ iff $S_E \in \min_{\Gamma}(\text{obj}_c(\min(\mathbf{T}_{(P \cup \{\leftarrow \text{not } O\})_F^\kappa} \uparrow \omega)))$.

PROOF. The proof can be given in the same way as the proof of Theorem 3.22 using Lemma 4.6. \square

For abductive extended logic programs, the results of Theorem 4.7 are further simplified so that we need not compute the minimal models of the disjunctive fixpoint. This is similar to the case of abductive normal logic programs in Theorem 3.22.

Corollary 4.8. Let $\langle P, \Gamma \rangle$ be an abductive extended logic program. Then

$$\min\text{-BS}_{\langle P, \Gamma \rangle} = \min_{\Gamma}(\text{obj}_c(\mathbf{T}_{P_F^\kappa} \uparrow \omega)).$$

Example 4.9. Consider the abductive extended disjunctive program $\langle P, \Gamma \rangle$, where

$$\begin{aligned} P &= \{p \vee q \leftarrow \text{not } r, \\ &\quad r \leftarrow \text{not } a, \\ &\quad \neg q \leftarrow b\}, \\ \Gamma &= \{a, b\}. \end{aligned}$$

This program has five belief sets: $S_1 = \{r\}$, $S_2 = \{a, p\}$, $S_3 = \{a, q\}$, $S_4 = \{r, b, \neg q\}$, and $S_5 = \{a, p, b, \neg q\}$, and S_1 is the Γ -minimal belief set. Let p be an observation. Then, S_2 and S_5 are those belief sets containing p , and $E_2 = S_2 \cap \Gamma = \{a\}$ is the minimal explanation of p , but $E_5 = S_5 \cap \Gamma = \{a, b\}$ is its nonminimal explanation.⁶ Now, the program with the observation is transformed

⁶Note, however, that the program $P \cup E_2$ has another answer set S_3 that does not contain p , while $P \cup E_5$ has the unique answer set S_5 that contains p . Hence, some alternative definitions of explanations can be considered according to credulous or skeptical reasoning for the observation. In this respect, see Inoue [16], for example.

into $(P \cup \{\leftarrow \text{not } p\})_{\neq}^{\kappa}$ which includes

$$\begin{aligned}
 & (p \wedge \neg Kr) \vee (q \wedge \neg Kr) \vee Kr \leftarrow, \\
 & (r \wedge \neg Ka) \vee a \leftarrow, \\
 & (\neg q \wedge b) \vee \neg Kb \leftarrow, \\
 & \leftarrow \neg Kr \wedge r, \\
 & \leftarrow \neg Ka \wedge a, \\
 & \leftarrow \neg Kb \wedge b, \\
 & \leftarrow \neg q \wedge q, \\
 & Kp \leftarrow.
 \end{aligned}$$

Then, $\mathbf{T}_{(P \cup \{\leftarrow \text{not } p\})_{\neq}^{\kappa}} \uparrow \omega$ contains two canonical sets, $S_2^{\kappa} = \{p, \neg Kr, a, \neg Kb\}$ and $S_5^{\kappa} = \{p, \neg Kr, a, \neg q, b\}$, which correspond to the explanations E_2 and E_5 .

5. BOTTOM-UP EVALUATION OF ABDUCTIVE PROGRAMS

In this section, we investigate the procedural aspect of the fixpoint theory for abductive programs in the context of a particular inference system called the *model generation theorem prover* (MGTP) [11, 17]. MGTP is a parallel and refined version of SATCHMO [26], which is a bottom-up forward-reasoning system that uses *hyperresolution* and *case-splitting* on nonunit hyperresolvents.

Let P be a positive disjunctive program consisting of clauses of the form

$$(H_{1,1} \wedge \cdots \wedge H_{1,k_1}) \vee \cdots \vee (H_{l,1} \wedge \cdots \wedge H_{l,k_l}) \leftarrow B_1 \wedge \cdots \wedge B_m \quad (21)$$

where B_i s ($1 \leq i \leq m$; $m \geq 0$) and $H_{j,l}$ s ($1 \leq j \leq l$; $1 \leq l \leq k_j$; $k_j \geq 1$; $l \geq 0$) are atoms, and all variables are assumed to be universally quantified at the front of the clause. Given an interpretation I , MGTP applies the following two operations to I and either expands I or rejects I :

1. (Interpretation Extension) If there is a nonnegative clause of the form (21) in P and a substitution σ such that $I \models (B_1 \wedge \cdots \wedge B_m)\sigma$ and $I \not\models (H_{i,1} \wedge \cdots \wedge H_{i,k_i})\sigma$ for all $i = 1, \dots, l$, then I is expanded in l ways by adding $H_{i,1}\sigma, \dots, H_{i,k_i}\sigma$ to I for each $i = 1, \dots, l$.
2. (Interpretation Rejection) If there is a negative clause $\leftarrow B_1, \dots, B_m$ in P and a substitution σ such that $I \models (B_1 \wedge \cdots \wedge B_m)\sigma$, then I is discarded.

Starting from the empty interpretation $I_0 = \emptyset$, MGTP repeats to apply the above two operations as long as a new interpretation can be expanded or some interpretation can be pruned. Here, in obtaining a substitution σ in each operation, it is sufficient to consider *matching* instead of full unification if every clause is *range-restricted* [26], that is, if every variable in the clause has at least one occurrence in the body. In this case, every set I of atoms constructed by MGTP contains only ground atoms. Furthermore, when a program is function-free, MGTP always terminates in a finite step.

Thus, a program input to MGTP is usually assumed to be a finite, function-free set of range-restricted clauses. For example, let C be a clause of the form (19) in an extended disjunctive program, and C^{κ} the MGTP clause of the form (20) that is translated from C . In order that C^{κ} may be range-restricted, every variable in C has an occurrence in a nonabducible literal B_i ($1 \leq i \leq m$) that is not preceded

by *not* in the body of C . Note that clauses can be converted in order to satisfy this kind of range restriction [26]. MGTP gives high inference rates for range-restricted clauses by avoiding computation relative to their useless ground instances [11].

The connection between closure computation by SATCHMO/MGTP and the fixpoint semantics with the mapping T_P given in Section 3 is obvious, which can be regarded as an extension of the relation between hyperresolution and van Emden and Kowalski's fixpoint semantics for definite programs [37, sect. 8]. In fact, for each split interpretation constructed by MGTP, hyperresolution is applied in the same way as in the case of definite programs. Then, since we have presented correct transformations of abductive programs into semantically equivalent positive disjunctive programs in the previous sections, the soundness and completeness of MGTP mentioned above imply that MGTP is also sound and complete to compute belief models/sets of function-free, range-restricted abductive programs.

We summarize the advantages of MGTP for computing belief models/sets of abductive programs as follows. Other additional merits of MGTP computation that are compared with other styles of implementation will be discussed in Section 6.2.

1. Since we keep believed literals KL 's and $\neg KL$'s in each interpretation, when new clauses are added to the program, the previous fixpoint closure can be used as the input to the next computation. Hence, computation is *incremental*.
2. Our program transformation is *modular* in the sense that adding new clauses to a program is reflected by adding new transformed *not*-free clauses to the corresponding transformed program.
3. While case-splitting is the place where nondeterminism arises in our procedure, those split interpretations can be dealt with independently without future backtracking. This means that, for every generated interpretation, each ground instance of any clause is evaluated only once.
4. For abductive Horn, normal, and extended (disjunctive) programs, our program translations are especially suitable for *OR-parallelism* of MGTP because, for each negation-as-failure formula as well as an abducible, we make guesses to believe or disbelieve it. Inoue et al. [18] have shown that model generation for abductive Horn programs using the translation in Section 3.3 successfully extracts a great amount of parallelism of MGTP in solving a logic circuit design problem.
5. While MGTP is a bottom-up abductive procedure, it is equipped with various devices for reducing the number of combinations of ground hypotheses from Γ in generating belief models (see Section 6.2.3).
6. Inoue et al. [18] have shown how to recover the "goal-oriented" feature within the above parallel abductive procedure by applying the *magic set* method [1] to Horn abduction. Our bottom-up abductive procedure can thus avoid naive computation.

6. COMPARISON WITH OTHER APPROACHES

This section compares the proposed abductive theory to related work. Our fixpoint theory gives a *new, uniform framework for characterizing minimal models, stable models, belief models, answer sets, and belief sets* of abductive/nonabductive, normal/extended, logic/disjunctive programs. Since there have been no algorithms to compute the belief sets of arbitrary form of abductive programs, our procedural

semantics also provides *the most general abductive procedure in the class of function-free and range-restricted programs.*

6.1. Fixpoint Characterization for Disjunctive and Normal Programs

Here, we summarize the differences between other approaches and our fixpoint construction for positive disjunctive programs and normal logic programs.

A fixpoint semantics for positive disjunctive programs has been studied by several researchers. Minker and Rajasekar [28] consider a mapping over the set of positive disjunctions (called *state*), while our fixpoint construction is based on the manipulation of standard Herbrand interpretations and directly computes models. Fernandez and Minker [9] present a fixpoint semantics for stratified disjunctive programs using a fixpoint operator over the sets of *minimal* interpretations. To this end, their fixpoint operator computes minimal sets of atoms at every stage of closure computation. With our fixpoint operator, on the other hand, each interpretation can be treated in a different, independent process in closure computation, so that split interpretations can be taken as the source for exploiting OR-parallelism of MGTP.

For normal and extended disjunctive programs, Gelfond and Lifschitz originally defined the stable model semantics [13] and the answer set semantics [14] by means of guesses and reducts of programs. On the other hand, our fixpoint is constructively defined. In contrast to another constructive approach like [31], our fixpoint construction is performed in parallel based on case-splitting on derived disjunctions, and does not need any selection strategies or future backtracking during the computation of stable models. Sakama and Inoue [33, 34] also present yet another fixpoint semantics for positive and extended disjunctive programs. They use similar fixpoint constructions, but the semantics dealt with in [33] is the *possible model semantics* and that in [34] is the *paraconsistent stable/possible model semantics*.

In [10], Fernandez et al. independently develop a method of computing stable models by using a similar but different program transformation from ours in Section 3.2. In our transformation (4), each head H is associated with its *prerequisite condition* $\neg KB_{m+1} \wedge \cdots \wedge \neg KB_n$ in an explicit way, while this is not the case in their transformation. Then, it is not clear whether Fernandez et al.'s transformation can be naturally extended to deal with abductive programs. In this regard, our translation appears to be more suitable for handling abducibles. Since the prerequisite condition in Definitions 3.18 and 4.5

$$PRE = A_1 \wedge \cdots \wedge A_n \wedge B_{m+1} \wedge \cdots \wedge B_s \wedge \neg KA_{n+1} \wedge \cdots \wedge \neg KA_t$$

contains abduced literals A_1, \dots, A_n explicitly, we can easily identify abducibles from other atoms in each obtained model, and negative clauses can be used to test the consistency of abducibles in each interpretation.

6.2. Various Characterizations for Abductive Programs

6.2.1. ABDUCTION AS DEDUCTION. Console et al. [4] characterize abduction by deduction (called the *object-level abduction*) through Clark's completion semantics of a program [3]. According to their framework, abduction is characterized as follows: For an abductive logic program $\langle P, \Gamma \rangle$, let $comp^{-\Gamma}(P)$ be the completion of nonabducible predicates in P . For an atom O (observation), if E is a formula from Γ satisfying the conditions

1. $comp^{-\Gamma}(P) \cup \{O\} \models E$, and
2. no other E' from Γ satisfying the above condition subsumes E ,

then a minimal set of literals $S \subseteq \Gamma$ such that $S \models E$ is called an *explanation* of O .

The object-level abduction coincides with the meta-level characterization of abduction in terms of SLDNF proof procedure for *hierarchical logic programs*⁷ [4]. Note here that the restriction of hierarchical programs is necessary not only for assuring the completeness of SLDNF resolution, but also for characterizing abduction in terms of completion (see also [24]).

Example 6.1. Let us consider an abductive program containing cyclic clauses

$$P = \{p \leftarrow q, \\ q \leftarrow p, \\ q \leftarrow a\},$$

and

$$\Gamma = \{a\}.$$

Then

$$comp^{-\Gamma}(P) = \{p \equiv q, \quad q \equiv p \vee a\},$$

and for an observation $O = p$, $P \cup \{a\} \models p$, while $comp^{-\Gamma}(P) \cup \{O\} \not\models a$.

On the other hand,

$$P_{\Gamma}^{\kappa} = \{p \leftarrow q, \\ q \leftarrow p, \\ (q \wedge a) \vee \neg \mathcal{K}a \leftarrow, \\ \leftarrow \neg \mathcal{K}a \wedge a\}$$

is obtained by our transformation in Section 3.3, and $\{q, a, p\}$ is in $\mathbf{T}_{P_{\Gamma}^{\kappa}} \uparrow \omega$.

Denecker and De Schreye [5] propose a model generation procedure for Console et al.'s object-level abduction. In contrast to ours, their procedure computes the models of the only-if part of a completed program that is not range-restricted in general, even if the original definite clauses are range-restricted. To this end, they extend the model generation method by incorporating term rewriting techniques, while we can use the original MGTP without any change. Furthermore, the application of their procedure is limited to definite programs, whereas we allow negative and disjunctive clauses as well as negation as failure in programs. Bry [2] first considered abduction by model generation, but his abduction is defined in terms of a meta-theory.

6.2.2. ABDUCTIVE INTERPRETATION OF NEGATION AS FAILURE. The idea of dealing with negation as failure and abduction in a uniform way was first proposed by Eshghi and Kowalski [8], and further developed by Kakas and Mancarella [21]. Our transformation also realizes a uniform approach, but is entirely original and

⁷Normal logic programs containing no predicates defined via positive/negative cycles.

has the advantage of providing a uniform framework for yet another extension of logic programming, including disjunction and classical negation.

Eshghi and Kowalski [8] give an abductive interpretation of negation as failure in normal logic programs. For each negation-as-failure formula $not\ B(\mathbf{x})$, the formula $B^*(\mathbf{x})$ is associated where B^* is a new predicate symbol not appearing anywhere in the program. A program P is thereby transformed into the definite program P^* together with the set Γ^* of abducibles with the predicates B^* s. Then, an atom O is true in a stable model of P iff there is a set E^* of abducibles from Γ^* such that

1. $P^* \cup E^* \models O$, and
2. $P^* \cup E^*$ satisfies the integrity constraints

$$\neg(B(\mathbf{x}) \wedge B^*(\mathbf{x})) \text{ and } B(\mathbf{x}) \vee B^*(\mathbf{x}) \text{ for every abducible predicate } B^*.$$

In this abductive characterization, the difficulty arises in dealing with the disjunctive constraints that cannot be checked without actually computing models in general. Thus, it is hard to design an elegant top-down proof procedure which is sound with respect to the stable model semantics. In fact, Eshghi and Kowalski [8] show an abductive proof procedure for normal logic programs by incorporating consistency tests into SLD resolution, but its soundness with respect to the stable model semantics is not guaranteed in general.⁸

For an abductive normal logic program $\langle P, \Gamma \rangle$, Kakas and Mancarella [22] show a top-down abductive procedure for the transformed program $\langle P^*, \Gamma \cup \Gamma^* \rangle$, where P^* and Γ^* are obtained by the transformation of [8]. This transformation inherits the difficulty of computation from Eshghi and Kowalski's abductive interpretation of negation as failure, and their procedure suffers from the soundness problem with respect to the belief model semantics. Satoh and Iwayama [36] develop an abductive procedure which is sound with respect to the belief model semantics by incorporating a special integrity checking into the procedure of [8, 22]. To our best knowledge, no procedure other than ours has been developed so far as a sound procedure for abductive extended disjunctive programs.

6.2.3. COMPUTATION WITH TMS. Satoh and Iwayama [35] and Inoue [16] independently show that any abductive normal logic program $\langle P, \Gamma \rangle$ can be transformed into a *single* extended (or normal) logic program. For each atom A in Γ , they introduce the negative literal $\neg A$ and a pair of clauses

$$\begin{aligned} A &\leftarrow not\ \neg A, \\ \neg A &\leftarrow not\ A. \end{aligned} \tag{22}$$

Then, there is a 1-1 correspondence between the belief models of $\langle P, \Gamma \rangle$ and the answer sets (or stable models if $\neg A$ is considered as a new atom) of the transformed program. Using this transformation, Satoh and Iwayama [35] propose a bottom-up, TMS-style procedure for computing stable models of a normal logic program, which is similar to Sacca and Zaniolo's [31] procedure and performs an exhaustive search with backtracking. At this point, we can use any procedure other than TMS-style procedures for computing stable models. For instance, Dressler's nonmonotonic

⁸For Example 3.10, the top-down abductive procedure of [8] gives a proof for $O = p$, but no stable model satisfies p . However, Eshghi and Kowalski's abductive proof procedure is sound with respect to the *preferred extension* semantics by Dung [7].

ATMS [6] can also be used to compute belief models. Comparing each procedure, the MGTP-based procedure by Inoue et al. [17] has the following advantages over the procedures of [31, 35]. First, MGTP can deal with disjunctive programs, while TMS and ATMS cannot. Second, MGTP gives high inference rates for range-restricted clauses by avoiding computation relative to their useless ground instances, while TMS and ATMS generally deal only with the propositional case and have to prepare all the ground instances of a program in advance. Third, MGTP performs a backtrack-free search and more easily parallelized than others.

Although the simulation (22) of abducibles is theoretically correct, this technique has the drawback that it may generate $2^{|\Gamma|}$ interpretations, even for an abductive Horn program, and is, therefore, often explosive for a number of practical applications. The program transformation methods proposed in this paper avoid this problem in two aspects. First, for each epistemic hypothesis which is either a positive hypothesis from abducibles or a negative hypothesis through negation as-failure, case-splitting is delayed as long as possible since an interpretation is expanded with a ground clause only when the body of the transformed clause becomes true. Second, by using MGTP, a ground instance of hypothesis is introduced only when there is a ground substitution for each clause with variables such that the body of the clause is satisfied. Hence, hypotheses are introduced when they are necessary, and the number of generated interpretations is reduced as much as possible.

6.2.4. OTHER CHARACTERIZATIONS. Finally, it is worth noting that abductive programs can be formalized in other existing logic programming frameworks. Inoue and Sakama [20] recently showed that abductive extended disjunctive programs can be transformed into extended disjunctive programs with *positive occurrences of negation as failure*, and then into ordinary extended disjunctive programs. Their translation is complete with respect to the all belief sets of any abductive program, while the translation in this paper is complete with respect to the Γ -minimal belief sets. On the other hand, Sakama and Inoue [32] recently developed a translation from abductive normal logic/disjunctive programs into disjunctive programs, as well as a converse translation from disjunctive programs into abductive normal logic programs in the context of the *possible model semantics*, so that these two classes of programs are shown to be equivalent. Both works [20, 32] have contributed to the theory of the computational complexity of abductive normal logic/disjunctive programs.

7. CONCLUSION

We have established a uniform framework for fixpoint characterization of abductive (and nonabductive) Horn, normal, and extended logic (and disjunctive) programs. Based on a fixpoint operator over the sets of Herbrand interpretations, the belief model semantics of an abductive normal logic program can be characterized by the fixpoint of a suitably transformed positive disjunctive program. In the proposed transformations, both negative hypotheses through negation as failure and positive hypotheses from the abducibles are dealt with uniformly.

The result has also been directly applied to the belief set semantics of abductive extended disjunctive programs. Compared with other approaches, our fixpoint theory provides a constructive way to give explanations for observations. We also showed that a bottom-up model generation procedure can be used for computing

belief models or belief sets, and has a computational advantage from the viewpoint of parallelism. Since there has been no algorithm which can compute the belief sets of arbitrary form of abductive programs, our procedural semantics also provides the most general abductive procedure in the class of function-free and range-restricted programs.

The transformation method in this paper is also applicable to other semantics of abductive programs. For example, the paraconsistent, multivalued semantics for extended disjunctive programs [34] can be extended to incorporate abducible literals, and then the corresponding belief sets can be directly characterized by the translation and the fixpoint semantics in this paper.

REFERENCES

1. Bancilhon, F., Maier, D., Sagiv, Y., and Ullman, J. D., Magic Sets and Other Strange Ways to Implement Logic Programs, in: *Proc. 5th ACM SIGMOD-SIGACT Symposium on Principles of Database Systems*, 1986, pp. 1-15.
2. Bry, F., Intensional Updates: Abduction via Deduction, in: D. H. D. Warren and P. Szeredi (eds.), *Logic Programming: Proc. 7th International Conference*, MIT Press, 1990, pp. 561-575.
3. Clark, K. L., Negation as Failure, in: H. Gallaire and J. Minker (eds.), *Logic and Data Bases*, Plenum Press, 1978, pp. 293-322.
4. Console, L., Dupre, D. T., and Torasso, P., On the Relationship between Abduction and Deduction, *J. Logic and Computation* 1(5):661-690 (1991).
5. Denecker, M., and De Schreye, D., On the Duality of Abduction and Model Generation, in: *Proc. International Conference on Fifth Generation Computer Systems 1992*, Ohmsha, 1992, pp. 650-657.
6. Dressler, O., Problem Solving with the NM-ATMS, in: *Proc. 9th European Conference on Artificial Intelligence*, Pitman, 1990, pp. 253-258.
7. Dung, P. M., Negations as Hypotheses: An Abductive Foundation for Logic Programming, in: K. Furukawa (ed.), *Logic Programming: Proc. 8th International Conference*, MIT Press, 1991, pp. 3-17.
8. Eshghi, K., and Kowalski, R. A., Abduction Compared with Negation by Failure, in: G. Levi and M. Martelli (eds.), *Logic Programming: Proc. 6th International Conference*, MIT Press, 1989, pp. 234-254.
9. Fernandez, J. A., and Minker, J., Computing Perfect Models of Disjunctive Stratified Databases, in: *Proc. ILPS'91 Workshop on Disjunctive Logic Programs*, San Diego, CA, 1991.
10. Fernandez, J. A., Lobo, J., Minker, J., and Subrahmanian, V. S., Disjunctive LP + Integrity Constraints = Stable Model Semantics, *Annals of Mathematics and Artificial Intelligence* 8:449-474 (1993).
11. Fujita, H., and Hasegawa, R., A Model Generation Theorem Prover in KL1 Using a Ramified-Stack Algorithm, in: K. Furukawa (ed.), *Logic Programming: Proc. 8th International Conference*, MIT Press, 1991, pp. 494-500.
12. Gelfond, M., Epistemic Approach to Formalization of Commonsense Reasoning, Research Report, Computer Science Department, University of Texas at El Paso, El Paso, TX, 1990.
13. Gelfond, M., and Lifschitz, V., The Stable Model Semantics for Logic Programming, in: R. A. Kowalski and K. A. Bowen (eds.), *Logic Programming: Proc. 5th International Conference and Symposium*, MIT Press, 1988, pp. 1070-1080.
14. Gelfond, M., and Lifschitz, V., Classical Negation in Logic Programs and Disjunctive Databases, *New Generation Computing* 9(3,4):365-385 (1991).

15. Inoue, K., Linear Resolution for Consequence Finding, *Artificial Intelligence* 56(2,3):301–353 (1992).
16. Inoue, K., Hypothetical Reasoning in Logic Programs, *J. Logic Programming* 18(3):191–227 (1994).
17. Inoue, K., Koshimura, M., and Hasegawa, R., Embedding Negation as Failure into a Model Generation Theorem Prover, in: D. Kapur (ed.), *Automated Deduction: Proc. 11th International Conference, Lecture Notes in Artificial Intelligence 607*, Springer-Verlag, 1992, pp. 400–415.
18. Inoue, K., Ohta, Y., Hasegawa, R., and Nakashima, M., Bottom-Up Abduction by Model Generation, in: *Proc. 13th International Joint Conference on Artificial Intelligence*, Morgan Kaufmann, 1993, pp. 102–108.
19. Inoue, K., and Sakama, C., Transforming Abductive Logic Programs to Disjunctive Programs, in: D. S. Warren (ed.), *Logic Programming: Proc. 10th International Conference*, MIT Press, 1993, pp. 335–353.
20. Inoue, K., and Sakama, C., On Positive Occurrences of Negation as Failure, in: J. Doyle, E. Sandewall, and P. Torasso (eds.), *Principles of Knowledge Representation and Reasoning: Proc. 4th International Conference*, Morgan Kaufmann, 1994, pp. 293–304.
21. Kakas, A. C., and Mancarella, P., Generalized Stable Models: A Semantics for Abduction, in: *Proc. 9th European Conference on Artificial Intelligence*, Pitman, 1990, pp. 385–391.
22. Kakas, A. C., and Mancarella, P., Knowledge Assimilation and Abduction, in: J. P. Martins and M. Reinfrank (eds.), *Truth Maintenance Systems: Proc. ECAI-90 Workshop, Lecture Notes in Artificial Intelligence 515*, Springer-Verlag, 1991, pp. 54–70.
23. Kakas, A. C., Kowalski, R. A., and Toni, F., Abductive Logic Programming, *J. Logic and Computation* 2(6):719–770 (1992).
24. Konolige, K., Abduction versus Closure in Causal Theories, *Artificial Intelligence* 53:255–272 (1992).
25. Lloyd, J. W., and Topor, R. W., Making Prolog More Expressive, *J. Logic Programming* 3:225–240 (1984).
26. Manthey, R., and Bry, F., SATCHMO: A Theorem Prover Implemented in Prolog, in: *Proc. 9th International Conference on Automated Deduction, Lecture Notes in Computer Science 310*, Springer-Verlag, 1988, pp. 415–434.
27. Minker, J., On Indefinite Data Bases and the Closed World Assumption, in: *Proc. 6th International Conference on Automated Deduction, Lecture Notes in Computer Science 138*, Springer-Verlag, 1982, pp. 292–308.
28. Minker, J., and Rajasekar, A., A Fixpoint Semantics for Disjunctive Logic Programs, *J. Logic Programming* 9:45–74 (1990).
29. Poole, D., Goebel, R., and Aleliunas, R., Theorist: A Logical Reasoning System for Defaults and Diagnosis, in: N. Cercone and G. McCalla (eds.), *The Knowledge Frontier: Essays in the Representation of Knowledge*, Springer, 1987, pp. 331–352.
30. Preist, C., and Eshghi, E., Consistency-Based and Abductive Diagnoses as Generalized Stable Models, in: *Proc. International Conference on Fifth Generation Computer Systems 1992*, Ohmsha, 1992, pp. 514–521.
31. Sacca, D., and Zaniolo, C., Stable Models and Non-Determinism in Logic Programs with Negation, in: *Proc. 9th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, 1990, pp. 205–229.
32. Sakama, C., and Inoue, K., On the Equivalence between Disjunctive and Abductive Logic Programs, in: P. V. Hentenryck (ed.), *Logic Programming: Proc. 11th International Conference*, MIT Press, 1994, pp. 489–503.
33. Sakama, C., and Inoue, K., An Alternative Approach to the Semantics of Disjunctive Logic Programs and Deductive Databases, *J. Automated Reasoning* 13(1):145–172 (1994).

34. Sakama, C., and Inoue, K., Paraconsistent Stable Semantics for Extended Disjunctive Programs, *J. Logic and Computation* 5(3):265–285 (1995).
35. Satoh, K., and Iwayama, N., Computing Abduction by Using the TMS, in: K. Furukawa (ed.), *Logic Programming: Proc. 8th International Conference*, MIT Press, 1991, pp. 505–518.
36. Satoh, K., and Iwayama, N., A Query Evaluation Method for Abductive Logic Programming, in: K. Apt (ed.), *Logic Programming: Proc. Joint International Conference and Symposium*, MIT Press, 1992, pp. 671–684.
37. van Emden, M. H., and Kowalski R. A., The Semantics of Predicate Logic as a Programming Language, *J. ACM* 23:733–742 (1976).