ICOT Technical Report: TR-0824

TR-0824

Transforming Abductive Logic Programs to Disjunctive Programs

by

K. Inoue & C. Sakama (ASTEM)

December, 1992

ICOT

Mita Kokusai Bldg. 21F
4-28 Mita 1-Chome
Minato-ku Tokyo 108 Japan

(03)3456-3191 ~5
Telex ICOT J32964

Institute for New Generation Computer Technology

Transforming Abductive Logic Programs to Disjunctive Programs

Katsumi Inoue
ICOT
Mita Kokusai Bldg., 21F
1-4-28 Mita, Minato-ku
Tokyo 108, Japan
phone: +81-3-3456-4365
inoue@icot.or.jp

Chiaki Sakama
ASTEM
Research Institute of Kyoto
17 Chudoji Minami-machi
Shimogyo, Kyoto 600, Japan
phone: +81-75-315-8651
sakama@astem.or.jp

November 30, 1992

Abstract

A new fixpoint semantics for abductive logic programs is provided, in which the generalized stable 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. This characterization allows us to have a parallel bottom-up model generation procedure for computing abductive explanations from arbitrary (range-restricted and function-free) general, extended, and disjunctive programs with integrity constraints.

1 Introduction

Abduction, an inference to explanation, has recently been recognized as a very important form of reasoning for logic programming as well as various AI problems. In [EK89, KM90, Gel90, Ino91], abduction is expressed as an extension of logic programming. Eshghi and Kowalski [EK89] give an abductive interpretation of *negation as failure* [Cla78] in the class of *general logic programs*, and show a 1-1 correspondence between the *stable models* [GL88] of a general logic program and the extensions of its associated abductive framework. Their approach is extended by several literatures such as [KM90, Dun91, KM91] (a comprehensive survey is in [KKT92]). Kakas and Mancarèlla [KM90] propose a framework of *abductive logic programming* which supplies abduction not only for negation as failure but in a general way: Their abductive framework is defined as a triple $\langle P, \Gamma, \mathcal{I} \rangle$, where P is a general logic program, Γ is a set of *abducible predicates*, and $\mathcal{I}$ is a set of *integrity constraints*. Then, a *generalized stable model* of $\langle P, \Gamma, \mathcal{I} \rangle$ 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

[Gel90] proposes an abductive framework within *extended disjunctive programs* [GL91] that allow disjunctions in heads and classical negation along with negation as failure. Further, Inoue [Ino91] proposes a more general framework for hypothetical reasoning, called a *knowledge system*, by allowing any extended logic program as candidate hypotheses Γ, and shows its applicability to default reasoning, closed world assumption, and inconsistency resolution as well as abduction. Moreover, every knowledge system is shown to be transformed into a semantically equivalent abductive logic programming framework.

However, difficulty arises in designing procedures for these abductive frameworks since (generalized) stable models of a program were not originally defined in a constructive way. Moreover, we cannot have any elegant top-down proof procedure which is sound with respect to the (generalized) stable model semantics. In fact, Eshghi and Kowalski [EK89] show an abductive proof procedure for general logic programs by incorporating consistency tests into SLD resolution, but its soundness with respect to the stable model semantics is not guaranteed in general. Also, Kakas and Mancarella [KM91] define a top-down procedure for abductive logic programming as an extension of [EK89]'s proof procedure, but their procedure inherits the unsoundness problem with respect to the stable model semantics.

In order to compute stable models of a general logic program or *answer sets* [GL91] of an extended disjunctive program, Inoue et al [IKH92] have shown a *constructive* definition of stable models and answer sets, and provided a bottom-up procedure based on *model generation* techniques [MB88, FH91]. Recently, we pointed out that this proposal can also be regarded as a new fixpoint semantics for general and extended (disjunctive) logic programs [IS92]. The basic idea of this technique is to transform a program into a semantically equivalent *positive disjunctive program* that does not contain negation as failure.

In this paper, we generalize Inoue et al's program transformation technique for non-abductive programs [IKH92, IS92] to deal with abductive frameworks. Namely, we transform an abductive logic programming framework into a positive disjunctive program not containing negation as failure. We then show that the generalized stable models of an abductive framework can be characterized by the fixpoint closure of the transformed program. This fixpoint characterization is further extended to an abductive framework with extended disjunctive programs. For a procedural aspect of our fixpoint semantics, we also show that a model generation method for positive disjunctive programs can be used as a sound and complete procedure for computing generalized stable models for function-free, 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, general logic programs and abductive logic programming. In Section 4, we extend the results to extended disjunctive programs with abducibles. Section 5 presents a procedure for computing generalized stable models. Some comparisons between our fixpoint theory and previously proposed approaches such as [CDT91, KM91, SI91] are discussed in Section 6.

2 Model Theory for Abductive Logic Programs

There are several definitions of abduction [PGA87, EK89, KM90, Bry90, Gel90, Ino91, CDT91, Ino92]. The semantics of abduction we use here is based on the generalized stable models defined by Kakas and Mancarella [KM90]. As stated in the previous section, their abductive framework is an extension of Eshghi and Kowalski's framework [EK89], and is given by a triple $\langle P, \Gamma, \mathcal{I} \rangle$, where P is a general 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 [PGA87, Ino92], Kakas and Mancarella define a program P not as first-order formulas but as a *general logic program* with negation as failure. This definition covers a more general class of programs than Console et al's *object-level* abduction [CDT91] that is defined for hierarchical logic programs. Two different definitions by Gelfond [Gel90] and Inoue [Ino91] are more general than that by [KM90] 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 general logic program* 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 P, we include them in a program 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 following sections. For this purpose, we first give the syntax and stable model semantics of general logic programs.

Definition 2.1 A *general logic program* is a finite set of *clauses*, which are either of the form:

$$H \leftarrow B_1 \wedge \ldots \wedge B_m \wedge not\, B_{m+1} \wedge \ldots \wedge not\, B_n\,, \tag{1}$$

or of the form:

$$\leftarrow B_1 \wedge \ldots \wedge B_m \wedge not\, B_{m+1} \wedge \ldots \wedge not\, B_n\,, \tag{2}$$

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 *integrity constraint*. An integrity constraint is called a *negative clause* if $m = n$.

A general logic program not containing *not* is called a *Horn program*. A Horn program not containing negative clauses is called a *definite program*.

In the above definition, we allow in a program integrity constraints as clauses with empty heads, which are not explicitly defined in [GL88]. While [KM90] defines integrity constraints as first-order formulas separated from programs, every integrity constraint in the form of a first-order formula F can be first characterized as a clause without a head, $\leftarrow not\, F$, then translated into clauses using the transformation of [LT84]. For instance, an integrity constraint $p \supset q$ can be expressed by $\leftarrow p \wedge not\, q$. This translation is also employed by [SK87].

In the semantics of a general logic program, a clause containing variables stands for the possibly infinite set of its ground instances obtained by instantiating every variable by the elements of the Herbrand universe of the program in every possible way. An *interpretation* of a program is defined as a subset of the Herbrand base $\mathcal{HB}$ of the program. An interpretation I *satisfies* a ground Horn clause $H \leftarrow B_1 \wedge \ldots \wedge B_m$ if $\{B_1, \ldots, B_m\} \subseteq I$ implies $H \in I$. Especially, I satisfies a ground negative clause $\leftarrow B_1 \wedge \ldots \wedge B_m$ if $\{B_1, \ldots, B_m\} \not\subseteq I$. For a Horn program P, the smallest interpretation satisfying every ground clause from P is called the *least model* of P.

Definition 2.2 Let P be a general logic program, and I an interpretation. The *reduct* P^I of P by I is defined as follows: A clause $H \leftarrow B_1 \wedge \ldots \wedge B_m$ (resp. $\leftarrow B_1 \wedge \ldots \wedge B_m$) is in P^I if there is a ground clause $H \leftarrow B_1 \wedge \ldots \wedge B_m \wedge \mathit{not}\, B_{m+1} \wedge \ldots \wedge \mathit{not}\, B_n$ (resp. $\leftarrow B_1 \wedge \ldots \wedge B_m \wedge \mathit{not}\, B_{m+1} \wedge \ldots \wedge \mathit{not}\, B_n$) from P such that $\{B_{m+1}, \ldots, B_n\} \cap I = \emptyset$.

Then, I is a *stable model* [GL88] of P if I is the least model of P^I.

We say that a general logic program P is *consistent* if it has a stable model; otherwise, it is called *inconsistent*.

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

Definition 2.3 An *abductive general logic program* is a pair $\langle P, \Gamma \rangle$, where P is a general logic program, and Γ is a set of predicate symbols from P called the *abducible predicates*. The set of all ground atoms $\mathcal{A}_\Gamma$ ($\subseteq \mathcal{HB}$) having abducible predicates from Γ is called the *abducibles*. When P is a Horn program, $\langle P, \Gamma \rangle$ is called an *abductive Horn program*.

Note that the above definition is an extension of that by Kakas and Mancarella [KM90] to allow any general logic program (with integrity constraints) in P, while [KM90] requires that abducibles do not appear in heads of ground clauses.

Definition 2.4 Let $\langle P, \Gamma \rangle$ be an abductive general logic program and E a subset of $\mathcal{A}_\Gamma$. An interpretation $I(E)$ is a *generalized stable model* of $\langle P, \Gamma \rangle$ if it is a stable model of the general logic program $P \cup E$[1] and satisfies $E = I(E) \cap \mathcal{A}_\Gamma$.

Each generalized stable model in the above definition reduces to a stable model of P when $\Gamma = \emptyset$. In Definition 2.4, the condition $E = I(E) \cap \mathcal{A}_\Gamma$ is necessary since an abducible appearing in the head of a ground clause may become true when other abducibles from E are true (see Example 2.1 below). In this way, each generalized stable model $I(E)$ can be uniquely associated with its "generating" abducibles E. A similar extension has been proposed by Preist and Eshghi [PE92].

Definition 2.5 Let $\langle P, \Gamma \rangle$ be an abductive general logic program and O an atom. A set $E \subseteq \mathcal{A}_\Gamma$ is an *explanation* of O (*with respect to* $\langle P, \Gamma \rangle$) if there is a generalized stable model $I(E)$ which satisfies O.

[1] For each abducible $A \in \mathcal{A}_\Gamma$, we identify the atom A with the clause $A \leftarrow$ in E.

In the following, we assume that an observation O is a *non-abducible ground atom*. This restriction is not essential for the following reasons. First, if O is an abducible, every its explanation trivially contains O. 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 $O_1, \ldots, O_m$ are observed and other atoms $O_{m+1}, \ldots, O_n$ are not observed, by introducing a clause $O \leftarrow O_1 \wedge \ldots \wedge O_m \wedge \mathit{not}\, O_{m+1} \wedge \ldots \wedge \mathit{not}\, O_n$ and computing explanations of O.

We usually prefer those explanations which contain abducibles as few as possible.

Definition 2.6 Let $\langle P, \Gamma \rangle$ be an abductive general logic program and E a subset of $\mathcal{A}_\Gamma$. A generalized stable model $I(E)$ is *minimal* if no generalized stable model $I(E')$ satisfies that $E' \subset E$.

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

Example 2.1 Consider an abductive Horn program $\langle P, \Gamma \rangle$ where

$$P = \{\ \mathit{sore}(\mathit{leg}) \leftarrow \mathit{broken}(\mathit{leg}),\ \ \mathit{broken}(\mathit{leg}) \leftarrow \mathit{broken}(\mathit{tibia})\ \}$$

and $\Gamma = \{\mathit{broken}\}$. Let $O = \mathit{sore}(\mathit{leg})$ be an observation. Then, $E = \{\mathit{broken}(\mathit{leg})\}$ is a minimal explanation of O. While $E' = \{\mathit{broken}(\mathit{tibia}), \mathit{broken}(\mathit{leg})\}$ is a (non-minimal) explanation of O, $E'' = \{\mathit{broken}(\mathit{tibia})\}$ is not an explanation of O, since $\mathit{broken}(\mathit{tibia})$ causes $\mathit{broken}(\mathit{leg})$ so that there is no generalized stable model $I(E'')$ satisfying $E'' = I(E'') \cap \mathcal{A}_\Gamma$. 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 $\mathit{broken}(\mathit{leg})$ is more likely than that of $\mathit{broken}(\mathit{tibia})$.

Lemma 2.1 Let $\langle P, \Gamma \rangle$ be an abductive general logic program, E a subset of $\mathcal{A}_\Gamma$, and O an atom. Then, E is a minimal explanation of O with respect to $\langle P, \Gamma \rangle$ iff $I(E)$ is a minimal generalized stable model of $\langle P \cup \{ \leftarrow \mathit{not}\, O\}, \Gamma \rangle$.

Proof: First, observe that the addition of $\leftarrow \mathit{not}\, O$ to P imposes the integrity constraint that O should be derived. Then,

E is a minimal explanation of O with respect to $\langle P, \Gamma \rangle$
$\Leftrightarrow$ no $E' \subset E$ is an explanation of O with respect to $\langle P, \Gamma \rangle$
$\Leftrightarrow$ no generalized stable model $I(E')$ of $\langle P, \Gamma \rangle$ in which O is true satisfies $E' \subset E$
$\Leftrightarrow$ no generalized stable model $I(E')$ of $\langle P \cup \{ \leftarrow \mathit{not}\, O\}, \Gamma \rangle$ satisfies $E' \subset E$
$\Leftrightarrow$ $I(E)$ is a minimal generalized stable model of $\langle P \cup \{ \leftarrow \mathit{not}\, O\}, \Gamma \rangle$. □

Example 2.2 Consider an abductive general logic program $\langle P, \Gamma \rangle$ where

$$P = \{\ p \leftarrow r \wedge b \wedge \mathit{not}\, q\,,\quad q \leftarrow a\,,\quad r \leftarrow\,,\quad \leftarrow \mathit{not}\, p\ \}$$

and $\Gamma = \{a, b\}$. The unique generalized stable model of $\langle P, \Gamma \rangle$ is $I(E) = \{r, p, b\}$. If we regard $\leftarrow \mathit{not}\, p$ as an observation, $E = I(E) \cap \mathcal{A}_\Gamma = \{b\}$ 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 general logic programs. First, we introduce (a) a fixpoint semantics for positive disjunctive programs [IS92], then (b) a fixpoint semantics for general logic programs [IS92] using a transformation to positive disjunctive programs by [IKH92]. Next, (c) a fixpoint semantics for abductive Horn programs is given using another program transformation, then finally it is extended to (d) a fixpoint semantics for abductive general logic programs by combining the transformations of (b) and (c).

3.1 Fixpoint Semantics for Positive Disjunctive Programs

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

$$H_1 \vee \ldots \vee H_l \leftarrow B_1 \wedge \ldots \wedge B_m \quad (l, m \geq 0) \tag{3}$$

where H_i's and B_j's are atoms. An interpretation I *satisfies* a ground clause of the form (3) if $\{B_1, \ldots, B_m\} \subseteq I$ implies $H_i \in I$ for some $1 \leq i \leq l$. Then, the semantics of a positive disjunctive program P is given by its *minimal models* [Min82] each of which is defined by a minimal interpretation satisfying all ground clauses from P.

To characterize the nondeterministic behavior of a disjunctive program, Inoue and Sakama [IS92] have introduced an ordering and a closure operator over a lattice of the sets of Herbrand interpretations $2^{2^{\mathcal{HB}}}$ as follows.

Definition 3.1 Let $\mathbf{I}$ and $\mathbf{J}$ be sets of interpretations. Then, $\mathbf{I} \sqsubseteq \mathbf{J}$ iff $\mathbf{I} = \mathbf{J}$ or $\forall J \in \mathbf{J} \setminus \mathbf{I}$, $\exists I \in \mathbf{I} \setminus \mathbf{J}$ such that $I \subset J$.

Note that $\sqsubseteq$ is a partial order and each element in $2^{2^{\mathcal{HB}}}$ makes a complete lattice under the ordering $\sqsubseteq$ with the top element $\emptyset$ and the bottom element $2^{\mathcal{HB}}$.

Definition 3.2 Let P be a positive disjunctive program and $\mathbf{I}$ be a set of interpretations. Then a mapping $\mathbf{T}_P : 2^{2^{\mathcal{HB}}} \to 2^{2^{\mathcal{HB}}}$ is defined by

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

where the mapping $T_P : 2^{\mathcal{HB}} \to 2^{2^{\mathcal{HB}}}$ is defined as follows:

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

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

Definition 3.2 says that, if an interpretation I does not satisfy a ground negative clause then $T_P(I) = \emptyset$, else $T_P(I)$ contains every interpretation obtained from I by adding each single disjunct from every ground clause that is not satisfied by I.

Definition 3.3 The *ordinal powers* of T_P are defined as follows.

$$\begin{aligned} \mathrm{T}_P \uparrow 0 &= \{\emptyset\} \\ \mathrm{T}_P \uparrow n+1 &= \mathrm{T}_P(\mathrm{T}_P \uparrow n) \\ \mathrm{T}_P \uparrow \omega &= lub\{\mathrm{T}_P \uparrow n \mid n < \omega\} \end{aligned}$$

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

Example 3.1 Let $P = \{\ p \vee q \leftarrow r, \quad s \leftarrow r, \quad r \leftarrow\ , \quad \leftarrow q \wedge s\ \}$. Then, we get $\mathbf{T}_P \uparrow 1 = \{\{r\}\}$, $\mathbf{T}_P \uparrow 2 = \{\{r,s,p\},\{r,s,q\}\}$, and $\mathbf{T}_P \uparrow 3 = \{\{r,s,p\}\} = \mathbf{T}_P \uparrow \omega$.

Theorem 3.1 [IS92] (a) $\mathrm{T}_P \uparrow \omega$ is a fixpoint. We call it a *disjunctive fixpoint* of P.
(b) Each element in $\mathrm{T}_P \uparrow \omega$ is a model of P.
(c) Let $\mathcal{MM}_P$ be the set of all minimal models of P. Then, $\mathcal{MM}_P = min(\mathrm{T}_P \uparrow \omega)$, where $min(\mathrm{I}) = \{\, I \in \mathrm{I} \mid \nexists J \in \mathbf{I}$ such that $J \subset I \,\}$.
(d) A positive disjunctive program P is inconsistent iff $\mathrm{T}_P \uparrow \omega = \emptyset$.
(e) If P is a definite program, $\mathrm{T}_P \uparrow \omega$ contains a unique element I which is the least model of P.

Theorem 3.1 (c) characterizes the *minimal model semantics* [Min82] of a positive disjunctive program. On the other hand, (d) can be used as a test for the consistency of a positive disjunctive program. Furthermore, (e) says that, for a definite program, our fixpoint construction reduces to van Emden and Kowalski's fixpoint semantics [vEK76].

3.2 Fixpoint Semantics for General Logic Programs

To characterize the stable model semantics of a general logic program, Inoue et al have proposed a program transformation which transforms a general logic program into a semantically equivalent *not*-free disjunctive program [IKH92].

Definition 3.4 [IKH92] Let P be a general logic program and $\mathcal{HB}$ be its Herbrand base. Then P^κ is the program obtained as follows.

(i) For each clause $H \leftarrow B_1 \wedge \ldots \wedge B_m \wedge not\, B_{m+1} \wedge \ldots \wedge not\, B_n$ in P, P^κ contains

$$(H \wedge \neg \mathsf{K}B_{m+1} \wedge \ldots \wedge \neg \mathsf{K}B_n) \vee \mathsf{K}B_{m+1} \vee \ldots \vee \mathsf{K}B_n \ \leftarrow\ B_1 \wedge \ldots \wedge B_m . \qquad (4)$$

Especially, each integrity constraint becomes $\mathsf{K}B_{m+1} \vee \ldots \vee \mathsf{K}B_n \leftarrow B_1 \wedge \ldots \wedge B_m$.

(ii) For each atom B in $\mathcal{HB}$, P^κ contains a clause $\leftarrow \neg \mathsf{K}B \wedge B$.

Here, $\mathsf{K}B$ (resp. $\neg\mathsf{K}B$) is a new *atom* which denotes B *is believed* (resp. *disbelieved*). In the transformation (i), each *not* B_i is rewritten in $\neg\mathsf{K}B_i$ and shifted to the head of the clause. Moreover, since the head H becomes true when each $\neg\mathsf{K}B_i$ in the body is true, the condition $\neg\mathsf{K}B_{m+1} \wedge \ldots \wedge \neg\mathsf{K}B_n$ is added to H. The constraint (ii) says that each atom B cannot be true and disbelieved at the same time.

An *interpretation* I^κ is now defined as a subset of the new Herbrand base: $\mathcal{HB}^\kappa = \mathcal{HB} \cup \{\mathsf{K}B \mid B \in \mathcal{HB}\} \cup \{\neg\mathsf{K}B \mid B \in \mathcal{HB}\}$. An atom in $\mathcal{HB}^\kappa$ is called *objective* if it is in $\mathcal{HB}$, and the set of objective atoms in an interpretation I^κ is denoted as $obj(I^\kappa)$.

In [IKH92], it is shown that the stable models of a program can be produced *constructively* from the transformed program. Here, we characterize the result using the disjunctive fixpoint of the transformed program. For this purpose, we slightly modify a mapping presented in Definition 3.2 to allow a disjunction of conjunctions of atoms in the head of a clause. For a conjunction of atoms $F = H_1 \wedge \ldots \wedge H_k$, we denote the set of its conjuncts as $conj(F) = \{H_1, \ldots, H_k\}$. Let P^κ be a program, and I^κ an interpretation. A mapping $T_{P^\kappa} : 2^{\mathcal{HB}^\kappa} \rightarrow 2^{2^{\mathcal{HB}^\kappa}}$ is now defined as:

$$T_{P^\kappa}(I^\kappa) = \begin{cases} \emptyset, & \text{if } I^\kappa \models B_1 \wedge \ldots \wedge B_m \text{ for some ground negative clause} \\ & \quad \leftarrow B_1 \wedge \ldots \wedge B_m \text{ from } P^\kappa; \\ \{\, J^\kappa \mid & \text{for each ground clause } C_i:\ F_1^i \vee \ldots \vee F_{l_i}^i \leftarrow B_1^i \wedge \ldots \wedge B_{m_i}^i \\ & \text{from } P^\kappa \text{ such that } \{B_1^i, \ldots, B_{m_i}^i\} \subseteq I^\kappa \text{ and} \\ & conj(F_j^i) \not\subseteq I^\kappa \text{ for any } j = 1, \ldots, l_i, \\ & \quad J^\kappa = I^\kappa \cup \bigcup_{C_i} conj(F_j^i)\ (1 \leq j \leq l_i)\ \}, \quad \text{otherwise}. \end{cases}$$

Using this definition, the mapping $\mathbf{T}_{P^\kappa}$ and its disjunctive fixpoint are also defined in the same way as in Section 3.1 and those properties presented there still hold.

Definition 3.5 An interpretation I^κ is called *canonical* if it satisfies the condition: for each ground atom A, if $\mathsf{K}A \in I^\kappa$ then $A \in I^\kappa$. For a set $\mathbf{I}^\kappa$ of interpretations, we write: $obj_c(\mathbf{I}^\kappa) = \{\, obj(I^\kappa) \mid I^\kappa \in \mathbf{I}^\kappa \text{ and } I^\kappa \text{ is canonical} \,\}$.

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

Theorem 3.2 [IS92] Let P be a general logic program, P^κ its transformed form, and $\mathcal{ST}_P$ the set of all stable models of P. Then, $\mathcal{ST}_P = obj_c(\mathbf{T}_{P^\kappa} \uparrow \omega)$.
Especially, P is inconsistent iff $obj_c(\mathbf{T}_{P^\kappa} \uparrow \omega) = \emptyset$.

Example 3.2 Let $P = \{\, p \leftarrow not\, q,\ \ q \leftarrow not\, p,\ \ r \leftarrow q,\ \ r \leftarrow not\, r \,\}$. Then,

$$\begin{aligned} P^\kappa \ &= \ \{\, (p \wedge \neg\mathsf{K}q) \vee \mathsf{K}q \leftarrow,\ \ (q \wedge \neg\mathsf{K}p) \vee \mathsf{K}p \leftarrow,\ \ r \leftarrow q,\ \ (r \wedge \neg\mathsf{K}r) \vee \mathsf{K}r \leftarrow \,\} \\ &\cup \ \{\, \leftarrow \neg\mathsf{K}B \wedge B \mid B \in \{p, q, r\}\} . \end{aligned}$$

Now, $\mathbf{T}_{P^\kappa} \uparrow \omega = \{\, \{p, \neg\mathsf{K}q, \mathsf{K}p, \mathsf{K}r\}, \{\mathsf{K}q, q, \neg\mathsf{K}p, \mathsf{K}r, r\}, \{\mathsf{K}q, \mathsf{K}p, \mathsf{K}r\} \,\}$, in which only the second element is canonical. Hence, $obj_c(\mathbf{T}_{P^\kappa} \uparrow \omega) = \{\{q, r\}\}$, 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 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* $\neg \mathsf{K}B$ is checked through the integrity constraint $\leftarrow \neg \mathsf{K}B \wedge B$ during the fixpoint construction, while for the *positive hypothesis* $\mathsf{K}B$, the integrity checking is carried out by the *canonical constraint* that all the "assumed" literals are actually "derived" at the fixpoint.

Now, we move on to abduction. 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. The only difference is that for the positive hypothesis $\mathsf{K}A$ for each abducible A, *we do not need the canonical constraint.* We first present a transformation of an abductive Horn program.

Definition 3.6 Let (P, Γ) be an abductive Horn program. Then, P^{ϵ}_{Γ} is the program obtained as follows.

(i) For each Horn clause from P: $H \leftarrow B_1 \wedge \ldots \wedge B_m \wedge A_1 \wedge \ldots \wedge A_n$ $(m, n \geq 0)$, where B_i's are non-abducibles and A_j's are abducibles, P^{ϵ}_{Γ} contains the clause:

$$(H \wedge \mathsf{K}A_1 \wedge \ldots \wedge \mathsf{K}A_n) \vee \neg \mathsf{K}A_1 \vee \ldots \vee \neg \mathsf{K}A_n \leftarrow B_1 \wedge \ldots \wedge B_m . \qquad (5)$$

Especially, each negative clause becomes $\neg \mathsf{K}A_1 \vee \ldots \vee \neg \mathsf{K}A_n \leftarrow B_1 \wedge \ldots \wedge B_m$.

(ii) For each abducible A in $\mathcal{A}_{\Gamma}$, P^{ϵ}_{Γ} contains the following two clauses:

$$\leftarrow \neg \mathsf{K}A \wedge A , \qquad (6)$$

$$A \leftarrow \mathsf{K}A . \qquad (7)$$

We can see that the clause (5) transformed from an abductive Horn program and the clause (4) transformed from a general logic program are *dual* in the sense that an abduced atom A is dealt with as a positive hypothesis $\mathsf{K}A$, while a negation-as-failure formula *not* B is dealt with as a negative hypothesis $\neg \mathsf{K}B$. Here, however, we have the additional clause (7) for each abducible A. Since this clause derives A whenever an interpretation contains the positive hypothesis $\mathsf{K}A$, it makes every interpretation in $\mathrm{T}_{P^{\epsilon}_{\Gamma}} \uparrow \omega$ satisfy the canonical condition defined in Definition 3.5. Hence, for each Horn clause from P, we can replace the transformed clause (5) in P^{ϵ}_{Γ} with the clause

$$(H \wedge A_1 \wedge \ldots \wedge A_n) \vee \neg \mathsf{K}A_1 \vee \ldots \vee \neg \mathsf{K}A_n \leftarrow B_1 \wedge \ldots \wedge B_m \qquad (8)$$

and omit each clause (7) for each abducible A in $\mathcal{A}_{\Gamma}$. We denote as P^{κ}_{Γ} the program obtained from P by this alternative transformation. Since this change does not affect the result of the fixpoint of P_Γ^ε as far as objective atoms are concerned, we can identify P_Γ^κ with P_Γ^ε. [2] In this way, each abduced atom can be added to an interpretation without imposing the condition that it should be derived.

Lemma 3.3 Let $\langle P, \Gamma \rangle$ be an abductive Horn program.
(a) For any $I^\kappa \in \mathbf{T}_{P_\Gamma^\kappa} \uparrow \omega$, $obj(I^\kappa)$ is a generalized stable model of $\langle P, \Gamma \rangle$.
(b) For any generalized stable model $I(E)$ of $\langle P, \Gamma \rangle$, there is a generalized stable model $I(E')$ of $\langle P, \Gamma \rangle$ such that $E' \subseteq E$, $I(E') \setminus E' = I(E) \setminus E$, and $I(E') = obj(I^\kappa)$ for some $I^\kappa \in \mathbf{T}_{P_\Gamma^\kappa} \uparrow \omega$.
(c) If $E \subseteq \mathcal{A}_\Gamma$ is an explanation of an atom O, then there is an explanation E' of O such that $E' \subseteq E$ and $I(E') = obj(I^\kappa)$ for some $I^\kappa \in \mathbf{T}_{P_\Gamma^\kappa} \uparrow \omega$.

Proof: (a) Let $E = obj(I^\kappa) \cap \mathcal{A}_\Gamma$, and P' be the definite program obtained from P by removing every negative clause. By Theorem 3.1 (e), $\mathbf{T}_{P' \cup E} \uparrow \omega$ contains the unique element I. Then, for each ground clause of the form $H \leftarrow B_1 \wedge \ldots \wedge B_m \wedge A_1 \wedge \ldots \wedge A_n$ (A_j's are abducibles) from P', if $\{B_1, \ldots, B_m\} \subseteq I$ then either $\{A_1, \ldots, A_n, H\} \subseteq I$ or $\exists j \ (1 \leq j \leq n)$ such that $A_j \not\in I$, and for the corresponding clause of the form (8), if $\{B_1, \ldots, B_m\} \subseteq I$ then either $\{A_1, \ldots, A_n, H\} \subseteq I^\kappa$ or $\exists j \ (1 \leq j \leq n)$ such that $\neg \mathsf{K} A_j \in I^\kappa$. Hence, $I = obj(I^\kappa)$. Since I is the least model of $P' \cup E$ and $P \cup E$ is a consistent Horn program, I is also the stable model of $P \cup E$. By definition, I is a generalized stable model of $\langle P, \Gamma \rangle$.

(b) For any atom $H^i \in I(E) \setminus E$, there is a ground clause C^i: $H^i \leftarrow B_1^i \wedge \ldots \wedge B_{m_i}^i \wedge A_1^i \wedge \ldots \wedge A_{n_i}^i$ (A_j^i's are abducibles) from P such that $\{B_1^i, \ldots, B_{m_i}^i\} \subseteq I(E) \setminus E$ and $\{A_1^i, \ldots, A_{n_i}^i\} \subseteq E$. Let $E' = \bigcup_{H^i \in I(E) \setminus E} \{A_1^i, \ldots, A_{n_i}^i\}$. Since for the clause C^i, there is the corresponding clause $(H^i \wedge A_1^i \wedge \ldots \wedge A_{n_i}^i) \vee \neg \mathsf{K} A_1^i \vee \ldots \vee \neg \mathsf{K} A_{n_i}^i \leftarrow B_1^i \wedge \ldots \wedge B_{m_i}^i$ is in P_Γ^κ, if $\{B_1^i, \ldots, B_{m_i}^i\} \subseteq J$ 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 $J \cup \{H_i, A_1^i, \ldots, A_{n_i}^i\} \subseteq J'$. Since $\{H_i, A_1^i, \ldots, A_{n_i}^i\} \subseteq I(E)$ 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 $E' \subseteq I^\kappa$. By (a), $obj(I^\kappa)$ is a generalized stable model of $\langle P, \Gamma \rangle$. It follows immediately that $E' \subseteq E$, $I(E') \setminus E' = I(E) \setminus E$, and $I(E') = obj(I^\kappa)$.

(c) If E is an explanation of O, then there is a generalized stable model $I(E)$ of $\langle P, \Gamma \rangle$ satisfying O. By (b), there is a generalized stable model $I(E')$ of $\langle P, \Gamma \rangle$ such that $E' \subseteq E$, $I(E') \setminus E' = I(E) \setminus E$, and $I(E') = obj(I^\kappa)$ 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. □

3.4 Fixpoint Semantics for Abductive General Logic Programs

Now, we show a transformation of abductive general logic programs by combining the two transformations shown in Sections 3.2 and 3.3. Each negation-as-failure formula ***not*** B for a non-abducible B is translated in the same way as Definition 3.4: it is split into $\neg \mathsf{K} B$ and

[2] In the transformation (8), each hypothesis A_j can be considered to be *skipped* instead of being resolved. Thus, this operation is a bottom-up counterpart of the Skip rule in SOL resolution [Ino92] that is defined for top-down abduction.

$\mathsf{K}B$. On the other hand, when a negation-as-failure formula *not* A mentions an abducible A, it should be split into $\neg\mathsf{K}A$ and A. This is because for each abducible A, we can deal with it as if the axiom (7) $A \leftarrow \mathsf{K}A$ is present.

Definition 3.7 Let $\langle P, \Gamma \rangle$ be an abductive general program. Then, P^{κ}_{Γ} is the program obtained as follows.

(i) For each clause from P:

$$H \leftarrow B_1 \wedge \ldots \wedge B_m \wedge A_1 \wedge \ldots \wedge A_n \wedge \mathit{not}\, B_{m+1} \wedge \ldots \wedge \mathit{not}\, B_s \wedge \mathit{not}\, A_{n+1} \wedge \ldots \wedge \mathit{not}\, A_t$$

where $s \geq m \geq 0$, $t \geq n \geq 0$, B_j's are non-abducibles, and A_k's are abducibles,

$$\begin{aligned}(H \wedge \bigwedge_{i=1}^{n} A_i \wedge \bigwedge_{j=m+1}^{s} \neg\mathsf{K}B_j \wedge \bigwedge_{k=n+1}^{t} \neg\mathsf{K}A_k) \vee \bigvee_{i=1}^{n} \neg\mathsf{K}A_i \vee \bigvee_{j=m+1}^{s} \mathsf{K}B_j \vee \bigvee_{k=n+1}^{t} A_k \\ \leftarrow B_1 \wedge \ldots \wedge B_m \qquad (9)\end{aligned}$$

is in P^{κ}_{Γ}. Especially, each integrity constraint is transformed to:

$$\neg\mathsf{K}A_1 \vee \ldots \vee \neg\mathsf{K}A_n \vee \mathsf{K}B_{m+1} \vee \ldots \vee \mathsf{K}B_s \vee A_{n+1} \vee \ldots \vee A_t \leftarrow B_1 \wedge \ldots \wedge B_m \,.$$

(ii) For each atom H in $\mathcal{HB}$, P^{κ}_{Γ} contains the clause $\leftarrow \neg\mathsf{K}H \wedge H$.

Notice that a transformed program P^{κ}_{Γ} in Definition 3.7 reduces to the program P^{κ} in Section 3.2 when Γ is empty, and reduces to the program P^{κ}_{Γ} in Section 3.3 when P is a Horn program.

Lemma 3.4 Let $\langle P, \Gamma \rangle$ be an abductive general logic program, and E a subset of $\mathcal{A}_{\Gamma}$. Then, $I(E)$ is a generalized stable model of $\langle P, \Gamma \rangle$ iff $I(E)$ is a generalized stable model of $\langle P^{I(E)}, \Gamma \rangle$.

Proof: $I(E)$ is a generalized stable model of $\langle P, \Gamma \rangle$
$\Leftrightarrow$ $I(E)$ is a stable model of $P \cup E$ and $E = I(E) \cap \mathcal{A}_{\Gamma}$
$\Leftrightarrow$ $I(E)$ is the least (and stable) model of $P^{I(E)} \cup E^{I(E)}$ and $E = I(E) \cap \mathcal{A}_{\Gamma}$
$\Leftrightarrow$ $I(E)$ is a generalized stable model of $\langle P^{I(E)}, \Gamma \rangle$ (because $E^{I(E)} = E$). □

Lemma 3.5 Let $\langle P, \Gamma \rangle$ be an abductive general logic program.
(a) For any $I \in \mathit{obj}_c(\mathrm{T}_{P^{\kappa}_{\Gamma}} \uparrow \omega)$, I is a generalized stable model of $\langle P, \Gamma \rangle$.
(b) For any generalized stable model $I(E)$ of $\langle P, \Gamma \rangle$, a generalized stable model $I(E')$ of $\langle P, \Gamma \rangle$ is in $\mathit{obj}_c(\mathrm{T}_{P^{\kappa}_{\Gamma}} \uparrow \omega)$ such that $E' \subseteq E$ and $I(E') \setminus E' = I(E) \setminus E$.
(c) If $E \subseteq \mathcal{A}_{\Gamma}$ is an explanation of an atom O, then there is an explanation E' of O such that $E' \subseteq E$ and $I(E') \in \mathit{obj}_c(\mathrm{T}_{P^{\kappa}_{\Gamma}} \uparrow \omega)$.

Proof: (a) Let $I^\kappa \in \mathbf{T}_{P^\kappa_\Gamma} \uparrow \omega$ such that I^κ is canonical, and $I(E) = obj(I^\kappa)$. For each ground clause of the form (9) from P^κ_Γ, if $\{B_1, \ldots, B_m\} \subseteq I(E) \setminus E$, then either (i) $H \in I(E)$, $\{A_1, \ldots, A_n\} \subseteq E$ and $\{\neg\mathsf{K}B_{m+1}, \ldots, \neg\mathsf{K}B_s, \neg\mathsf{K}A_{n+1}, \ldots, \neg\mathsf{K}A_t\} \subseteq I^\kappa$, (ii) $\exists i\ (1 \le i \le n)$ such that $\neg\mathsf{K}A_i \in I^\kappa$, (iii) $\exists j\ (m+1 \le j \le s)$ such that $\mathsf{K}B_j \in I^\kappa$, or (iv) $\exists k\ (n+1 \le k \le t)$ such that $A_k \in E$. Now, consider the abductive Horn program $\langle P^{I(E)}, \Gamma \rangle$, and let $J^\kappa \in \mathbf{T}_{(P^{I(E)})^\kappa_\Gamma} \uparrow \omega$. For each ground clause of the form (9) from P^κ_Γ, if (iii′) $\mathsf{K}B_j \notin I^\kappa$ (then $\neg\mathsf{K}B_j \in I^\kappa$ and $B_j \notin I(E) \setminus E$ since $I^\kappa \in \mathbf{T}_{P^\kappa_\Gamma} \uparrow \omega$) for any $j = m+1, \ldots, s$ and (iv′) $A_k \notin E$ for any $k = n+1, \ldots, t$, then there is a ground clause of the form (8) from $(P^{I(E)})^\kappa_\Gamma$, and it holds that, if $\{B_1, \ldots, B_m\} \subseteq J^\kappa$ then either (i′) $\{H, A_1, \ldots, A_n\} \subseteq J^\kappa$ or (ii′) $\exists i\ (1 \le i \le n)$ such that $\neg\mathsf{K}A_i \in J^\kappa$. On the other hand, if (iii″) $\mathsf{K}B_j \in I^\kappa$ (then $B_j \in I(E) \setminus E$ since I^κ is canonical) for some $j\ (m+1 \le j \le s)$ or (iv″) $A_k \in E$ for some $k\ (n+1 \le k \le t)$, then no corresponding clause exists in $(P^{I(E)})^\kappa_\Gamma$. Hence, there exists a J^κ satisfying $obj(J^\kappa) = I(E)$. Then, $I(E)$ is a generalized stable model of $\langle P^{I(E)}, \Gamma \rangle$ by Lemma 3.3 (a), and is a generalized stable model of $\langle P, \Gamma \rangle$ by Lemma 3.4.

Part (b) and (c) can be proved in a similar way to Lemma 3.3 (b) and (c). □

The next theorem characterizes the generalized stable model semantics of an abductive general logic program and the minimal explanations of an observation in terms of the disjunctive fixpoint of the transformed program. In the following, when $\mathbf{I}$ is a set of interpretations, we write: $min_\Gamma(\mathbf{I}) = \{\, I(E) \in \mathbf{I} \mid \not\exists I(E') \in \mathbf{I}$ such that $E' \subset E \,\}$. Note that $min_\Gamma(\mathbf{I}) = \{\, I(E) \in \mathbf{I} \mid E \in min(\{E' \mid I(E') \in \mathbf{I}\}) \,\}$.

Theorem 3.6 Let $\langle P, \Gamma \rangle$ be an abductive general logic program.
(a) Let $min\text{-}\mathcal{GST}_{\langle P,\Gamma \rangle}$ be the set of all minimal generalized stable models of $\langle P, \Gamma \rangle$. Then, $min\text{-}\mathcal{GST}_{\langle P,\Gamma \rangle} = min_\Gamma(obj_c(\mathbf{T}_{P^\kappa_\Gamma} \uparrow \omega))$.
(b) Let E be a subset of $\mathcal{A}_\Gamma$, and O an atom. Then, E is a minimal explanation of O with respect to $\langle P, \Gamma \rangle$ iff $I(E) \in min_\Gamma(obj_c(\mathbf{T}_{(P \cup \{\leftarrow not\, O\})^\kappa_\Gamma} \uparrow \omega))$.

Proof: (a) By Lemma 3.5 (b), it follows immediately that $min\text{-}\mathcal{GST}_{\langle P,\Gamma \rangle} \subseteq obj_c(\mathbf{T}_{P^\kappa_\Gamma} \uparrow \omega)$, and hence $min\text{-}\mathcal{GST}_{\langle P,\Gamma \rangle} \subseteq min_\Gamma(obj_c(\mathbf{T}_{P^\kappa_\Gamma} \uparrow \omega))$ holds. On the other hand, by Lemma 3.5 (a), every $I(E) \in obj_c(\mathbf{T}_{P^\kappa_\Gamma} \uparrow \omega)$ is a generalized stable model of $\langle P, \Gamma \rangle$. If $I(E) \in min_\Gamma(obj_c(\mathbf{T}_{P^\kappa_\Gamma} \uparrow \omega))$ is not in $min\text{-}\mathcal{GST}_{\langle P,\Gamma \rangle}$, then $\exists I(E') \in min\text{-}\mathcal{GST}_{\langle P,\Gamma \rangle}$ such that $E' \subset E$. However, by the above discussion, $I(E') \in min_\Gamma(obj_c(\mathbf{T}_{P^\kappa_\Gamma} \uparrow \omega))$, a contradiction. Therefore, the result follows.

(b) By Lemma 3.5 (c), for every minimal explanation E of O, there is a generalized stable model $I(E)$ of $\langle P, \Gamma \rangle$ in $obj_c(\mathbf{T}_{P^\kappa_\Gamma} \uparrow \omega)$ such that $I(E)$ satisfies O. Then, by Lemma 2.1, $I(E) \in min\text{-}\mathcal{GST}_{\langle P \cup \{\leftarrow not\, O\}, \Gamma \rangle}$. By (a), $min\text{-}\mathcal{GST}_{\langle P \cup \{\leftarrow not\, O\}, \Gamma \rangle}$ is given by $min_\Gamma(obj_c(\mathbf{T}_{(P \cup \{\leftarrow not\, O\})^\kappa_\Gamma} \uparrow \omega))$. Hence, the result follows. □

Example 3.3 (cont. from Example 2.2) The abductive general logic program $\langle P, \Gamma \rangle$, where $P = \{\ p \leftarrow r \wedge b \wedge not\, q, \quad q \leftarrow a, \quad r \leftarrow\ , \quad \leftarrow not\, p\ \}$ and $\Gamma = \{a, b\}$, is transformed to P^κ_Γ that contains:

$$(p \wedge b \wedge \neg\mathsf{K}q) \vee \neg\mathsf{K}b \vee \mathsf{K}q \leftarrow r\,, \quad (q \wedge a) \vee \neg\mathsf{K}a \leftarrow\ , \quad r \leftarrow\ , \quad \mathsf{K}p \leftarrow\ ,$$

and $\leftarrow \neg \mathrm{K}H \wedge H$ for every $H \in \mathcal{HB}$. Then, $\{r, p, b, \neg \mathrm{K}q, \neg \mathrm{K}a, \mathrm{K}p\}$ is the unique canonical set in $\mathrm{T}_{P^{\kappa}_{\Gamma}} \uparrow \omega$, and hence $\mathit{min}\text{-}\mathcal{GST}_{\langle P,\Gamma \rangle} = \{\{r, p, b\}\}$.

4 Abductive Extended Disjunctive Programs

Gelfond [Gel90] and Inoue [Ino91] proposed more general frameworks for abduction than that in [KM90] by allowing *classical negation* and disjunctions in a program. Now, we consider a fixpoint theory for such extended classes of abductive programs.

An *extended disjunctive program* is a disjunctive program which contains classical negation ($\neg$) as well as negation as failure (*not*) in the program [GL91], and is defined as a finite set of clauses of the form:

$$L_1 \vee \ldots \vee L_l \leftarrow L_{l+1} \wedge \ldots \wedge L_m \wedge \mathit{not}\, L_{m+1} \wedge \ldots \wedge \mathit{not}\, L_n \quad (n \geq m \geq l \geq 0) \qquad (10)$$

where each L_i is a positive or negative *literal*.

The semantics of extended disjunctive programs is given by the notion of *answer sets*. We denote the set of all ground literals from a program as $\mathcal{L} = \mathcal{HB} \cup \{\neg B \mid B \in \mathcal{HB}\}$. Let P be an extended disjunctive program and $S \subseteq \mathcal{L}$. Then, the *reduct* P^S of P by S is defined as follows: A clause $L_1 \vee \ldots \vee L_l \leftarrow L_{l+1} \wedge \ldots \wedge L_m$ is in P^S if there is a ground clause of the form (10) from P such that $\{L_{m+1}, \ldots, L_n\} \cap S = \emptyset$. Then, S is a *consistent answer set* of P, if S is a minimal set satisfying the conditions:

1. For each clause $L_1 \vee \ldots \vee L_l \leftarrow L_{l+1} \wedge \ldots \wedge L_m$ $(l \geq 1)$ in P^S, if $\{L_{l+1}, \ldots, L_m\} \subseteq S$, then $L_i \in S$ for some $1 \leq i \leq l$.

2. For each integrity constraint $\leftarrow L_1 \wedge \ldots \wedge L_m$ in P^S, $\{L_1, \ldots, L_m\} \not\subseteq S$.

3. S does not contain both B and $\neg B$ for any atom B.

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 general logic programs, the results presented in Sections 3.1 and 3.2 can be naturally extended. The only extra requirement we consider is the constraint that an atom B and its negation $\neg B$ cannot be in a consistent answer set. Now, for an extended disjunctive program P, the transformed program P^{κ} is defined as follows [IKH92]: For each clause of the form (10) from P, P^{κ} contains

$$\begin{aligned}(L_1 \wedge \neg \mathrm{K}L_{m+1} \wedge \ldots \wedge \neg \mathrm{K}L_n) \vee \ldots \vee (L_l \wedge \neg \mathrm{K}L_{m+1} \wedge \ldots \wedge \neg \mathrm{K}L_n)& \\ \vee \mathrm{K}L_{m+1} \vee \ldots \vee \mathrm{K}L_n \;\leftarrow\; L_{l+1} \wedge \ldots \wedge L_m\,,& \qquad (11)\end{aligned}$$

for each literal L in $\mathcal{L}$, a clause $\leftarrow \neg \mathrm{K}L \wedge L$ is in P^{κ}, and for each atom B in $\mathcal{HB}$, a clause $\leftarrow \neg B \wedge B$ is in P^{κ}.

Theorem 4.1 [IS92] Let P be an extended disjunctive program, and $\mathcal{AS}_P$ the collection of all consistent answer sets of P. Then, $\mathcal{AS}_P = \mathit{obj}_c(\mathit{min}(\mathrm{T}_{P^{\kappa}} \uparrow \omega))$.

Now, we define abduction within extended disjunctive programs.

Definition 4.1 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 positive/negative predicate symbols from P. The *abducibles* $\mathcal{A}^{\pm}_{\Gamma}$ ($\subseteq \mathcal{L}$) is the set of all ground literals with the predicates from Γ.

Let E be a subset of $\mathcal{A}^{\pm}_{\Gamma}$. A set of literals $S(E)$ is a *belief set* of $\langle P, \Gamma \rangle$ if it is a consistent answer set of the extended disjunctive program $P \cup E$ and satisfies $E = S(E) \cap \mathcal{A}^{\pm}_{\Gamma}$. A *minimal* belief set and a (*minimal*) *explanation* are defined in the same way as in Definitions 2.5 and 2.6.

The transformation for an abductive extended disjunctive program P is defined in the same way as Definition 3.7: For each clause in P of the form:

$$H_1 \vee \ldots \vee H_l \leftarrow B_1 \wedge \ldots \wedge B_m \wedge A_1 \wedge \ldots \wedge A_n \wedge not\ B_{m+1} \wedge \ldots \wedge not\ B_s \wedge not\ A_{n+1} \wedge \ldots \wedge not\ A_t$$

where $l \geq 0$, $s \geq m \geq 0$, $t \geq n \geq 0$, H_i's are literals, B_j's are non-abducible literals, and A_k's are abducible literals, P^{κ}_{Γ} contains the clause:

$$\begin{aligned} &(H_1 \wedge PRE) \vee \ldots \vee (H_l \wedge PRE) \vee \neg \mathsf{K} A_1 \vee \ldots \vee \neg \mathsf{K} A_n \\ &\qquad\qquad \vee \mathsf{K} B_{m+1} \vee \ldots \vee \mathsf{K} B_s \vee A_{n+1} \vee \ldots \vee A_t \leftarrow B_1 \wedge \ldots \wedge B_m \, , \qquad (12) \\ &\text{where } PRE = A_1 \wedge \ldots \wedge A_n \wedge \neg \mathsf{K} B_{m+1} \wedge \ldots \wedge \neg \mathsf{K} B_s \wedge \neg \mathsf{K} A_{n+1} \wedge \ldots \wedge \neg \mathsf{K} A_t \, , \end{aligned}$$

for each literal L in $\mathcal{L}$, a clause $\leftarrow \neg \mathsf{K} L \wedge L$ is in P^{κ}_{Γ}, and for each atom H in $\mathcal{HB}$, a clause $\leftarrow \neg H \wedge H$ is in P^{κ}_{Γ}.

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

Theorem 4.2 Let $\langle P, \Gamma \rangle$ be an abductive extended disjunctive program.
(a) Let $min\text{-}\mathcal{BS}_{\langle P, \Gamma \rangle}$ be the set of all minimal belief sets of $\langle P, \Gamma \rangle$. Then,
$min\text{-}\mathcal{BS}_{\langle P, \Gamma \rangle} = min_{\Gamma}(obj_c(min(\mathrm{T}_{P^{\kappa}_{\Gamma}} \uparrow \omega)))$.
(b) Let E be a subset of $\mathcal{A}^{\pm}_{\Gamma}$, and O a literal. Then, E is a minimal explanation of O with respect to $\langle P, \Gamma \rangle$ iff $S(E) \in min_{\Gamma}(obj_c(min(\mathbf{T}_{(P \cup \{\leftarrow not\, O\})^{\kappa}_{\Gamma}} \uparrow \omega)))$.

Proof: The proof can be given in a similar way to the proof of Theorem 3.6 except that, according to the existence of disjunctions in P, each I^{κ} is taken from $min(\mathrm{T}_{P^{\kappa}_{\Gamma}} \uparrow \omega)$ (as in Theorem 3.1 (c) and Theorem 4.1) instead of $\mathrm{T}_{P^{\kappa}_{\Gamma}} \uparrow \omega$ when proving the result corresponding to Lemma 3.5 (a). □

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) [FH91, IKH92]. MGTP is a parallel and refined version of SATCHMO

[MB88], which is a bottom-up forward-reasoning system that uses *hyperresolution* and *case-splitting* on non-unit hyperresolvents.

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

$$(H_{1,1} \wedge \ldots \wedge H_{1,k_1}) \vee \ldots \vee (H_{l,1} \wedge \ldots \wedge H_{l,k_l}) \leftarrow B_1 \wedge \ldots \wedge B_m \qquad (13)$$

where B_i's ($1 \leq i \leq m$; $m \geq 0$) and H_{j,k_j}'s ($k_j \geq 1$; $1 \leq j \leq l$) 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 non-negative clause of the form (13) in P and a substitution σ such that $I \models (B_1 \wedge \ldots \wedge B_m)\sigma$ and $I \not\models (H_{i,1} \wedge \ldots \wedge H_{i,k_i})\sigma$ for any $i = 1, \ldots, l$, then I is expanded in l ways by adding $H_{i,1}\sigma, \ldots, H_{i,k_i}\sigma$ to I for each $i = 1, \ldots, l$.

2. **(Interpretation Rejection)** If there is a negative clause $\leftarrow B_1, \ldots, B_m$ in P and a substitution σ such that $I \models (B_1 \wedge \ldots \wedge B_m)\sigma$, then I is discarded.

Here, the process of obtaining a substitution σ does not need full unification if every clause is *range-restricted* [MB88], that is, if every variable in the clause has at least one occurrence in its body. Since every set I of atoms constructed by MGTP in such a case contains only ground atoms, it is sufficient to consider matching instead of full unification. Thus, we assume that a program P input to MGTP is a finite, function-free program and satisfies the range-restrictedness for each clause in P. MGTP has been implemented in KL1 [UC90] on the PIM/m distributed-memory multiprocessor machine developed at ICOT, and split interpretations caused by the interpretation extension operation for a rule with $l \geq 2$ are automatically allocated to distributed memories so that they are taken as the source for exploiting OR-parallelism of MGTP.

The connection between closure computation by 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 [vEK76, Section 8]. In fact, for each split interpretation constructed by MGTP, hyperresolution is applied in the same way as in the case of definite programs.

We have prepared rules to reject inconsistent interpretations that contain both $\neg\mathsf{K}H$ and H (or both $\neg H$ and H) for any atom H. To implement these rules, we can simply use an axiom *schema* like: $\leftarrow \neg\mathsf{K}X \wedge X$ (or $\leftarrow \neg X \wedge X$) in which X can be matched with any atom by regarding $\neg\mathsf{K}$ and $\neg$ as functors in KL1. Furthermore, we can add additional schemas like $\leftarrow \neg\mathsf{K}X \wedge \mathsf{K}X$ and $\leftarrow \mathsf{K}\neg X \wedge \mathsf{K}X$ to improve the efficiency (see [IKH92]).

For abductive Horn, general 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 [IOHN92] 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.

6 Comparison with Other Approaches

This section compares the proposed theory to related work. Our fixpoint theory can give *a new uniform framework for a lot of previous work on characterizing stable models or answer sets* of general or extended (disjunctive) programs as well as proposals on computing explanations from these programs together with abducibles. Since there have been no algorithm which can compute the belief sets of arbitrary forms 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 General Programs

Here, we summarize the differences between other approaches and our fixpoint construction for positive disjunctive programs and general logic programs. For a more detailed comparison, see [IS92].

A fixpoint semantics for positive disjunctive programs has been studied by several researchers [MR90, FM91, Dec92]. Minker and Rajasekar [MR90] consider a mapping over the set of positive disjunctions (called *states*), while our fixpoint construction is based on the manipulation of standard Herbrand interpretations and directly computes models. Fernandez and Minker [FM91] present a fixpoint semantics for stratified disjunctive programs using a fixpoint operator over the sets of *minimal* interpretations. Decker [Dec92] also develops a fixpoint semantics for positive disjunctive programs based on another mapping over the sets of interpretations.

For general and extended disjunctive programs, the stable model semantics [GL88] and the answer set semantics [GL91] were originally defined by means of reducts of programs so that their fixpoints are not constructively given. On the other hands, our fixpoint is constructively defined, and for the propositional case its computational complexity is the same as that of computing the minimal models of a positive disjunctive program. In contrast to another constructive approach by [SZ90], our fixpoint construction is performed in parallel based on case-splitting on derived disjunctions and does not need any selection strategies nor future backtracking during the computation of stable models.

In [FLMS91], Fernandez et al develop a method of computing stable models by using a similar but different program transformation from ours. In our transformation (4), each head H is associated with its prerequisite condition $\neg \mathsf{K} B_{m+1} \wedge \ldots \wedge \neg \mathsf{K} B_n$ in an explicit way, while this is not the case in their transformation. Furthermore, we effectively use negative clauses to prune away improper extensions, while their transformation does not use any such negative clauses. Although we could extend [FLMS91]'s transformation to deal with abductive general logic programs, our translation appears to be more suitable for handling abducibles. Since the prerequisite condition in this case contains abduced atoms, 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 Computing Abduction

Console et al [CDT91] characterize abduction by deduction (called the *object-level abduction*) through Clark's completion semantics of a program [Cla78]. 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 non-abducible predicates in P. For an observed atom O, 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 $S \subseteq \mathcal{A}^{\pm}_{\Gamma}$ such that $S \models E$ is called an *explanation* of O.

In [CDT91], they show that the object-level abduction coincides with the meta-level characterization of abduction in terms of SLDNF proof procedure for *hierarchical logic programs.* [3] 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 [Kon92]).

Example 6.1 Consider a program containing cyclic clauses: $P = \{\, p \leftarrow q,\ \ q \leftarrow p,\ \ q \leftarrow a \,\}$ where a is an abducible atom. Then, $comp^{-\Gamma}(P) = \{\, p \leftrightarrow q,\ \ q \leftrightarrow p \vee a \,\}$, and for an observation $O = p$, $P \cup \{a\} \models p$, while $comp^{-\Gamma}(P) \cup \{O\} \not\models a$.

On the other hand, $P^{\kappa}_{\Gamma} = \{\, p \leftarrow q,\ \ q \leftarrow p,\ \ (q \wedge a) \vee \neg \mathsf{K}a \leftarrow,\ \ \leftarrow \neg \mathsf{K}a \wedge a \,\}$ is obtained by our transformation in Section 3.3, and $\{q, a, p\}$ is in $\mathrm{T}_{P^{\kappa}_{\Gamma}} \uparrow \omega$.

Denecker and De Schreye [DDS92] recently proposed a model generation procedure for Console et al's object-level abduction. In contrast to us, they compute 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 have to 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. Bry [Bry90] firstly considered abduction by model generation, but his abduction is defined in terms of a meta-theory.

Eshghi and Kowalski [EK89] give an abductive interpretation of negation as failure in general logic programs. For each negation-as-failure formula $not\, B(\mathrm{x})$, the formula $B^{*}(\mathrm{x})$ 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 abducible 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 following integrity constraints:

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

[3] General logic programs which contain no predicates defined via positive/negative cycles.

However, difficulty arises in dealing with the disjunctive constraints that cannot be checked without actually computing models in general. Thus, we cannot have any elegant top-down proof procedure which is sound with respect to the stable model semantics. For Example 3.2, the top-down abductive procedure of [EK89] gives a proof for $O = p$, but no stable model satisfies p. [4] For an abductive general logic program $\langle P, \Gamma \rangle$, Kakas and Mancarella [KM91] also transform the negation-as-failure formulas in P, and 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 [EK89]. However, this transformation inherits the difficulty of computation from Eshghi and Kowalski's abductive interpretation of negation as failure, and suffers from the soundness problem with respect to the generalized stable model semantics. Gelfond [Gel90] defines abductive extended disjunctive programs, but no proof procedure is given.

Alternatively, [Ino91] and [SI91] show that an abductive general logic program $\langle P, \Gamma \rangle$ can be transformed into a *single* general logic program. For each atom $A(\mathrm{x})$ from Γ, they introduce the negative literal $\neg A(\mathrm{x})$ and a pair of new clauses: [5]

$$\begin{array}{rcl} A(\mathrm{x}) & \leftarrow & not\, \neg A(\mathrm{x})\,, \\ \neg A(\mathrm{x}) & \leftarrow & not\, A(\mathrm{x})\,. \end{array} \tag{14}$$

Then, there is a 1-1 correspondence between the generalized stable models of $\langle P, \Gamma \rangle$ and the stable models of the transformed program. Using this transformation, again, we cannot have a top-down proof procedure which is sound with respect to the generalized stable model semantics because we remain in the stable model semantics. Therefore, Satoh and Iwayama [SI91] propose a bottom-up, TMS-style procedure for computing stable models of a general logic program, which is similar to [SZ90]'s procedure and performs an exhaustive search with backtracking. At this point, we can use any procedure to compute stable models. Comparing each procedure, the MGTP-based procedure by [IKH92] has the following advantages over procedures of [SZ90, SI91]. First, MGTP can deal with disjunctive programs, while TMS-like algorithms cannot. Second, MGTP gives high inference rates for range-restricted clauses by avoiding computation relative to their useless ground instances, while a TMS generally deals only with the propositional case. Third, MGTP performs a backtrack-free search and more easily parallelized than others.

Although the simulation (14) of abducibles is theoretically correct, this technique has the drawback that it may generate $2^{|\mathcal{A}_\Gamma|}$ 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

[4] However, Eshghi and Kowalski's abductive proof procedure is sound with respect to the *preferred extension* semantics by Dung [Dun91].

[5] Satoh and Iwayama do not use the negative literal $\neg A$ but use the new predicate symbol $\tilde{A}$ for each predicate symbol A appearing in Γ because they do not allow classical negation in a program.

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.

7 Conclusion

We have presented a uniform framework for fixpoint characterization of abductive Horn, general, and extended (disjunctive) programs. Based on a fixpoint operator over a complete lattice consisting of the sets of Herbrand interpretations, the generalized stable model semantics of an abductive general logic program can be characterized by the fixpoint of a suitably transformed 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 is different from any other characterization of abductive programs, and provides a constructive way to give explanations for observations. We also showed that a bottom-up model generation procedure can be used for computing generalized stable models or belief sets and has a computational advantage from the viewpoint of parallelism.

References

[Bry90] F. Bry. Intensional updates: abduction via deduction. In: *Proc. 7th Int. Conf. Logic Programming*, pages 561–575, 1990.

[Cla78] K.L. Clark. Negation as failure. In: H. Gallaire and J. Minker, editors, *Logic and Data Bases*, pages 293–322, Plenum, 1978.

[CDT91] L. Console, D.T. Dupre and P. Torasso. On the relationship between abduction and deduction. *J. Logic and Computation*, 1:661–690, 1991.

[Dec92] H. Decker. Foundations of first-order databases. Research Report, Siemens, Munich, 1992.

[DDS92] M. Denecker and D. De Schreye. On the duality of abduction and model generation. In: *Proc. Int. Conf. Fifth Generation Computer Systems 1992*, pages 650–657, 1992.

[Dun91] P.M. Dung. Negations as hypotheses: an abductive foundation for logic programming. In: *Proc. 8th Int. Conf. Logic Programming*, pages 3–17, 1991.

[EK89] K. Eshghi and R.A. Kowalski. Abduction compared with negation by failure. In: *Proc. 6th Int. Conf. Logic Programming*, pages 234–254, 1989.

[FM91] J.A. Fernandez and J. Minker. Computing perfect models of disjunctive stratified databases. In: *Proc. ILPS'91 Workshop on Disjunctive Logic Programs*, 1991.

[FLMS91] J.A. Fernandez, J. Lobo, J. Minker, and V.S. Subrahmanian. Disjunctive LP + integrity constraints = stable model semantics. In: *Proc. ILPS'91 Workshop on Deductive Databases*, 1991.

[FH91] H. Fujita and R. Hasegawa. A model generation theorem prover in KL1 using a ramified-stack algorithm. In: *Proc. 8th Int. Conf. Logic Programming*, pages 494–500, 1991.

[Gel90] M. Gelfond. Epistemic approach to formalization of commonsense reasoning. Research Report, Computer Science Department, University of Texas at El Paso, El Paso, 1990.

[GL88] M. Gelfond and V. Lifschitz. The stable model semantics for logic programming. In: *Proc. 5th Int. Conf. Symp. Logic Programming*, pages 1070-1080, 1988.

[GL91] M. Gelfond and V. Lifschitz. Classical negation in logic programs and disjunctive databases. *New Generation Computing*, 9:365–385, 1991.

[Ino91] K. Inoue. Extended logic programs with default assumptions. In: *Proc. 8th Int. Conf. Logic Programming*, pages 490–504, 1991.

[Ino92] K. Inoue. Linear resolution for consequence finding. *Artificial Intelligence*, 56:301–353, 1992.

[IKH92] K. Inoue, M. Koshimura and R. Hasegawa. Embedding negation as failure into a model generation theorem prover. In: *Proc. 11th Int. Conf. Automated Deduction*, Lecture Notes in Artificial Intelligence, 607, pages 400–415, Springer-Verlag, 1992.

[IOHN92] K. Inoue, Y. Ohta, R. Hasegawa and M. Nakashima. Bottom-up abduction by model generation. Technical Report TR-816, ICOT, Tokyo, October 1992.

[IS92] K. Inoue and C. Sakama. A uniform approach to fixpoint characterization of disjunctive and general logic programs. Technical Report TR-817, ICOT, Tokyo, October 1992.

[KM90] A.C. Kakas and P. Mancarella. Generalized stable models: a semantics for abduction. In: *Proc. 9th European Conf. Artificial Intelligence*, pages 385–391, 1990.

[KM91] A.C. Kakas and P. Mancarella. Knowledge assimilation and abduction. In: *Proc. ECAI-90 Workshop on Truth Maintenance Systems*, Lecture Notes in Artificial Intelligence, 515, pages 54–70, Springer-Verlag, 1991.

[KKT92] A.C. Kakas, R.A. Kowalski and F. Toni. Abductive logic programming. A survey paper, Imperial College of Science, Technology and Medicine, London, 1992.

[Kon92] K. Konolige. Abduction versus closure in causal theories. *Artificial Intelligence*, 53:255–272, 1992.

[LT84] J.W. Lloyd and R.W. Topor. Making Prolog more expressive. *J. Logic Programming*, 3:225–240, 1984.

[MB88] R. Manthey and F. Bry. SATCHMO: a theorem prover implemented in Prolog. In: *Proc. 9th Int. Conf. Automated Deduction*, Lecture Notes in Computer Science, 310, pages 415–434, Springer-Verlag, 1988.

[Min82] J. Minker. On indefinite data bases and the closed world assumption. In: *Proc. 6th Int. Conf. Automated Deduction*, Lecture Notes in Computer Science, 138, pages 292–308, Springer-Verlag, 1982.

[MR90] J. Minker and A. Rajasekar. A fixpoint semantics for disjunctive logic programs. *J. Logic Programming*, 9:45–74, 1990.

[PGA87] D. Poole, R. Goebel and R. Aleliunas. Theorist: a logical reasoning system for defaults and diagnosis. In: N. Cercone and G. McCalla, editors, *The Knowledge Frontier: Essays in the Representation of Knowledge*, pages 331–352, Springer-Verlag, 1987.

[PE92] C. Preist and K. Eshghi. Consistency-based and abductive diagnoses as generalized stable models. In: *Proc. Int. Conf. Fifth Generation Computer Systems 1992*, pages 514–521, 1992.

[SZ90] D. Sacca and C. Zaniolo. Stable models and non-determinism in logic programs with negation. In: *Proc. 9th ACM SIGACT-SIGMOD-SIGART Symp. Principles of Database Systems*, pages 205–229, 1990.

[SK87] F. Sadri and R. Kowalski. A theorem proving approach to database integrity. In: J. Minker, editor, *Foundations of Deductive Databases and Logic Programming*, pages 313–362, Morgan Kaufmann, 1987.

[SI91] K. Satoh and N. Iwayama. Computing abduction by using the TMS. In: *Proc. 8th Int. Conf. Logic Programming*, pages 505–518, 1991.

[UC90] K. Ueda and T. Chikayama. Design of the kernel language for the parallel inference machine. *Computer J.*, 33:494–500, 1990.

[vEK76] M.H. van Emden and R.A. Kowalski. The semantics of predicate logic as a programming language. *J. ACM*, 23:733–742, 1976.