On the Equivalence between Disjunctive and Abductive Logic Programs*

Chiaki Sakama
ASTEM Research Institute of Kyoto
17 Chudoji Minami-machi
Shimogyo, Kyoto 600 Japan
sakama@astem.or.jp

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

Abstract

This paper presents the equivalence relationship between disjunctive and abductive logic programs. We show that the generalized stable model semantics of abductive logic programs can be translated into the possible model semantics of disjunctive programs, and vice versa. It is also proved that abductive disjunctive programs can be expressed by abductive logic programs under the possible model semantics. Furthermore, when considering the disjunctive stable model semantics instead of the possible model semantics, it is unlikely that disjunctive programs can be efficiently expressed in terms of the generalized stable model semantics. The results of this paper reveal that disjunctive programs and abductive logic programs are just different ways of looking at the same problem if we choose the appropriate semantics.

1 Introduction

Disjunctive programs and abductive logic programs are two extensions of logic programming which draw much attention recently. Disjunctive programs provide us with the framework of reasoning with indefinite information. The framework was firstly studied by Minker [Min82] in which he introduced the *minimal model semantics* of positive disjunctive programs. In the last decade, various extensions of this framework have been studied

*In *Proceedings of the 11th International Conference on Logic Programming* (ICLP'94), MIT Press, pp. 489-503, 1993.

for disjunctive programs with negation. On the other hand, abductive logic programs supply the ability to perform reasoning with hypotheses. Such a framework was firstly investigated by Eshghi and Kowalski [EK89] and then generalized by Kakas and Mancarella [KM90] who extended Gelfond and Lifschitz's stable model semantics [GL88] to the *generalized stable model semantics*. Further extensions have been studied by several researchers in the last few years [KKT92].

Disjunctive programs and abductive logic programs have been independently developed and have different syntax and semantics from each other. However, in disjunctive programs, each disjunction is considered to represent knowledge about possible alternative beliefs, and such beliefs can also be regarded as a kind of hypotheses. In abductive logic programs, on the other hand, each candidate hypothesis is examined whether it is adopted or not, and this situation can be considered as meta-level disjunctive knowledge that either a hypothesis is true or not. Thus, each formalism appears to deal with very similar problems from different viewpoints. Then the question naturally arises whether there is any formal correspondence between these two frameworks.

There are some works which can be related to the above question. Dung [Dun92] presents a program transformation from acyclic disjunctive programs to normal logic programs under the stable model semantics and uses Eshghi and Kowalski's abductive proof procedure for such programs. However, Dung's transformation is restricted to acyclic disjunctive programs and not applicable in general. Inoue and Sakama [IS93] present a program transformation from abductive logic programs to disjunctive programs under the stable model semantics and use a bottom-up model generation proof procedure for computing abduction. While their transformation is fairly general, it is a one-way transformation from abductive logic programs to disjunctive programs.

This paper investigates a general correspondence between disjunctive programs and abductive logic programs. For the part from abductive logic programs to disjunctive programs, we show that the generalized stable models of an abductive logic program are characterized by the *possible models* [Sak89, SI93] of the transformed disjunctive program. Conversely, from disjunctive programs to abductive logic programs, we show that the possible models of a disjunctive program are exactly the generalized stable models of the transformed abductive logic program. Moreover, if the *disjunctive stable model semantics* [Prz91] is taken as the underlying semantics instead of the possible model semantics, it is unlikely that disjunctive programs can be efficiently expressed in terms of the generalized stable model semantics. It is also shown that abductive disjunctive programs can be expressed by abductive logic programs under the possible model semantics.

The rest of this paper is organized as follows. In Section 2, we introduce notations used in this paper. In Section 3, we present a program transformation from abductive logic programs to disjunctive programs and show that the generalized stable models of an abductive logic program are characterized by the possible models of the transformed disjunctive program. In Section 4, we present that a converse transformation is also possible and show that the possible models of a disjunctive program are the generalized stable models of the transformed abductive logic program. In Section 5, abductive disjunctive programs are also translated into disjunctive programs and it is shown that abductive logic programs are as expressive as abductive disjunctive programs under the possible model semantics. Section 6 discusses the relation between disjunctive programs and abductive logic programs from the complexity point of view. Section 7 concludes this paper.

2 Definitions

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

$$A_1 \vee \ldots \vee A_l \leftarrow B_1 \wedge \ldots \wedge B_m \wedge not\, B_{m+1} \wedge \ldots \wedge not\, B_n \quad (l \geq 0,\ n \geq m \geq 0) \quad (1)$$

where A_i's and B_j's are atoms and *not* denotes *negation as failure*. The left-hand side of the clause is called the *head*, while the right-hand side of the clause is called the *body*. In this paper, we often use the Greek letter Γ to denote the conjunction in the body of a clause. A clause is called *disjunctive* (resp. *normal*) if its head contains more than one atom (resp. exactly one atom). A clause with the empty head is called an *integrity constraint*. A normal disjunctive program containing no *not* is called a *positive disjunctive program*, while a program containing no disjunctive clause is called a *normal logic program*. As usual, we semantically identify a program with its *ground program*, which is the possibly infinite set of all ground clauses from the program.

An *interpretation* of a program P is a subset of the Herbrand base $\mathcal{HB}_P$ of the program. An interpretation I is called a *disjunctive stable model* [Prz91] of P if I coincides with a minimal model of the positive disjunctive program P^I defined as

$$P^I = \{ A_1 \vee \ldots \vee A_l \leftarrow B_1 \wedge \ldots \wedge B_m \mid \text{there is a ground clause of the form (1) from } P \text{ such that } \{B_{m+1}, \ldots, B_n\} \cap I = \emptyset \}.$$

In particular, when P is a normal logic program, I is called a *stable model* of P [GL88].

The *possible model semantics* proposed in [Sak89, SI93][1] is an alternative semantics for disjunctive programs. It is introduced to enable one to specify both inclusive and exclusive disjunctions in a program.

Given a normal disjunctive program P, a *split program* is defined as a ground normal logic program obtained from P by replacing each ground

[1] In [SI93], it is called the *possible world semantics*.

disjunctive clause of the form (1) with the following ground normal clauses (called *split clauses*):

$$A_i \leftarrow B_1 \wedge \ldots \wedge B_m \wedge not B_{m+1} \wedge \ldots \wedge not B_n \;\text{ for every } A_i \in S$$

where S is some non-empty subset of $\{A_1, \ldots, A_l\}$. Then a *possible model* of P is defined as a stable model of any split program of P. Clearly, possible models reduce to stable models in normal logic programs. In positive disjunctive programs, the notion of possible models also coincides with the *possible worlds* presented in [Cha93].

Example 2.1 Let P be the program:

$$\{\ a \vee b \leftarrow not\, c, \quad d \leftarrow a \wedge b\ \}.$$

Then the split programs of P are

$$\{\ a \leftarrow not\, c, \quad d \leftarrow a \wedge b\ \},$$

$$\{\ b \leftarrow not\, c, \quad d \leftarrow a \wedge b\ \},$$

$$\{\ a \leftarrow not\, c, \quad b \leftarrow not\, c, \quad d \leftarrow a \wedge b\ \},$$

and $\{a\}$, $\{b\}$, $\{a, b, d\}$ are the possible models of P. □

Note that $\{a\}$ and $\{b\}$ are also disjunctive stable models, while $\{a, b, d\}$ is not. Thus the disjunctive stable models exclude the possibility of an inclusive interpretation of the disjunction in the above example, while the possible models consider both exclusive and inclusive interpretations. Notice that if one does not want the inclusive interpretation $\{a, b, d\}$ under the possible model semantics, it is enough to insert the integrity constraint $\leftarrow a \wedge b$ in P. Intuitively speaking, each possible model presents an interpretation in which each atom has its possible justification in a program, and both exclusive and inclusive interpretations of disjunctions are considered whenever there is no integrity constraint to inhibit inclusive ones. In [Sak89, SI93], it is shown that the possible model semantics can provide a flexible mechanism for inferring negation in disjunctive programs.

An *abductive logic program* is a pair $\langle P, \mathcal{A} \rangle$ where P is a normal logic program and $\mathcal{A}$ is a finite set of atoms called the *abducibles*.[2] Let $\langle P, \mathcal{A} \rangle$ be an abductive logic program and E be a subset of $\mathcal{A}$. An interpretation I is a *generalized stable model* of $\langle P, \mathcal{A} \rangle$ if I is a stable model of the normal logic program $P \cup E$. A generalized stable model I is called $\mathcal{A}$-*minimal* if there is no generalized stable model J such that $J \cap \mathcal{A} \subset I \cap \mathcal{A}$. Clearly, ($\mathcal{A}$-minimal) generalized stable models coincide with stable models if $\mathcal{A} = \emptyset$.

[2] We slightly modified the original definition of [KM90] by including integrity constraints in a program and considering abducible atoms instead of abducible predicates. Here, an abducible containing variables is identified with its ground instances.

Let $\langle P, \mathcal{A} \rangle$ be an abductive logic program and O be an atom which represents *observation*.[3] Then a set $E \subseteq \mathcal{A}$ is an *explanation* of O if there is a generalized stable model I of $\langle P, \mathcal{A} \rangle$ such that I satisfies O and $E = I \cap \mathcal{A}$. An explanation E of O is *minimal* if no $E' \subset E$ is an explanation of O.

Note that the problem of finding explanations is essentially equivalent to the problem of finding generalized stable models since E is a (minimal) explanation of O with respect to $\langle P, \mathcal{A} \rangle$ iff I is a ($\mathcal{A}$-minimal) generalized stable model of $\langle P \cup \{ \leftarrow not\, O\},\ \mathcal{A} \rangle$ such that $I \cap \mathcal{A} = E$ [IS93].

Example 2.2 Let $\langle P, \mathcal{A} \rangle$ be an abductive logic program such that

$$P = \{\ p(x) \leftarrow q(x) \wedge not\ r(x), \quad q(x) \leftarrow s(x), \ \ q(x) \leftarrow t(x)\ \}$$

and $\mathcal{A} = \{\ s(x),\ \ t(b)\ \}$. Then, for a given observation $O = p(a)$, the ($\mathcal{A}$-minimal) generalized stable model $I = \{\ p(a),\ q(a),\ s(a)\ \}$ of $\langle P, \mathcal{A} \rangle$ satisfies O and its (minimal) explanation is $E = I \cap \mathcal{A} = \{\ s(a)\ \}$. Here, I is also the unique generalized stable model of $\langle P \cup \{\leftarrow not\, p(a)\}, \mathcal{A} \rangle$. □

3 Generalized Stable Models are Possible Models

In this section, we present a program transformation from abductive logic programs to normal disjunctive programs and show that the generalized stable models of an abductive logic program can be expressed by the possible models of the transformed normal disjunctive program.

In an abductive logic program, each candidate hypothesis is either assumed or not. Such a situation is naturally expressed by disjunctions in a program.

Definition 3.1 Let $\langle P, \mathcal{A} \rangle$ be an abductive logic program. Then its *dlp-transformation* is defined by a normal disjunctive program $dlp(\langle P, \mathcal{A} \rangle)$ which is obtained from P by adding the following disjunctive clauses for each abducible $A \in \mathcal{A}$:

$$A \vee \varepsilon \leftarrow \tag{2}$$

where ε is an atom not appearing elsewhere in P. □

The intuitive meaning of the *dlp*-transformation is that when an abducible A is assumed in an abductive logic program $\langle P, \mathcal{A} \rangle$, the corresponding disjunct A is chosen from (2) in the transformed normal disjunctive program $dlp(\langle P, \mathcal{A} \rangle)$. Else when A is not assumed, the newly introduced atom ε is chosen from (2). Thus the *dlp*-transformation specifies meta-level knowledge representing whether each abducible is assumed or not.[4]

[3] As discussed in [IS93], without loss of generality an observation is assumed to be a non-abducible ground atom.

[4] A similar transformation is also presented in [IS94] in the context of extended disjunctive programs with *positive occurrences of negation as failure*.

Now we express the generalized stable model semantics in terms of $dlp(\langle P, \mathcal{A} \rangle)$. Let I be a possible model of a normal disjunctive program P. We say that I is $\mathcal{A}$-*minimal* if there is no possible model J of P such that $J \cap \mathcal{A} \subset I \cap \mathcal{A}$. In the following, an atom A is identified with the unit clause $A \leftarrow$ in E.

Theorem 3.1 Let $\langle P, \mathcal{A} \rangle$ be an abductive logic program. Then,

(i) $I \setminus \{\varepsilon\}$ is a generalized stable model of $\langle P, \mathcal{A} \rangle$ iff I is a possible model of $dlp(\langle P, \mathcal{A} \rangle)$.

(ii) $I \setminus \{\varepsilon\}$ is an $\mathcal{A}$-minimal generalized stable model of $\langle P, \mathcal{A} \rangle$ iff I is an $\mathcal{A}$-minimal possible model of $dlp(\langle P, \mathcal{A} \rangle)$.

Proof: (i) Let I' be a generalized stable model of $\langle P, \mathcal{A} \rangle$. Then I' is a stable model of $P \cup E$ for some E from $\mathcal{A}$. Now let us consider the transformed program $dlp(\langle P, \mathcal{A} \rangle)$. Then there is a split program P' of $dlp(\langle P, \mathcal{A} \rangle)$ such that for each disjunctive clause (2), $A \leftarrow$ is in P' if $A \in E$; $\varepsilon \leftarrow$ is in P', otherwise. When $\varepsilon \leftarrow$ is in P', $I' \cup \{\varepsilon\}$ is a stable model of P' and also a possible model of $dlp(\langle P, \mathcal{A} \rangle)$. Else when $\varepsilon \leftarrow$ is not in P', I' is a stable model of P' and also a possible model of $dlp(\langle P, \mathcal{A} \rangle)$. Hence the result of only-if part follows.

Conversely, when I is a possible model of $dlp(\langle P, \mathcal{A} \rangle)$, it is a stable model of some split program P' of $dlp(\langle P, \mathcal{A} \rangle)$. Let E be the set of all split clauses included in P'. Then I is a stable model of $P \cup E$. Since $E \setminus \{\varepsilon \leftarrow\}$ consists of instances from $\mathcal{A}$, $I \setminus \{\varepsilon\}$ is a generalized stable model of $\langle P, \mathcal{A} \rangle$.

(ii) The result directly follows from (i) and the definitions of $\mathcal{A}$-minimal generalized stable models/$\mathcal{A}$-minimal possible models. $\Box$

Corollary 3.2 Let $\langle P, \mathcal{A} \rangle$ be an abductive logic program. Then, for a given observation O, there is a (minimal) explanation E of O iff there is an ($\mathcal{A}$-minimal) possible model I of $dlp(\langle P, \mathcal{A} \rangle)$ satisfying O and $I \cap \mathcal{A} = E$. $\Box$

Example 3.1 Let $\langle P, \mathcal{A} \rangle$ be an abductive logic program such that

$$\begin{aligned} P \;=\; \{ & \textit{wet-shoes} \leftarrow \textit{wet-grass} \wedge \textit{not driving-car}, \\ & \textit{wet-grass} \leftarrow \textit{rained}, \\ & \textit{wet-grass} \leftarrow \textit{sprinkler-on} \}, \end{aligned}$$

and $\mathcal{A} = \{ \textit{rained}, \; \textit{sprinkler-on} \}$. Then,

$$dlp(\langle P, \mathcal{A} \rangle) = P \cup \{ \textit{rained} \vee \varepsilon \leftarrow, \quad \textit{sprinkler-on} \vee \varepsilon \leftarrow \}$$

which has the five possible models:

$$\{\textit{rained}, \; \textit{sprinkler-on}, \; \textit{wet-grass}, \; \textit{wet-shoes}\},$$

$$\{\varepsilon\},$$

$$\{\varepsilon,\ rained,\ wet\text{-}grass,\ wet\text{-}shoes\},$$

$$\{\varepsilon,\ sprinkler\text{-}on,\ wet\text{-}grass,\ wet\text{-}shoes\},$$

$$\{\varepsilon,\ rained,\ sprinkler\text{-}on,\ wet\text{-}grass,\ wet\text{-}shoes\}.$$

Thus, the generalized stable models of $\langle P, \mathcal{A} \rangle$ coincide with the sets which are obtained by removing ε from each possible model. In particular, $\{\varepsilon\}$ is the $\mathcal{A}$-minimal possible model and it corresponds to the $\mathcal{A}$-minimal generalized stable model $\emptyset$ of $\langle P, \mathcal{A} \rangle$. $\Box$

The result of this section indicates that abductive logic programs are also considered as disjunctive programs. In the next section, we present that the converse also holds.

4 Possible Models are Generalized Stable Models

As presented in introduction, indefinite information in disjunctive programs is viewed as possible hypotheses in a program. Then it is natural to represent disjuncts in terms of abducibles in an abductive logic program. However, the problem is that disjunctive clauses possibly have conditions in their bodies, while abductive logic programs introduced in Section 2 lack the ability of expressing assumptions with preconditions. Then our first task is to extend the framework of abductive logic programs to possibly include such hypothetical rules.

An abductive logic program considering in this section is a pair $\langle P, \mathcal{C} \rangle$ where P is a normal logic program and $\mathcal{C}$ is a finite set of normal clauses called the *abducible rules*. The abducible rule intuitively means that if the rule is abduced then it is used for inference together with the background knowledge from P. In this sense, abductive logic programs presented in the previous sections are considered as a special case where each abducible rule has the empty precondition. The generalized stable model semantics of such an extended framework is defined as follows.

Definition 4.1 Let $\langle P, \mathcal{C} \rangle$ be an abductive logic program and F be a subset of $\mathcal{C}$. An interpretation I is a *generalized stable model* of $\langle P, \mathcal{C} \rangle$ if it is a stable model of the normal logic program $P \cup F$. $\Box$

The generalized stable models introduced above are a direct extension of those presented in the previous sections, and they reduce to the usual notion when $\mathcal{C} = \mathcal{A}$.

Next we provide a program transformation which translates normal disjunctive programs into abductive logic programs. For a normal disjunctive program P, we define $P = disj(P) \cup \overline{disj}(P)$ where $disj(P)$ is the set of all disjunctive clauses from P and $\overline{disj}(P)$ is the set of all normal clauses and integrity constraints from P.

Definition 4.2 Given a normal disjunctive program P, let us consider the set of normal clauses

$$\mathcal{C} = \{\ A_i \leftarrow \Gamma \mid A_1 \vee \ldots \vee A_l \leftarrow \Gamma \in disj(P) \text{ and } 1 \le i \le l\ \} \tag{3}$$

and the integrity constraints

$$IC = \{\ \leftarrow \Gamma \wedge not A_1 \wedge \ldots \wedge not A_l \mid A_1 \vee \ldots \vee A_l \leftarrow \Gamma \in disj(P)\ \}. \tag{4}$$

Then we define the *alp-transformation* of P by $alp(P) = \langle\, \overline{disj}(P) \cup IC,\, \mathcal{C}\,\rangle$. □

The intuitive meaning of the *alp*-transformation is that each disjunctive clause in a program is replaced with a set of abducible rules (3) in $\mathcal{C}$. The integrity constraints (4) in IC impose the condition that at least one of disjuncts are chosen as an abducible whenever the body of a disjunctive clause is true. In this way, by the *alp*-transformation each disjunctive clause is rewritten by a set of abducible rules.

Now we present the relationship between the possible models of a normal disjunctive program P and the generalized stable models of the transformed abductive logic program $alp(P)$.

Theorem 4.1 Let P be a normal disjunctive program. Then I is a possible model of P iff I is a generalized stable model of $alp(P)$.

Proof: Let I be a possible model of P. Then there is a split program P' of P such that I is a stable model of P'. Suppose that each ground disjunctive clause $C^k:\ A_1 \vee \ldots \vee A_{l_k} \leftarrow \Gamma_k$ from P is replaced with the split clauses in $C^k_S = \{A_i \leftarrow \Gamma_k \mid A_i \in S\}$ in P' where S is a non-empty subset of $\{A_1, \ldots, A_{l_k}\}$. Then I is a stable model of $\overline{disj}(P) \cup \bigcup_k C^k_S$. Since $\bigcup_k C^k_S$ consists of instances from $\mathcal{C}$ and I satisfies integrity constraints IC, I is also a generalized stable model of $alp(P)$.

Conversely, let I be a generalized stable model of $alp(P)$. Then I is a stable model of $P \cup F$ where F is a subset of $\mathcal{C}$. For each normal clause $A_i \leftarrow \Gamma$ in F, there is a corresponding disjunctive clause $C:\ A_1 \vee \ldots \vee A_l \leftarrow \Gamma$ in $disj(P)$ such that $1 \le i \le l$. Also, since I satisfies integrity constraints IC, when I satisfies Γ, at least one normal clause $A_i \leftarrow \Gamma$ is included in F. In this case, there is a split program P' of P in which each ground instance of a disjunctive clause C is split into a corresponding ground instance of a normal clause $A_i \leftarrow \Gamma$. Thus I is also a stable model of P', hence a possible model of P. □

Example 4.1 [Cha93] Let

$$\begin{aligned} P \ = \ \{\ & violent \vee psychopath \leftarrow suspect, \\ & dangerous \leftarrow violent \wedge psychopath, \\ & suspect \leftarrow \ \}. \end{aligned}$$

Then, $alp(P) = \langle\, \overline{disj}(P) \cup IC,\, \mathcal{C}\, \rangle$ where

$$\begin{aligned}
\overline{disj}(P) \cup IC \;&=\; \{\; dangerous \leftarrow violent \wedge psychopath, \\
&\quad\;\; suspect \leftarrow, \\
&\quad\;\; \leftarrow suspect \wedge not\, violent \wedge not\, psychopath \;\}, \\
\mathcal{C} \;&=\; \{\; violent \leftarrow suspect, \quad psychopath \leftarrow suspect \;\}.
\end{aligned}$$

Thus, $alp(P)$ has three generalized stable models:

$$\{suspect,\ violent\},$$

$$\{suspect,\ psychopath\},$$

$$\{suspect,\ violent,\ psychopath,\ dangerous\},$$

which coincide with the possible models of P. $\square$

Note that in the above example there is no minimal model of P containing *dangerous*. By contrast, $alp(P)$ has a generalized stable model in which *dangerous* is true, which corresponds to a possible model in which the disjunction is inclusively true.

The abductive logic programming framework presented in this section is also introduced by Inoue [Ino91] in the context of the *knowledge system* for *extended logic programs* [GL90]. He also shows that an abductive logic program $\langle\, P, \mathcal{C}\, \rangle$ can be translated into a semantically equivalent usual abductive logic program $\langle\, P, \mathcal{A}\, \rangle$. Given an abductive logic program $\langle\, P, \mathcal{C}\, \rangle$, let us consider a program P' which is obtained from P by including the clause $A \leftarrow A' \wedge \Gamma$ for each abducible rule $A \leftarrow \Gamma$ in $\mathcal{C}$. Here A' is a newly introduced atom not appearing elsewhere in P and is uniquely associated with each A. Also let $\mathcal{A}'$ be a set of abducibles which consists of every newly introduced atom A'. Then he proves that there is a one-to-one correspondence between the generalized stable models of $\langle\, P, \mathcal{C}\, \rangle$ and the generalized stable models of $\langle\, P', \mathcal{A}'\, \rangle$. This fact implies that the possible models of a normal disjunctive program are also expressed by the generalized stable models of a usual abductive logic program.

5 Generalized Possible Models are Generalized Stable Models

This section presents a connection between abductive disjunctive programs and abductive logic programs.

Abductive disjunctive programs are normal disjunctive programs with abducibles. The definition of an *abductive disjunctive program* $\langle\, P, \mathcal{A}\, \rangle$ is the same as an abductive logic program except that P is a normal disjunctive program. For a given set $E \subseteq \mathcal{A}$, an interpretation I is a *generalized disjunctive stable model* of $\langle\, P, \mathcal{A}\, \rangle$ if it is a disjunctive stable model of the normal disjunctive program $P \cup E$. On the other hand, I is a *generalized possible model* of $\langle P, \mathcal{A} \rangle$ if it is a possible model of the normal disjunctive program $P \cup E$. A generalized disjunctive stable model (resp. generalized possible model) I is $\mathcal{A}$-*minimal* if there is no generalized disjunctive stable model (resp. generalized possible model) J such that $J \cap \mathcal{A} \subset I \cap \mathcal{A}$.

The above definitions are direct extensions of the previously proposed notions. In fact, generalized disjunctive stable models (resp. generalized possible models) reduce to disjunctive stable models (resp. possible models) in normal disjunctive programs with $\mathcal{A} = \emptyset$, and both generalized disjunctive stable models and generalized possible models reduce to generalized stable models in abductive logic programs.

A difference between generalized disjunctive stable models and generalized possible models is illustrated in the following example.

Example 5.1 Let $\langle P, \mathcal{A} \rangle$ be an abductive disjunctive program such that

$$P = \{ \, a \vee b \leftarrow c, \quad d \leftarrow a \wedge b \, \}$$

and $\mathcal{A} = \{ \, c \, \}$. Then, $\emptyset$, $\{c, a\}$, $\{c, b\}$, $\{c, a, b, d\}$ are all generalized possible models, while $\{c, a, b, d\}$ is not a generalized disjunctive stable model. Thus, for a given observation $O = d$, it has an explanation c under the generalized possible models, while no explanation is available under the generalized disjunctive stable models. $\square$

In this way, the generalized possible model semantics can provide explanations which come from inclusive disjunctions, while the generalized disjunctive stable model semantics cannot in general.

Inoue and Sakama [IS93] present that generalized disjunctive stable models of abductive disjunctive programs are translated into disjunctive stable models of normal disjunctive programs. That is, normal disjunctive programs are as expressive as abductive disjunctive programs under the disjunctive stable model semantics. We first show that this fact also holds for the possible model semantics.

For an abductive disjunctive program $\langle P, \mathcal{A} \rangle$, we define its *dlp-transformation* $dlp(\langle P, \mathcal{A} \rangle)$ in the same manner as presented in Definition 3.1. Then the following results hold.

Theorem 5.1 Let $\langle P, \mathcal{A} \rangle$ be an abductive disjunctive program. Then,

(i) $I \setminus \{\varepsilon\}$ is a generalized possible model of $\langle P, \mathcal{A} \rangle$ iff I is a possible model of $dlp(\langle P, \mathcal{A} \rangle)$.

(ii) $I \setminus \{\varepsilon\}$ is an $\mathcal{A}$-minimal generalized possible model of $\langle P, \mathcal{A} \rangle$ iff I is an $\mathcal{A}$-minimal possible model of $dlp(\langle P, \mathcal{A} \rangle)$.

Proof: Similar to the proof of Theorem 3.1. $\square$

The above theorem, together with Theorem 4.1, implies the following result.

Corollary 5.2 Let $\langle P, \mathcal{A} \rangle$ be an abductive disjunctive program. Then $I \setminus \{\varepsilon\}$ is a generalized possible model of $\langle P, \mathcal{A} \rangle$ iff I is a generalized stable model of $alp(dlp(\langle P, \mathcal{A} \rangle))$. $\square$

By Theorem 5.1, normal disjunctive programs are also as expressive as abductive disjunctive programs under the possible model semantics. Moreover, Corollary 5.2 presents that abductive disjunctive programs can be expressed even by abductive logic programs under the generalized possible model semantics.

6 Discussion

In this section, we discuss relationships between disjunctive programs and abductive logic programs from the computational complexity point of view. Throughout this section, programs are assumed to be propositional programs.

When abductive logic programs do not contain negation as failure, Selman and Levesque [SL90] and Eiter and Gottlob [EG92] show that the decision problem of the existence of explanations for a given observation in an abductive Horn program is NP-complete. In other words, in an abductive Horn program, deciding whether there is a generalized stable model satisfying an observation is NP-complete.

Inoue [Ino91] and Satoh and Iwayama [SI91] show that an abductive logic program can be translated into a semantically equivalent normal logic program. For an abductive logic program $\langle P, \mathcal{A} \rangle$, consider a normal logic program obtained from P by adding the following clauses for each abducible A in $\mathcal{A}$:

$$A \leftarrow not\, A' \,,$$

$$A' \leftarrow not\, A \,,$$

where A' is a newly introduced atom not appearing elsewhere in P and is uniquely associated with each A. Then these authors show that there is a one-to-one correspondence between the generalized stable models of $\langle P, \mathcal{A} \rangle$ and the stable models of the transformed normal logic program. Since it is known that deciding whether an atom is true in a stable model is NP-complete [MT91], the above translation implies that deciding whether there is a generalized stable model satisfying a given observation is also NP-complete.[5]

[5] More precisely, the generalized stable models include the stable models as a special case, then its set-membership problem is NP-hard. Since the polynomial-time transformation translates the decision problem for a generalized stable model into the corresponding problem for a stable model which is in NP, the membership in NP also follows.

Table 1: Comparison of Computational Complexity

Program	Semantics	Complexity
Abductive LP	Horn Abduction	NP-complete
	Generalized Stable Model	NP-complete
Normal DLP	Possible Model	NP-complete
	Disjunctive Stable Model	Σ_2^P-complete
Abductive DLP	Generalized Possible Model	NP-complete
	Generalized Disjunctive Stable Model	Σ_2^P-complete

Sakama and Inoue [SI93] have shown that possible models of a normal disjunctive program can be efficiently expressed by stable models of a normal logic program. For a normal disjunctive program P, let us consider a normal logic program obtained from P by replacing each disjunctive clause:

$$A_1 \vee \ldots \vee A_l \leftarrow \Gamma$$

in P with the following normal clauses and an integrity constraint:

$$\begin{array}{l} A_i \leftarrow \Gamma \wedge \mathit{not}\, A'_i \quad \text{for } i = 1, \ldots, l\,, \\ A'_i \leftarrow \Gamma \wedge \mathit{not}\, A_i \quad \text{for } i = 1, \ldots, l\,, \\ \leftarrow \Gamma \wedge A'_1 \wedge \ldots \wedge A'_l\,, \end{array}$$

where each A'_i is a new atom not appearing in P and is uniquely introduced for each A_i in $\mathcal{HB}_P$. Then they show that there is a one-to-one correspondence between the possible models of P and the stable models of the transformed normal logic program. Therefore, deciding the existence of a possible model satisfying a given atom is NP-complete. Furthermore, from discussion presented in Section 5, since generalized possible models can be efficiently translated into possible models, the corresponding decision problem for generalized possible models is also NP-complete.

On the other hand, deciding the existence of a disjunctive stable model containing a given atom is known to be Σ_2^P-complete [EG93a]. Inoue and Sakama [IS93] present a polynomial-time transformation from abductive disjunctive programs to normal disjunctive programs under the disjunctive stable model semantics. Therefore, deciding whether there is a generalized disjunctive stable model satisfying a given observation is also Σ_2^P-complete. These results are summarized in Table 1.

The above complexity measures verify the results of this paper that the generalized stable model semantics of abductive logic programs can be expressed in terms of the possible model semantics of normal disjunctive programs by a polynomial-time transformation, and vice versa. Moreover, we can observe that *there is no efficient way to express the disjunctive stable model semantics in terms of the generalized stable model semantics unless* *the polynomial hierarchy collapses*. This observation extends the fact that disjunctive stable models cannot be expressed by stable models of a normal logic program in polynomial time [EG93b]. This explains the reason why we have chosen the possible model semantics in this paper. Also we can observe that when considering to extend the framework of abductive logic programs to abductive disjunctive programs, *the generalized possible model semantics enables us to extend the framework without increasing computational complexity, while this is not the case for the generalized disjunctive stable model semantics.* The fact that the complexity of the (generalized) disjunctive stable model semantics is higher than the complexity of the (generalized) possible model semantics is explained as follows: computation of disjunctive stable models introduces an additional source of complexity for its minimality-checking, while this is not the case for computation of possible models due to its "non-minimal" feature.

The possible model semantics is originally introduced in order to provide a flexible mechanism for closed world assumptions in disjunctive programs. However, the results of this paper reveal that the possible model semantics also contributes to bridge the gap between disjunctive programs and abductive logic programs. As presented earlier in this section, under the possible model semantics normal disjunctive programs are reducible to normal logic programs. Since abductive disjunctive programs are reducible to normal disjunctive programs, abductive disjunctive programs are also reducible to normal logic programs under the possible model semantics. Moreover, in [SI93] we have shown that normal disjunctive/logic programs are also transferable to semantically equivalent positive disjunctive programs. These facts, together with the results presented in this paper, indicate a somewhat surprising fact that *all "extensions" of logic programming, i.e., normal logic programs, disjunctive programs, and abductive logic programs, are essentially equivalent under the possible model semantics. That is, negation as failure, disjunctions, and abducibles can be used interchangeably under the possible model semantics.*

7 Conclusion

This paper has investigated the relationship between disjunctive programs and abductive logic programs. The contributions of this paper are summarized as follows.

1. A program transformation from abductive logic programs to normal disjunctive programs was presented. It was shown that the generalized stable models of an abductive logic program are characterized by the possible models of the transformed normal disjunctive program.

2. A converse transformation from normal disjunctive programs to abductive logic programs was presented. It was shown that the possible

models of a normal disjunctive program are exactly the generalized stable models of the transformed abductive logic program.

3. Normal disjunctive programs were proved to be as expressive as abductive disjunctive programs. Moreover, it was shown that abductive disjunctive programs, normal disjunctive programs, abductive logic programs, and normal logic programs are all equivalent under the possible model semantics.

4. From the computational complexity point of view, we have argued that expressing the disjunctive stable model semantics in terms of generalized stable models is most unlikely possible in polynomial time. Also, it was shown that the generalized possible model semantics can extend the framework of abductive logic programs to abductive disjunctive programs without increasing the computational complexity.

Disjunctive knowledge in disjunctive programs and abductive hypotheses in abductive logic programs appear to deal with very similar problems from different viewpoints. This paper verified this conjecture and revealed close relationships between disjunctive programs and abductive logic programs. That is, *both formalisms are just different ways of looking at the same problem if we choose the appropriate semantics*. The results of this paper verify the usefulness of the possible model semantics as a theoretical tool not only for disjunctive programs but also for abductive logic programs. Moreover, we have argued that the possible model semantics can provide a unifying framework for various extensions of logic programming, and in spite of its usefulness, it does not increase the computational complexity more than the classical propositional satisfiability. The possible model semantics also has a close relation to autoepistemic logic [IS94].

Acknowledgements

We are grateful to anonymous referees for their helpful comments.

References

[Cha93] Chan, E. P. F., A Possible World Semantics for Disjunctive Databases, *IEEE Trans. on Data and Knowledge Engineering* 5(2), 282-292, 1993.

[Dun92] Dung, P. M., Acyclic Disjunctive Logic Programs with Abductive Procedure as Proof Procedure, *Proc. Int. Conf. on Fifth Generation Computer Systems*, 555-561, 1992.

[EG92] Eiter, T. and Gottlob, G., The Complexity of Logic-Based Abduction, Research Report CD-TR 92/35, Technical University Vienna, 1992, to appear in *J. ACM*. Extended abstract in *Proc. 10th Ann. Symp. on Theoretical Aspects of Computer Science*, LNCS 665, 70-79, 1993.

[EG93a] Eiter, T. and Gottlob, G., Complexity Aspects of Various Semantics for Disjunctive Databases, *Proc. 12th ACM SIGACT-SIGMOD-SIGART Symp. on Principles of Database Systems*, 158-167, 1993.

[EG93b] Eiter, T. and Gottlob, G., Complexity Results for Disjunctive Logic Programming and Application to Nonmonotonic Logics, *Proc. Int. Logic Programming Symp.*, 266-278, 1993.

[EK89] Eshghi, K. and Kowalski, R. A., Abduction Compared with Negation by Failure, *Proc. 6th Int. Conf. on Logic Programming*, 234-254, 1989.

[GL88] Gelfond, M. and Lifschitz, V., The Stable Model Semantics for Logic Programming, *Proc. 5th Int. Conf./Symp. on Logic Programming*, 1070-1080, 1988.

[GL90] Gelfond, M. and Lifschitz, V., Logic Programs with Classical Negation, *Proc. 7th Int. Conf. on Logic Programming*, 579-597, 1990.

[Ino91] Inoue, K., Extended Logic Programs with Default Assumptions, *Proc. 8th Int. Conf. Logic Programming*, 490-504, 1991. Extended version: Hypothetical Reasoning in Logic Programs, *J. Logic Programming* 18, to appear, 1994.

[IS93] Inoue, K. and Sakama, C., Transforming Abductive Logic Programs to Disjunctive Programs, *Proc. 10th Int. Conf. on Logic Programming*, 335-353, 1993.

[IS94] Inoue, K. and Sakama, C., On Positive Occurrences of Negation as Failure, *Proc. 4th Int. Conf. on Principles of Knowledge Representation and Reasoning*, to appear, 1994.

[KM90] Kakas, A. C. and Mancarella, P., Generalized Stable Models: A Semantics for Abduction, *Proc. 9th European Conf. on Artificial Intelligence*, 385-391, 1990.

[KKT92] Kakas, A. C., Kowalski, R. A. and Toni, F., Abductive Logic Programming, *J. Logic and Computation* 2(6), 719-770, 1992.

[MT91] Marek, W. and Truszczynski, M., Autoepistemic Logic, *J. ACM* 38(3), 588-619, 1991.

[Min82] Minker, J., On Indefinite Data Bases and the Closed World Assumption, *Proc. 6th Int. Conf. on Automated Deduction*, Lecture Notes in Computer Science 138, Springer-Verlag, 292-308, 1982.

[Prz91] Przymusinski, T. C., Stable Semantics for Disjunctive Programs, *New Generation Computing*, 401-424, 1991.

[Sak89] Sakama, C., Possible Model Semantics for Disjunctive Databases, *Proc. 1st Int. Conf. on Deductive and Object-Oriented Databases*, 337-351, 1989.

[SI93] Sakama, C. and Inoue, K., An Alternative Approach to the Semantics of Disjunctive Logic Programs and Deductive Databases. Research

Report, ASTEM Research Institute of Kyoto. Shorter version: Negation in Disjunctive Logic Programs, *Proc. 10th Int. Conf. on Logic Programming*, 703-719, 1993.

[SI91] Satoh, K. and Iwayama, K., Computing Abduction by using the TMS, *Proc. 8th Int. Conf. on Logic Programming*, 505-518, 1991.

[SL90] Selman, B. and Levesque, H. J., Abductive and Default Reasoning: A Computational Core, *AAAI-90*, 343-348, 1990.