ICOT Technical Report: TR-0818

TR-0818

Relating Disjunctive Logic Programs to Default Theories

by

C. Sakama (ASTEM) & K. Inoue

November, 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

Relating Disjunctive Logic Programs to Default Theories

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

Katsumi Inoue
ICOT
Mita Kokusai Bldg., 21F
1-4-28 Mita, Minato-ku
Tokyo 108, Japan
inoue@icot.or.jp

November 4, 1992

Abstract

This paper presents the relationship between disjunctive logic programs and default theories. We first show that Bidoit and Froidevaux's positivist default theory causes a problem in the presence of disjunctive information in a program. Then we present a correct transformation of disjunctive logic programs into default theories and show a one-to-one correspondence between the stable models of a program and the extensions of its associated default theory. We also extend the results to extended disjunctive programs and investigate their connection to Gelfond et al's disjunctive default theory.

1 Introduction

A default theory initially introduced by Reiter [Rei80] is well-known as one of the major formalism of nonmonotonic reasoning in AI. Recent studies have shed light on the relationship between nonmonotonic reasoning and logic programming, and the default theory is also turned out to be closely related to the declarative semantics of logic programming [Prz88, BF91a, BF91b, MT89a, LY91, MS92, LS92].

Bidoit and Froidevaux [BF91a, BF91b] have firstly investigated the relationship between logic programming and default theories and introduced a *positivist default theory* for stratifiable and non-stratifiable logic programs. Marek and Truszczynski [MT89a] have also developed transformations of general logic programs into default theories and shown a one-to-one correspondence between stable models of a logic program and its corresponding default extensions. While, Li and You [LY91] have presented a method of translating some specific defaults into logic programs and shown its connection to the stable and the well-founded semantics of logic programs. Marek and Subrahmanian [MS92] have also shown the relationship between supported models of general logic programs and weak extensions of default theories. On the other hand, Gelfond et al [GLPT91] have recently proposed a new framework called a *disjunctive default theory* which is devised to treat default reasoning with disjunctive information. The disjunctive default theory is also closely related to the answer set semantics of extended disjunctive programs.

It is often said that the difficulty of Reiter's default theory arises when one considers default reasoning with disjunctive information. Using a popular example from [Poo89], when we consider default rules:

$$\frac{: lh\text{-}usable \wedge \neg lh\text{-}broken}{lh\text{-}usable}, \quad \frac{: rh\text{-}usable \wedge \neg rh\text{-}broken}{rh\text{-}usable}$$

with a disjunctive formula:

$$lh\text{-}broken \vee rh\text{-}broken$$

they have a single extension containing both *lh-usable* and *rh-usable*, which is unintuitive.

From the point of view of disjunctive logic programming, Lobo and Subrahmanian [LS92] present a one-to-one correspondence between minimal model semantics of a positive disjunctive program P and extensions of a default theory which is obtained from P by adding defaults $\frac{:\neg A}{\neg A}$ for each atom A. In the presence of negation in a program, Bidoit and Froidevaux [BF91a] present the relationship between a stratified disjunctive program and its associated positivist default theory. However, we will point out in this paper that Bidoit and Froidevaux's positivist default theory contains a flaw and cannot be applicable to a disjunctive program with negation even if it is stratifiable. Hence modification and extension are needed to relate disjunctive logic programs and default theories in general.

In this paper, we study the relation between disjunctive logic programs and default theories. In Section 3, we revisit Bidoit and Froidevaux's study and point out its problem in disjunctive logic programs. Then in Section 4 we introduce a transformation of a disjunctive logic program into a default theory and show a one-to-one correspondence between stable models of the program and extensions of its associated default theory. In Section 5, we extend the results to extended disjunctive logic programs, and their connection to Gelfond et al's disjunctive default theory is presented in Section 6. Finally, in Section 7 we discuss connections to autoepistemic logic and circumscription.

2 Disjunctive Logic Programs and Default Theories

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

$$A_1 \vee ... \vee A_l \leftarrow A_{l+1} \wedge ... \wedge A_m \wedge not A_{m+1} \wedge ... \wedge not A_n \quad (n \geq m \geq l \geq 1)$$

where A_i's are atoms and *not* is a negation by failure operator,[1] and all the variables are assumed to be universally quantified at the front of the clause. A clause is called *disjunctive* if $l > 1$, else if $l = 1$, it is called *normal*. When a clause contains no *not* ($m = n$), it is called *positive*. The disjunction $A_1 \vee \ldots \vee A_l$ is called the *head* and the conjunction $A_{l+1} \wedge \ldots \wedge not A_n$

[1] While the operator $\neg$ denotes classical negation in this paper.

is called the *body* of the clause. Each predicate in the head is said to be *defined* by the predicates in the body.

A program which contains at least one disjunctive clause is called a *disjunctive (logic) program* and a program containing no disjunctive clause is called a *general logic program*. A program consists of only positive clauses is called a *positive program*. A program containing no predicate recursively defined through its negation by failure is called a *stratified program*. A *ground clause* is a clause which contains no variable. A *ground program* is a program in which every variable is instantiated by the elements of the Herbrand universe of the program in every possible way. A ground program is a possibly infinite set of ground clauses. From the semantical point of view, a program is equivalent to its ground program, so we consider a ground program in this paper. As usual, we consider an interpretation and a model of a program P to be subsets of the Herbrand base $\mathcal{HB}_P$ of the program.

As for the semantics of programs, we consider the *stable model semantics* which is introduced by Gelfond and Lifschitz in [GL88]. The definition of the stable model semantics was initially given for general logic programs, and it is also extended to programs possibly containing disjunctive clauses.

Definition 2.1 Let P be a program and M be an interpretation of P. Consider a positive program P^M obtained from P as follows:

$$P^M = \{A_1 \vee ... \vee A_l \leftarrow A_{l+1} \wedge ... \wedge A_m \mid \text{there is a ground clause } A_1 \vee ... \vee A_l \leftarrow A_{l+1} \wedge ... \wedge A_m \wedge not A_{m+1} \wedge ... \wedge not A_n \text{ from } P \text{ and } A_{m+1}, ..., A_n \notin M\}.$$

P^M is called the *reduct* of P with respect to M. Then if M coincides with a minimal model of P^M, M is called a *stable model* of P. □

A similar extension is also presented in [Prz90]. A program has none, one or multiple stable models in general. Especially, when a program is stratified, it has at least one stable model called a *perfect model*.

A *default theory* D is a set of default rules of the form:

$$\frac{\alpha : \beta_1, .., \beta_n}{\gamma}$$

where $\alpha, \beta_1, .., \beta_n$ and γ are closed first order formulas and respectively called the *prerequisite*, the *justifications* and the *consequent*. Especially, if α is empty, we call D a *prerequisite-free* default theory. Note here that the above definition, which is due to [GLPT91], is different from the standard one [Rei80] in which the theory is given by the pair (D, W) of defaults and first order formulas. As noted in [GLPT91], since a formula F in W is viewed as a special default with the prerequisite *true* and the empty justification $\frac{:}{F}$ in D, both definitions are equivalent. Hence, throughout of this paper, we do not distinguish W from D and such a special default is written by F, instead of $\frac{:}{F}$.

A set of sentences S is *deductively closed* if $S = Th(S)$ where Th is the deductive closure operator as usual. An extension of a default theory is defined as follows.

Definition 2.2 [GLPT91] Let D be a default theory and E be a set of sentences. Then E is an *extension* of D if it coincides with the smallest deductively closed set of sentences E' satisfying the condition: for any ground instance of any default rule $\alpha : \beta_1, .., \beta_n / \gamma$ from D, if $\alpha \in E'$ and $\neg\beta_1, .., \neg\beta_n \notin E$ then $\gamma \in E'$. □

A default theory may have none, one or multiple extensions in general.

3 Positivist Default Theory Revisited

To relate logic programming with default theories, Bidoit and Froidevaux [BF91a, BF91b] have presented a transformation which translates logic programs into so-called *positivist default theories*. According to [BF91a], this transformation is presented as follows.

Definition 3.1 [BF91a] Let P be a program. Then its *positivist default theory* D associated with P is constructed as follows:

(i) For each positive clause $A_1 \vee ... \vee A_l \leftarrow A_{l+1} \wedge ... \wedge A_m$ from P, its corresponding formula $A_{l+1} \wedge ... \wedge A_m \Rightarrow A_1 \vee ... \vee A_l$ is in D,

(ii) Each clause containing negation by failure $A_1 \vee ... \vee A_l \leftarrow A_{l+1} \wedge ... \wedge A_m \wedge not A_{m+1} \wedge ... \wedge not A_n$ in P is transformed into the following default in D:

$$\frac{A_{l+1} \wedge ... \wedge A_m : \neg A_{m+1}, .., \neg A_n}{A_1 \vee ... \vee A_l}$$

(iii) For each atom A in $\mathcal{HB}_P$, the following *CWA-default* is in D:

$$\frac{: \neg A}{\neg A}$$

(iv) Nothing else is in D. □

Then [BF91a] claims that a positivist default theory associated with a stratified disjunctive program has always at least one extension (Theorem 3.5 in [BF91a]). Moreover,

(Theorem 4.1.3 in [BF91a]) Let P be a stratifiable logical database. Then M is a perfect model for P iff M is a default model for its positivist default theory.

In the above theorem, a default model means an Herbrand model of an extension and a "logical database" corresponds to a disjunctive program in our terminology. However, the following example shows that *there exists a stratified disjunctive program whose positivist default theory does not have any extension.*

Example 3.1 Let P be the stratified disjunctive program:

$$a \leftarrow b \wedge not\, c$$

$$b \leftarrow a \wedge not\, c$$

$$a \vee b \leftarrow$$

which has the perfect model $\{a, b\}$. While, consider its positivist default theory D:

$$\{\frac{b : \neg c}{a}, \frac{a : \neg c}{b}, a \vee b, \frac{: \neg a}{\neg a}, \frac{: \neg b}{\neg b}, \frac{: \neg c}{\neg c}\}$$

If we assume $E = Th(\{a, b, \neg c\})$, then $E' = Th(\{a \vee b, \neg c\})$ is the smallest deductively closed set satisfying each default in D. Since $E \neq E'$, D has no extension. □

The above example presents that the result reported in [BF91a] is problematic. Especially, when a program contains disjunctive information, the positivist default theory is of no use.[2] This observation also leads to the assertion that Theorem 5.2 in [Prz90], which presents the relationship between positivist default theories and the stable semantics of disjunctive programs, does not hold any more. Since previously presented results are now turned out to be incorrect, we now need modification and reconstruction of theories to relate disjunctive logic programs and default theories.

4 Translating Disjunctive Logic Programs into Default Theories

In this section, we present a transformation which translates disjunctive logic programs into default theories.

Definition 4.1 Let P be a disjunctive program. Then its *associated default theory* D_P is constructed as follows:

(i) Each clause $A_1 \vee ... \vee A_l \leftarrow A_{l+1} \wedge ... \wedge A_m \wedge not A_{m+1} \wedge ... \wedge not A_n$ in P is transformed into the following default in D_P:

$$\frac{: \neg A_{m+1}, .., \neg A_n}{A_{l+1} \wedge ... \wedge A_m \Rightarrow A_1 \vee ... \vee A_l}$$

(ii) For each atom A in $\mathcal{HB}_P$, the following *CWA-default* is in D_P:

$$\frac{: \neg A}{\neg A}$$

[2] According to our analysis, the proof of Lemma 3.3 in [BF91a] seems to contain a problem. However, if a disjunctive program contains no negation, the positivist default theory reduces to the defaults presented in [LS92] and it works well.

(iii) Nothing else is in D. $\Box$

Notice that D_P is a prerequisite-free default theory.

Remark: Marek and Truszczynski [MT89a] have developed three kinds of transformations tr_1, tr_2 and tr_3 which transform general logic programs into default theories. Considering these transformations in the context of disjunctive logic programs, a transformation presented in (i) is a direct extension of the transformation tr_2 except that we are considering CWA-defaults in (ii). While a transformation based upon tr_3 corresponds to the positivist default theory presented in the previous section, which is already turned out inappropriate to characterize disjunctive programs. While a tr_1-based transformation translates each clause into the following default:

$$\frac{A_{l+1} \wedge ... \wedge A_m : \neg A_{m+1}, .., \neg A_n}{A_1 \vee ... \vee A_l}$$

A difference between tr_1 and tr_3 is that in tr_1, each positive clause is transformed into a justification-free default in D. However, we cannot use this tr_1-based transformation as the following example shows.

Example 4.1 Consider the program $\{a \leftarrow b,\ b \leftarrow a,\ a \vee b \leftarrow\}$. Then by the above tr_1-based transformation, it is translated into the following set of defaults,

$$\{\frac{b:}{a},\ \frac{a:}{b},\ a \vee b,\ \frac{:\neg a}{\neg a},\ \frac{:\neg b}{\neg b}\}$$

which has no extension. $\Box$

These observations tell us that among three transformations in [MT89a], the tr_2-based transformation is the only candidate which can be used to characterize the semantics of disjunctive logic programs. Before verifying this expectation, we address some features of prerequisite-free defaults.

Lemma 4.1 Let D be a prerequisite-free default theory. Then E is an extension of D iff

$$E = Th(\{\gamma \mid \frac{:\beta_1, .., \beta_n}{\gamma} \in D \text{ where } \neg\beta_1, .., \neg\beta_n \notin E\})$$

Proof: If E is an extension of D, by Theorem 2.1 in [Rei80], $E = \bigcup_{i=0}^{\infty} E_i$ where

$$\begin{aligned} E_0 &= \{F \mid F \text{ is a first order formula in } D\}, \\ E_{i+1} &= Th(E_i) \cup \{\gamma \mid \frac{:\beta_1, .., \beta_n}{\gamma} \in D \text{ where } \neg\beta_1, .., \neg\beta_n \notin E\}. \end{aligned}$$

Then $E_i = Th(E_1)$ for $i \geq 2$, the result immediately follows. The only-if part follows from Theorem 2.5 in [Rei80]. $\Box$

The above lemma presents that prerequisite-free defaults are sufficient to assure the converse of Theorem 2.5 in [Rei80]. The above result is further simplified as follows. Let D be a default theory and E be a set of sentences. Then let D^E be a default theory which is obtained from D by

$$D^E = \{\frac{\alpha :}{\gamma} \mid \frac{\alpha : \beta_1, .., \beta_n}{\gamma} \in D \quad \text{and} \quad \neg\beta_1, .., \neg\beta_n \notin E\}$$

D^E is called the *reduct* of D with respect to E [GLPT91]. Then the following property holds.

Lemma 4.2 [GLPT91] A set of sentences E is an extension for a default theory D iff E is the minimal set E' closed under provability in propositional calculus and under the rules from D^E. □

From the above two lemmas, we get the following corollary.

Corollary 4.3 Let D be a prerequisite-free default theory. Then E is an extension of D iff $E = Th(D^E)$. □

The above corollary presents that for a prerequisite-free default theory D, its extensions are characterized by the set of first order formulas D^E.

Now we are in a position to prove the main result of this section. Before that, we recall the following result for positive disjunctive programs.

Lemma 4.4 [LS92] Let P be a positive disjunctive program. If E is an extension of D_P, $E \cap \mathcal{HB}_P$ is a minimal model of P. □

Theorem 4.5 Let P be a program and D_P be its associated default theory. Then

(i) if M is a stable model of P, there is an extension E of D_P such that $M = E \cap \mathcal{HB}_P$;

(ii) if E is an extension of D_P, then $M = E \cap \mathcal{HB}_P$ is a stable model of P.

Proof: (i) Suppose M is a stable model of P and let $E = Th(M \cup \neg\overline{M})$ where $\neg\overline{M} = \{\neg A \mid A \in \mathcal{HB}_P \setminus M\}$. Then for each clause $A_1 \vee .. \vee A_l \leftarrow A_{l+1} \wedge .. \wedge A_m$ in P^M, the corresponding formula $A_{l+1} \wedge .. \wedge A_m \Rightarrow A_1 \vee .. \vee A_l$ is in $D_P{}^E$. Since M is a minimal model of P^M and $D_P{}^E = P^M \cup \{\neg A \mid A \notin M\}$, M is also a minimal model of $D_P{}^E$. Then $Th(M \cup \neg\overline{M}) = Th(D_P{}^E)$ holds. Therefore, by Corollary 4.3, $Th(M \cup \neg\overline{M})$ is an extension of D_P, and since $Th(M \cup \neg\overline{M}) \cap \mathcal{HB}_P = M$, the result follows.

(ii) When E is an extension of D_P, $E = Th(D_P{}^E)$ holds by Corollary 4.3. Let $M = E \cap \mathcal{HB}_P$. Then for each formula $A_{l+1} \wedge .. \wedge A_m \Rightarrow A_1 \vee .. \vee A_l$ in $D_P{}^E$, the corresponding clause $A_1 \vee .. \vee A_l \leftarrow A_{l+1} \wedge .. \wedge A_m$ is in P^M. Since M is a minimal model of $D_P{}^E$ (by Lemma 4.4), it is also a minimal model of P^M. Hence the result follows. □

Corollary 4.6 A program P has no stable model iff D_P has no extension. □

The above theorem presents a one-to-one correspondence between the stable models of a program and its extensions. Especially for general logic programs, the above theorem reduces to the result in [MT89a].

Example 4.2 [GLPT91] Let P be the program consisting of the clauses:

$$lh\text{-}usable \leftarrow not\ ab_1$$

$$rh\text{-}usable \leftarrow not\ ab_2$$

$$ab_1 \leftarrow lh\text{-}broken$$

$$ab_2 \leftarrow rh\text{-}broken$$

$$lh\text{-}broken \vee rh\text{-}broken \leftarrow$$

Then each rule is transformed into the following defaults in D_P:

$$\frac{:\neg ab_1}{lh\text{-}usable},\ \ \frac{:\neg ab_2}{rh\text{-}usable},\ \ lh\text{-}broken \Rightarrow ab_1,\ \ rh\text{-}broken \Rightarrow ab_2,\ \ lh\text{-}broken \vee rh\text{-}broken$$

with the CWA-defaults:

$$\frac{:\neg lh\text{-}broken}{\neg lh\text{-}broken},\ \ \frac{:\neg rh\text{-}broken}{\neg rh\text{-}broken},\ \ \frac{:\neg lh\text{-}usable}{\neg lh\text{-}usable},\ \ \frac{:\neg rh\text{-}usable}{\neg lh\text{-}usable},\ \ \frac{:\neg ab_1}{\neg ab_1},\ \ \frac{:\neg ab_2}{\neg ab_2}$$

Then D_P has two extensions in which a collection of all atoms from each extension becomes

$$\{lh\text{-}usable, rh\text{-}broken, ab_2\} \text{ and } \{rh\text{-}usable, lh\text{-}broken, ab_1\}$$

which coincide with the stable models of P. □

The above example presents that Poole's paradox is eliminated in Reiter's default by considering CWA-defaults for each atom.

5 Default Translation of Extended Disjunctive Programs

An *extended disjunctive program* is a disjunctive program which contains *classical negation* as well as negation by failure in the program [GL91]. The definition of an extended disjunctive program is the same as that of a disjunctive program in Section 2 except that each clause in a program has the following form[3] :

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

[3] In [GL91], a connective | is used instead of $\vee$ to distinguish properties of an extended program from classical first order logic. But here we abuse the classical notation as far as no confusion arises.

where each L_i is a positive or negative literal. Especially, if a program contains no disjunctive clause, it is just called an *extended logic program.*

The semantics of an extended disjunctive program is defined in the same manner as the stable model semantics of disjunctive programs. Let P be an extended disjunctive program and S be a set of literals. Then the *reduct* P^S of P with respect to S is defined as follows:

$$P^S = \{L_1 \vee ... \vee L_l \leftarrow L_{l+1} \wedge ... \wedge L_m \mid \text{there is a ground clause } L_1 \vee ... \vee L_l \leftarrow L_{l+1} \wedge ... \wedge L_m \wedge not\, L_{m+1} \wedge ... \wedge not\, L_n \text{ from } P \text{ and } L_{m+1}, ..., L_n \notin S\}.$$

Then S is called an *answer set* of P, if S is a minimal set satisfying the conditions:

(i) for each clause in P^S, if $L_{l+1}, ..., L_m$ are in S, then some L_i $(1 \leq i \leq l)$ is in S,

(ii) if S contains both A and $\neg A$ for some atom A, then $S = \mathcal{L}$ where $\mathcal{L}$ is the set of all literals in the language.

A program has none, one or multiple answer sets in general. A program which has an answer set different from $\mathcal{L}$ is called *consistent.*

For an extended disjunctive program P, its *positive form* P' is obtained from P by replacing each negative literal $\neg A$ appearing in P with a newly introduced atom A' which has the same arity with A. Then P' is a disjunctive program containing no classical negation. For notational convenience, let S' be a positive form of an answer set S where each negative literal $\neg A$ in S is rewritten by A' in S'. Then the following relationship holds.

Lemma 5.1 [GL91] Let P be a *consistent* extended disjunctive program. Then P has an answer set S iff P' has a stable model S'. □

Note that in the above lemma, the consistency assumption is necessary (see Example 5.1). Since an extended disjunctive program reduces to a disjunctive program by considering its positive form, we can directly apply Definition 4.1 to give an associated default theory for an extended disjunctive program. We firstly rephrase Theorem 4.5 for our current use.

Lemma 5.2 Let P be an extended disjunctive program and P' be its positive form. Then

(i) if M' is a stable model of P', there is an extension E' of $D_{P'}$ such that $M' = E' \cap \mathcal{HB}_{P'}$;

(ii) if E' is an extension of $D_{P'}$, then $M' = E' \cap \mathcal{HB}_{P'}$ is a stable model of P'. □

The next theorem directly follows from the above two lemmas, which states a one-to-one correspondence between the answer sets of a program and its extensions.

Theorem 5.3 Let P be a *consistent* extended disjunctive program and P' be its positive form. Then

(i) if S is an answer set of P, there is an extension E' of $D_{P'}$ such that $S' = E' \cap \mathcal{HB}_{P'}$;

(ii) if E' is an extension of $D_{P'}$, then $S' = E' \cap \mathcal{HB}_{P'}$ is a positive form of an answer set S of P. □

Clearly the above results reduce to the case of extended logic programs in the absence of disjunction in a program.[4] It should be noted that when a program is not consistent, we cannot apply Theorem 5.3 in a straightforward way. To present such a case, let us introduce a couple of terminologies. An extension E' of $D_{P'}$ is said *contradictory* if it contains a pair of complementary atoms A and A', otherwise it is called *consistent*. Then the following relation holds.

Corollary 5.4 Let P be an extended disjunctive program and P' be its positive form. If $\mathcal{L}$ is the unique answer set of P, $D_{P'}$ has no consistent extension. □

The converse of the above corollary does not hold in general.

Example 5.1 Let P be the extended program,

$$a \leftarrow \neg b \wedge \neg c$$

$$\neg a \leftarrow$$

$$\neg b \leftarrow not\ b$$

$$\neg c \leftarrow not\ c$$

which has no answer set. While its positive form P' becomes

$$a \leftarrow b' \wedge c'$$

$$a' \leftarrow$$

$$b' \leftarrow not\, b$$

$$c' \leftarrow not\, c$$

and its associated default $D_{P'}$ consists of

$$b' \wedge c' \Rightarrow a,\ \ a',\ \ \frac{:\neg b}{b'},\ \ \frac{:\neg c}{c'},\ \ \frac{:\neg a}{\neg a},\ \ \frac{:\neg b}{\neg b},\ \ \frac{:\neg c}{\neg c},\ \ \frac{:\neg a'}{\neg a'},\ \ \frac{:\neg b'}{\neg b'},\ \ \frac{:\neg c'}{\neg c'}$$

which has a contradictory extension $Th(\{a, \neg b, \neg c, a', b', c'\})$. □

To characterize a program having no answer set, consider a program $P^{\mathcal{L}}$ which is the reduct of P with respect to $\mathcal{L}$. Then $P^{\mathcal{L}}$ is a collection of all *not*-free clauses from P. Now the following property holds.

[4] [GL91] presents a default translation of extended logic programs, which corresponds to tr_1 and differs from ours.

Lemma 5.5 [Ino91] Let P be an extended disjunctive program. Then P has a unique answer set $\mathcal{L}$ iff $P^{\mathcal{L}}$ has a unique answer set $\mathcal{L}$. □

Using this property, the following result holds.

Theorem 5.6 Let P be an extended disjunctive program, $P^{\mathcal{L}}$ be its *not*-free subprogram, and $P^{\mathcal{L}'}$ be its positive form. Then

(i) P has a unique answer set $\mathcal{L}$ iff $D_{P^{\mathcal{L}'}}$ has a unique contradictory extension;

(ii) P has no answer set iff $D_{P^{\mathcal{L}'}}$ has a consistent extension and $D_{P'}$ has no consistent extension. □

6 Relationship to Disjunctive Default Theory

A disjunctive default theory, recently proposed by Gelfond et. al [GLPT91], is known as one of the extension of Reiter's default theory which is devised to treat default reasoning with disjunctive information. In this section, we investigate the connection between the associated default theory presented in the previous sections and the disjunctive default theory.

A *disjunctive default theory* Δ is a collection of defaults of the form:

$$\frac{\alpha : \beta_1, .., \beta_m}{\gamma_1 \mid ... \mid \gamma_n}$$

where $\alpha, \beta_1, .., \beta_m, \gamma_1, .., \gamma_n$ $(m, n \geq 0)$ are closed first order formulas and also respectively called the *prerequisite*, the *justifications* and the *consequents*.

An *extension* of a disjunctive default theory is defined in the same manner with that of a default theory except that it is a minimal deductively closed set E of sentences such that if E satisfies the prerequisite and justification conditions, E is required to contain one of the consequent γ_i $(1 \leq i \leq n)$ rather than the disjunction itself.

For a given extended disjunctive program P, its *associated disjunctive default theory* Δ_P is defined as follows: a clause $L_1 \vee ... \vee L_l \leftarrow L_{l+1} \wedge ... \wedge L_m \wedge not L_{m+1} \wedge ... \wedge not L_n$ in P is translated into the following disjunctive default

$$\frac{L_{l+1} \wedge .. \wedge L_m : \neg L_{m+1}, .., \neg L_n}{L_1 \mid ... \mid L_l}$$

Note here that any CWA-default is not included in Δ_P. The following lemma is due to [GLPT91] which presents the relation between an extended disjunctive program and its associated disjunctive default theory.

Lemma 6.1 [GLPT91] Let P be an extended disjunctive program and Δ_P be its associated disjunctive default theory. Then a set of literals S is an answer set of P iff S is the set of all literals from an extension of Δ_P. □

In the previous section, we have investigated the relationship between extended disjunctive programs and default theories. Then we get the following theorem from Theorem 5.3 and Lemma 6.1. Recall here that S' is a positive form of an answer set S.

Theorem 6.2 Let P be a consistent extended disjunctive program and P' be its positive form. Then

(i) if E_Δ is an extension of Δ_P and $S = E_\Delta \cap \mathcal{L}$, then there exists an extension E' of $D_{P'}$ such that $S' = E' \cap \mathcal{HB}_{P'}$;

(ii) if E' is an extension of $D_{P'}$ and $S' = E' \cap \mathcal{HB}_{P'}$, then there exists an extension E_Δ of Δ_P such that $S = E_\Delta \cap \mathcal{L}$. □

Corollary 6.3 Let P be an extended disjunctive program, $P^{\mathcal{L}}$ be its *not*-free subprogram, and $P^{\mathcal{L}'}$ be its positive form. Then

(i) Δ_P has a unique extension $Th(\mathcal{L})$ iff $D_{P^{\mathcal{L}'}}$ has a unique contradictory extension;

(ii) Δ_P has no extension iff $D_{P^{\mathcal{L}'}}$ has a consistent extension and $D_{P'}$ has no consistent extension. □

The above results bridge a gap between disjunctive defaults and Reiter's default theories in terms of disjunctive logic programs.

In [GLPT91], the difficulty of expressing disjunctive information in Reiter's default is discussed using some examples. However, we have already seen that Poole's paradox is eliminated by considering the CWA-defaults in its associated default theory. The following examples, which are also given in [GLPT91] to differentiate each formalism, present that we do not lose any information under Reiter's default theory in the presence of disjunctive information.

Example 6.1 Let $\Delta_P = \{a \Leftrightarrow b,\ \ a \mid b\}$. Then the default theory

$$D_P = \{a \Leftrightarrow b,\ \ a \vee b,\ \ \frac{:\neg a}{\neg a},\ \ \frac{:\neg b}{\neg b}\}$$

has the unique extension $Th(\{a,b\})$ which is equivalent to the extension of Δ_P. □

Example 6.2 Let Δ_P be the following disjunctive default theory:

$$\{a \mid b,\ \ \frac{a:}{b},\ \ \frac{:\neg a}{c}\}$$

Then the corresponding default theory

$$D_P = \{a \vee b,\ \ a \Rightarrow b,\ \ \frac{:\neg a}{c},\ \ \frac{:\neg a}{\neg a},\ \ \frac{:\neg b}{\neg b},\ \ \frac{:\neg c}{\neg c}\}$$

has the unique extension $Th(\{\neg a, b, c\})$ where $Th(\{\neg a, b, c\} \cap \mathcal{HB}_P)$ coincides with the unique extension of Δ_P. □

In is not clear whether there is a general correspondence between the disjunctive default and Reiter's default, however, the results presented in this section show that Reiter's default theory has the same expressiveness as the disjunctive default theory to characterize the stable and answer set semantics of disjunctive logic programs.

7 Connection to Autoepistemic Logic and Circumscription

From the point of view of autoepistemic logic, it is known that there is a correspondence between extensions of Reiter's default theory and expansions of Moore's autoepistemic logic [Moo85]. Marek and Truszczynski [MT89b] have shown that there is a one-to-one correspondence between a *weak* extension of a default theory and an expansion of its corresponding autoepistemic theory. Further, they showed that for prerequisite-free defaults, the notions of weak extensions and extensions coincide. These facts imply that the results presented in this paper are also rephrased under autoepistemic logic. That is, in Definition 4.1 (i), instead of translating each clause in a program into a corresponding default rule, we can transform it into the following autoepistemic formula:

$$A_{l+1} \wedge \ldots \wedge A_m \wedge \neg LA_{m+1} \wedge \ldots \wedge \neg LA_n \Rightarrow A_1 \vee \ldots \vee A_l$$

and instead of the CWA-defaults in (ii), we have

$$\neg LA \Rightarrow \neg A.$$

Thus we obtain the autoepistemic theory AE_P associated with a disjunctive logic program P. Then the following result holds.

Theorem 7.1 Let P be a disjunctive logic program and AE_P be its associated autoepistemic theory defined above. Then there is a one-to-one correspondence between stable models of P and expansions of AE_P. □

Such an autoepistemic translation is also presented in [Prz90] in the context of the 3-valued stable model semantics.[5] Moreover, by using the same technique presented in the previous sections, this autoepistemic translation is also extensible to the answer set semantics of extended disjunctive programs and their associated disjunctive default theories. These observation presents that the results presented in this paper also provide yet another epistemic characterization of extended disjunctive programs and disjunctive default theories, which is different from such as [Lif91, Tru91].

Circumscription [Mc80] is also closely related with default theories. Etherington [Eth87] has shown that a certain class of default theories is identified with its corresponding circumscriptive theories. Saying the relationship in the context of logic programming, for a positive disjunctive program P, the default theory $P \cup \{\frac{:\neg A}{\neg A} \mid A \in \mathcal{HB}_P\}$ coincides with the

[5] However, its default counterpart is incorrectly stated as is presented in Section 3.

circumscription $Circ(P;A)$ where $Circ(P;A)$ denotes circumscribing each predicate of A in P.

Now we consider a correspondence between disjunctive programs and circumscription in general. For a disjunctive program P, let P_L be a program obtained from P by replacing each $not A$ in P by $\neg LA$, where LA is a new atom meaning A *is believed*. Then $AE_P = P_L \cup \bigcup_{A\in\mathcal{HB}_P}\{A \Rightarrow LA\}$. In the following, a model means an Herbrand model. Now the following theorem holds.

Theorem 7.2 Let P be a disjunctive program and D_P be its associated default theory. Then E is an extension of D_P iff M is a model of $Circ(P_L;A) \wedge \bigwedge_{A\in\mathcal{HB}_P} LA \equiv A$ such that $M \cap \mathcal{HB}_P = E \cap \mathcal{HB}_P$.

Proof: An extension E of D_P is also an extension of ${D_P}^E$ (Lemma 4.2). Now put $M' = E \cap \mathcal{HB}_P$. Then ${D_P}^E$ is a default theory associated with a positive disjunctive program $P^{M'}$, and M' is equivalent to a model of $Circ(P^{M'};A)$ [Eth87]. (1)

Now consider $Circ(AE_P;LA;A)$ where the third argument in $Circ$ denotes *variables*. Put $M = M' \cup \{LA \mid A \in M'\}$. Then clearly M is a model of $Circ(AE_P;LA;A)$. Let ${P_L}^M$ be a positive disjunctive program which is obtained from P_L by deleting (i) each clause which has a negative literal $\neg LA$ in its body such that $LA \in M$, and (ii) all negative literals $\neg LA$ in the remaining clauses. Then it clearly holds that M is a model of $Circ(AE_P;LA;A)$ iff M is a model of $Circ({AE_P}^M;LA;A)$ where ${AE_P}^M$ is ${P_L}^M \cup \bigcup_{A\in\mathcal{HB}_P}\{A \Rightarrow LA\}$. Since an atom LA does not appear in ${P_L}^M$ and the minimization of LA implies that of A by $\neg LA \Rightarrow \neg A$, the extension of each predicate from A in a model M of $Circ({AE_P}^M;LA;A)$ coincides with that in a model $M \cap \mathcal{HB}_P$ of $Circ({P_L}^M;A)$.

Then it holds that M is a model of $Circ(AE_P;LA;A)$ iff $M \cap \mathcal{HB}_P$ is a model of $Circ({P_L}^M;A)$. Moreover, $Circ(AE_P;LA;A)$ is equivalent to $Circ(P_L;A) \wedge \bigwedge_{A\in\mathcal{HB}_P} LA \equiv A$ (by Proposition 3 in [Lif87]), it follows that $M \cap \mathcal{HB}_P$ is a model of $Circ({P_L}^M;A)$ iff M is a model of $Circ(P_L;A) \wedge \bigwedge_{A\in\mathcal{HB}_P} LA = A$. (2)

Since $P^{M'} = {P_L}^M$, by (1) and (2) the result follows. □

The above $Circ(P_L;A) \wedge \bigwedge_{A\in\mathcal{HB}_P} LA \equiv A$ corresponds to the so-called *introspective circumscription* in [Lif89].[6] Then the above theorem extends the result presented in [Lif89] for general logic programs to the case of general and extended disjunctive programs.

8 Conclusion

This paper has presented the relationship between disjunctive logic programs and default theories. The contributions of this paper are summarized as follows:

1. The problem of Bidoit and Froidevaux's positivist default theory was pointed out. It was shown that we could not use the positivist default theory any more in the presence of disjunctive information in a program.

[6] A similar result is recently reported by Lin and Shoham in [LiS92].

2. A transformation of disjunctive logic programs into default theories was presented. This transformation is the one and only one extension of Marek and Truszczynski's transformations, and it was shown a one-to-one correspondence between the stable models of a disjunctive logic program and the default extensions of its associated default theory.

3. The above result was also extended to extended disjunctive programs. It was shown that the answer set semantics of an extended disjunctive program was also characterized by its associated default theory.

4. The connection between the associated default theory and Gelfond et al's disjunctive default theory is presented. Reiter's default theory was shown to be still expressive as well as the disjunctive default theory to characterize the semantics of disjunctive logic programs.

5. We have also discussed relationships between disjunctive logic programs and autoepistemic theories and circumscription. These results not only provide bridges between our work and previously proposed approaches, but also present yet another characterization of extended disjunctive programs and disjunctive default theories.

References

[BF91a] Bidoit, N. and Froidevaux, C., General Logic Databases and Programs: Default Logic Semantics and Stratification, *J. Information and Computation* 91, 15-54, 1991.

[BF91b] Bidoit, N. and Froidevaux, C., Negation by Default and Unstratifiable Logic Programs, *Theoretical Computer Science* 78, 85-112, 1991.

[Eth87] Etherington, D. W., Relating Default Logic and Circumscription, *Proc. IJCAI'87*, 489-494, 1987.

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

[GL91] Gelfond, M. and Lifschitz, V., Classical Negation in Logic Programs and Disjunctive Databases, *New Generation Computing* 9, 365-385, 1991.

[GLPT91] Gelfond, M., Lifschitz, V., Przymusinska, H. and Truszczynski, M., Disjunctive Defaults, *Proc. 2nd Int. Conf. on Principles of Knowledge Representation and Reasoning*, Cambridge, 230-237, 1991.

[Ino91] Inoue, K., Extended Logic Programs with Default Assumptions, *Proc. 8th Int. Conf. on Logic Programming*, Paris, 490-504, 1991.

[Lif87] Lifschitz, V., Pointwise Circumscription, in *Readings in Nonmonotonic Reasoning* (M. Ginsberg ed.), 179-193, Morgan Kaufmann, 1987.

[Lif89] Lifschitz, V., Between Circumscription and Autoepistemic Logic, *Proc. 1st Int. Conf. on Principles of Knowledge Representation and Reasoning*, Toronto, 235-244, 1989.

[Lif91] Lifschitz, V., Nonmonotonic Databases and Epistemic Queries, *Proc. IJCAI'91*, 381-386, 1991.

[LY91] Li, L. and You, J-H., Making Default Inference from Logic Programs, *Computational Intelligence* 7, 142-153, 1991.

[LiS92] Lin F. and Shoham, Y. A Logic of Knowledge and Justified Assumptions, *Artificial Intelligence* 57, 271-289, 1992.

[LS92] Lobo, J. and Subrahmanian, V. S., Relating Minimal Models and Pre Requisite-Free Normal Defaults, *to appear in Information Processing Letter*, 1992.

[Mc80] McCarthy, J., Circumscription - a form of nonmonotonic reasoning, *Artificial Intelligence* 13, 27-39, 1980.

[Moo85] Moore, R. C., Semantical Considerations on Nonmonotonic Logic, *Artificial Intelligence* 25, 75-94, 1985.

[MT89a] Marek, W. and Truszczynski, M., Stable Semantics for Logic Programs and Default Theories, *Proc. North American Conf. on Logic Programming*, 243-256, 1989.

[MT89b] Marek, W. and Truszczynski, M., Relating Autoepistemic and Default Logics, *Proc. 1st Int. Conf. on Principles of Knowledge Representation and Reasoning*, Toronto, 276-288, 1989.

[MS92] Marek, W. and Subrahmanian, V. S., The Relationship between Stable, Supported, Default and Autoepistemic Semantics for General Logic Programs, *Theoretical Computer Science* 103, 365-386, 1992.

[Poo89] Poole, D., What the Lottery Paradox Tells us about Default Reasoning, *Proc. 1st Int. Conf. on Principles of Knowledge Representation and Reasoning*, 333-340, 1989.

[Prz88] Przymusinski, T. C., On the Relationship between Nonmonotonic Reasoning and Logic Programming, *Proc. AAAI'88*, 444-448, 1988.

[Prz90] Przymusinski, T. C., Extended Stable Semantics for Normal and Disjunctive Programs, *Proc. 7th Int. Conf. on Logic Programming*, Jerusalem, 459-477, 1990.

[Rei80] Reiter, R., A Logic for Default Reasoning, *Artificial Intelligence* 13, 81-132, 1980.

[Tru91] Truszczynski, M., Modal Interpretations of Default Logic, *Proc. IJCAI'91*, 393-398, 1991.