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
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 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 connections with Gelfond et al's disjunctive default theory, autoepistemic logic, and circumscription.

1 Introduction

Default logic 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 default logic has also turned out to be closely related to the declarative semantics of logic programming [BH86, BF91a, BF91b, Prz88, MT89a, GL91, LY91, MS92, LS92].

Bidoit and Froidevaux [BF91a, BF91b] have firstly investigated the relationship between logic programming and default logic and introduced a

*In *Proceedings of the 2nd International Workshop on Logic Programming and Non-monotonic Reasoning (LPNMR'93)*, MIT Press, pp. 266-282, 1993.

positivist default theory for stratifiable and non-stratifiable logic programs. Marek and Truszczynski [MT89a] have also developed transformations of logic programs into default theories and shown a one-to-one correspondence between the stable models of a logic program and its corresponding default extensions. The result was further extended by Gelfond and Lifschitz [GL91] to programs with classical negation, in which they present a connection between answer sets of a program and its corresponding default extensions.

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

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

with a disjunctive formula:

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

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

From the viewpoint of *disjunctive logic programming*, Bidoit and Hull [BH86] present a one-to-one correspondence between the minimal models of a positive disjunctive program P and the 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 a 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 problem and cannot be applicable to a disjunctive program with negation even if it is stratifiable. Recently, Gelfond et al. [GLPT91] proposed a new framework called *disjunctive default logic* which is a direct extension of Reiter's default logic. While the disjunctive default logic is closely related to the answer set semantics of extended disjunctive programs, it remains open whether there is a correspondence between Reiter's default logic and disjunctive logic programs 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 programs. Then in Section 4, we introduce a transformation of a disjunctive program into a default theory and show a one-to-one correspondence between the stable models of a disjunctive program and the extensions of its associated default theory. In Section 5, we extend the results to extended disjunctive programs, and their connection with Gelfond et al's disjunctive default theory is presented in Section 6. Finally, in Section 7 we discuss connections with autoepistemic logic and circumscription.

2 Disjunctive Logic Programs and Default Theories

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

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

where A_i's are atoms and *not* is the negation-by-failure operator.[1] A clause is called *disjunctive* if $l > 1$, else if $l = 1$, it is called *normal*. The left-hand side of the clause is called the *head*, while the right-hand side of the clause is called the *body*.

A program possibly containing disjunctive clauses is called a *disjunctive program* and a program containing no disjunctive clause is called a *normal program*. A program containing no *not* is called a *positive program*. A program containing no self-recursive predicate through its negation is called a *stratified program*. Given a program P, its *ground program* consists of all (possibly infinite) ground instances of the clauses from P. As usual, we consider an interpretation of a program P as a subset of the Herbrand base $\mathcal{HB}_P$ of the program, and identify any program with its ground program.

As for the semantics of programs, we consider the *stable model semantics* by Gelfond and Lifschitz [GL88]. The stable model semantics was initially given for normal programs, and we directly extend the definition to disjunctive programs.

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:

$$P^M = \{A_1 \vee \ldots \vee A_l \leftarrow A_{l+1} \wedge \ldots \wedge A_m \mid \text{ there is a ground clause of the form: } A_1 \vee \ldots \vee A_l \leftarrow A_{l+1} \wedge \ldots \wedge A_m \wedge not A_{m+1} \wedge \ldots \wedge not A_n \text{ from } P \text{ such that } \{A_{m+1}, \ldots, A_n\} \cap M = \emptyset \ \}.$$

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

Note that the above definition reduces to the notion of stable models in [GL88] in normal programs. A similar extension is also presented in [Prz90].

A program has none, one or multiple stable models in general. In particular, 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:

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

[1] The connective $\neg$ denotes classical negation in this paper.

where $\alpha, \beta_1, \ldots, \beta_n$ and γ are quantifier-free first-order formulas and respectively called the *prerequisite*, the *justifications* and the *consequent*.[2] In particular, if every α in D is empty (or α is *true*), 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 this paper, we do not distinguish W from D, and such a special default is written by F instead of $\frac{:}{F}$. As usual, we assume a default rule with variables as a shorthand for the set of all its ground instances.

A set of formulas 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 formulas. Then E is an *extension* of D if it coincides with the smallest deductively closed set of formulas E' satisfying the condition: for any ground instance $\alpha : \beta_1, \ldots, \beta_n / \gamma$ of any default rule from D, if $\alpha \in E'$ and $\neg\beta_1, \ldots, \neg\beta_n \notin E$ then $\gamma \in E'$. $\square$

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] 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 the *positivist default theory* D associated with P is constructed as follows:

(i) For each *not*-free clause $A_1 \vee \ldots \vee A_l \leftarrow A_{l+1} \wedge \ldots \wedge A_m$ in P, its corresponding formula $A_{l+1} \wedge \ldots \wedge A_m \Rightarrow A_1 \vee \ldots \vee A_l$ is in D.

(ii) Each clause containing *not* in its body $A_1 \vee \ldots \vee A_l \leftarrow A_{l+1} \wedge \ldots \wedge A_m \wedge$ $notA_{m+1} \wedge \ldots \wedge notA_n$ in P is transformed into the following default in D:
$$\frac{A_{l+1} \wedge \ldots \wedge A_m : \neg A_{m+1}, \ldots, \neg A_n}{A_1 \vee \ldots \vee A_l}.$$

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

[2] As in [GLPT91], we consider quantifier-free defaults in this paper.

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

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:

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

which has the perfect model $\{a, b\}$. Then 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 $E \neq E'$, E is not an extension. In fact, D has no extension. $\Box$

The above example presents that the result reported in [BF91a] is incorrect. In fact, when a program contains disjunctive information as well as negation, the positivist default theory causes a problem.[3] This observation also leads to the assertion that the result [Prz90, Theorem 5.2], which presents the relationship between positivist default theories and the stable semantics of disjunctive programs, does not hold too. Since previously presented results turned out to be incorrect, we now need modification and reconstruction of theories to relate disjunctive programs and default theories.

4 Translating Disjunctive Logic Programs into Default Theories

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

[3] According to our analysis, the proof of Lemma 3.3 in [BF91a] seems to contain a problem. However, if a disjunctive program contains no *not*, the positivist default theory reduces to the defaults presented in [BH86] and it works well.

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

(i) Each clause $A_1 \vee \ldots \vee A_l \leftarrow A_{l+1} \wedge \ldots \wedge A_m \wedge notA_{m+1} \wedge \ldots \wedge notA_n$ in P is transformed into the following default in D_P:

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

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

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

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

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 normal programs into default theories. Considering these transformations in the context of disjunctive programs, the transformation presented in (i) can be regarded as an extension of the transformation tr_2 except that we are considering the CWA-default in (ii). While a transformation based upon tr_3 corresponds to the positivist default theory presented in the previous section, it has already turned out inappropriate to characterize disjunctive programs. A tr_1-based transformation translates each clause into the default:

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

The difference between tr_1 and tr_3 is that in tr_3 each *not*-free clause is transformed into a first-order formula in D. However, this tr_1-based transformation is also inappropriate 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 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, from the viewpoint of extending three transformations in [MT89a], the tr_2-based transformation is the only candidate that can be used to characterize the semantics of disjunctive programs.

Then we verify the correctness of our transformation. First, we address some features of prerequisite-free default theories.

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

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

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

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

Then $E_i = Th(E_1)$ for $i \geq 2$, and the result immediately follows. □

The above lemma presents that prerequisite-free default theories 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 formulas. Then let D^E be a default theory which is obtained from D by

$$\begin{aligned} D^E \;=\; & \{\frac{\alpha:}{\gamma} \mid \frac{\alpha:\beta_1,\ldots,\beta_n}{\gamma} \text{ is a ground instance of a default in } D \\ & \text{and } \neg\beta_1,\ldots,\neg\beta_n \notin E\}. \end{aligned}$$

where 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 formulas E is an extension of 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 $E = Th(D^E)$. □

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 [BH86, LS92] Let P be a positive disjunctive program. If E is an extension of D_P, then $E \cap \mathcal{HB}_P$ is a minimal model of P. □

Theorem 4.5 Let P be a program and D_P be its associated default theory. Then the following relationships hold.

(i) If M is a stable model of P, then 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 $A_1 \vee \ldots \vee A_l \leftarrow A_{l+1} \wedge \ldots \wedge A_m$ in P^M, the corresponding formula $A_{l+1} \wedge \ldots \wedge A_m \Rightarrow A_1 \vee \ldots \vee A_l$ 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 $A_{l+1} \wedge \ldots \wedge A_m \Rightarrow A_1 \vee \ldots \vee A_l$ in ${D_P}^E$, the corresponding clause $A_1 \vee \ldots \vee A_l \leftarrow A_{l+1} \wedge \ldots \wedge A_m$ 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 the extensions of its associated default theory. Especially for normal programs, the above theorem reduces to the result in [MT89a].

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

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

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

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

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

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

These clauses are transformed into the following defaults in D_P:

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

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

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, and the sets of all atoms from them become

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

which coincide with the stable models of P. □

The above example presents that Poole's paradox can be 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* along with 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:[4]

$$L_1 \vee \ldots \vee L_l \leftarrow L_{l+1} \wedge \ldots \wedge L_m \wedge notL_{m+1} \wedge \ldots \wedge notL_n \quad (n \geq m \geq l \geq 1)$$

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

The semantics of an extended disjunctive program is defined in a similar way to the stable model semantics of disjunctive programs.

Let P be an extended disjunctive program containing no *not*, and $\mathcal{L}_P$ be the set of all ground literals from the language of P. Then, an *answer set* of P is defined as the minimal subset S of $\mathcal{L}_P$ satisfying the conditions:

1. For each ground clause $L_1 \vee \ldots \vee L_l \leftarrow L_{l+1} \wedge \ldots \wedge L_m$ from P, $\{L_{l+1}, \ldots, L_m\} \subseteq S$ implies $L_i \in S$ for some i $(1 \leq i \leq l)$; and
2. If S contains a pair of complementary literals L and $\neg L$, then $S = \mathcal{L}_P$.

Next, let P be an extended disjunctive program and $S \subseteq \mathcal{L}_P$. The *reduct* P^S of P with respect to S is defined as

$$\begin{aligned} P^S = \{ L_1 \vee \ldots \vee L_l \leftarrow L_{l+1} \wedge \ldots \wedge L_m \mid \ & \text{there is a ground clause of} \\ & \text{the form: } L_1 \vee \ldots \vee L_l \leftarrow L_{l+1} \wedge \ldots \wedge L_m \wedge notL_{m+1} \wedge \ldots \wedge notL_n \\ & \text{from } P \text{ such that } \{L_{m+1}, \ldots, L_n\} \cap S = \emptyset \ \}. \end{aligned}$$

Then S is an *answer set* of P if S is an answer set of P^S.

A program has none, one or multiple answer sets in general. 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 which is a straightforward extension of the result for extended logic programs [GL91, Proposition 2].

---

[4]In [GL91], the connective | is used instead of $\vee$ to distinguish properties of an extended disjunctive program from classical first-order logic. But here we abuse the connective $\vee$ for notational unification with previous sections.

Lemma 5.1 Let P be an extended disjunctive program. Then, a consistent set S is is an answer set of P iff S^+ is a stable model of P^+. $\square$

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.

(i) If M is a stable model of P^+, then 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^+. $\square$

The next theorem directly follows from the above two lemmas, which presents a one-to-one correspondence between the consistent answer sets of a program and the extensions of its associated default theory. In the following, an extension E of D_{P^+} is said *consistent* if it does not contain a pair of complementary atoms A and A'; otherwise it is called *contradictory*.

Theorem 5.3 Let P be an extended disjunctive program.

(i) If S is a consistent answer set of P, then there is an extension E of D_{P^+} such that $S^+ = E \cap \mathcal{HB}_{P^+}$.

(ii) If E is a consistent extension of D_{P^+}, then $S^+ = E \cap \mathcal{HB}_{P^+}$ is a positive form of an answer set S of P. $\square$

Clearly the above results reduce to the case of extended logic programs in the absence of disjunctions in a program.[5] It should be noted that when a program has no consistent answer set, we cannot apply Theorem 5.3 in a straightforward way.

Corollary 5.4 Let P be an extended disjunctive program. If $\mathcal{L}$ is the unique answer set of P, then D_{P^+} has no consistent extension. $\square$

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

Example 5.1 Let P be the extended program:

$$\{a \leftarrow \neg b, \quad \neg a \leftarrow, \quad \neg b \leftarrow not\, b\},$$

which has no answer set. On the other hand, its positive form P^+ becomes

[5] [GL91] presents a default translation of extended logic programs, which is an extension of tr_1 of [MT89a] and different from ours.

$$\{a \leftarrow b', \quad a' \leftarrow, \quad b' \leftarrow not\, b\},$$

and its associated default theory D_{P^+} is

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

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

To characterize a program having no consistent answer set, consider a program $P^{\mathcal{L}_P}$ which is the reduct of P with respect to $\mathcal{L}_P$. By the definition of answer sets, $\mathcal{L}_P$ is the answer set of P iff $\mathcal{L}_P$ is the answer set of $P^{\mathcal{L}_P}$. Let ${P^{\mathcal{L}_P}}^+$ be a positive form of $P^{\mathcal{L}_P}$. Then the following result holds.

Theorem 5.5 Let P be an extended disjunctive program. Then,

(i) P has the unique answer set $\mathcal{L}$ iff $D_{P^{\mathcal{L}+}}$ has no consistent 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 extensions of Reiter's default theory which is devised to treat default reasoning with disjunctive information. In this section, we investigate the connection between the disjunctive default theory and the associated default theory presented in the previous sections.

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

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

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

An *extension* E of a disjunctive default theory is defined in the same manner as a default theory except that it is a minimal deductively closed set E' of formulas such that if E' satisfies the prerequisite and E is consistent with the justifications, then E' is required to contain some 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 $L_1 \vee \ldots \vee L_l \leftarrow L_{l+1} \wedge \ldots \wedge L_m \wedge notL_{m+1} \wedge \ldots \wedge notL_n$ in P is translated into the disjunctive default:

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

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. $\Box$

In the previous section, we have investigated the relationship between extended disjunctive programs and default theories. Now 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 an extended disjunctive program.

(i) If E_Δ is an extension of Δ_P and $S = E_\Delta \cap \mathcal{L}$ is consistent, then there is an extension E of D_{P^+} such that $S^+ = E \cap \mathcal{HB}_{P^+}$.

(ii) If E is a consistent extension of D_{P^+} and $S^+ = E \cap \mathcal{HB}_{P^+}$, then there is an extension E_Δ of Δ_P such that $S = E_\Delta \cap \mathcal{L}$. $\Box$

Corollary 6.3 Let P be an extended disjunctive program and $P^{\mathcal{L}^+}$ be a positive form of $P^{\mathcal{L}}$. Then

(i) Δ_P has a unique extension $Th(\mathcal{L})$ iff $D_{P^{\mathcal{L}^+}}$ has no consistent extension;

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

The above results bridge the gap between disjunctive default theories and Reiter's default theories in terms of extended disjunctive programs.

In [GLPT91], the difficulty of expressing disjunctive information in Reiter's default theory 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. $\Box$

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. □

It remains open whether there is a general correspondence between the disjunctive default theory and Reiter's default theory. 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 extended disjunctive programs.

7 Connections with Autoepistemic Logic and Circumscription

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. Furthermore, they showed that for prerequisite-free default theories, the notions of weak extensions and extensions coincide. These facts imply that the results presented in the previous sections are also rephrased under autoepistemic logic. That is, in Definition 4.1 (i), instead of translating each clause in a program into the corresponding default rule, we can transform it into the following autoepistemic formula:

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

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

Such an autoepistemic translation is also presented in [Prz90] in the context of the 3-valued stable model semantics. However, by using the same technique presented in the previous sections, it is easily shown that this autoepistemic translation is extensible to the answer set semantics of extended disjunctive programs and their associated disjunctive default theories. These observations present 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 formalisms in such as [Lif91, Tru91].

Finally, we characterize the stable model semantics of disjunctive programs in terms of *circumscription* [Mc80]. For a disjunctive program P, let P_L be a first-order theory which is obtained from P by replacing each $notA$ in P by $\neg LA$, where LA is a new atom meaning A *is believed*. Then P_L is a set of formulas of the same form as $(*)$, but this time LA is interpreted not as an epistemic formula but as a first-order atom, and its Herbrand base is defined by $\mathcal{HB}_P \cup \{LA \mid A \in \mathcal{HB}_P\}$.

Let Π be the set of all predicates appearing in the language of a disjunctive program P. The circumscription $Circ(P_L; \Pi)$ means circumscribing the predicates Π in P_L with the *fixed* predicates $L\Pi$, where $L\Pi = \{Lp \mid p \in \Pi\}$. Any model of $Circ(P_L; \Pi)$ is a Π-*minimal* model of P_L, in which the extension of each predicate from Π is minimized with a fixed interpretation for each predicate from $L\Pi$. In the following, for $M \subseteq \mathcal{HB}_P$ we write $LM = \{LA \mid A \in M\}$, and $L\Pi \equiv \Pi$ means $\bigwedge_{p\in\Pi} \forall \mathbf{x}(p(\mathbf{x}) \equiv Lp(\mathbf{x}))$. Then, the following theorem holds.

Theorem 7.2 Let P be a disjunctive program. Then M is a stable model of P iff $M \cup LM$ is an Herbrand model of $Circ(P_L; \Pi) \wedge L\Pi \equiv \Pi$.

Proof: Let ${P_L}^{LM}$ be a first-order theory which is obtained from P_L by deleting (i) each formula which has a negative literal $\neg LA$ in its antecedent such that $LA \in LM$, and (ii) all negative literals $\neg LA$ in the antecedents of the remaining formulas. Then we first show the following lemma.

Lemma 7.3 $M \cup LM$ is an Herbrand model of $Circ({P_L}^{LM}; \Pi) \wedge L\Pi \equiv \Pi$ iff $M \cup LM$ is an Herbrand model of $Circ(P_L; \Pi) \wedge L\Pi \equiv \Pi$.

Proof: Let $M \cup LM$ be an Herbrand model of $Circ({P_L}^{LM}; \Pi) \wedge L\Pi \equiv \Pi$. Then for each ground formula $A_{l+1} \wedge \ldots \wedge A_m \Rightarrow A_1 \vee \ldots \vee A_l$ in ${P_L}^{LM}$, $\{A_{l+1}, \ldots, A_m\} \subseteq M$ implies $A_i \in M$ for some i $(1 \leq i \leq l)$. In this case, there is a corresponding ground formula $A_{l+1} \wedge \ldots \wedge A_m \Rightarrow A_1 \vee \ldots \vee A_l \vee LA_{m+1} \vee \ldots \vee LA_n$ from P_L such that $\{A_{l+1}, \ldots, A_m\} \subseteq M$ implies $A_i \in M$. Since M is Π-minimal, $M \cup LM$ is also an Herbrand model of $Circ(P_L; \Pi) \wedge L\Pi \equiv \Pi$.

Conversely, let $M \cup LM$ be an Herbrand model of $Circ(P_L; \Pi) \wedge L\Pi \equiv \Pi$. Then for each ground formula $A_{l+1} \wedge \ldots \wedge A_m \Rightarrow A_1 \vee \ldots \vee A_l \vee LA_{m+1} \vee \ldots \vee LA_n$ from P_L, $\{A_{l+1}, \ldots, A_m\} \subseteq M$ implies either (i) $A_i \in M$ for some i $(1 \leq i \leq l)$ or (ii) $LA_j \in LM$ for some j $(m+1 \leq j \leq n)$. In case of (i), when $LA_j \notin LM$, there is a corresponding ground formula $A_{l+1} \wedge \ldots \wedge A_m \Rightarrow A_1 \vee \ldots \vee A_l$ in ${P_L}^{LM}$ such that $\{A_{l+1}, \ldots, A_m\} \subseteq M$ implies $A_i \in M$. In case of (ii), there is no corresponding ground formula in ${P_L}^{LM}$. In each case, $M \cup LM$ is also a model of ${P_L}^{LM}$. Since M is Π-minimal, $M \cup LM$ is an Herbrand model of $Circ({P_L}^{LM}; \Pi) \wedge L\Pi \equiv \Pi$. □

Proof of Theorem 7.2: M is a stable model of P
iff M is a minimal model of P^M
iff M is an Herbrand model of $Circ(P^M; \Pi)$
iff M is an Herbrand model of $Circ({P_L}^{LM}; \Pi)$
iff $M \cup LM$ is an Herbrand model of $Circ({P_L}^{LM}; \Pi) \wedge L\Pi \equiv \Pi$
iff $M \cup LM$ is an Herbrand model of $Circ(P_L; \Pi) \wedge L\Pi \equiv \Pi$ (by Lemma 7.3).
□

For normal logic programs, the above relationship is also investigated under the name of *introspective circumscription* [Lif89] or *autoepistemic circumscription* [YY92]. The above theorem generalizes those results to disjunctive programs,[6] and using the techniques presented in the previous section, it also enables us to characterize extended disjunctive programs in terms of circumscription.

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 a disjunctive program with negation even if it is stratifiable.

2. An alternative transformation of disjunctive programs into default theories was presented. This transformation is also one of the extensions of Marek and Truszczynski's transformations, and it was shown a one-to-one correspondence between the stable models of a disjunctive 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 Reiter's default theory and Gelfond et al's disjunctive default theory was presented. Reiter's default theory was

[6] A similar result is also reported in [LiS92] without proof.

shown to be still expressive as well as the disjunctive default theory to characterize the semantics of disjunctive programs.

5. A disjunctive program was also characterized in terms of autoepistemic logic and circumscription. These results naturally extend the previously proposed results for normal programs and 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.

[BH86] Bidoit, N. and Hull, R., Positivism vs. Minimalism in Deductive Databases, *Proc. 5th ACM Symp. on Principles of Database Systems*, 123-132, 1986.

[GL88] Gelfond, M. and Lifschitz, V., The Stable Model Semantics for Logic Programming, *Proc. 5th Int. Conf./Symp. on Logic Programming*, 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*, 230-237, 1991.

[Lif89] Lifschitz, V., Between Circumscription and Autoepistemic Logic, *Proc. 1st Int. Conf. on Principles of Knowledge Representation and Reasoning*, 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, *Information Processing Letter* 44, 129-133, 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*, 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*, 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.

[YY92] Yuan, L. Y. and You, J-H., Autoepistemic Circumscription and Logic Programming, *J. of Automated Reasoning* (to appear), 1992.