Embedding Circumscriptive Theories in General Disjunctive Programs*

Chiaki Sakama[1]** and Katsumi Inoue[2]

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

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

Abstract. This paper presents a method of embedding circumscriptive theories in general disjunctive programs. In a general disjunctive program, negation as failure occurs not only in the body but in the head of a rule. In this setting, minimized predicates of a circumscriptive theory are specified using the negation in the body, while fixed and varying predicates are expressed by the negation in the head. Moreover, the translation implies a close relationship between circumscription and abductive logic programming. That is, fixed and varying predicates in a circumscriptive theory are also viewed as abducible predicates in an abductive disjunctive program. Our method of translating circumscription into logic programming is fairly general compared with the existing approaches and exploits new applications of logic programming for representing commonsense knowledge.

1 Introduction

It is well-known that logic programming semantics have close relationships to circumscription. In early studies, Reiter [Rei82] presented that Clark's predicate completion is characterized by circumscription in Horn theories. Lifschitz [Lif85a] showed that the closed world assumption is equivalent to circumscription when the CWA is consistent and circumscription minimizes every predicate in function-free first-order theories. Lifschitz [Lif88] and Przymusinski [Prz88] characterized the perfect model semantics of stratified logic programs and stratified disjunctive logic programs by prioritized circumscription. The results were further generalized by Gelfond *et al.* [GPP89] who introduced various forms of CWAs in terms of circumscription. Recent studies show that the stable model

* In *Proceedings of the 3rd International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR'95), Lecture Notes in Artificial Intelligence* 928, Springer-Verlag, pp. 344-357, 1995.

** Address after April 1995: Department of Computer and Communication Sciences, Wakayama University, 930 Sakaedani, Wakayama 640, Japan.

semantics of normal and disjunctive logic programs are also characterized by epistemic circumscription [Lif89,LS92,YY93,SI93].

On the other side, when we consider representing circumscriptive theories in logic programming, difficulties arise due to their axiomatic differences. According to [GL88a], there are three main differences between circumscription and logic programming as follows:

(a) In logic programming, the unique name assumption and the domain closure assumption are usually presumed, while no corresponding assumption exists in the definition of circumscription.
(b) Circumscription contains not only minimized predicates but fixed and varying predicates, while logic programming minimizes predicates by the CWA but lacks the mechanism of representing fixed and varying predicates.
(c) The meaning of logic programs is syntax-dependent and priorities are specified by negation as failure, while circumscription handles classical first-order theories and priorities are specified by the circumscriptive policy.

Such differences usually impose some restriction on formulas included in a circumscriptive theory. For instance, Gelfond and Lifschitz [GL88a] introduce a method of translating circumscriptive theories into normal logic programs, where theories are restricted to stratifiable ones without fixed predicates.

In this paper, we take the above points into consideration and explore a new method of embedding circumscriptive theories in logic programs. As for the part (a), we consider the Herbrand models of circumscriptive theories instead of incorporating both assumptions into the theories as in [GL88a], since each assumption is automatically satisfied in Herbrand models [BS85]. To fill the gap of (b), we introduce a class of *general disjunctive programs*, in which a rule possibly contains negation-as-failure formulas not only in the body but in the head of the rule. Then we show that fixed and varying predicates of circumscriptive theories are expressed by negation-as-failure formulas appearing in the heads of rules. For the part (c), we translate priorities specified by the circumscriptive policy into negation as failure in the body of a rule, together with the introduction of *characteristic clauses*. Our method is fairly general and faithfully translates a large class of circumscriptive theories into logic programs. Moreover, this translation implies a connection between circumscription and abductive logic programming. That is, from the viewpoint of abduction, fixed and varying predicates in a circumscriptive theory are considered as abducible predicates in an abductive disjunctive program.

The rest of this paper is organized as follows. In Section 2, we introduce a class of general disjunctive programs and present their semantics. In Section 3, we give a translation from circumscriptive theories into general disjunctive programs. It is shown that there is an equivalence relationship between the PZ-minimal Herbrand models of a circumscriptive theory and the stable models of a translated general disjunctive program. In Section 4, we show that circumscriptive theories are also expressed in terms of abductive disjunctive programs. Section 5 discusses related issues and applications to other nonmonotonic formalisms, and Section 6 summarizes the paper.

2 General Disjunctive Programs

Logic programs considering in this paper are disjunctive logic programs, which possibly contain negation-as-failure formulas not only in the body but in the head of a rule.

A *general disjunctive program* (GDP) is a set of rules of the form

$$A_1 \mid \ldots \mid A_k \mid not\, A_{k+1} \mid \ldots \mid not\, A_l \\ \leftarrow\; B_{l+1},\, \ldots,\, B_m,\, not\, B_{m+1},\, \ldots,\, not\, B_n\,, \tag{1}$$

where A_i's and B_j's are atoms and $n \geq m \geq l \geq k \geq 0$. The disjunction in the left-hand side is the *head* and the conjunction in the right-hand side is the *body* of the rule. Each rule with variables stands for the set of its ground instances as usual. Intuitive reading of the rule (1) is that if each $B_{l+1}, \ldots, B_m$ holds and each $B_{m+1}, \ldots, B_n$ does not hold then some A_i $(1 \leq i \leq k)$ holds or some A_j $(k+1 \leq j \leq l)$ does not hold.

Logic programs with such positive occurrences of negation as failure were initially introduced by Lifschitz and Woo as a subset of the logic of *minimal belief and negation as failure* (MBNF) [LW92,Lif94a]. Recently, its applications to commonsense reasoning in logic programming were exploited by Inoue and Sakama [IS94]. In [IS94], programs called *general extended disjunctive programs* (GEDPs) are introduced, which generalize *extended disjunctive programs* (EDPs) of [GL91] by introducing positive *not*. GDPs considering here are a subclass of GEDPs, i.e., GEDPs without classical negation. A GDP reduces to a *normal disjunctive program* (NDP) when each rule contains no *not* in the head. An NDP is called a *normal logic program* (NLP) if each rule contains at most one atom in the head, while an NDP is called a *positive disjunctive program* (PDP) if each rule contains no *not* in the body.

An *interpretation* of a program is a subset of the Herbrand base of the program. An interpretation I *satisfies* a ground rule of the form (1) iff $\{B_{l+1}, \ldots, B_m\} \subseteq I$ and $\{B_{m+1}, \ldots, B_n\} \cap I = \emptyset$ imply either $\{A_1, \ldots, A_k\} \cap I \neq \emptyset$ or $\{A_{k+1}, \ldots, A_l\} \not\subseteq I$.

A declarative meaning of a GDP is given by the *stable model semantics*. First, let Π be a PDP and M an interpretation. Then M is a *minimal model* of Π iff M is a minimal set satisfying each rule in Π. Next, let Π be a GDP and M an interpretation. The PDP Π^M is defined as follows: a rule

$$A_1 \mid \ldots \mid A_k \leftarrow B_{l+1}, \ldots, B_m \tag{2}$$

is in Π^M iff there is a ground rule of the form (1) from Π such that

$$\{A_{k+1}, \ldots, A_l\} \subseteq M \;\text{ and }\; \{B_{m+1}, \ldots, B_n\} \cap M = \emptyset\,.$$

Then, M is a *stable model* of Π iff M is a minimal model of Π^M.

By definition, any stable model of a GDP Π satisfies every ground rule from Π. For any ground formula F, we write $\Pi \models F$ iff $M \models F$ holds in every stable model M of Π.

The above definition of stable models is a special case of the definition of answer sets in a GEDP [LW92,IS94], and it reduces to the notion of stable models of Przymusinski [Prz91] when Π is an NDP, and that of Gelfond and Lifschitz [GL88b] when Π is an NLP.

It should be noted that in contrast with the case of NDPs or NLPs, a stable model of a GDP is not always minimal.

Example 2.1 Let Π be a program consisting of the single rule

$$A \mid not\, A \leftarrow .$$

Then Π has two stable models $\emptyset$ and $\{A\}$.

In the next section, we will see that this unique feature of stable models of GDPs is useful to represent fixed and varying predicates in circumscriptive theories.

3 Embedding Circumscriptive Theories in GDPs

Circumscription is a form of nonmonotonic reasoning initially introduced by McCarthy [Mc80] and thoroughly developed by Lifschitz [Lif94b]. We first review the framework of circumscription. The following definition is due to [Lif85b].

Given a first-order theory T, let P and Z be disjoint tuples of predicates from T. Then *circumscription* of P in T with *variable* Z is defined as the second-order formula:

$$Circ(T;P;Z) = T(P,Z) \wedge \neg\exists P'\, Z'(T(P',Z') \wedge P' < P), \tag{3}$$

where $T(P,Z)$ is a theory containing predicate constants P, Z, and P', Z' are tuples of predicate variables similar to P, Z. The set of all predicates other than P, Z from T is denoted by Q, which is called the *fixed* predicates.

A model of $Circ(T;P;Z)$ is called a PZ-*minimal model* of T, in which extensions of each predicate from P are minimized with those from Z varied and those from Q fixed. In particular, when Z is empty it is called a P-*minimal model*. For any first-order formula F, $Circ(T;P;Z) \models F$ iff $M \models F$ holds for every PZ-minimal model M of T [Lif85b].

We introduce some notations which will be useful. P^+, Z^+, and Q^+ respectively denote the sets of all positive literals with predicates from P, Z, and Q, while P^-, Z^-, and Q^- denote the sets of all negative literals correspondingly. We also use the letters P, Z, and Q to denote the sets of all (positive or negative) literals whose predicates belong to P, Z, and Q, respectively. The small letters $p_1, \ldots, p_l$, $z_1, \ldots, z_m$, and $q_1, \ldots, q_n$ are used to denote atoms from P^+, Z^+, and Q^+, respectively.

In this paper, we consider a first-order theory T as a set of *clauses* of the form:

$$A_1 \vee \ldots \vee A_l \vee \neg B_1 \vee \ldots \vee \neg B_m$$

where each A_i ($1 \leq i \leq l$; $l \geq 0$) and B_j ($1 \leq j \leq m$; $m \geq 0$) are atoms and every variable in the formula is assumed to be universally quantified at the front. Each clause is also identified with the set of its literals. A first-order theory is simply said a theory hereafter.

Given a theory T we restrict our attention to its *Herbrand models*, since we are interested in a semantic relationship to logic programming. Note that such a restriction has an effect to incorporate both the *domain closure assumption* and the *unique name assumption* into T [BS85].

To express circumscription in terms of logic programming, we first introduce the notion of characteristic clauses. We say a clause C_1 *subsumes* a clause C_2 if $C_1\theta \subseteq C_2$ holds for some substitution θ, and C_1 does not have literals more than C_2.[3]

Definition 3.1 Let T be a theory. Then a *characteristic clause* of T is defined as a clause C such that

(i) $T \vdash C$ where C consists of literals from $P^+ \cup Q$, and
(ii) for any clause C' such that $T \vdash C'$, if C' subsumes C then C subsumes C'.

The set of all characteristic clauses of T is denoted as $cc(T)$.

The notion of characteristic clauses is also introduced in [BS85,GPP89,Prz89,HIP91] and discussed in [Ino92] in a general setting.

Next we define a transformation from a first-order theory to a GDP.

Definition 3.2 Given a theory T, a GDP Π_T is constructed as follows.

1. For any clause in T of the form:

$$\begin{aligned} & p_1 \vee \ldots \vee p_l \vee z_1 \vee \ldots \vee z_m \vee q_1 \vee \ldots \vee q_n \\ & \quad \vee \neg p_{l+1} \vee \ldots \vee \neg p_s \vee \neg z_{m+1} \vee \ldots \vee \neg z_t \vee \neg q_{n+1} \vee \ldots \vee \neg q_u \,, \end{aligned} \tag{4}$$

   Π_T has the rule:

$$\begin{aligned} & z_1 \mid \ldots \mid z_m \mid q_1 \mid \ldots \mid q_n \\ & \leftarrow p_{l+1}, \ldots, p_s, z_{m+1}, \ldots, z_t, q_{n+1}, \ldots, q_u, not\, p_1, \ldots, not\, p_l \,. \end{aligned} \tag{5}$$

2. For every characteristic clause of T of the form:

$$p_1 \vee \ldots \vee p_l \vee q_1 \vee \ldots \vee q_n \vee \neg q_{n+1} \vee \ldots \vee \neg q_u \,, \tag{6}$$

   Π_T has the rule:

$$p_1 \mid \ldots \mid p_l \mid q_1 \mid \ldots \mid q_n \leftarrow q_{n+1}, \ldots, q_u \,. \tag{7}$$

[3] This subsumption is called the θ-subsumption in [Lov78].

3. For any atom $A \in Q \cup Z$, Π_T has the rule:

$$A \mid not\, A \leftarrow \ . \tag{8}$$

The intuitive meaning of the above transformation is explained as follows. First, for any clause (4) from T, predicates from P have higher priorities to Z for minimizing, hence each p_i $(i = 1, \ldots, l)$ is shifted to the body in its negated form $not\, p_i$ in (5). Second, any characteristic clause (6) from T is added to Π_T as a rule (7). This transformation retains each rule that can make atoms from P true. The rule (8) says that each atom A from Q or Z is either true or not.

The rest of this section shows that the above transformation exactly embeds a circumscriptive theory $Circ(T; P; Z)$ in a GEDP Π_T. In the following, PZ (or P)-minimal models mean PZ (or P)-minimal *Herbrand* models. Also any atom A is identified with the rule $A \leftarrow$ in a program. We begin with preliminary lemmas.

Lemma 3.1 Let T be a theory such that $Q = Z = \emptyset$. Then M is a P-minimal model of T iff M is a stable model of Π_T.

Proof. In this case, there is a characteristic clause $p_1 \vee \ldots \vee p_l$ from T iff there is a corresponding rule of the form $p_1 \mid \ldots \mid p_l \leftarrow$ in ${\Pi_T}^M$. Then, M is a P-minimal model of T iff M is a P-minimal model of $cc(T)$ and satisfies each rule of the form (5) iff M is a stable model of Π_T.

Lemma 3.2 [Lif94b, (3.2)] Let Ψ be a set of closed formulas containing no predicate from P, Z. Then, $Circ(T \wedge \Psi; P; Z) = Circ(T; P; Z) \wedge \Psi$. □

Lemma 3.3 Let T be a theory such that $Z = \emptyset$. Then M is a P-minimal model of T iff M is a stable model of Π_T.

Proof. M is a P-minimal model of T such that $M \cap Q = \Psi$ iff M is a model of $Circ(T; P) \wedge \Psi$. Here, $Circ(T; P) \wedge \Psi = Circ(T \wedge \Psi; P)$ by Lemma 3.2. Since Ψ is all the true atoms from Q in M, M is a model of $Circ(T \wedge \Psi; P)$ iff M is a model of $Circ(T \wedge \Psi; P, Q)$. In this case, the fixed predicates are considered empty, so that M is a model of $Circ(T \wedge \Psi; P, Q)$ iff M is a stable model of $\Pi_{T \wedge \Psi}$ by Lemma 3.1. For each $A \in \Psi$, there is a rule of the form (8) and $A \in {\Pi_T}^M$. Then, ${\Pi_{T \wedge \Psi}}^M = {\Pi_T}^M \wedge \Psi = {\Pi_T}^M$. Thus, M is a minimal model of ${\Pi_{T \wedge \Psi}}^M$ iff M is a minimal model of ${\Pi_T}^M$ iff M is a stable model of Π_T.

Let $M_P = M \cap P$, $M_Q = M \cap Q$, $M_Z = M \cap Z$, and Σ a disjunction of literals.

Lemma 3.4 Let T be a theory and M a P-minimal model of $cc(T)$. Then there is a model N of T such that $N_P = M_P$ and $N_Q = M_Q$.

Proof. When M is a P-minimal model of $cc(T)$, M satisfies any clause $\Sigma \subset P^+ \cup Q$ such that $T \vdash \Sigma$. Now we show the result by two steps.

(I) First, suppose that M does not satisfy some ground clause $C : \Sigma_0 \vee \neg p_1 \vee \ldots \vee \neg p_l$ such that $T \vdash C$, $T \not\vdash \Sigma_0$, and $\Sigma_0 \subset P^+ \cup Q$. Then $M \not\models \Sigma_0$ and

$M \models p_i$ $(i = 1, \ldots, l)$. We first show that for each p_i there is a ground clause $C_i : p_i \vee \Sigma_i$ from $cc(T)$ such that $M \not\models \Sigma_i$. Suppose to the contrary that there is no such clause for some p_i. Then, for any ground clause Σ_i, $cc(T) \vdash p_i \vee \Sigma_i$ implies $M \models \Sigma_i$. Hence, $M \setminus \{p_i\}$ is a model of $cc(T)$, which contradicts the fact that M is a P-minimal model of $cc(T)$. Next, by C and each C_i, $T \vdash \Sigma_0 \vee \Sigma_1 \vee \ldots \vee \Sigma_l$. However, since $M \not\models \Sigma_i$ for $i = 0, \ldots, l$, $M \not\models \Sigma_0 \vee \Sigma_1 \vee \ldots \vee \Sigma_l$. This contradicts the fact that M satisfies any clause $\Sigma \subset P^+ \cup Q$ such that $T \vdash \Sigma$. Therefore, M satisfies any clause $\Sigma' \subset P \cup Q$ such that $T \vdash \Sigma'$.

(II) Next, let us consider the set $\mathcal{Z}_M(T)$ of ground clauses defined by

$$\mathcal{Z}_M(T) = \{\, l_1 \vee \ldots \vee l_n \mid \Sigma \vee l_1 \vee \ldots \vee l_n \in \mathit{ground}(T), \\ \Sigma \subset P \cup Q,\ M \not\models \Sigma,\ \text{and } l_i \in Z \text{ for } i = 1, \ldots, n \,\},$$

where $\mathit{ground}(T)$ is the set of ground clauses from T. Suppose that $\mathcal{Z}_M(T)$ is unsatisfiable. By Herbrand's Theorem, there is a finite subset Γ of $\mathcal{Z}_M(T)$ such that Γ is unsatisfiable. Then, let

$$\Gamma_M(T) = \{\, \Sigma \mid \Sigma \vee l_1 \vee \ldots \vee l_n \in \mathit{ground}(T) \text{ and } l_1 \vee \ldots \vee l_n \in \Gamma \,\}.$$

By $\Gamma \vdash \Box$ and $T \wedge \bigwedge_{\Sigma_i \in \Gamma_M(T)} \neg \Sigma_i \vdash \Gamma$, $T \wedge \bigwedge_{\Sigma_i \in \Gamma_M(T)} \neg \Sigma_i \vdash \Box$ holds, thereby $T \vdash \bigvee_{\Sigma_i \in \Gamma_M(T)} \Sigma_i$. However, $M \not\models \Sigma_i$ for each $\Sigma_i \in \Gamma_M(T)$ by the construction of $\Gamma_M(T)$ and $\mathcal{Z}_M(T)$. Hence, $M \not\models \bigvee_{\Sigma_i \in \Gamma_M(T)} \Sigma_i$, which contradicts the result of (I) that M satisfies every clause $\Sigma' \subset P \cup Q$ such that $T \vdash \Sigma'$. Therefore, $\mathcal{Z}_M(T)$ is satisfiable. Let N_Z be a model of $\mathcal{Z}_M(T)$, and put $N = M \cup N_Z$. Then, N satisfies every clause of T, and $N_P = M_P$ and $N_Q = M_Q$.

Lemma 3.5 Let T be a theory and M a model of T. Then, M is a PZ-minimal model of T iff $M_P \cup M_Q$ is a P-minimal model of $cc(T)$.

Proof. ($\Rightarrow$) Suppose that M is a PZ-minimal model of T. Since $T \vdash C$ for any $C \in cc(T)$, M is also a model of $cc(T)$. Also $cc(T)$ does not include any literal from Z, $M_P \cup M_Q$ is a model of $cc(T)$. Suppose that $M_P \cup M_Q$ is not a P-minimal model of $cc(T)$. Then there is a P-minimal model N of $cc(T)$ such that $N_Q = M_Q$ and $N_P \subset M_P$. By Lemma 3.4, there is a model M' of T such that $M'_P = N_P$ and $M'_Q = N_Q$. However, since $M'_P \subset M_P$ and $M'_Q = M_Q$, this contradicts the fact that M is a PZ-minimal model of T. Therefore, $M_P \cup M_Q$ is a P-minimal model of $cc(T)$.

($\Leftarrow$) Suppose that $M_P \cup M_Q$ is a P-minimal model of $cc(T)$. By Lemma 3.4, there is a model N of T such that $N_P = M_P$ and $N_Q = M_Q$. If N is not a PZ-minimal model of T, there is a PZ-minimal model N' of T such that $N'_P \subset N_P$ and $N'_Q = N_Q$. In this case, however, $N'_P \cup N'_Q$ is a P-minimal model of $cc(T)$ by the ($\Rightarrow$) part, which contradicts the fact that $M_P \cup M_Q = N_P \cup N'_Q$ is a P-minimal model of $cc(T)$. Hence, $N = M_P \cup M_Q \cup N_Z$ is a PZ-minimal model of T. Since N and M differ only on predicates from Z, M is also a PZ-minimal model of T.

Given a theory T and an interpretation M, let us define a theory T^{M_P} such that the clause

$$z_1 \vee \ldots \vee z_m \vee q_1 \vee \ldots \vee q_n \\ \vee \neg p_{l+1} \vee \ldots \vee \neg p_s \vee \neg z_{m+1} \vee \ldots \vee \neg z_t \vee \neg q_{n+1} \vee \ldots \vee \neg q_u \tag{9}$$

is in T^{M_P} iff a clause

$$p_1 \vee \ldots \vee p_l \vee z_1 \vee \ldots \vee z_m \vee q_1 \vee \ldots \vee q_n \\ \vee \neg p_{l+1} \vee \ldots \vee \neg p_s \vee \neg z_{m+1} \vee \ldots \vee \neg z_t \vee \neg q_{n+1} \vee \ldots \vee \neg q_u \tag{10}$$

is in T and $M_P \not\models p_1 \vee \ldots \vee p_l$. Then the following property holds.

Lemma 3.6 If M is a PZ-minimal model of T, $M_Q \cup M_Z$ is a P-minimal model of T^{M_P}.

Proof. Since T^{M_P} contains no literals from P^+, any model N of T^{M_P} such that $N_P = \emptyset$ is a P-minimal model of T^{M_P}. On the other hand, if M is a model of T, $M_Q \cup M_Z$ is a model of T^{M_P} by definition. Therefore, $M_Q \cup M_Z$ is a P-minimal model of T^{M_P}.

Now we are ready to show the main result of this paper.

Theorem 3.1 Let T be a theory. Then M is a PZ-minimal model of T iff M is a stable model of Π_T.

Proof. Put $\Pi_T = \Pi_{sft} \cup \Pi_{cc} \cup \Pi_{pnf}$ where Π_{sft} is the set of rules of the form (5), Π_{cc} is the set of rules of the form (7), and Π_{pnf} is the set of rules of the form (8). Then, ${\Pi_T}^M = {\Pi_{sft}}^{M_P} \cup \Pi_{cc} \cup {\Pi_{pnf}}^{M_Q} \cup {\Pi_{pnf}}^{M_Z}$. Since ${\Pi_{pnf}}^{M_Q} = M_Q$ and ${\Pi_{pnf}}^{M_Z} = M_Z$, it becomes ${\Pi_T}^M = {\Pi_{sft}}^{M_P} \cup \Pi_{cc} \cup M_Q \cup M_Z$. Suppose that M is a PZ-minimal model of T. By Lemma 3.5, $M_P \cup M_Q$ is a P-minimal model of $cc(T)$, thereby a minimal model of $\Pi_{cc} \cup M_Q$ (by Lemma 3.3). Since $\Pi_{cc} \cup M_Q$ contains no atom from Z, $M_P \cup M_Q \cup M_Z$ is a minimal model of $\Pi_{cc} \cup M_Q \cup M_Z$. On the other hand, by Lemma 3.6, $M_Q \cup M_Z$ is a P-minimal model of T^{M_P}. Hence M satisfies T^{M_P}, so does ${\Pi_{sft}}^{M_P}$. Therefore, M is a minimal model of ${\Pi_{sft}}^{M_P} \cup \Pi_{cc} \cup M_Q \cup M_Z = {\Pi_T}^M$, hence a stable model of Π_T.

Conversely, suppose that M is a stable model of Π_T. Then M is a minimal model of ${\Pi_T}^M$. For each rule of the form:

$$z_1 \mid \ldots \mid z_m \mid q_1 \mid \ldots \mid q_n \leftarrow p_{l+1}, \ldots, p_s, z_{m+1}, \ldots, z_t, q_{n+1}, \ldots, q_u \tag{11}$$

in ${\Pi_{sft}}^{M_P}$, there is a corresponding clause of the form (9) in T^{M_P}. Then if M satisfies (11), so does (9). Also, for each rule not included in ${\Pi_{sft}}^{M_P}$ but in Π_{sft}, $M_P \models p_1 \vee \ldots \vee p_l$ holds, hence M also satisfies the corresponding clause (10) in T. Thus, M is a model of T. On the other hand, $M_P \cup M_Q$ is a minimal model of $\Pi_{cc} \cup M_Q$, thereby a P-minimal model of $cc(T)$ (by Lemma 3.3). Therefore, M is a PZ-minimal model of T by Lemma 3.5.

Corollary 3.7 Let T be a theory and F a ground formula. Then, $Circ(T; P; Z) \models F$ iff $\Pi_T \models F$. □

Note that $Circ(T; P; Z)$ is satisfiable iff Π_T has a stable model. When $Circ(T; P; Z)$ is unsatisfiable, T is also unsatisfiable [Lif94b, Corollary 6.3.3]. In this case, $cc(T)$ contains the empty clause only [Ino92, Proposition 2.5], then Π_T contains "$\leftarrow$" meaning the falsity and has no stable model.

Example 3.1 Let T be the theory:

$$bird(x) \vee \neg penguin(x), \tag{12}$$
$$\neg fly(x) \vee \neg penguin(x), \tag{13}$$
$$fly(x) \vee ab(x) \vee \neg bird(x), \tag{14}$$
$$bird(Joe), \tag{15}$$

where $P = \{ab\}$, $Z = \{fly\}$, and $Q = \{bird, penguin\}$. Then, $Circ(T; P; Z)$ has two PZ-minimal models: $M_1 = \{bird(Joe), fly(Joe)\}$ and $M_2 = \{bird(Joe), ab(Joe), penguin(Joe)\}$. In this case, Π_T becomes

$$bird(x) \leftarrow penguin(x), \tag{16}$$
$$\leftarrow fly(x), penguin(x), \tag{17}$$
$$fly(x) \leftarrow bird(x), not\, ab(x), \tag{18}$$
$$ab(x) \leftarrow penguin(x), \tag{19}$$
$$bird(Joe) \leftarrow, \tag{20}$$
$$fly(x) \mid not\, fly(x) \leftarrow, \tag{21}$$
$$bird(x) \mid not\, bird(x) \leftarrow, \tag{22}$$
$$penguin(x) \mid not\, penguin(x) \leftarrow, \tag{23}$$

which has the stable models M_1 and M_2.

In the above translation, by (12), (13), and (14) the characteristic clause

$$ab(x) \vee \neg penguin(x)$$

is obtained, which is translated into (19) in Π_T. Note that in this theory the translation of [GL88a] cannot be applied, since T contains the fixed predicates.

Example 3.2 Our translation can also handle theories representing *multiple inheritance*. Consider the theory T consisting of the clauses

$$pacifist(x) \vee ab_1(x) \vee \neg republican(x),$$
$$\neg pacifist(x) \vee ab_2(x) \vee \neg quaker(x),$$

where $P = \{ab_1, ab_2\}$, $Z = \{pacifist\}$, and $Q = \{republican, quaker\}$. Then Π_T becomes

$$pacifist(x) \leftarrow republican(x), not\, ab_1(x),$$

$$\begin{aligned}
&\leftarrow \mathit{pacifist}(x), \mathit{quaker}(x),\ \mathit{not}\, ab_2(x),\\
&ab_1(x) \mid ab_2(x) \leftarrow \mathit{republican}(x),\ \mathit{quaker}(x),\\
&\mathit{pacifist}(x) \mid \mathit{not}\, \mathit{pacifist}(x) \leftarrow,\\
&\mathit{republican}(x) \mid \mathit{not}\, \mathit{republican}(x) \leftarrow,\\
&\mathit{quaker}(x) \mid \mathit{not}\, \mathit{quaker}(x) \leftarrow .
\end{aligned}$$

Then, given the facts

$$\mathit{republican}(Nixon) \leftarrow, \quad \mathit{quaker}(Nixon) \leftarrow,$$

Π_T has two stable models: $\{\, \mathit{republican}(Nixon),\ \mathit{quaker}(Nixon),\ ab_1(Nixon) \,\}$ and $\{\, \mathit{republican}(Nixon),\ \mathit{quaker}(Nixon),\ \mathit{pacifist}(Nixon),\ ab_2(Nixon) \,\}$, which are exactly the PZ-minimal models of T. Note that the above theory is not represented by a stratified NLP, hence cannot be handled by [GL88a].

4 Connection with Abductive Disjunctive Programs

Abductive logic programming [KKT92] is an extension of logic programming, which realizes abductive reasoning in AI by incorporating a set of hypotheses into programs. In this section, we show that circumscriptive theories are expressed by abductive disjunctive programs using the result of the previous section.

An *abductive disjunctive program* (ADP) is a pair $\langle\, \Pi, \Gamma \,\rangle$ where Π is an NDP and Γ is a finite set of predicates called the *abducible predicates*. The set of all ground atoms $\mathcal{A}_\Gamma$ having abducible predicates from Γ is called the *abducibles*. Let $\langle\, \Pi, \Gamma \,\rangle$ be an ADP. An interpretation M is called a *belief set* of $\langle\, \Pi, \Gamma \,\rangle$ iff M is a stable model of the NDP $\Pi \cup E$ for some set $E \subseteq \mathcal{A}_\Gamma$. By definition, stable models of an NDP are belief sets of an ADP with $E = \emptyset$.

Inoue and Sakama [IS94] observe that a rule of the form:

$$A \mid \mathit{not}\, A \leftarrow$$

in a GDP is viewed as a rule expressing an abductive hypothesis, i.e., an abducible A is true or not. The correspondence is formally presented as follows.

Lemma 4.1 [IS94] Let $\langle\, \Pi, \Gamma \,\rangle$ be an ADP. Suppose that $\mathit{gdp}(\Pi, \Gamma)$ is a GDP such that

$$\mathit{gdp}(\Pi, \Gamma) = \Pi \cup \{\ A \mid \mathit{not}\, A \leftarrow \ \mid A \in \mathcal{A}_\Gamma\ \}.$$

Then, M is a belief set of $\langle\, \Pi, \Gamma \,\rangle$ iff M is a stable model of $\mathit{gdp}(\Pi, \Gamma)$. □

Since GDPs Π_T introduced in the previous section are of the form $\mathit{gdp}(\Pi, \Gamma)$, the above result implies that circumscriptive theories can be expressed in terms of ADPs by identifying fixed and varying predicates as abducible predicates.

Given a theory T, let $\mathit{adp}(T; P; Z) = \langle\, \Pi, \Gamma \,\rangle$ where (i) for each clause in T of the form (4), Π contains the rule of the form (5); (ii) for every characteristic clause of T of the form (6), Π contains the rule of the form (7); and $\Gamma = Q \cup Z$. Then the following theorem holds by Theorem 3.1 and Lemma 4.1.

Theorem 4.1 Let T be a theory. Then M is a PZ-minimal model of T iff M is a belief set of $adp(T;P;Z)$. □

Example 4.1 Let T be the theory of Example 3.1. Then $adp(T;P;Z)$ becomes $\langle \Pi, \Gamma \rangle$ where Π consists of rules (16) – (20), and $\Gamma = \{ fly, bird, penguin \}$. Then it is easy to see that M_1 and M_2 become the belief sets of $\langle \Pi, \Gamma \rangle$.

5 Discussion

In this section, we discuss related issues and further applications to nonmonotonic reasoning.

1. (*Comparison with GL-translation*)
Gelfond and Lifschitz [GL88a] introduce a translation from circumscriptive theories into NLPs. Comparing both translations, the GL-translation is restricted to stratifiable theories, and every clause is assumed to contain at most one variable predicate and no fixed predicates. By contrast, we have no such restriction and consider GDPs instead of NLPs. On the other hand, under the above condition they provide a translation from *prioritized circumscription* into NLPs, while we do not consider the issue here. The possibility of extending our translation to prioritized circumscription is currently under investigation.

It is known that circumscription with fixed and varying predicates can be reduced to circumscription without them [dKK89,CEG92]. Using the techniques, it might be possible to translate circumscriptive theories into disjunctive logic programs without handling fixed and varying predicates directly. However, these reductions introduce extra predicates which are not included in the original theory. In this sense, our method faithfully translates the original circumscriptive theory into a logic program by representing fixed and varying predicates in an appropriate manner.

2. (*Connection with EDPs*)
In this paper, we have used the rule of the form (8) for each atom A having fixed or varying predicates. On the other hand, if we use the *CWA-rule*

$$\neg A \leftarrow not\, A$$

for each atom A with a minimizing predicate, it is possible to use the rule

$$A \mid \neg A \leftarrow \tag{24}$$

instead of (8). Then there will be a one-to-one correspondence between the PZ-minimal Herbrand models of a first-order theory T and the answer sets of the translated EDP. Note that in this case an answer set contains negative literals explicitly, which are false in the corresponding PZ-minimal model. The rule of the form (24) is called a *completeness rule* in [LT94].

3. (*Proof procedure*)
Some proof procedures are known for computing circumscription directly

[Prz89,Gin89,HIP91]. On the other hand, the translation presented in this paper enables us to use proof procedures for GDPs also for circumscriptive theories. For example, a simple model generation procedure for GEDPs in [IS94] is used for this purpose.

Note that our translation is different from the approach of [Lif85b,Lif94b], which reduces the second-order circumscription formula (3) to an equivalent first-order formula. In our translation, computation of characteristic clauses is achieved by first-order deduction and always possible as long as the number of characteristic clauses is finite. Procedures for computing characteristic clauses are given in [Prz89,Ino92].

4. (*Relation to other NMR formalisms*)
Circumscription is related to other nonmonotonic formalisms. According to [GLPS94], circumscription $Circ(T;P;Z)$ is represented by *autoepistemic logic* (AEL) as follows:

$$T \wedge (P \supset \mathsf{B}P) \wedge (Q \supset \mathsf{B}Q) \wedge (\mathsf{B}Q \supset Q)\,.$$

Also, by analogy with the above translation, circumscription is represented by the MBNF formula[4]

$$\mathsf{B}T \wedge (not\,P \supset \mathsf{B}\neg P) \wedge (not\,Q \supset \mathsf{B}\neg Q) \wedge (not\,Q \vee \mathsf{B}Q)\,.$$

Comparing these AEL/MBNF translations with our GDP translations, some interesting correspondences are observed. First, the last formulas in the AEL/MBNF translations correspond to the rule (8) in our GDP translation.[5] Second, the AEL/MBNF translations include the CWA-formulas for each atom from both P and Q, while we have no such rules but include rules (7) for handling P. Third, in the AEL/MBNF translations no care is taken for each Z, while our GDP translation have rules (5) and (8) to cope with Z properly.

The result of this paper is also extended to another translation from circumscription into AEL/MBNF by combining the translation from GDPs into AEL/MBNF in [IS94,LW92]. These correspondences will generalize the result of this paper to the non-Herbrand model case.

5. (*Application to parametric knowledge*)
Lifschitz [Lif93] argues that in nonmonotonic systems some additional assumption makes the domain under consideration smaller, and consequently the class of conclusions true in all the domains becomes larger. He calls such assumptions as *parameters*, and the monotonicity property as *restricted monotonicity*. In circumscription, fixed predicates have the property of such parameters. For instance, in Example 3.1, $bird(Joe)$ and $penguin(Joe)$ are considered as parameters, therefore all theorems of T are included in the theorems

[4] Lifschitz [Lif94a] gives a similar translation without Q.

[5] Using a different formalization [Kon89], the formula $\mathsf{B}Q \supset Q$ in the AEL translation can be replaced with $\neg Q \supset \mathsf{B}\neg Q$.

of $T \cup \{penguin(Joe)\}$. Lifschitz provides a way to realize restricted monotonicity in EDPs. According to [Lif93], T is translated into the EDP Π'_T:

$$bird(x) \leftarrow penguin(x), \tag{16}$$

$$\neg fly(x) \leftarrow penguin(x), \tag{25}$$

$$fly(x) \leftarrow bird(x),\ not\, ab(x), \tag{18}$$

$$ab(x) \leftarrow not\, \neg penguin(x), \tag{26}$$

$$bird(Joe) \leftarrow . \tag{20}$$

The differences between Π'_T and Π_T are the existence of the rule (25) instead of (17), the rule (26) instead of (19), and the absence of (21), (22) and (23). Note here that the above translation realizes restricted monotonicity, while there is no correspondence between the answer sets of Π'_T and the PZ-minimal models of the original theory. In fact, Π'_T has the unique answer set $\{\, bird(Joe),\ ab(Joe) \,\}$, while no corresponding PZ-minimal model of T exists. On the other hand, in our translation there is an exact matching between the the stable models of Π_T and the PZ-minimal models of T by the presence of (21) – (23). Moreover, Lifschitz does not provide a method of encoding parametric knowledge in logic programming in general, while our translation illustrates that such knowledge is represented using the positive occurrences of negation as failure. Since parametric knowledge plays an important role in theories of action [Lif93], our translation will contribute to those theories.

6 Summary

This paper has presented a method of embedding circumscriptive theories in general disjunctive programs. We introduced a translation from circumscriptive theories into general disjunctive programs, and showed that the PZ-minimal Herbrand models of a circumscriptive theory are exactly the stable models of the transformed GDP. The result is also applied to abductive characterization of circumscription. That is, by identifying fixed and varying predicates with abducible predicates, the PZ-minimal Herbrand models of a circumscriptive theory are expressed by the belief sets of an abductive disjunctive program. We finally addressed extensions and applications to other nonmonotonic formalisms.

The result of this paper implies that general disjunctive programs are as expressive as circumscriptive theories under the Herbrand model semantics. Moreover, the positive occurrences of negation as failure increase the expressive power of logic programming as a nonmonotonic formalism, and exploit new applications of logic programming as a knowledge representation tool.

References

[BS85] Bossu, G. and Siegel, P., Saturation, Nonmonotonic Reasoning and the Closed World Assumption, *Artificial Intelligence* 25, 13-63, 1985.

[CEG92] Cadoli, M., Eiter, T. and Gottlob, G., An Efficient Method for Eliminating Varying Predicates from a Circumscription, *Artificial Intelligence* 54, 397-410, 1992.

[dKK89] de Kleer, J. and Konolige, K., Eliminating the Fixed Predicates from a Circumscription, *Artificial Intelligence* 39, 391-398, 1989.

[GL88a] Gelfond, M. and Lifschitz, V., Compiling Circumscriptive Theories into Logic Programs, *Proc. AAAI-88*, 455-459. Extended version in: *Proc. 2nd Int. Workshop on Nonmonotonic Reasoning*, LNAI 346, 74-99, 1988.

[GL88b] Gelfond, M. and Lifschitz, V., The Stable Model Semantics for Logic Programming, *Proc. ICLP'88*, 1070-1080.

[GPP89] Gelfond, M., Przymusinska, H. and Przymusinski, T., On the Relation between Circumscription and Negation as Failure, *Artificial Intelligence* 38, 75-94, 1989.

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

[GLPS94] Gelfond, M., Lifschitz, V., Przymusinska, H. and Schwarz, G., Autoepistemic Logic and Introspective Circumscription, *Proc. TARK'94*, 197-207.

[Gin89] Ginsberg, M. L., A Circumscriptive Theorem Prover, *Artificial Intelligence* 39, 209-230, 1989.

[HIP91] Helft, N., Inoue, K. and Poole, D., Query Answering in Circumscription, *Proc. IJCAI-91*, 426-431.

[Ino92] Inoue, K., Linear Resolution for Consequence Finding, *Artificial Intelligence* 56, 301-353, 1992.

[IS94] Inoue, K. and Sakama, C., On Positive Occurrences of Negation as Failure, *Proc. KR'94*, 293-304.

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

[Kon89] Konolige, K., On the Relation between Autoepistemic Logic and Circumscription, *Proc. IJCAI-89*, 1213-1218.

[Lif85a] Lifschitz, V., Closed World Databases and Circumscription, *Artificial Intelligence* 27, 229-235, 1985.

[Lif85b] Lifschitz, V., Computing Circumscription, *Proc. IJCAI-85*, 121-127.

[Lif88] Lifschitz, V., On the Declarative Semantics of Logic Programs with Negation, in *Foundations of Deductive Databases and Logic Programming* (J. Minker ed.), Morgan Kaufmann, 177-192, 1988.

[Lif89] Lifschitz, V., Between Circumscription and Autoepistemic Logic, *Proc. KR'89*, 235-244.

[LW92] Lifschitz, V. and Woo, T. Y. C., Answer Sets in General Nonmonotonic Reasoning (preliminary report), *Proc. KR'92*, 603-614.

[Lif93] Lifschitz, V., Restricted Monotonicity, *Proc. AAAI-93*, 432-437.

[Lif94a] Lifschitz, V., Minimal Belief and Negation as Failure, *Artificial Intelligence* 70, 53-72, 1994.

[Lif94b] Lifschitz, V., Circumscription, in *Handbook of Logic in Artificial Intelligence and Logic Programming* (D. M. Gabbay, *et al.* eds.), Clarendon Press, 297-352, 1994.

[LT94] Lifschitz, V. and Turner, H., From Disjunctive Programs to Abduction, *Proc. ICLP'94 Workshop on Nonmonotonic Extensions of Logic Programming*, 111-125.

[LS92] Lin F. and Shoham, Y., A Logic of Knowledge and Justified Assumptions, *Artificial Intelligence* 57, 271-289, 1992.

[Lov78] Loveland, D. W., *Automated Theorem Proving: A Logical Basis*, North-Holland, 1978.

[Mc80] McCarthy, J., Circumscription – A Form of Nonmonotonic Reasoning, *Artificial Intelligence* 13, 27-39, 1980.

[Prz88] Przymusinski, T. C., On the Declarative Semantics of Deductive Databases and Logic Programs, in the same source book as [Lif88], 193-216, 1988.

[Prz89] Przymusinski, T., An Algorithm to Compute Circumscription, *Artificial Intelligence* 38, 49-73, 1989.

[Prz91] Przymusinski, T. C., Stable Semantics for Disjunctive Programs, *New Generation Computing* 9, 401-424, 1991.

[Rei82] Reiter, R., Circumscription implies Predicate Completion (sometimes), *Proc. AAAI-82*, 418-420.

[SI93] Sakama, C. and Inoue, K., Relating Disjunctive Logic Programs to Default Theories, *Proc. LPNMR'93*, 266-282.

[YY93] Yuan, L. Y. and You, J-H., Autoepistemic Circumscription and Logic Programming, *J. Automated Reasoning* 10, 143-160, 1993.