Complenteness of Diamond-Resolution in Or-type Knowledge Bases

Sakai, Hiroshi
Department of Computer Engineering Kyushu Institute of Technology

https://doi.org/10.5109/3124

Bulletin of Informatics and Cybernetics, Vol. 24, No. 1 ~ 2, 1990

COMPLETENESS OF DIAMOND-RESOLUTION IN OR-TYPE KNOWLEDGE BASES

By

Hiroshi SAKAI*

Abstract

The framework of the *or-type knowledge base* is proposed to deal with disjunctive information in knowledge bases. Recently, it is an important problem how effectively we use information which may have some incompleteness. In or-type knowledge bases, the predicate symbols are only restricted to or^m, where the superscript m implies the arity, and any disjunctive information is included in a predicate as arguments.

The two systems are naturally defined for the incompleteness of the disjunctive information. The one is a *Box-system*, where the incompleteness does not influence the deduction and the refutation. The other is a *Diamond-system*, where the incompleteness influences them. We have already developed the fixpoint theorem and the completeness of resolution in the Box-system.

In this paper, we show the fixpoint theorem, and the completeness of resolution in the Diamond-system. Furthermore, we present an actual question-answering in an or-type knowledge base by a realized prover.

1. Introduction

A disjunctive information such as "At least, either A or B holds, but it is not known which one holds" is a kind of knowledge, which we especially call an or-type knowledge. The or-type knowledge can be used to deduce some facts. However, it is not easy to deal with the or-type knolwedge in knowledge bases, because the semantics of the Horn logic does not hold for clauses containing or-type knowledge. For example, let us consider the following program P,

$$P = \{C \leftarrow A; B, A; B\},$$

where A, B and C are propositional variables, and A; B is an or-type knowledge. The Herbrand models of the program P are $\{A, C\}$, $\{B, C\}$ and $\{A, B, C\}$. The intersection of all models, which is $\{C\}$, assigns *false* to A; B. Namely, the model intersection property does not hold for the program P. Likewise, the atomic formula C is a logical consequence of P, but $P \cup \{\leftarrow C\}$ does not have any *SLD*-refutation. Because, neither the fact A nor the fact B can be deduced by the fact "A or B".

* Department of Computer Engineering, Kyushu Institute of Technology, Tobata Kitakyushu 804, Japan

The near-Horn Prolog [1] and the disjunctive logic program [2] are proposed to deal with disjunctive information in knowledge bases. Likewise, we have proposed the framework '*or-type knowledge base*' [3]. In our framework, the predicate symbols are only restricted to or^m, where m implies the arity of the predicate. An or-type knowledge base is a finite set of the rules and the facts of the forms:

$$\text{rule: } or^m(t_1, \ldots, t_m) \leftarrow or^n(s_1, \ldots, s_n), \ldots, or^h(r_1, \ldots, r_h)\,.$$

$$\text{fact: } or^k(t_1, \ldots, t_k)\,.$$

For example, we represent an or-type knowledge "*Tom* lives at least in *Tokyo*, *New york* or *Paris*" by an atomic formula,

$$or^3(\text{live}(tom, tokyo), \text{live}(tom, new\_york), \text{live}(tom, paris))\,.$$

The terms live(*tom*, *tokyo*), live(*tom*, *new_york*) and live(*tom*, *paris*) are compound terms. An example of an or-type knowledge base and the real question-answering are presented in the appendix.

We assume the following two properties of the or-type knowledge in the subsequent discussion.

(1) In case a ground atom $or^n(t_1, \ldots, t_n)$ is *true*, at least one $or^1(s)$ is *true* for a term $s \in \{t_1, \ldots, t_n\}$, and $or^{n+k}(t_1, \ldots, t_n, s_1, \ldots, s_k)$ is also *true* for any terms $\{s_1, \ldots, s_k\}$.

(2) In case a ground atom $or^n(t_1, \ldots, t_n)$ is *false*, $or^k(s_1, \ldots, s_k)$ is also *false* for any terms $\{s_1, \ldots, s_k\} \subset \{t_1, \ldots, t_n\}$.

2. Problems and Reviews of Or-type Knowledge Bases

If all predicate symbols in an or-type knowledge base are or^1, then the program is a Horn logic program, and the framework of the or-type knowledge base contains the framework of Horn logic. However, it is necessary to redefine the interpretation, the model, the logical consequence, the deduction and the derivation in or-type knowledge bases. It is also necessary to consider the following two systems for the incompleteness of or-type knowledge.

(1) *Box-system*, where the incompleteness of or-type knowledge does not influence the deduction and refutation,

(2) *Diamond-system*, where the incompleteness of or-type knowledge influences the deduction and refutation.

The words box and diamond are used in the Modal logic, and they imply the modalities '*certainty*' and '*possibility*', respectively. We also use the words box and diamond to express the modalities '*certainty*' and '*possibility*'. It is necessary for us to develop the semantics in each system. We prefix '*BOX-*' and '*DMD-*' to each technical term in Box-system and Diamond-system, respectively.

Let us consider the following or-type knowledge bases S_1 and S_2.

$$S_1 = \{or^1(a) \leftarrow or^2(b, c), or^1(c)\}\,,$$

$$S_2 = \{or^1(a) \leftarrow or^2(b, c), or^2(c, d)\}\,.$$

In S_1, $or^1(c)$ holds and so does $or^2(b, c)$ by the properties of or-type knowledge. The $or^1(a)$ can certainly be deduced, and $S_1 \cup \{\leftarrow or^1(a)\}$ can certainly be refuted (by *BOX*-resolution [3]). Namely, the incompleteness of knowledge does not influence the deduction of $or^1(a)$ and the refutation of $S_1 \cup \{\leftarrow or^1(a)\}$. However in S_2, $or^2(c, d)$ holds and $or^1(c)$ may hold. In case $or^1(c)$ holds, $or^1(a)$ can be deduced and $S_2 \cup \{\leftarrow or^1(a)\}$ can be refuted in S_2. The incompleteness of knowledge influences the deduction and the refutation in S_2.

In [3], we have developed the semantics in the Box-system. Here, we will briefly review the main results in the Box-system. Let S and U_S be an or-type knowledge base and the Herbrand universe of S, respectively. Let F_S be the set of compound terms constructed by functors in S except or^m $(m = 1, 2, \ldots)$ and U_S.

$$B_S = \{or^1(t) | t \in F_S\}$$

is the standard Herbrand base. We defined EB_S as follows;

$$EB_S = \{or^m(t_1, \ldots, t_m) | \{t_1, \ldots, t_m\} \subset F_S, m \text{ is arbitrary}\} ,$$

and called it the *extended Herbrand base of S*. In relation to EB_S, we define the interpretation of S, which assigns *true*, *false* or *undefined* to any atom in EB_S. The *model of S* is the interpretation which assigns *true* to all clauses in S, and the *least model of S* can be defined by the intersection of the restricted models [3] of S. We also defined the *BOX*-derivation as follows. Let goal G be $\leftarrow A_1, A_2, \ldots, A_k$, and rule C be $A \leftarrow B_1, B_2, \ldots, B_q$, where A_i, A and B_j are in the form of $or^m(t_1, \ldots, t_m)$. $ELE(or^m(t_1, \ldots, t_m))$ implies a set $\{or^1(t_1), \ldots, or^1(t_m)\}$, and R is a computation rule. If there exists a substitution θ such that

$$ELE(A\theta) \subset ELE(A_i\theta) \tag{1}$$

for a selected atom A_i by the computation rule R, then we derive a new goal G'

$$G' : \leftarrow (A_1, \ldots, A_{i-1}, B_1, \ldots, B_q, A_{i+1}, \ldots, A_k)\theta .$$

We call G', a *BOX-resolvent of G and C*. We also call (1) and θ, a *BOX-unification* and a *BOX-unifier*, respectively.

We have already shown the fixpoint theorem and the completeness of *BOX*-resolution in the Box-system. In this paper, we show the fixpoint theorem and the completeness of *DMD*-resolution in the Diamond-system.

3. Derivation in the Diamond-system

We first define the derivation in the Diamond-system. The problem in this paper is to establish the model theory and the deduction corresponding to the following derivation.

Definition 1. Let G be $\leftarrow A_1, A_2, \ldots, A_k$, C be $A \leftarrow B_1, B_2, \ldots, B_q$ and R be a computation rule. If there exists a substitution θ such that

$$ELE(A\theta) \cap ELE(A_i\theta) \neq \varnothing \tag{2}$$

for a selected atom A_i by the computation rule R, then we derive new goal G'

$$G'\colon \leftarrow (A_1, \ldots, A_{i-1}, B_1, \ldots, B_q, A_{i+1}, \ldots, A_k)\theta\,.$$

We call G', a *DMD-resolvent of G and C*. We also call (2) and θ, a *DMD-unification* and a *DMD-unifier*, respectively.

Now we present an algorithm to calculate a *DMD*-unifier of two atoms by using the standard unification algorithm [4].

DMD-**unification algorithm** (for two atoms $or^k(t_1, \ldots, t_k)$ and $or^q(s_1, \ldots, s_q)$)

1. Let i be 1.
2. If $i > k$ then stop. $or^k(t_1, \ldots, t_k)$ and $or^q(s_1, \ldots, s_q)$ are not *DMD*-unifiable.
3. Take a term $t_i \in \{t_1, \ldots, t_m\}$.
4. Successively unify the following pairs of atoms;

$$\begin{aligned} &or^q(t_i, \_\_, \_\_, \ldots, \_\_) \quad \text{and} \quad or^q(s_1, \ldots, s_q)\,, \\ &or^q(\_\_, t_i, \_\_, \ldots, \_\_) \quad \text{and} \quad or^q(s_1, \ldots, s_q)\,, \\ &or^q(\_\_, \_\_, \ldots, \_\_, t_i) \quad \text{and} \quad or^q(s_1, \ldots, s_q) \end{aligned}$$

   by using the unification algorithm [4], where ,__, implies an anonymous variable. If a unification succeeds and a unifier θ is calculated, then stop. $or^k(t_1, \ldots, t_k)$ and $or^q(s_1, \ldots, s_q)$ is *DMD*-unifiable. θ is a *DMD*-unifier. Otherwise, let i be $i + 1$, and goto 2.

A *DMD*-unifier is a *most general unifier* (mgu) for two atoms in step 4, because it is calculated by the unification algorithm [4]. In the unification algorithm, an mgu of two atoms is uniquely decided except variants. However, in the *DMD*-unification algorithm, the mgu of two atoms except variants may not be uniquely decided. In the *SLD*-refutation, a backtracking occurs due to the inappropriate selection of input clauses. However, in the *DMD*-refutation, a backtracking may occur due to the inappropriate selection of *DMD*-unifiers.

Let S be an or-type knowledge base, G be a goal and R be a computation rule. A *DMD-derivation of* $S \cup \{G\}$ *via* R and a *DMD-refutation of* $S \cup \{G\}$ *via* R are the same as in [4]. An *unrestricted DMD-refutation* is a *DMD*-refutation whose *DMD*-unifier may not be an mgu. According to the *DMD*-unification algorithm and the *DMD*-refutation, we can conclude the following two lemmas.

Lemma 1. (Mgu lemma) *Let S be an or-type knowledge base and G be a goal. Suppose that $S \cup \{G\}$ has an unrestricted DMD-refutation. Then $S \cup \{G\}$ has a DMD-refutation of the same length.*

Proof. By definition of the *DMD*-unification algorithm, Lemma 8.1 in [4] is directly applicable to this proof. ■

Lemma 2. (Lifting lemma) *Let S be an or-type knowledge base, G be a goal and θ be a substitution. If there exists a DMD-refutation of $S \cup \{G\theta\}$, then there exists a DMD-refutation of $S \cup \{G\}$ of the same length.*

Proof. It can be similarly proved by Lemma 8.2 in [4] and Lemma 1. ■

Definition 2. Let S be an or-type knowledge base. The *DMD-success set of* S is the set of all $or^h(t_1, \ldots, t_h)$ in EB_S such that $S \cup \{\leftarrow or^h(t_1, \ldots, t_h)\}$ has a *DMD*-refutation.

4. Model theory in the Diamond-system

In order to discuss the model theory of the Diamond-system, we first consider the following or-type knowledge base S_3.

$$S_3 = \{or^1(a) \leftarrow or^1(b), or^2(b, c)\},$$

In S_3, the least model in the Box-system, denoted by $\Box(S_3)$, is $\{or^2(b, c)\}$. $S_3 \cup \{\leftarrow or^2(a, e)\}$ has a *DMD*-refutation. However, the truth value assigned to $or^2(a, e)$ is *false* by the interpretation $\bigcup_{A \in \Box(S_3)} ELE(A) (= \{or^1(b), or^1(c)\} \subset B_{S_3})$. On the other hand, let $or^k(t_1, \dots, t_k)$ be any element in $(\bigcup_{A \in \Box(S)} ELE(A))^*$ for an or-type knowledge base S, where

$$(A)^* = \{B \in EB_S | \text{the truth value assigned to } B \text{ by the interpretation } A \text{ is } true\}.$$

Then, there exists at least one $or^1(t)$ in $ELE(or^k(t_1, \dots, t_k))$ such that the truth value of $or^1(t)$ is *true* by the interpretation $\bigcup_{A \in \Box(S)} ELE(A)$, i.e., $or^1(t) \in ELE(A)$ for some $A \in \Box(S)$. Since $A \in \Box(S)$, $S \cup \{\leftarrow A\}$ has a *BOX*-refutation, and $ELE(or^k(t_1, \dots, t_k)) \cap ELE(A) \neq \emptyset$, so $S \cup \{\leftarrow or^k(t_1, \dots, t_k)\}$ has a *DMD*-refutation. Namely,

$$(\bigcup_{A \in \Box(S)} ELE(A))^* \subset DMD\text{-success set of } S.$$

Definition 3. Let S be an or-type knowledge base. A *DMD-interpretation I of S* is an Herbrand interpretation of S which satisfies the following condition.

Condition: Let A be a ground atom, which is a fact or a head in a ground instance of a rule. If the truth value assigned to A by the interpretation I is *true*, then the truth value of each element in $ELE(A)$ is also *true*, i.e., $ELE(A) \subset I$.

For example, the *DMD*-interpretations of a fact $or^2(b, c)$ are $\emptyset$ and $\{or^1(b), or^1(c)\}$. An interpretation $\{or^1(b)\}$ assigns *true* to $or^2(b, c)$, but it is not a *DMD*-interpretation. Let L be the set of all *DMD*-interpretations which assign *true* to the following rule

$$or^m(t_1, \dots, t_m) \leftarrow or^n(s_1, \dots, s_n), \dots, or^h(r_1, \dots, r_h).$$

Furthermore, let N be the set of all *DMD*-interpretations which assign *true* to the following all rules.

$$or^1(t_1) \leftarrow or^n(s_1, \dots, s_n), \dots, or^h(r_1, \dots, r_h).$$
$$\vdots$$
$$or^1(t_m) \leftarrow or^n(s_1, \dots, s_n), \dots, or^h(r_1, \dots, r_h).$$

For the set of *DMD*-interpretations L and N, $L \subset N$ holds.

For any atom A in EB_S and any *DMD*-interpretation I, if $ELE(A) \cap I \neq \emptyset$, then the truth value of the atom A is *true*. Otherwise, the truth value of the atom A is *false*. In this way, a *DMD*-interpretation assigns *true* or *false* to any atom in EB_S.

Proposition 3. (Model intersection property) Let S be an or-type knowledge base, M be a set of all *DMD*-interpretations which are models of S. Then, $\bigcap_{I \in M} I$ is also a *DMD*-interpretation and a model of S.

Proof. Since B_S is a model and a *DMD*-interpretation of S, M is always non empty set. If an empty set is a model of S, namely S is a set of rules, then $\bigcap_{I \in M} I$ is also an empty set and a model of S. Now we consider the case that S contains at least a fact. First we consider facts. Let $or^m(t_1, \ldots, t_m)$ be any fact in S. By definition of the DMD-interpretation,

$$ELE(or^m(t_1, \ldots, t_m)) = \{or^1(t_1), \ldots, or^1(t_m)\} \subset I \text{ for any } I \in M\,,$$

and therefore,

$$ELE(or^m(t_1, \ldots, t_m)) \subset \bigcap_{I \in M} I\,.$$

The interpretation $\bigcap_{I \in M} I$ is a DMD-interpretation and a model of the fact $or^m(t_1, \ldots, t_m)$. Secondly we consider rules. Let $or^h(t_1, \ldots, t_h) \leftarrow B_1, \ldots, B_k$ be any rule in S. We show that if the interpretation $\bigcap_{I \in M} I$ assigns *true* to a ground instance of the body $(B_1, \ldots, B_k)\theta$, then $\bigcap_{I \in M} I$ assigns *true* to the atom $or^h(t_1, \ldots, t_h)\theta$ and $\bigcap_{I \in M} I$ contains each element in $ELE(or^h(t_1, \ldots, t_h)\theta)$. Since the truth value assigned to $(B_1, \ldots, B_k)\theta$ by $\bigcap_{I \in M} I$ is *true*, the truth value assigned to $ELE(or^h(t_1, \ldots, t_h)\theta)$ by $\bigcap_{I \in M} I$ is *true*. Namely,

$$ELE(or^h(t_1, \ldots, t_h)\theta) = \{or^1(t_1)\theta, \ldots, or^1(t_h)\theta\} \subset I \text{ for any } I \in M\,,$$

which implies

$$ELE(or^h(t_1, \ldots, t_h)\theta) \subset \bigcap_{I \in M} I\,. \qquad \blacksquare$$

We denote $\bigcap_{I \in M} I$ by $\diamondsuit(S)$, and call it the *least model of S in the Diamond-system.*

DEFINITION 4. Let S be an or-type knowledge base. An atom $A \in EB_S$ is a *DMD-logical consequence of* S, if for every DMD-interpretation I of S, I is a model for S implies that I is a model for the atom A.

PROPOSITION 4. *Let S be an or-type knowledge base.*

$$(\diamondsuit(S))^* = \{A \in EB_S \mid A \text{ is a DMD-logical consequence of } S\}\,.$$

PROOF. $(\Rightarrow)$ Let $or^h(t_1, \ldots, t_h)$ be an element in $(\diamondsuit(S))^*$, then the truth value assigned to $or^h(t_1, \ldots, t_h)$ by the interpretation $\diamondsuit(S)$ is *true*. Thus the truth value of $or^h(t_1, \ldots, t_h)$ is *true* by any DMD-interpretation which is also a model of S. Namely, $or^h(t_1, \ldots, t_h)$ is a DMD-logical consequence of S.
$(\Leftarrow)$ Let $or^h(t_1, \ldots, t_h) \in EB_S$ be any DMD-logical consequence of S. By definition, the truth value of $or^h(t_1, \ldots, t_h)$ is *true* by any DMD-interpretation I which is a model of S. If no $or^1(t)$ in $ELE(or^h(t_1, \ldots, t_h))$ satisfies $or^1(t) \in I$ for any DMD-interpretation I which is a model of S, then $or^1(t) \notin \diamondsuit(S)$ for any $or^1(t)$ in $ELE(or^h(t_1, \ldots, t_h))$. Thus the truth value assigned to $or^h(t_1, \ldots, t_h)$ by $\diamondsuit(S)$ is *false*, and it contradicts that $or^h(t_1, \ldots, t_h)$ is a DMD-logical consequence. Therefore, at least one element $or^1(t)$ in $ELE(or^h(t_1, \ldots, t_h))$ satisfies $or^1(t) \in I$ for any DMD-interpretation I which is a model of S. Hence the truth value assigned to $or^h(t_1, \ldots, t_h)$ by the interpretation $\diamondsuit(S)$ is *true*. $\blacksquare$

5. Deduction in the Diamond-system

We define a mapping $\diamondsuit T_S$, then we show the fixpoint theorem and the completeness of DMD-resolution in the Diamond-system.

Definition 5. Let S be an or-type knowledge base. We define mappings $\Diamond T_S$ and $\Diamond R_S$ as follows;

$\Diamond T_S$: $2^{B_S} \to 2^{B_S}$, $\Diamond R_S$: $2^{B_S} \to 2^{EB_S}$,
$\Diamond R_S(I) = \{A \in EB_S|$
(1) $A \leftarrow B_1, B_2, \ldots, B_k$ is a gound instance of a clause in S,
(2) $ELE(B_i) \cap I \neq \varnothing$ holds for any B_i $(1 \leqq i \leqq k)\}$,
$\Diamond T_S(I) = \bigcup_{A \in \Diamond R_S(I)} ELE(A)$.

2^{B_S}, which is the set of all Herbrand interpretations of S, is the complete lattice under the partial order of set inclusion. Clearly, the mapping $\Diamond T_S$ is continuous, so $\Diamond T_S$ has the least fixpoint $lfp(\Diamond T_S)$, which corresponds to $\Diamond T_S \uparrow \omega$[4].

Proposition 5. *Let S be an or-type knowledge base. Then I is a DMD-interpretation and a model of S if and only if $\Diamond T_S(I) \subset I$.*

Proof. I is a *DMD*-interpretation and a model for S iff for each ground instance $A \leftarrow B_1, B_2, \ldots, B_k$ of each clause in S, $ELE(B_i\theta) \cap I \neq \varnothing$ for any i $(1 \leqq i \leqq k)$ implies $ELE(A) \subset I$ iff $\Diamond T_S(I) \subset I$. ∎

Theorem 6. (Fixpoint theorem) *Let S be an or-type knowledge base. Then,*

$$\Diamond(S) = lfp(\Diamond T_S) = \Diamond T_S \uparrow \omega .$$

Proof.

$$\begin{aligned} &\Diamond(S) \\ &= glb\{I \in 2^{B_S} | I \text{ is a } DMD\text{-interpretation and a model of } S\} \\ &= glb\{I \in 2^{B_S} | \Diamond T_S(I) \subset I\}, \quad \text{by Proposition 5} \\ &= lfp(\Diamond T_S), \quad \text{by Proposition 5.1 in [4]} \\ &= \Diamond T_S \uparrow \omega, \quad \text{by Proposition 5.4 in [4] and the continuity of } \Diamond T_S . \end{aligned}$$ ∎

Theorem 7. (Soundness of *DMD*-resolution) *Let S be an or-type knowledge base. If $S \cup \{G\}$ has a DMD-refutation with answer substitution θ*[4], *then $\forall(G\theta)$ is a DMD-logical consequences of S.*

Proof. Let G be a goal $\leftarrow A_1, A_2, \ldots, A_k$ and $\theta_1, \ldots, \theta_n$ be the sequence of mgu's used in a *DMD*-refutation of $S \cup \{G\}$ via a computation rule R. We show $\forall(G\theta)$ is a *DMD*-logical consequence of S by induction of the length of the *DMD*-refutation. Suppose first that $n = 1$. This means that G is a goal of the form $\leftarrow A_1$, and there exists a fact A and substitution θ_1 such that $ELE(A_1\theta_1) \cap ELE(A\theta_1) \neq \varnothing$. Since each element in $ELE(A_1\theta_1) \cap ELE(A\theta_1)$ is a *DMD*-logical consequence of S, $\forall(A_1\theta_1)$ is a *DMD*-logical consequence of S. Now suppose that the results holds for $n-1$. Suppose $\theta_1, \ldots, \theta_n$ is the sequence of mgu's used in a *DMD*-refutation of $S \cup \{G\}$ with length n. Let $A \leftarrow B_1, \ldots, B_q$ be the first input clause and A_m the selected atom of G, which satisfies $ELE(A_m\theta_1) \cap ELE(A\theta_1) \neq \varnothing$. By the induction hypothesis,

$$\forall((A_1, \ldots, A_{m-1}, B_1, \ldots, B_q, A_{m+1}, \ldots, A_k)\theta_1 \ldots \theta_n)$$

is a *DMD*-logical consequence of S. Thus, if $q > 0$, then $\forall((B_1, \ldots, B_q)\theta_1 \ldots \theta_n)$ is a *DMD*-logical consequence of S. Consequently, each element in $ELE(A\theta_1 \ldots \theta_n)$ is a

DMD-logical consequence of S. $\forall(A_m\theta_1\dots\theta_n)$ is a *DMD*-logical consequence of S, because $ELE(A_m\theta_1\dots\theta_n)\cap ELE(A\theta_1\dots\theta_n)\neq\varnothing$. Hence $\forall((A_1, A_2, \dots, A_k)\theta_1\dots\theta_n)$ is a *DMD*-logical consequence of S. ■

THEOREM 8. *Let S be an or-type knowledge base. Then,*

$$(\Diamond(S))^* = DMD\text{-}success\ set\ of\ S\,.$$

PROOF. Let $A(=or^m(t_1,\dots,t_m))$ be an element in EB_S and $S\cup\{\leftarrow A\}$ has a *DMD*-refutation via some computation rule. By Theorem 7, A is a *DMD*-logical consequence of S. Thus, by Proposition 4, A is in $(\Diamond(S))^*$. Now, we show that $(\Diamond(S))^*$ is contained in the *DMD*-success set of S. Suppose $A\in(\Diamond(S))^*$. By Theorem 6, $A\in(\Diamond T_S\uparrow n)^*$ for some $n\in\omega$. We prove by induction on n that $A\in(\Diamond T_S\uparrow n)^*$ implies $S\cup\{\leftarrow A\}$ has a *DMD*-refutation. Suppose first that $n=1$. $A\in(\Diamond T_S\uparrow 1)^*$ means that $ELE(A)\cap ELE(B)\neq\varnothing$ for some fact B in S. Namely, $S\cup\{\leftarrow A\}$ has a *DMD*-refutation. Suppose that the result holds for $n-1$. Let $A\in(\Diamond T_S\uparrow n)^*$. By definition of $\Diamond T_S$, there exists a ground instance of a clause $B\leftarrow B_1,\dots,B_k$ such that

$$ELE(A)\cap ELE(B\theta)\neq\varnothing \text{ and } \{B_1\theta,\dots,B_k\theta\}\subset(\Diamond T_S\uparrow n-1)^* \text{ for some } \theta\,.$$

By the induction hypothesis, $S\cup\{\leftarrow B_i\theta\}$ has a *DMD*-refutation for each $i=1,\dots,k$. Because each $B_i\theta$ is ground, these *DMD*-refutations can be combined into a *DMD*-refutation of $S\cup\{\leftarrow(B_1,\dots,B_k)\theta\}$. Thus $S\cup\{\leftarrow A\theta\}$ has an unrestricted *DMD*-refutation. The *DMD*-refutation of $S\cup\{\leftarrow A\}$ can be obtained by the mgu lemma. ■

LEMMA 9. *Let S be an or-type knowledge base and A be an atom. If $\forall(A)$ is a DMD-logical consequence of S, then there exists a DMD-refutation of $S\cup\{\leftarrow A\}$ with the identity substitution as the computed answer substitution* [4].

PROOF. It can be similarly proved by Lemma 8.5 in [4]. ■

THEOREM 10. (Completeness of *DMD*-resolution) *Let S be an or-type knowledge base. If $\forall(G\theta)$ is a DMD-logical consequence, then there exists a computation rule R, a computed answer substitution σ for $S\cup\{G\}$ and a substitution γ such that $\theta=\sigma\gamma$.*

PROOF. Let G be a goal $\leftarrow A_1, A_2, \dots, A_k$. Since $\forall(G\theta)$ is a *DMD*-logical consequence of S, $\forall(A_i\theta)$ is a *DMD*-logical consequence of S for $i=1,\dots,k$. By Lemma 9 there exists a *DMD*-refutation of $S\cup\{\leftarrow A_i\theta\}$ with the identity substitution as the computed answer substitution for $i=1,\dots,k$. We can combine these *DMD*-refutations into a *DMD*-refutation of $S\cup\{G\theta\}$ with identity substitution. Suppose the sequence of mgu's of the *DMD*-refutation of $S\cup\{G\theta\}$ is $\theta_1,\dots,\theta_n$. Then, $G\theta\theta_1\theta_2\dots\theta_n=G\theta$, because $\theta_1\theta_2\dots\theta_n$ is an identity substitution. By the lifting lemma, there exists a *DMD*-refutation of $S\cup\{G\}$ with mgu's $\theta_1', \theta_2', \dots, \theta_n'$ such that $\theta\theta_1\theta_2\dots\theta_n=\theta_1'\theta_2'\dots\theta_n'\gamma'$ for some substitution γ'. Let σ be $\theta_1'\theta_2'\dots\theta_n'$ restricted to the variables in G. Then $\theta=\sigma\gamma$, where γ is an appropriate restriction of γ'. ■

6. Concluding Remarks

We have proposed the framework of the or-type knowledge base in order to deal with or-type knowledge. We have already shown the fixpoint theorem and the completeness of *BOX*-resolution in the Box-system [3]. In this paper, we have mainly discussed the model theory, the deduction and the refutation in the Diamond-system. We have shown the fixpoint theorem and the completeness of DMD-resolution in the Diamond-system, which are the foundations of the or-type knowledge bases. We have also realized a prover by prolog, whose derivation depends on the BOX-derivation and the DMD-derivation.

However, another derivation which uses both BOX-unification and DMD-unification may also be important. In such a derivation, the least model in the Box-system is dynamically modified whenever the DMD-unification is used. The model theory and the deduction for the new derivation are another theoretical issues. The application of our theory to real expert systems is also an important theme.

Acknowledgments

The author is very grateful to Prof. S. Arikawa for his continual encouragement. This work is partially supported by the Grant-in-Aid for Encouragement of Young Scientists (No. 01780049), The Ministry of Education, Science and Culture of Japan.

References

[1] LOVELAND, D. W.: *Near-Horn prolog*, Proc. 4th Int. Conf. on Logic Programming, (1987), 456–469.

[2] LOBO, J., MINKER, J. and RAJASEKAR, A.: *Extending the semantics of logic programs to disjunctive logic programs*, Proc. 6th Int. Conf. on Logic Programming, (1989), 255–267.

[3] SAKAI, H.: *Inference methods and semantics on or-type knowledge bases*, Lecture Notes in Artificial Intelligence, Springer-Verlag, **383** (1989), 136–155.

[4] LLOYD, J. W.: *Foundations of logic programming*, Springer-Verlag, (1984).

Received September 14, 1989
Revised October 12, 1989
Communicated by S. Arikawa

Appendix

Let us show the real question-answering in an or-type knowledge base. The knowledge base, which deals with a knowledge of sessions in an annual conference, is as follows;

rule 1: If a man attends at least session a, b or c, then he is in the first building.
rule 2: If a man attends at least session d or e, then he is in the second building.
rule 3: If X and Y are in the same building, then X meets Y, and Y meets X.
fact 1: tanaka attends at least session a or b.
fact 2: suzuki attends at least session b or c.
fact 3: yamada attends at least session a or d.

```
/******************** Or-type knowledge base **********************/
or1(place(X,first)):-or3(attend(X,session(a)),attend(X,session(b)),
                         attend(X,session(c))).
or1(place(X,second)):-or2(attend(X,session(d)),attend(X,session(e))).
or1(meet(X,Y)):-or1(place(X,X_place)),or1(place(Y,X_place)),or1(not(X,Y)).
or2(attend(tanaka,session(a)),attend(tanaka,session(b))).
or2(attend(suzuki,session(b)),attend(suzuki,session(c))).
or2(attend(yamada,session(a)),attend(yamada,session(d))).

/************************** Execution *****************************/
| ?-set.                            /* initialization */
yes
| ?-box(or1(meet(tanaka,N))).       /* ask a man who certainly meets tanaka */
N     = suzuki;
no
| ?-dmd(or1(meet(tanaka,N))).       /* ask a man who may meet tanaka except
N     = yamada;                     suzuki */
no
| ?-set.                            /* initialization */
yes
| ?-dmd(or1(meet(tanaka,N))).       /* ask a man who may meet tanaka */
N     = suzuki;
N     = yamada;
no
| ?-set.
yes
| ?-box(or1(place(yamada,N))).      /* ask a place where yamada certainly is */
no
| ?-dmd(or1(place(yamada,N))).      /* ask a place where yamada may be */
N     = first;
N     = second;
no
| ?-set.
yes
```

```
| ?-mix(box(or1(place(X,first))),dmd(or1(attend(X,Y)))).
X    = tanaka,                    /* ask a man who certainly is in first
Y    = session(a);                building and a session the man may
X    = tanaka,                    attend */
Y    = session(b);
X    = suzuki,
Y    = session(b);
X    = suzuki,
Y    = session(c);
no
```

(∗) Comments were added to logging data after execution.