Complenteness of Diamond-Resolution in Or-type Knowledge Bases

Sakai, Hiroshi
Department of Computer Engineering Kyushu Institute of Technology

http://hdl.handle.net/2324/3124
Completeness of Diamond-Resolution in Or-type Knowledge Bases

Hiroshi Sakai

October 13, 1989

Research Institute of Fundamental Information Science
Kyushu University 33
Fukuoka 812, Japan
E-mail: sakai@comp.kyutech.ac.jp Phone: (093)-871-1931 ex.464
Completeness of Diamond-Resolution
in Or-type Knowledge Bases

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

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, 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

The 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 the 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 knowledge 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 $\{ A, C \}, \{ B, C \} \{ A, B, C \}$. 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

This work is partially supported by Grant-in-Aid for Encouragement of Young Scientists(No.01780049), The Ministry of Education, Science and Culture of Japan.
logical consequence of \( P \), but \( P \cup \{ \leftarrow C \} \) does not have an SLD-refutation. Because, neither the fact \( A \) nor the fact \( B \) can be deduced by the fact "\( A \) or \( B \)".

The near-Horn Prolog[1] and the disjunctive logic program[2] are proposed to deal with or-type knowledge in knowledge bases. Likewise, we have proposed the framework 'or-type knowledge base'[3]. In our framework, the predicate symbols are only restricted to \( \text{orm} \), 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:

- rule: \( \text{orm}(t_1, \ldots, t_m) \leftarrow \text{orm}(s_1, \ldots, s_n), \ldots, \text{orh}(r_1, \ldots, r_h). \)
- fact: \( \text{ork}(t_1, \ldots, t_k). \)

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

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

The terms \( \text{live}(\text{tom}, \text{tokyo}), \text{live}(\text{tom}, \text{new\_york}) \) and \( \text{live}(\text{tom}, \text{paris}) \) 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 \( \text{orm}(t_1, \ldots, t_n) \) is true, at least one \( \text{orl}(s) \) is true for a term \( s \in \{t_1, \ldots, t_n\} \), and \( \text{orm} + k(t_1, \ldots, t_n, s_1, \ldots, s_k) \) is also true for any terms \( \{s_1, \ldots, s_k\} \).
2. In case a ground atom \( \text{orm}(t_1, \ldots, t_n) \) is false, \( \text{ork}(s_1, \ldots, s_k) \) is also false for any terms \( \{s_1, \ldots, s_k\} \subseteq \{t_1, \ldots, t_n\} \).

2. Problems and reviews of or-type knowledge bases

If all predicate symbols in an or-type knowledge base are \( \text{orl} \), 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 \( S_2 \).

\[ S_1 = \{ \text{orl}(a) \leftarrow \text{orl}(b, c), \text{orl}(c) \}, \]
\[ S_2 = \{ \text{orl}(a) \leftarrow \text{orl}(b, c), \text{orl}(c, d) \}. \]

In \( S_1 \), \( \text{orl}(c) \) holds and so does \( \text{orl}(b, c) \) by the properties of or-type knowledge. The \( \text{orl}(a) \) can certainly be deduced, and \( S_1 \cup \{ \leftarrow \text{orl}(a) \} \) can certainly be refuted (by BOX-resolution [3]). Namely, the incompleteness of knowledge does not influence
the deduction of \( \text{or}^1(a) \) and the refutation of \( S_1 \cup \{ \leftarrow \text{or}^1(a) \} \). However, in \( S_2 \), \( \text{or}^2(c, d) \) holds and \( \text{or}^1(c) \) may hold. In case \( \text{or}^1(c) \) holds, \( \text{or}^1(a) \) can be deduced and \( S_2 \cup \{ \leftarrow \text{or}^1(a) \} \) 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 \( U_5 \) be an or-type knowledge base and the Herbrand universe of \( S \), respectively. Let \( F_5 \) be the set of compound terms constructed by functors in \( S \) except \( \text{orm}(m = 1, 2, \ldots) \) and \( U_5 \).

\[
B_5 = \{ \text{or}^1(t) \mid t \in F_5 \}
\]

is the standard Herbrand base. We defined \( EB_5 \) as follows;

\[
EB_5 = \{ \text{orm}(t_1, \ldots, t_m) \mid \{t_1, \ldots, t_m\} \subset F_5, m \text{ is arbitrary } \},
\]

and called it the extended Herbrand base of \( S \). In relation to \( EB_5 \), we define the interpretation of \( S \), which assigns true, false or undefined to any atom in \( EB_5 \). 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 \( \text{orm}(t_1, \ldots, t_m) \).

\[
ELE(\text{orm}(t_1, \ldots, t_m)) \text{ implies } \{ \text{or}^1(t_1), \ldots, \text{or}^1(t_m) \}, \text{ and } R \text{ is a computation rule. If there exists a substitution } \theta \text{ such that}
\]

\[
(1) \quad ELE(A) \subset ELE(A) \theta,
\]

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 \( \theta \), 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 \( \theta \) such that

\[
(2) \quad ELE(A) \cap ELE(A) \theta \neq \emptyset
\]

for a selected atom \( A_i \) by the computation rule \( R \), then we derive 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 DMD-resolvent of \( G \) and \( C \). We also call (2) and \( \theta \), 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 \( \text{or}^k(t_1, \ldots, t_k) \) and \( \text{or}^q(s_1, \ldots, s_q) \) )**

1. Let \( i \) be 1.
2. If \(i > k\) then stop. \(\text{or}^k(t_1, ..., t_k)\) and \(\text{orq}(s_1, ..., s_q)\) are not DMD-unifiable.

3. Take a term \(t_i \in \{t_1, ..., t_m\}\).

4. Successively unify the following pairs of atoms:
   \[
   \begin{align*}
   \text{orq}(t_i, ..., s_1, ..., s_q), \\
   \text{orq}(s_1, ..., s_q), \\
   \end{align*}
   \]
   by using the unification algorithm[4], where \(\_\) implies an anonymous variable. If a unification succeeds and a unifier \(\theta\) is calculated, then stop.

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 \(SU(G)\) via \(R\) and a DMD-refutation of \(SU(G)\) 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 \(SU(G)\) has an unrestricted DMD-refutation. Then \(SU(G)\) 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, \(G\) a goal and \(\theta\) a substitution. If there exists a DMD-refutation of \(SU(G\theta)\), then there exists a DMD-refutation of \(SU(G)\) 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 atom \(\text{or}^h(t_1, ..., t_h)\) in \(EB_S\) such that \(SU(\text{or}^h(t_1, ..., t_h))\) 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 bases \(S_3\).

\[
S_3 = \{ \text{or}^3(a) \leftarrow \text{or}^1(b), \text{or}^2(b, c), \}
\]
In $S_3$, the least model in the Box-system, denoted by $\square(S_3)$, is $\{\text{or}^2(b,c)\}$. $S_3 \cup \{\leftarrow \text{or}^2(a,e)\}$ has a DMD-refutation. However, the truth value assigned to \text{or}^2(a,e) is \textit{false} by the interpretation $\cup_{A \in \square(S_3)} \text{ELE}(A) = \{\text{or}^1(b), \text{or}^1(c)\} \subset B_5$. On the other hand, let $\text{or}^k(t_1, \ldots, t_k)$ be any element in $(\cup_{A \in \square(S_3)} \text{ELE}(A))^*$ for an or-type knowledge $S$, where

$$(A)^* = \{ B \in EB_5 \mid \text{the truth value assigned to } B \text{ by the interpretation } A \text{ is true} \}.$$  

Then, there exists at least one $\text{or}^1(t)$ in $\text{ELE}(\text{or}^k(t_1, \ldots, t_k))$ such that the truth value of $\text{or}^1(t)$ is \textit{true} by the interpretation $\cup_{A \in \square(S_3)} \text{ELE}(A)$, i.e., $\text{or}^1(t) \in \text{ELE}(A)$ for some $A \in \square(S)$. Since $A \in \square(S)$, $S \cup \{\leftarrow A\}$ has a BOX-refutation, and $\text{ELE}(\text{or}^k(t_1, \ldots, t_k)) \cap \text{ELE}(A) \neq \emptyset$, so $S \cup \{\leftarrow \text{or}^k(t_1, \ldots, t_k)\}$ has a DMD-refutation. Namely,

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

**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 \textit{true}, then the truth value of each element in $\text{ELE}(A)$ is also \textit{true}, i.e., $\text{ELE}(A) \subseteq I$.

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

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

Furthermore, let $N$ be the set of all DMD-interpretations which assign \textit{true} to the following all rules.

$$\text{or}^1(t_1) \leftarrow \text{or}^m(s_1, \ldots, s_n), \ldots, \text{or}^h(r_1, \ldots, r_h).$$

$$\vdots$$

$$\text{or}^1(t_m) \leftarrow \text{or}^m(s_1, \ldots, s_n), \ldots, \text{or}^h(r_1, \ldots, r_h).$$

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

For any atom $A$ in $EB_5$ and any DMD-interpretation $I$, if $\text{ELE}(A) \cap I \neq \emptyset$, then the truth value of the atom $A$ is \textit{true}. Otherwise, the truth value of the atom $A$ is \textit{false}. In this way, a DMD-interpretation assigns \textit{true} or \textit{false} to any atom in $EB_5$.

**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, $\cap_{I \in M} I$ is also a DMD-interpretation and a model of $S$.

Proof. Since $B_5$ is a model and a DMD-interpretation of $S$, $M$ is always nonempty set. If an empty set is a model of $S$, namely $S$ is a set of rules, then $\cap_{I \in M} I$ 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 $\text{orm}(t_1, \ldots, t_m)$ be any fact in $S$. By definition of the DMD-interpretation,

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

and therefore,

$$\text{ELE}(\text{orm}(t_1, \ldots, t_m)) \subseteq \cap_{I \in M} I.$$  

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

$$ELE(orh(t_1, ..., t_h)\theta) = \{orl(t_1)\theta, ..., orl(t_h)\theta\} \subseteq I$$

for any $I \in M$, which implies 

$$ELE(orh(t_1, ..., t_h)\theta) \subseteq \cap_{i\in M} I.$$

We denote $\cap_{i\in M} I$ by $\Diamond (S)$, 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. 

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

Proof.

( $\Rightarrow$ ) Let $orh(t_1, ..., t_h)$ be an element in $(\Diamond (S))^*$, then the truth value assigned to $orh(t_1, ..., t_h)$ by the interpretation $\Diamond (S)$ is true. Thus the truth value of $orh(t_1, ..., t_h)$ is true by any DMD-interpretation which is also a model of $S$. Namely, $orh(t_1, ..., t_h)$ is a DMD-logical consequence of $S$.

( $\Leftarrow$ ) Let $orh(t_1, ..., t_h) \in EB_S$ be any DMD-logical consequence of $S$. By definition, the truth value of $orh(t_1, ..., t_h)$ is true by any DMD-interpretation $I$ which is a model of $S$. If no $orl(t) \in ELE(orh(t_1, ..., t_h))$ satisfies $orl(t) \subseteq I$ for any DMD-interpretation $I$ which is a model of $S$, then $orl(t) \subseteq \Diamond (S)$ for any $orl(t) \in ELE(orh(t_1, ..., t_h))$.

Thus the truth value assigned to $orh(t_1, ..., t_h)$ by $\Diamond (S)$ is false, and it contradicts that $orh(t_1, ..., t_h)$ is a DMD-logical consequence. Therefore, at least one element $orl(t)$ in $ELE(orh(t_1, ..., t_h))$ satisfies $orl(t) \subseteq I$ for any DMD-interpretation $I$ which is a model of $S$. Hence the truth value assigned to $orh(t_1, ..., t_h)$ by the interpretation $\Diamond (S)$ is true.

### 5. Deduction in the Diamond-system

We define a mapping $\Diamond T_5$, and 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 $\Diamond T_5$ and $\Diamond R_5$ as follows;

$\Diamond T_5 : 2B_S \rightarrow 2B_S, \quad \Diamond R_5 : 2B_S \rightarrow 2EB_S$.

$\Diamond R_5(I) = \{A \in EB_S | \}

- (1) $A \leftarrow B_1, B_2, ..., B_q$ is a ground instance of a clause in $S$,
- (2) $ELE(B_i) \cap I \neq \emptyset$ holds for any $B_i (1 \leq i \leq q)$.

$\Diamond T_5(I) = \cup \{A \in \Diamond R_5(I) \mid ELE(A) \}.$

$2B_S$, which is the set of all Herbrand interpretations of $S$, is the complete lattice under the partial order of set inclusion. Clearly, the mapping $\Diamond T_5$ is continuous, so $\Diamond T_5$ has the least fixpoint $ftp(\Diamond T_5)$, which corresponds to $\Diamond T_5 \uparrow \omega[4]$. 


Proposition 5. Let S be an or-type knowledge base. Then l is a DMD-interpretation and a model of S if and only if \( \Diamond T_5(l) \subseteq I \).

Proof. l 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, \( ELE(B_i \theta) \cap I \neq \emptyset \) for any \( i(1 \leq i \leq k) \) implies \( ELE(A) \subseteq I \) iff \( \Diamond T_5(l) \subseteq I \).

Theorem 6. (Fixpoint theorem) Let S be an or-type knowledge base. Then,
\[ (\Diamond(S) = \text{ifp}(\Diamond T_5) = \Diamond T_5 \uparrow \omega. \]

Proof.
\( \Diamond(S) \)
\[ = \text{glb}\{ I \in 2^S | I \text{ is a DMD-interpretation and a model of S} \} \]
\[ = \text{glb}\{ I \in 2^S | \Diamond T_5(I) \subseteq I \}, \text{ by Proposition 5} \]
\[ = \text{ifp}(\Diamond T_5), \text{ by Proposition 5.1 in [4]} \]
\[ = \Diamond T_5 \uparrow \omega, \text{ by Proposition 5.4 in [4] and the continuity of } \Diamond T_5. \]

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 \( \theta[4] \), then \( \forall(G \theta) \) is a DMD-logical consequence 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 \( \theta_1 \) such that \( ELE(A \theta_1) \cap ELE(A \theta_1) \neq \emptyset \). Since each element in \( ELE(A \theta_1) \cap ELE(A \theta_1) \) is a DMD-logical consequence of S, \( \forall(A \theta_1) \) 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 \( A_m \) be the selected atom of G, which satisfies \( ELE(A_m \theta_1) \cap ELE(A \theta_1) \neq \emptyset \). 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. \( \forall(A_m \theta_1 \ldots \theta_n) \) is a DMD-logical consequence of S, because \( ELE(A_m \theta_1 \ldots \theta_n) \cap ELE(A \theta_1 \ldots \theta_n) \neq \emptyset \). Hence \( \forall((A_1, A_2, \ldots, A_k \theta_1 \ldots \theta_n) \) is a DMD-logical consequence of S.

Theorem 8. Let S be an or-type knowledge base. Then,
\[ (\Diamond(S))^* = \text{DMD-success set of S}. \]

Proof. Let \( A(= or^n(t_1, \ldots, t_n)) \) be an element in \( EB_5 \) 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 \( A \notin (\Diamond(S))^* \). By Theorem 6, \( A \notin (\Diamond T_5 \uparrow \omega)^* \) for some \( n \in \omega \). We prove by induction on \( n \) that \( A \notin (\Diamond T_5 \uparrow \omega)^* \) implies \( S \cup \{\leftarrow A\} \) has a DMD-refutation. Suppose first that \( n = 1 \). \( A \notin (\Diamond T_5 \uparrow \omega)^* \) means that \( ELE(A) \cap ELE(B) \neq \emptyset \) for some fact B in S. Namely, \( S \cup \{\leftarrow A\} \) has a DMD-refutation. Suppose that the result holds for \( n-1 \). Let \( A \notin (\Diamond T_5 \uparrow \omega)^* \). By definition of \( \Diamond T_5 \), there exists a ground instance of a clause \( B \leftarrow B_1, \ldots, B_k \) such that
By the induction hypothesis, $SU\{\leftarrow B_i\theta\}$ has a DMD-refutation for each $i=1,\ldots, k$. Because each $B_i\theta$ is ground, these DMD-refutations can be combined into a DMD-refutation of $SU\{\leftarrow (B_1,\ldots, B_k)\theta\}$. Thus $SU\{\leftarrow A\theta\}$ has an unrestricted DMD-refutation. The DMD-refutation of $SU\{\leftarrow A\}$ can be obtained by the mgu lemma.

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

**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 $\sigma$ for $SU\{G\}$ and a substitution $\gamma$ such that $\theta = \sigma \gamma$.

**Proof.** Let $G$ be a goal $\leftarrow A_1, A_2,\ldots, A_k$. Since $\forall (G\theta)$ is a DMD-logical consequence of $S$, $\forall (A_i\theta)$ is a DMD-logical consequence of $S$ for $i=1,\ldots, k$. By Lemma 9 there exists a DMD-refutation of $SU\{\leftarrow A_i\theta\}$ with the identity substitution as the computed answer substitution for $i=1,\ldots, k$. We can combine these DMD-refutations into a DMD-refutation of $SU\{G\theta\}$ with identity substitution.

Suppose the sequence of mgu's of the DMD-refutation of $SU\{G\theta\}$ is $\theta_1,\ldots, \theta_n$. Then, $G\theta_1\theta_2\ldots\theta_n = G\theta$, because $\theta_1\theta_2\ldots\theta_n$ is an identity substitution. By the lifting lemma, there exists a DMD-refutation of $SU\{G\}$ with mgu’s $\theta_1',\theta_2',\ldots, \theta_n'$ such that $\theta_1\theta_2\ldots\theta_n = \theta_1'\theta_2'\ldots\theta_n'\gamma'$ for some substitution $\gamma'$. Let $\sigma$ be $\theta_1'\theta_2'\ldots\theta_n'$ restricted to the variables in $G$. Then $\theta = \sigma \gamma'$, where $\gamma$ is an appropriate restriction of $\gamma'$.

**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 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 foundation 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.
Acknowledgment

The author is very grateful to Prof. S. Arikawa for his continual encouragement.

References


Appendix

Let us show the real question-answering in an or-type knowledge base. The knowledge base, which deals with a knowledge for 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. 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 suzuki */
N = yamada;
no
/* initialization */
| ?-set. 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))))). /* ask a man who certainly is in first building and a session the man may attend */
X = tanaka,
Y = session(a);
X = tanaka,
Y = session(b);
X = suzuki,
Y = session(b);
X = suzuki,
Y = session(c);
no