An Algorithm for Constructing Extensions of Propositional Autoepistemic Logic

NOBORU BABAGUCHI*, JUN-ICHI MORIUMA* and YOSHIKAZU TEZUKA*

Moore's Autoepistemic Logic(AL) would be one of the most promising formalizations of nonmonotonic reasoning for its desirable properties as a logic. AL is intended to model the beliefs of an ideally rational agent reflecting upon his own beliefs. An important notion for AL is an extension which is the possible set of conclusions drawn by autoepistemic reasoning. Constructing extensions may be viewed as realizing autoepistemic reasoning. However, one of AL's theoretical problems is nonconstructive character: Given a set of premises, it is difficult to construct extensions straightforwardly.

In this paper, we propose an algorithm for constructing extensions for propositional AL. In order to resolve the difficulty as stated above, we consider AL on possible world semantics, and present a new important theorem specifying the relationship between extensions and possible world interpretation. Based on this theorem, we embody the algorithm. Also we clarify that this algorithm is sound and complete, as well as that its complexity is $O(n2^n)$. Finally shown are some execution results of the implemented algorithm.

1. Introduction

A major reasoning scheme used in current knowledge processing systems is deduction based on first-order predicate logic. Deduction is of form as: from "P" and "if P then Q", infer "Q". It may be viewed as one of the most fundamental reasoning schemes in human reasoning. The conclusions drawn by deduction are the facts which logically follow from a set of current knowledge, i.e. completely valid propositions. From this property, a deduction system is required to provide all of valid knowledge about the domain to solve a problem. In other words, completeness of knowledge is needed. Accordingly the disadvantage of deduction system is that the reasoning potential is limited within the knowledge which it inherently possesses.

In recent years, a number of approaches to extend the reasoning beyond the framework of deduction have been reported. Such reasoning is sometimes referred to as higher order reasoning [1]. Nonmonotonic reasoning is one of formalizations of higher order reasoning. The term "nonmonotonic" stems from the following property: knowledge drawn by reasoning does not increase monotonically. On the other hand, reasoning with the property that given knowledge increases monotonically is called monotonic reasoning. It is known that deduction is, in fact, monotonic reasoning. Nonmonotonic reasoning is concerned with the reasoning from incomplete knowledge, e.g. knowledge including exceptions, and with the reasoning that allows tentative conclusions by defaults. Since nonmonotonic reasoning system has no need of completeness of knowledge, it is expected to make up for the disadvantage of deduction system as stated above.

Nonmonotonic logic is a logic to formalize nonmonotonic reasoning. In the 1980's, a variety of formalizations have been attempted. Autoepistemic Logic (AL) [2], which we will consider in this paper, was proposed by Moore in order to solve the problems of Nonmonotonic Logic by McDermott and Doyle [6]. Recently, several researchers have been focusing attention on AL as one of nonmonotonic logics of interest [3-5], because of its clear correspondence between syntax and semantics.

In discussing a reasoning system in terms of a logic, the process of drawing conclusions from a set of premises is of great importance. For example, resolution is used for monotonic logics to obtain theorems to be conclusions drawn. Extensions[1] in AL will approximate to theorems in monotonic logics. Therefore, constructing extensions can be viewed as realizing autoepistemic reasoning, and will be a prerequisite to investigate the logical feature of AL. As is often pointed out, however, it is difficult to straightforwardly construct extensions for given premises [3].

In order to treat this difficulty, Moore has derived useful theorems from the viewpoint of possible world

This is a translation of the paper that appeared originally in Japanese in Transactions of IPSJ, Vol. 31, No. 7 (1990), pp. 979-987.

*Department of Communication Engineering, Faculty of Engineering, Osaka University, 2-1 Yamadaoka, Suita, 565 Japan.

[1]An extension is just identical to a stable expansion named by Moore in [2].

semantics[3]. In this paper, extending Moore's theorems, we present an algorithm for constructing extensions [7]. In addition, we discuss the correctness and complexity of the algorithm and show a couple of execution examples.

2. Overview of Autoepistemic Logic [2, 5]

AL is a logic on the basis of the notion of beliefs, and is intended to model the reasoning of an ideal agent who recognizes both what he believes and what he does not believe. This type of reasoning is called autoepistemic reasoning in the sense that an agent will derive conclusions, reflecting upon his own beliefs. Even in the case that there is no confidence, humans will be able to act based on beliefs. Moreover, the beliefs will be revised under some situation. As is well understood, it is impossible to formalize this sort of reasoning on ordinary logics such as propositional and first-order predicate logics which are the base of monotonic deduction system [1].

AL is extended to augment a belief operator L to the language of ordinary logics. That is, AL allows an expression Lp as a formula on AL language. In what follows, we call a formula Lp an L formula. The belief operator L is interpreted as "believed". Then, a formula Lp is interpreted as "p is believed" or "the agent believes that p is true". AL's nonmonotonicity results from the changeability of the truth value of L formulae.

Now we show the definitions about AL to be required in the following discussions.

[Definition 1] An autoepistemic theory T is a set of formulae.

An autoepistemic theory, simply called a theory, specifies the beliefs of an agent.

[Definition 2] An autoepistemic interpretation I of a theory T is a truth value assignment to the formulae of T satisfying,

1. I conforms to the usual truth recursion for ordinary logics,
2. A formula Lp is true in I iff $p \in T$.

In context, a truth value assignment may be called an assignment. Next, we proceed to explain the two syntactical properties which characterize a theory: stability and groundedness.

[Definition 3] A theory T is stable iff it satisfies the following three conditions:

1. $T = \mathrm{Th}[T]$.
2. If $p \in T$ then $Lp \in T$.
3. If $p \notin T$ then $\sim Lp \in T$.

The notation $\mathrm{Th}[T]$ indicates a set of formulae logically following from T, i.e. a set of theorems. Condition 1 indicates that the set of agent's beliefs is closed under logical consequence. Conditions 2 and 3 indicate that an agent knows what is believed as well as what is disbelieved. Stability of a theory points out the state where no further conclusions can be drawn. A stable theory T semantically means that T contains every formulae which is true in all autoepistemic interpretations of T which satisfies all formulae of T.

Regarding a finite set of premises Δ as the knowledge and belief given to an agent, we define groundedness between premises and a theory.

[Definition 4] A theory T is grounded in a set of premises Δ

iff $T \subseteq \mathrm{Th}[\Delta \cup \mathrm{PB}(T) \cup \mathrm{NB}(T)]$,

where $\mathrm{PB}(T) = \{Lp \mid p \in T\}$ and $\mathrm{NB}(T) = \{\sim Lp \mid p \notin T\}$.

Here $\Delta \cup \mathrm{PB}(T) \cup \mathrm{NB}(T)$ is called a basis set. Lp and $\sim Lp$ are called a positive belief and a negative belief, respectively. We will discuss about the sets of $\mathrm{PB}(T)$ and $\mathrm{NB}(T)$ in Section 3.1. Groundedness semantically means that every autoepistemic interpretation of a theory T which satisfies all formulae of Δ also satisfies all formulae of T.

Finally we define an extension.

[Definition 5] A theory T is an extension of Δ, denoted by $\mathrm{EX}[\Delta]$

$$\text{iff} \quad T = \mathrm{Th}[\Delta \cup \mathrm{PB}(T) \cup \mathrm{NB}(T)]. \tag{1}$$

From the definition of stability, it can be easily proven that if a stable theory T contains a set of premises Δ, then $T \supseteq \mathrm{Th}[\Delta \cup \mathrm{PB}(T) \cup \mathrm{NB}(T)]$. The sufficient and necessary condition of $T = \mathrm{EX}[\Delta]$ is as follows: T is stable, contains Δ, and is grounded in Δ. Although an extension may be similar to a set of theorems in ordinary logics, it should be noted that for some premises, there are no extensions or multiple extensions.

3. Algorithm for Constructing Extensions

3.1 Nonconstructive Character of AL

Obviously from equation (1), an extension is define as a fixed point equation with respect to a theory T. This means that enumerating all the possible theories, we should determine whether or not each of them will satisfy the equation.

Now let us consider the basis set: $\Delta \cup \mathrm{PB}(T) \cup \mathrm{NB}(T)$. If a formula p is a logical consequence of T, i.e. p is true in all ordinary interpretations which satisfy T, then a formula Lp is a member of $\mathrm{PB}(T)$. On the other hand, if a formula p is not a logical consequence of T, i.e. p is false in at least one ordinary interpretation which satisfies T, then a formula $\sim Lp$ is a member of $\mathrm{NB}(T)$. The truth value of a formula Lp depends on whether p logically follows from T, in other words, whether p is derived from T. This suggests that there is no direct correspondence between the truth values of p and Lp.

To be a logical consequence and not to be a logical consequence are equivalent to unsatisfiability and satisfiability, respectively. On first-order predicate calculus, there is a procedure to check the unsatisfiability, while there is no procedure to check the satisfiability. Therefore, first-order predicate AL is unfortunately not even semi-decidable [9].

In order to construct an extension, for an enumerated theory T, we must investigate the coincidence between T and the set of formulae of logical consequences of the basis set obtained from T. The difficulty with straightforward construction of an extension from given premises is known as nonconstructive character of AL.

As described above, Moore gave valuable theorems from the viewpoint of possible world semantics (PWS) [3], as a strategy to deal with AL's nonconstructive character. However, he did not show the procedure for constructing extensions in a definite form. Extending Moore's theorems, we proceed to embody such a procedure by deriving a new theorem concerning the relationship between extensions and PWS [7].

3.2 AL on Possible World Semantics

Before discussing AL on PWS, we briefly state the fundamental notion about possible worlds. It is Kripke that first introduces the idea of possible worlds. He defined the interpretation of formulae of form Lp, Mp in modal logic by means of possible worlds. Intuitively, we will take account of a logical model using possible worlds corresponding to the situations we will be able to imagine. We now explain PWS as far as the following discussion is concerned; see [8] for details.

An interpretation on PWS is determined by a set of possible worlds, accessibility, and truth value assignment. A binary relation $R(w1, w2)$, representing that a world $w2$ is accessible from a world $w1$, means that one, who is in $w1$, can refer to the truth values in $w2$. As for accessibility, the next properties are considered.
Reflexivity: $R(w1, w1)$.
Transitivity: If $R(w1, w2)$ and $R(w2, w3)$ then $R(w1, w3)$.
Symmetricity: If $R(w1, w2)$ then $R(w2, w1)$.
A system characterized by a set of possible worlds, called a structure, may change according to which property will hold in it. For example, a system in which only two properties of reflexivity and transitivity hold is called S4 system. A system in which all properties hold is called S5 system. The structures in S4 and S5 systems are called S4 and S5 structures.

For brevity, let us consider propositional AL (PAL). The following theorem guarantees that a stable theory can be expressed as a complete S5 structure [3]. The complete S5 structure particularly indicates an S5 structure in which every world is accessible from every world.

[Theorem 1] A theory T is stable iff T is the set of formulae that are true in every world of some complete S5 structure.

This theorem enables us to give an autoepistemic interpretation of a stable theory in terms of a complete S5 structure. Hence, a possible world interpretation can be defined next [3]. From the feature of complete S5 structure, we are allowed to consider interpretation solely by structure and truth value assignment, without accessibility.

[Definition 6] Let a stable theory T be represented as a complete S5 structure K. A possible world interpretation of T is a pair (K, V) satisfying the following conditions. V denotes a truth value assignment to propositional constants, which conforms to propositional logic.

1. A propositional constant is true in (K, V) iff it is true in V.
2. A formula Lp is true in (K, V) iff p is true in all worlds in K.
3. A formula Lp is false in (K, V) iff p is false in at least one world in K.

This definition states that K and V will define the assignments to L formulae and propositional constants, respectively. As an example, consider the possible world interpretation of a formula $LP \rightarrow Q$ under $K=\{\{P, Q\}, \{P, \sim Q\}\}$, $V=\{P, \sim Q\}$. Positive and negative literals of propositional constants indicate assigning true and false values, respectively. In this case, P is true in all worlds in K; LP is true; Q is false in V. Then $LP \rightarrow Q$ is interpreted false in this (K, V).

Note that for a stable theory, an autoepistemic interpretation is entirely equivalent to a possible world interpretation. A possible world interpretation stipulates the relationship between the truth values of p and Lp through possible worlds. This contributes to constructing extensions. Further, there has been proposed the next theorem about satisfiability of possible world interpretation [3].

[Theorem 2] If (K, V) is an autoepistemic interpretation of T, then (K, V) satisfies all formulae of T iff the truth value assignment V is identified with an assignment provided by one of the possible worlds in K.

Based on Moore's theorems, we will derive a new theorem about extensions and possible world interpretations. Its proof will be described in Appendix 1.

[Theorem 3] Let K denote a complete S5 structure representing a stable theory T. T is an extension of a set of premises Δ iff the following two conditions are satisfied:
Containing condition: For every possible world interpretation (K, V) which satisfies all formulae of Δ, K contains a possible world w which is identified with a truth value assignment V.
Satisfying condition: All formulae of Δ are true in every possible world w in K.

3.3 Description of Algorithm

From the preceding discussions, constructing an extension EX[Δ] can be reduced to constructing a complete S5 structure K satisfying the containing and satisfying conditions of Theorem 3. Let us describe the details of the proposed algorithm.

[Algorithm for constructing extensions]

Table 1 A table for constructing extensions.

Ui\Vj	V_1	V_2	······	Vjmax	K	Label
U_1						
U_2						
⋮						
Uimax						

(1) Let Δ_L denote the set of all L formulae appearing in a finite set of premises Δ. For each P of form LP included in Δ_L, if P has a subformula of form LQ, then LQ should also be a member of Δ_L. Let Δ_C denote the set of all propositional constants appearing in Δ.

(2) Assign truth values U_i $(i=1,\cdots,i\max)$ to each formula of Δ_L. Similarly, assign truth values V_j $(j=1,\cdots,j\max)$ to each formula of Δ_C. Letting the numbers of members of Δ_L and Δ_C be denoted by N and M, respectively, $i\max=2^N$ and $j\max=2^M$.

(3) Fill the blank columns of Table 1, according to the following procedure.

(3a) If the truth value assignment given by (U_i, V_j) satisfies all formulae of Δ, then write "1" on a column of (U_i, V_j), otherwise write "0". This step iterates for all of (U_i, V_j).

(3b) For each U_i, form a set of V_j's such that "1" is on the column of (U_i, V_j), and write it on a column of K.

(3c) For each U_i, check whether K satisfies the next conditions:
For each $Lp_k \in \Delta_L$,
If Lp_k is assigned true in U_i, then P_k is true in all V's included in K.
If Lp_k is assigned false in U_i, then P_k is false in at least one V included in K.
In the above case, if p_k has a subformula of form LQ, an assignment to LQ conforms to U_i. If these conditions are satisfied, then write "C (consistent)" on a column of Label, otherwise write "I (inconsistent)".

Steps (3a) and (3b) in this algorithm means selecting a set of possible worlds satisfying premises Δ. The conditions at step (3c) are those for verifying that K should be a complete S5 structure.

3.4 Correctness of Algorithm

Since the proposed algorithm works for PAL, its haltness is obviously guaranteed from the finiteness of propositional calculus. We give a theorem about the correctness of the algorithm.

[Theorem 4] Let Δ be a set of premises, and K_i be a complete S5 structure labeled "C" by this algorithm. T is a theory represented as K_i iff T is an extension of Δ.

Theorem 4 is proven by showing both that a complete S5 structure K_i labeled "C" by this algorithm satisfies the containing and satisfying conditions of Theorem 3, and that a structure K_i satisfying such conditions is always labeled "C" by this algorithm. Namely, Theorem 4 states that this algorithm is sound and complete. Its proof is precisely described in Appendix 2.

When there are consistent multiple extensions, this algorithm will yield the corresponding non-empty structures labeled "C". When there is no extension, it labels "I" to all structures. For inconsistent premises which may be viewed as a peculiar case, there is only one inconsistent extension to be the entire language of PAL. In such a case, this algorithm yields an only empty structure labeled "C".

3.5 Computational Complexity of Algorithm

In this section, we will consider the computational complexity of the proposed algorithm. We begin by considering an input size n for this problem. The input is a finite set of premises Δ; More strictly, the input is a set of L formulae Δ_L and a set of propositional constants Δ_C as shown in step (1) of the algorithm. Letting

$$\Delta_L=\{\delta_{Li} \mid i=1,\cdots,N\},$$
$$\Delta_C=\{\delta_{Cj} \mid j=1,\cdots,M\},$$

the input size n is the total number of elements in Δ_L and Δ_C, i.e. $N+M$.

Next, the procedure at steps (2) and (3a) can be regarded as evaluating a formula $\Delta'(\delta_{L1},\cdots,\delta_{LM},\delta_{C1},\cdots,\delta_{CN})$ which is a conjunction of premises. There are 2^M and 2^N kinds of way to assign truth values to propositional constants and L formulae. Thus, Δ' can be evaluated with $2^{M+N}=2^n$ combinations. Since there are at most finite number of elements δ_{Li}, δ_{Cj} appearing in Δ', evaluating Δ' for each assignment will take a computation cost of $O(n)$[1]. The computation cost at steps (2) and (3a) is, therefore, $O(n2^n)$. Each cost at steps (3b) and (3c) is at most 2^{M+N} and N^{2n}, respectively. Since $N \leq n$, $M+N=n$, the cost at these steps will not exceed $O(n2^n)$. Consequently, the computational complexity of this algorithm can be concluded as $O(n2^n)$. Following computation theory, the problem discussed in this paper may be said intractable [10].

From practical point of view, we may provide the following three strategies.

1. Restrict V from the interpretations for ordinary (propositional) formulae.

In case that Δ contains a single literal such as p or $\sim p$, for instance p appears in Δ, we need not investigate the interpretations of V in which p is false, then we are allowed to deal with only half of all interpretations. Accordingly, if there are k single literals, we will be able to restrict 2^{-k} times interpretations. A usual knowledge base is likely to have these formulae, so we will expect to get efficient computation. For other formulae such as $p \rightarrow q$, we can ignore the interpretations V in which p is true and q is false. Thus, we can get efficient computation by prioritizing the interpretations for ordinary formulae.

[1]Strictly, supposing that l is the number of clauses in conjunctive normal form of Δ', the computation cost is $O(nl)$. Here we assume that l is constant order of n.

2. Restrict U from the relation between p and Lp. The definition of extensions guarantees that Lp should be true in an extension which contains p. If p appears in Δ and $Lp \in \Delta_L$, then we need not investigate the interpretations of U in which Lp is false. In such a case, we are allowed to consider half of all interpretations of U.
3. Restrict U from the relation between Lp and $L \sim p$. In case that $Lp \in \Delta_L$ and $L \sim p \in \Delta_L$, both formulae are never true in an extension, so we can ignore such an interpretation of U.

3.6 Related Work

The algorithm proposed in this paper is, in fact, an algorithm for searching logical models of extensions. Since a logical model of an extension can be viewed as a set of formulae satisfied in an extension, it is no problem to regard this algorithm as a kind of decision procedure.

So far, a method which utilizes a semantic tableau has been proposed as a decision procedure of PAL [5]. Basic operation of semantic tableau method is making a proof for a given formula in order to determine whether or not it is included in an extension. The major difference between semantic tableau method and this algorithm is where the theoretical base is stressed. The former is based on autoepistemic interpretation (cf. Definition 2), whereas the latter on possible world interpretation (cf. Definition 6). Semantic tableau method is essentially to investigate the membership relation between a given formula and extensions. Although the relation between other formulae and extensions are given as a side effect, the content and the number of extensions will not be given in such a clear form as this algorithm.

Recently Moore has proposed an algorithm for constructing extensions on the basis of Theorems 1 and 2 [11]. Moore's algorithm is, in principle, similar to ours, but the two algorithms differ in the details or the efficiency. After describing the outline of Moore's algorithm, we will compare each other.

[Moore's algorithm]

(1) From all possible truth value assignments V to propositional constants appearing in a set of premises Δ, generate complete S5 structures.

(2) Select the structures, generated in (1), which satisfy Δ in every possible world.

(3) Generate a possible world interpretation (K, V) for a selected structure K and every assignment V.

(4) Of all generated interpretations, select a (K, V) satisfying all formulae of Δ, check whether or not K contains every V. A structure representing an extension is K satisfying the above conditions.

The input size n of Moore's algorithm is the number of propositional constants appearing in Δ. Now let us consider the stepwise computation cost. At step (1), there are 2^n kinds of all possible truth value assignments for n constants. In addition, there are 2^{2^n} kinds of generated complete S5 structures which correspond to a power set of a set of truth value assignments. Thus, the computation cost at step (1) is $O(2^{2^n})$. The cost at step (2) is $O(n2^{2^n})$, because all formulae of Δ are interpreted for every generated structure. Each cost at steps (3) and (4) does not exceed $O(n2^{2^n})$. Therefore, the complexity of Moore's algorithm can be concluded as $O(n2^{2^n})$.

Table 2 Comparison between Moore's algorithm and this algorithm.

	Moore's algorithm		This algorithm	
Input Size	M	(3)	$M+N$	(5)
No. of Structures	2^{2^M}	(256)	2^N	(4)
No. of PW Interpretations	$\alpha 2^{2^M+M}$	(1856)	2^{M+N}	(32)
Complexity	$O(M2^{2^M})$		$O((M+N)2^{M+N})$	

Table 2 shows the comparison between Moore's algorithm and this algorithm with respect to the input size, the complexity, and the numbers of generated structures and possible world interpretations. When $\Delta = \{LP \rightarrow Q, LQ \rightarrow R\}$, as an example, each value for the items is also indicated in parentheses in Table 2. Symbols in this table are identical to those used in Sec.3.5. That is, M and N denote the numbers of propositional constants and L formulae, respectively, appearing in Δ, and $0 \leq \alpha \leq 1$. For this example, propositional constants are P, Q and R ($M=3$), and L formulae are LP and LQ ($N=2$).

Table 2 suggests that this algorithm will generate less structures and interpretations than Moore's. The latter enumerates the structures exhaustively, while the former largely reduces the structures to be considered, by taking account of the truth value of L formulae which is necessary to evaluate a set of premises Δ. Note that if the number N of L formulae exceeds 2^M, this algorithm will contrary get worse with respect to the efficiency.

3.7 Execution Examples of Algorithm

Execution examples of the proposed algorithm on a computer for the next Examples 1 to 3 are shown in Fig. 1(a) to (c). In this figure, the input to a computer is underlined. As you can see, a formula representing a premise is inputted as a list form. Positive and negative literals in interpretations of U and V represent to assign true and false values to L formulae and propositional constants. Symbols of "−" and "&" denote negation and conjunction, respectively.

This algorithm is implemented with Common Lisp language on SUN work station. The size of a program, including input/output part, is 9K bytes.

[Example 1] Input: $\Delta = \{\sim LP \rightarrow Q, \sim LQ \rightarrow P\}$
Output: EX1 $= \{\{P, Q\}, \{P, \sim Q\}\}$
EX2 $= \{\{P, Q\}, \{\sim P, Q\}\}$
No. of extensions $= 2$

These premises state the beliefs of "if P is not believed then Q" and "if Q is not believed then P". We obtain two extensions EX1, EX2 as a result. In EX1, since P is true in all worlds, formulae such as P, LP, and $\sim L \sim P$

```
Input Premises
? (-> (~ (L q)) q)
? (-> (~ (L q)) p)
? end
Δ= { ~LP->Q, ~LQ->P }
OK? (y/n) y
ΔL = {LP, LQ}
ΔC = {P, Q}
U1 = {LP, LQ}          V1 = {P, Q}
U2 = {LP, ~LQ}         V2 = {P, ~Q}
U3 = {~LP, LQ}         V3 = {~P, Q}
U4 = {~LP, LQ}         V4 = {~P, ~Q}
      V1  V2  V3  V4  Label        K
U1     1   1   1   1    I    {V1, V2, V3, V4}
U2     1   1   0   0    C    {V1, V2}
U3     1   0   1   0    C    {V1, V3}
U4     1   0   0   0    I    {V1}
EX1 = { {P, Q}, {P, ~Q} }
EX2 = { {P, Q}, {~P, Q} }
No. of EX is . . . 2
```

(a)

```
Input Premises
? (-> (~ (L p)) p)
? end
Δ = {~LP->P}
OK? (y/n) y
ΔL = {LP}
ΔC = {P}
U1 = {LP}     V1 = {P}
U2 = {~LP}    V2 = {~P}
      V1  V2  Label   K
U1     1   1    I   {V1, V2}
U2     1   0    I   {V1}
No. of EX is . . . 0
```

(b)

```
Input Premises
? (-> p q)
? (-> q r)
? (-> (& (L p) (~ (L r))) (~ r))
? p
? end
Δ = { P->Q, Q->R, LP & ~LR->~R, P}
OK? (y/n) y
ΔL = {LP, LR}
ΔC = {P, Q, R}
U1 = {LP, LR}     V1 = {P, Q, R}      V5 = {~P, Q, R}
U2 = {LP, ~LR}    V2 = {P, Q, ~R}     V6 = {~P, Q, ~R}
U3 = {~LP, LR}    V3 = {P, ~Q, R}     V7 = {~P, ~Q, R}
U4 = {~LP, ~LR}   V4 = {P, ~Q, ~R}    V8 = {~P, ~Q, ~R}
      V1  V2  V3  V4  V5  V6  V7  V8  Label  K
U1     1   0   0   0   0   0   0   0    C   {V1}
U2     0   0   0   0   0   0   0   0    I   {}
U3     1   0   0   0   0   0   0   0    I   {V1}
U4     1   0   0   0   0   0   0   0    I   {V1}
EX1 = { {P, Q, R} }
No. of EX is . . . 1
```

(c)

Fig. 1 Execution examples of the implemented algorithm.

are true. Since the truth value of Q is distinct in the two worlds, formulae such as $\sim LQ$ and $\sim L \sim Q$ are true. This means that logical consequence formulae of such formulae are also included in EX1. In EX2, formulae which is replaced P by Q for EX1 are true. This example is a well known multiple extension case which is specific to nonmonotonic logics.

[Example 2] Input: $\Delta = \{\sim LP \rightarrow P\}$
Output: No. of extensions $= 0$

These premises state agent's inconsistent beliefs of "if P is not believed then P". The execution result indicates that there is no extension; no consistent conclusion is obtained.

[Example 3] Input: $\Delta = \{P \rightarrow Q, Q \rightarrow R, LP \wedge \sim LR \rightarrow \sim R, P\}$
Output: EX1 $= \{\{P, Q, R\}\}$
No. of extensions $= 1$

The third formula of premises states a default representing "if P then normally $\sim R$" [4]. In this example, we may derive an inconsistent conclusion: both R from ordinary formulae and $\sim R$ from defaults. The execution result indicates that only one extension consisting of a single world is given, and formulae such as P, Q, R, LP, LQ, LR are true.

In general, an extension is expressed as a theory which is a set of formulae. An expression in terms of a complete S5 structure, which is an output form of the proposed algorithm, enables us to understand the formulae to be true in an extension easily. This advantage is due to expressing a theory which will become infinite as a finite form.

4. Conclusion

In this paper, we have discussed about the nonconstructive character of AL, and have stated the difficulty in constructing extensions which are conclusions drawn by autoepistemic reasoning. In order to resolve this difficulty, we have considered AL on possible world semantics, and have presented a new important theorem stipulating the relationship between extensions and possible world interpretation. Based on this theorem, we have given an algorithm for constructing extensions for PAL. Also we have clarified that this algorithm is sound and complete, as well as that its complexity is $O(n2^n)$.

This algorithm depends theoretically on the fact that a correspondence relation between truth values of p and Lp is given through possible worlds. Its advantages are as follows: we can straightforwardly construct extensions; we can intuitively understand the contents of each extension. It should be noted that this algorithm will not halt for first order AL because there are infinite models.

Remaining work is as follows: improving algorithm efficiency, developing proof theoretic method, applying this framework to knowledge base management and knowledge acquisition. We are now developing a nonmonotonic knowledge processing system [12, 13] by introducing the proposed algorithm. It will be reported in a forthcoming paper.

Acknowledgement

This work was supported in part by the Grant-in-Aid for scientific research from the Ministry of Education.

References

1. Special issue: Paradigm towards Advanced Artificial Intelligence, *J. JSAI*, 2, 1 (1987) (in Japanese).

2. MOORE, R. C. Semantical Considerations on Nonmonotonic Logic, *Artif. Intell.*, **25**, 1 (1985), 75-94.
3. MOORE, R. C. Possible World Semantics for Autoepistemic Logic, *AAAI Non-Monotonic Reasoning Workshop* (1984), 344-354.
4. KONOLIGE, K. On the Relation between Default Theories and Autoepistemic Logic, *Artif. Intell.*, **35**, 3 (1988), 343-382.
5. BABAGUCHI, N., MURAKAMI, K. and AIBARA, T. Decision Procedure for propositional Autoepistemic Logic, *J. JSAI*, **2**, 3 (1987), 359-366 (in Japanese).
6. MCDERMOTT, D. and DOYLE, J. Non-Monotonic Logic I, *Artif. Intell.*, **13**, 1/2 (1980), 41-72.
7. MORIUMA, J-I., BABAGUCHI, N. and TEZUKA, Y. On Constructing Autoepistemic Extension, *Proc. 37th Annual Convention IPSJ* (1988) (in Japanese), 1459-1460.
8. HUGHES, G. E. and CRESSWELL, M. J. An Introduction to Modal Logic, Methuen, London, 1968.
9. BABAGUCHI, N. and TEZUKA, Y. Autoepistemic Reasoning on First-Order Predicate Calculus, *Proc. 36th Annual Convention IPSJ* (1988) (in Japanese), 1343-1344.
10. OHTSUKI, T. Computational Complexity of Algorithms, *J. IECEJ*, **62**, 7 (1979) (in Japanese), 787-798.
11. MOORE, R. C. Autoepistemic Logic, in Non-Standard Logics for Automated Reasoning, Smets P. et al. eds., Academic Press, 1988.
12. MORIUMA, J-I., BABAGUCHI, N. and TEZUKA, Y. BMS: Belief Based Non-monotonic Knowledge Processing System, *Proc. 39th Annual Convention IPSJ* (1989) (in Japanese), 325-326.
13. MORIUMA, J-I., BABAGUCHI, N. and TEZUKA, Y. Nonmonotonic Knowledge Processing System BMS and its Application, *IPSJ SIG Knowledge Engineering and Artificial Intelligence Reports*, 68-1 (1990) (in Japanese).

Appendix 1 Proof of Theorem 3

From Definition 5, a theory T is an extension of a set of premises Δ iff T contains Δ, is stable and is grounded in Δ. First from Theorem 1, if T is an stable theory, it is obvious that T contains Δ iff the satisfying condition is satisfied. On the other hand, T is grounded in Δ iff every autoepistemic interpretation satisfying Δ also satisfies T. From the semantical equivalence between autoepistemic interpretation and possible world interpretation, and Theorem 2, the above condition is equivalent to the containing condition.

Appendix 2 Proof of Theorem 4

First assume that a structure K_i is labeled "C" by this algorithm. In order to let a stable theory T represented as K_i be an extension of Δ, K_i should satisfy the containing and satisfying conditions of Theorem 3. Now in case that K_i is labeled "C", K_i satisfies the condition at step (3c) of the algorithm, so U_i is a truth value assignment as: if P is true in every world of K_i, then LP is true, otherwise LP is false. Accordingly, for every truth value assignment V to propositional constants, an assignment $\langle U_i, V\rangle$, generated by a pair of U_i and V, can be viewed as a possible world interpretation of T. Then each of $\langle U_i, V\rangle$ marked "1" at step (3a) is a possible world interpretation of T in which every formulae of Δ is true, so there exist no other interpretations. Since K_i is defined, at step (3b), as a set of V such that $\langle U_i, V\rangle$ is marked "1", for every possible interpretation (K_i, V) of T in which every formula of Δ is true, a world w which is identified with V is included. Hence the containing condition of Theorem 3 is satisfied.

In a world w, the truth values of L formulae of Δ_L agree with U_i. Letting V be an assignment to propositional constants in w, all formulae of Δ are true in $\langle U_i, V\rangle$ by the definition of K. Therefore, all formulae of Δ are true in every world of K_i, so the satisfying condition of Theorem 3 is satisfied.

In the opposite direction, assume that a theory T is an extension of Δ. Let K be a complete S5 structure representing T. K satisfies the containing and satisfying conditions of Theorem 3. Here we prove that K is constructed by this algorithm, and is identified with some K_i labeled "C".

The truth value assignment to L formulae of Δ_L, given by K, must agree with any of $U_1, \cdots, U_{i\max}$. denoted by U_i. For any interpretation V, then, $\langle U_i, V\rangle$ is a possible world interpretation of T. Now let us prove K_i constructed from U_i is equal to K, namely $K_i=K$.

Assuming $K_i\neq K$, we show that a contradiction will be derived. If $K_i\neq K$, then there must exist a possible world w such that either $w\in K_i$, $w\notin K$ or $w\notin K_i$, $w\in K$.

(i) Assume that there is a world w such that $w\in K_i$, $w\notin K$. Since $w\in K_i$, by steps (3a) and (3b) of the algorithm, an interpretation $\langle U_i, V\rangle$ satisfies Δ, where V is an assignment to propositional constants, identified with w. In addition, $\langle U_i, V\rangle$ is a possible world interpretation of T. From these, the containing condition is satisfied. Hence $w\in K$, which contradicts the assumption.

(ii) Assume that there is a world w such that $w\notin K_i$, $w\in K$. Since $w\in K$, all formulae of Δ are true in w, from the satisfying condition of Theorem 3. For an assignment V in w, $\langle U_i, V\rangle$ must satisfy Δ. Then, $w\in K_i$ by steps (3a) and (3b). This is a contradiction.

Next we show that such K_i is labeled "C". An assignment to L formulae of Δ_L, given by K, agrees with U_i. Since $K_i=K$, K_i satisfies the condition at step (3c). Therefore, K_i is certainly labeled "C".