NP-complete Problems on Label Updating Calculation in ATMS

Furuya, Shinji
Department of Information Systems, Interdisciplinary Graduate School of Engineering Science, Kyushu University

Miyano, Satoru
Research Institute of Fundamental Information Science Kyushu University

https://doi.org/10.5109/3146

出版情報：Bulletin of informatics and cybernetics. 25 (1/2), pp.1-5, 1991-03-27. Research Association of Statistical Sciences
バージョン：

Bulletin of Informatics and Cybernetics, Vol. 25, No. 1~2, 1992

NP-COMPLETE PROBLEMS ON LABEL UPDATING CALCULATION IN ATMS

By

Shinji FURUYA* and Satoru MIYANO**

Abstract

We investigate the computational complexity of the label updating calculation in ATMS and show that the following two decision problems are NP-complete.

(1) The problem of deciding whether there exists a consistent environment calculated from a new justification for a node.

(2) The problem of deciding whether there exists an environment calculated from a new justification for a node such that it is not a superset of any environment in the label of the node.

1. Introduction

ATMS is a truth maintenance system on the basis of assumptions [2, 3]. It consists of nodes, assumptions, justifications and nogoods. In ATMS, incomplete knowledge is treated as assumption and each node has a collection of sets of assumptions that support it. The task of ATMS is to maintain the contexts in which data hold. The database is allowed to contain inconsistent data, and ATMS provides a function which gives reasonings in multiple contexts for a problem solver.

The complexity involved in ATMS seems fairly large since it executes the truth maintenance of a knowledge base by combining assumptions used in the system. Various approaches have been proposed to speed up the truth maintenance [1, 5, 6, 7, 9]. Provan [8] showed that the problem of constructing maximal consistent sets of assumptions in ATMS is NP-complete. Also, it is shown that the membership problem is NP-complete [10]. Okuno [7] mentioned that the label updating calculation seems NP-complete. In this paper we first show that deciding whether there exists a consistent environment calculated from a new justification for a node is NP-complete. We also prove in a similar way that the problem of deciding whether there exists an environment calculated from a new justification for a node such that it is not a superset of any environment in the label of the node is NP-complete.

* Department of Information Systems, Interdisciplinary Graduate School of Engineering Science, Kyushu University 39, Kasuga 816, Japan. (current affiliation: Matsushita Electric Co., Ltd)

** Research Institute of Fundamental Information Science, Kyushu University 33, Fukuoka 812, Japan. E-mail: miyano@rifis.sci.kyushu-u.ac.jp

2. ATMS

ATMS consists of the following components:

Node: It corresponds to a problem-solver datum and represents knowledge.

Assumption: It is a special kind of node which is assumed without any commitment as to what is assumed.

Justification: It describes how a node is derivable from other nodes.

Nogood: It represents a contradiction.

EXAMPLE 1.

$$\begin{aligned} N &= \{A,\ B,\ C,\ D,\ E,\ F,\ G\} \\ A_s &= \{A,\ B,\ C,\ D\} \\ J &= \{E \leftarrow A \quad F \leftarrow B,\ C \quad F \leftarrow D \quad G \leftarrow E,\ F\} \\ NG &= \{nogood(A,\ B)\} \end{aligned}$$

Here N, A_s, J and NG are the sets of nodes, assumptions, justifications and a nogood, respectively.

DEFINITION 1. An *environment* E is a set of assumptions. An environment E is said to be *consistent* if no contradictions are derived from E and the current set J of justifications, that is, $E,\ J \nvdash \perp$, where the symbol $\vdash$ represents the derivation in propositional logic. Also, a node v is said to *hold* in an environment E if node v can be derived from E and J, that is, $E,\ J \vdash v$. An ATMS *label* L_v for node v is a set of environments associated with the node v.

DEFINITION 2. A label L_v for node v is said to be *consistent* if each environment $E \in L_v$ is consistent. The label L_v for node v is *sound* if node v holds in each environment $E \in L_v$. The label L_v for node v is *complete* if every consistent environment E such that $E, J \vdash v$ holds is a superset of some environment $E' \in L_v$. Moreover, the label L_v for node v is *minimal* if no environment in L_v is a superset of any other.

Hereafter we assume that a label for a node is consistent, sound, complete and minimal. In Example 1, the label for assumption A is $\{\{A\}\}$ and the labels for nodes E, F, G are $\{\{A\}\}$, $\{\{B,\ C\}\}$, $\{D\}\}$, $\{\{A,\ D\}\}$, respectively. If there are no such environments in the label for a node, then the label is said to be *empty* and is denoted by $\{\ \}$.

When a new justification is added to a node, the label for the node and ones which the justification affects must be updated. We call this update the *label updating calculation*.

Let L_H be the current label for node H and we assume that a new justification $r: H \leftarrow F,\ G$ is added to H. The basic algorithm for this label updating calculation is described as follows:

1. Calculated $L'_H = \{E_F \cup E_G | E_F \in L_F,\ E_G \in L_G\}$, where L_F and L_G are the labels for F and G, respectively.
2. Remove environments which are not consistent.
3. Add environments in L'_H to L_H.

4. Remove environments which are supersets of others from L_H.

The labels of the nodes affected by H are updated in a similar way.

We call an environment in L'_H an *environment calculated from the justification r*.

3. Computational Complexity

In this section, we consider the computational complexity of label updating calculation by analyzing two kinds of decision problems arising the computation. The first decision problem is formulated as follows:

GOODENV (Good Environment)

Instance: A set N of nodes, a set $A_s \subseteq N$ of assumptions, a set $\mathcal{L}$ of labels for nodes in N, a set NG of nogoods, a node $t \in N$ and a justification r for t.

Problem: Decide whether there exists a consistent environment calculated from the justification r.

Then we obtain the following theorem.

THEOREM 1. *GOODENV is NP-complete.*

PROOF. First, we show that GOODENV is in NP. A nondeterministic Turing machine which accepts GOODENV moves as follows. It chooses an environment from the label for each node in the right side of r nondeterministically and checks if the environment constructed by their union is consistent.

Next, we give a reduction from 3-SAT (3-satisfiability problem) (see [4]) to GOODENV. For a Boolean formula $F = C_1C_2\cdots C_m$ in three conjunctive normal form with variables $x_1, \ldots, x_n$. Then we define N, A_s, NG, t, r using $2n$ nodes corresponding to the literals $x_1, \bar{x}_1, \ldots, x_m, \bar{x}_n$ and m nodes corresponding to the clauses $C_1, C_2, \ldots, C_m$ together with a special node t as follows:

$$
\begin{aligned}
N &= \{a_1, \bar{a}_1, \ldots, a_n, \bar{a}_n, c_1, c_2, \ldots, c_m, t\},\\
A_s &= \{a_1, \bar{a}_1, \ldots, a_n, \bar{a}_n\},\\
NG &= \{nogood(a_1, \bar{a}_1), \ldots, nogood(a_n, \bar{a}_n)\}.
\end{aligned}
$$

The set $\mathcal{L} = \{L_v \mid v \in N\}$ of the labels for the nodes are defined as follows:

(1) For $v \in \{a_1, \bar{a}_1, \ldots, a_n, \bar{a}_n\}$, let $L_v = \{\{v\}\}$.

(2) For t, let $L_t = \{\ \}$.

(3) For c_j ($j = 1, \ldots, m$), the label L_{c_j} for c_j is defined as follows: For a literal α, let $\tilde{\alpha}$ be the node corresponding to the literal α, i.e., $\tilde{\alpha} = a_u$ if $\alpha = x_u$ and $\tilde{\alpha} = \bar{a}_u$ if $\alpha = \bar{x}_u$. Then $L_{c_j} = \{\tilde{\alpha} \mid \alpha \in C_j\}$.

The justification r for node t is

$$t \leftarrow c_1, c_2, \ldots, c_m.$$

Then we show that F is satisfiable if and only if there exists a consistent environment E calculated from r.

First, we assume that there is a consistent environment E which is calculated from r. Then both a_i and $\overline{a}_i$ are not in the environment since $nogood(a_i, \overline{a}_i)$ for $i = 1, \ldots, n$. We define a truth assignment $\hat{x}_1, \ldots, \hat{x}_n$ to the variables $x_1, \ldots, x_n$ as follows: If the environment E contains a_i, then let $\hat{x}_i = 1$. If E contains $\overline{a}_i$, then let $\hat{x}_i = 0$. Otherwise $\hat{x}_i$ is arbitrary. Then it can be easily seen that each clause of F is satisfied by this truth assignment $\hat{x}_1, \ldots, \hat{x}_n$.

Conversely, if F is satisfiable, then let $\hat{x}_1, \ldots, \hat{x}_n$ be a truth assignment to the variables $x_1, \ldots, x_n$ which satisfies all clauses in F. We can construct a consistent environment E from r as follows. For a literal α, we denote by $\hat{\alpha}$ the value under the truth assignment $\hat{x}_1, \ldots, \hat{x}_n$. For each clause C_j, C_j contains a literal α with $\hat{\alpha} = 1$. Then we choose $\widetilde{\alpha}$ from L_{C_j}. In this way, we can obtain a consistent environment calculated from r by choosing these environments. It is not hard to see that this reduction is computable in polynomial time or log space. Hence GOODENV is NP-complete. □

By Theorem 1, the problem of searching consistent environments calculated from a new justification added to a node is intractable in general.

We also consider the following decision problem:

NOSUPER

Instance: A set N of nodes, a set $A_s \subseteq N$ of assumptions, a set $\mathcal{L}$ of labels for nodes in N, a node $t \in N$ and a justification r for t.

Problem: Decide whether there exists an environment calculated from r which is not a superset of any environment in the label for t.

THEOREM 2. *NOSUPER is NP-complete.*

PROOF. NOSUPER is accepted in polynomial time by a nondeterministic Turing machine as follows: It chooses an environment from the label of each node in the right side of the justification r nondeterministically and checks whether the environment E formed from these environments is not a superset of any environment in the label for node t. This is computable in polynomial time. Hence NOSUPER is in NP.

Now we give a reduction from 3-SAT to NOSUPER. For a given Boolean formula $F = C_1 C_2 \cdots C_m$ in three conjunctive normal form, we construct N, A_s, $\mathcal{L}$, t and a justification r as follows. Let $x_1, \ldots, x_n$ be the variables in F. Then we set

$$N = \{a_1, \overline{a}_1, \ldots, a_n, \overline{a}_n, d_1, d_2, \ldots, d_n, t\},$$
$$A_s = \{a_1, \overline{a}_1, \ldots, a_n, \overline{a}_n\}.$$

The labels for nodes are defined as follows:

(1) For $v \in \{a_1, \overline{a}_1, \ldots, a_n, \overline{a}_n\}$, let $L_v = \{\{v\}\}$.

(2) For d_i $(i = 1, \ldots, n)$, let $L_{d_i} = \{\{a_i\}, \{\overline{a}_i\}\}$.

(3) For each clause C_j of F, the label L_t for t contains $\{\widetilde{\alpha} \mid \alpha \in C_j\}$.

The justification r for t is

$$t \leftarrow d_1, \ldots, d_n.$$

We show that F is satisfiable if and only if there exists an environment E calculated from r which is not a superset of any environment in the label L_t for node t. Assume that there exists such an enviornment E calculated from r. Then either $\{a_i\}$ or $\{\bar{a}_i\}$ is chosen from d_i and either a_i or $\bar{a}_i$ is contained in the environment E for each $i = 1, \ldots, n$. Then we define a truth assignment $\hat{x}_1, \ldots, \hat{x}_n$ to the variables $x_1, \ldots, x_n$ as follows: If the enviornment E contains a_i, then let $\hat{x}_i = 1$. If E contains $\bar{a}_i$, then let $\hat{x}_i = 0$. We can see that each clause of F is satisfied by the truth assignment $\hat{x}_1, \ldots, \hat{x}_n$.

Conversely, If F is satisfiable, then let $\hat{x}_1, \ldots, \hat{x}_n$ be a truth assignment to the variables. For each $i = 1, \ldots, n$, if $\hat{x}_1 = 1$, then we choose $\{a_i\}$ from L_{d_i}. If $\hat{x}_1 = 0$, then we choose $\{\bar{a}_i\}$ from L_{d_i}. We can obtain an environment that satisfies the condition by choosing these environments from L_{d_i} for $i = 1, \ldots, n$. It is also easy to see that this reduction is computable in polynomial time or log space. Hence NOSUPER is NP-complete. □

By Theorem 2, the computational complexity of the label updating calculation is hard even in the case without any nogoods.

4. Conclusion

In this paper we showed that two decision problems on label updating calculation are NP-complete. These results show that the label updating calculation involves computationally intractable problems. It means that it is hard to handle many data in general. ATMS is applied widely in Artificial Intelligence, but it seems that some restrictions are required to cope with general real-time use.

References

[1] Chimura, F., Numaoka, C. and Tokoro, M.: *An ATMS considering assumption's priority*, Proc. 4th Annual Conf. JSAI, (1991) 61–64.

[2] de Kleer, J.: *An assumption-based TMS*, Artif. Intell. **28** (2), (1986) 127–162.

[3] de Kleer, J.: *Extending the ATMS*, Artif. Intell. **28** (2), (1986) 163–196.

[4] Garey, M.R. and Johnson, D.S.: *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W.H. Freeman and Company, San Francisco, 1979.

[5] Lodi, W.L.: *A new algorithm for truth maintenance*, Proc. Annual AI Systems in Government Conference, (1989) 14–21.

[6] Okuno, H.: *AMI: a new implementation of the ATMS and its parallel execution on shared-memory multiprocessor*, J. Japanese Soc. Artif. Intell. **5** (3), (1990) 333–342 (in Japanese).

[7] Okuno, H.: *Speed-up techniques of ATMS and its applications*, J. Japanese Soc. Artif. Intell. **6** (1), (1991) 24–34 (in Japanese).

[8] Provan, G.M.: *Efficiency analysis of multiple-context TMSs in scene representation*, Proc. AAAI-87, (1987) 173–177.

[9] Rothberg, E. and Gupta, A.: *Experiences implementing a parallel ATMS on a shared-memory multiprocessor*, Proc. 11th IJCAI-89, (1989) 199–205.

[10] Stillman, J.: *It's not my default: the complexity of membership problems in restricted propositional default logics*, Proc. AAAI-90, (1990) 571–578.

Received March 29, 1991