Deduction and Abduction with Ordered Binary Decision Diagrams

Takashi HORIYAMA[†]   Toshihide IBARAKI[‡]

Abstract. We consider problems of reasoning with a knowledge-base, which is represented by an ordered binary decision diagram (OBDD), for two special cases of general and Horn knowledge-bases. Our main results say that both finding a model of a knowledge-base and deducing from a knowledge-base can be done in linear time for general case, but that abduction is NP-complete even if the knowledge-base is restricted to be Horn. Then, we consider the abduction when its assumption set consists of all propositional literals (i.e., an answer for a given query is allowed to include any positive literals), and show that it can be done in polynomial time if the knowledge-base is Horn. Some other solvable cases are also discussed.

1 Introduction

Logical formulae are one of the traditional means of representing knowledge-bases in AI. However, it is known that deduction from a knowledge-base that consists of a set of propositional clauses is co-NP-complete and abduction is $\sum_2^P$-complete [2]. Recently, an alternative way of representing a knowledge-base has been proposed; i.e. it uses a subset of its models called characteristic models (see e.g., [4, 5]). Deduction from a Horn knowledge-base in this model-based approach can be performed in linear time, and abduction is also performed in polynomial time [4].

In this paper, we consider yet another method of knowledge representation, i.e., the use of ordered binary decision diagrams (OBDDs) [1, 7]. An OBDD is a directed acyclic graph representing a Boolean function. By restricting the order of variable appearances and by sharing isomorphic subgraphs, OBDDs have the following useful properties: 1) When an ordering of variables is specified, an OBDD has the unique reduced canonical form for each Boolean function. 2) Many Boolean functions appearing in practice can be compactly represented. 3) There are efficient algorithms for many Boolean operations on OBDDs. As a result of these properties, OBDDs are widely used for various practical applications, especially in computer-aided design and verification of digital systems.

The manipulation of knowledge-bases by OBDDs (e.g. deduction and abduction) was first discussed by Madre and Coudert [6], and then basic theoretical questions were examined by the authors [3]. For example, it was shown that, in some cases, an OBDD-based representation requires exponentially smaller space than the other two, i.e., formula-based and model-based representations, while there are also cases in which each of the other two requires exponentially smaller space than that of an OBDD. However, the computational complexity of the operations for reasoning, such as deduction and abduction, was still open.

In this paper, we consider the complexity of reasoning with general and Horn knowledge-bases of OBDDs. We first consider the problem of finding a model of a given OBDD. Although it is obvious that the problem can be solved in polynomial time for any OBDD, we show that the least model can be output in polynomial time if the given OBDD represents a Horn knowledge-base. As to the deduction with OBDDs, Madre and Coudert discussed the case in which both knowledge and query are given in OBDD-based representations [6]. It is also natural to assume that the query is given in CNF, because formula-based queries are easier to understand. We show that deduction in this case can be done in polynomial time, though a naive algorithm may require exponential time with respect to the input size.

We also discuss abduction with OBDDs. Although enumerating all possible outputs (i.e., explanations for a given query) may require exponential time [6], it was unknown whether or not generating only one

[†]Graduate School of Information Science, Nara Institute of Science and Technology, Nara, 630-0101 Japan. E-mail: horiyama@is.aist-nara.ac.jp

[‡]Department of Applied Mathematics and Physics, Kyoto University, Kyoto 606-8501, Japan. E-mail: ibaraki@i.kyoto-u.ac.jp

explanation can be done in polynomial time. Unfortunately, this problem is shown to be NP-complete even if knowledge-bases are restricted to be Horn. However, by introducing some constraints on assumption set, we show that abduction from Horn OBDDs can be done in polynomial time.

The rest of this paper is organized as follows. The next section gives definitions and basic concepts. The problems of reasoning with general and Horn OBDDs are discussed in Sections 3 and 4, respectively.

2 Preliminaries

2.1 Notations and Basic Concepts

We consider a Boolean function $f : \{0,1\}^n \rightarrow \{0,1\}$. An *assignment* is a vector $a \in \{0,1\}^n$, whose i-th coordinate is denoted by a_i. A *model of f* is a satisfying assignment a of f, and the *theory $\Sigma(f)$* *representing f* is the set of all models of f. Given $a, b \in \{0,1\}^n$, we denote by $a \leq b$ the usual bitwise (i.e., componentwise) ordering of assignments; $a_i \leq b_i$ for all $i = 1, 2, \ldots, n$, where $0 < 1$. A model a is *minimal* in Σ if no $b \in \Sigma$ satisfies $b < a$.

Let $x_1, x_2, \ldots, x_n$ be the n variables of f, where each x_i corresponds to the i-th coordinate of assignments. Negation of a variable x_i is denoted by $\overline{x}_i$. Any Boolean function can be represented by some CNF, which may not be unique. We sometimes do not make a distinction among a function f, its theory $\Sigma(f)$, and a CNF φ that represents f, unless confusion arises. We define a *restriction* of f by replacing a variable x_i by a constant $a_i \in \{0,1\}$, and denote it by $f|_{x_i=a_i}$. Namely, $f|_{x_i=a_i}(x_1, \ldots, x_n)$ $= f(x_1, \ldots, x_{i-1}, a_i, x_{i+1}, \ldots, x_n)$ holds. A *smoothing* of f, denoted by $\exists_{x_i} f$, is defined as $f|_{x_i=0} \vee f|_{x_i=1}$. Restriction and smoothing may be applied to many variables. We also define $f \leq g$ (resp., $f < g$) by $\Sigma(f) \subseteq \Sigma(g)$ (resp., $\Sigma(f) \subset \Sigma(g)$).

A theory Σ is *Horn* if Σ is closed under operation $\wedge_{bit}$, where $a \wedge_{bit} b$ is bitwise AND of models a and b. For example, if $a = (0011)$ and $b = (0101)$, then $a \wedge_{bit} b = (0001)$. Any Horn theory Σ has the *least* (i.e., *unique minimal*) *model* $a = \bigwedge_{bit \ b \in \Sigma} b$. We also use the operation $\wedge_{bit}$ as a set operation; $\Sigma(f) \wedge_{bit} \Sigma(g)$ $= \{a \mid a = b \wedge_{bit} c$ holds for some $b \in \Sigma(f)$ and $c \in \Sigma(g)\}$. We often denotes $\Sigma(f) \wedge_{bit} \Sigma(g)$ by $f \wedge_{bit} g$, for convenience. Note that the two functions $f \wedge g$ and $f \wedge_{bit} g$ are different.

A Boolean function f is *Horn* if $\Sigma(f)$ is Horn; equivalently if $f \wedge_{bit} f = f$ holds (as sets of models). A clause is *Horn* if the number of positive literals in it is at most one, and a CNF is *Horn* if it contains only Horn clauses. It is known that a theory Σ is Horn if and only if Σ can be represented by some Horn CNF.

Given Boolean functions f (called a *background theory*) and α (called a *query*), *deduction* is the problem of deciding whether $f \models \alpha$ holds or not. *Abduction* is the problem of generating an explanation for a given query. Given a Boolean functions f on n variables $X = \{x_1, x_2, \ldots, x_n\}$, a set $A \subseteq X$ (called an *assumption set*) and a positive literal $x_q \in X$ (called a *query letter*), an *explanation* for (f, A, x_q) is a set $E \subseteq A$ satisfying (i) $f \wedge E \models x_q$ (i.e., $f \wedge E \wedge \overline{x}_q \equiv 0$) and (ii) $f \wedge E$ is consistent (i.e., $f \wedge E \not\equiv 0$), where $f \wedge E$ denotes $f \wedge (\wedge_{x_i \in E} x_i)$. Some elements of the triple (f, A, x_q) may be omitted unless confusion arises. Since the set $\{x_q\}$ always satisfies both of the restrictions, it is called a *trivial explanation*. Our interest is computing *non-trivial explanation* efficiently. An explanation is *minimal* if none of whose subsets do not satisfy the constraints.

2.2 Ordered Binary Decision Diagrams

An *ordered binary decision diagram* (OBDD) is a directed acyclic graph that represents a Boolean function. It has two sink nodes 0 and 1, called the *0-node* and the *1-node*, respectively (which are together called the *constant nodes*). Other nodes are called *variable nodes*, and each variable node v is labeled by one of the variables $x_1, x_2, \ldots, x_n$. Let $var(v)$ denote the label of node v. Each variable node has exactly two outgoing edges, called a *0-edge* and a *1-edge*, respectively. One of the variable nodes becomes the unique source node, which is called the *root node*. Let $X = \{x_1, x_2, \ldots, x_n\}$ denote the set of n variables. A *variable ordering* is a total ordering $(x_{\pi(1)}, x_{\pi(2)}, \ldots, x_{\pi(n)})$, associated with each OBDD, where π is a permutation $\{1, 2, \ldots, n\} \rightarrow \{1, 2, \ldots, n\}$. The *level* of a node v, denoted by $level(v)$, is defined by its label;

if node v has label $x_{\pi(i)}$, $level(v)$ is defined to be $n - i + 1$. The level of the constant nodes is defined to be 0. On every path from the root node to a constant node in an OBDD, each variable appears at most once in the decreasing order of their levels.

Every node v of an OBDD also represents a Boolean function f_v, defined by the subgraph consisting of those edges and nodes reachable from v. If node v is a constant node, f_v equals to its label. If node v is a variable node, f_v is defined as $\overline{var(v)}\, f_{0\text{-}succ(v)} \vee var(v)\, f_{1\text{-}succ(v)}$ by Shannon's expansion, where $0\text{-}succ(v)$ and $1\text{-}succ(v)$, respectively, denote the nodes pointed by the 0-edge and the 1-edge of node v. The function f represented by an OBDD is the one represented by the root node. Given an assignment a, the value of $f(a)$ is determined by following the corresponding path from the root node to a constant node in the following manner: at each variable node v, one of the outgoing edges is selected according to the assignment $a_{var(v)}$. The value of the function is the label of the final constant node.

When two nodes u and v in an OBDD represent the same function, and their levels are the same, they are called *equivalent*. A node whose 0-edge and 1-edge both point to the same node is called *redundant*. An OBDD is called *reduced* if it has no equivalent nodes and no redundant nodes. In the following, we assume that all OBDDs are reduced, unless otherwise stated. The *size* of an OBDD is the number of nodes in the OBDD. Given a function f and a variable ordering, its reduced OBDD is unique and has the minimum size among all OBDDs with the same variable ordering. The minimum sizes of OBDDs representing a given Boolean function depends on the variable orderings [1].

3   Reasoning with General OBDDs

3.1   Finding a Model and Deduction with OBDDs

In this section, we consider the complexity of reasoning with general knowledge-bases, which are represented by OBDDs. In particular, we first consider the problems of finding a model of a knowledge-base and deducing from a knowledge-base. We assume, without loss of generality, that the variable ordering of a given OBDD is always $(x_n, x_{n-1}, \ldots, x_1)$.

We first consider finding a model of a knowledge-base. By definition, paths from the root node to the 1-node of a given OBDD correspond to the models of the theory represented by the OBDD. Algorithm FIND-MODEL in Fig. 1 outputs one of the models by finding a path to the 1-node. It follows a path by assigning 0 or 1 to variable x_ℓ in level ℓ. In Step 3, if $level(v) < \ell$ (i.e., f_v does not depend on x_ℓ), both 0 and 1 can be assigned. Otherwise, node v exists in level ℓ. Since node v has the 0-edge and the 1-edge, at least one of them consists a path to the 1-node. If the 0-edge points to the 0-node, the 1-edge is selected as a component of the path to the 1-node. Otherwise, the 0-edge is selected. Since FIND-MODEL takes as many 0's as possible at all levels, the output is obviously one of the minimal models. Moreover, it is the least model if a given theory is Horn. (Recall that any Horn theory has the unique least model.) Since each step can be done in constant time, the computation time of Algorithm FIND-MODEL is $O(n)$.

Theorem 3.1 *Given an OBDD of a theory $\Sigma(f)$, its minimal model can be generated in $O(n)$ time. Moreover, the output is the least model if $\Sigma(f)$ is Horn.*

Now, we discuss deduction with OBDDs. Madre and Coudert gave the following appealing result on the case when both background theory $\Sigma(f)$ and query α are given as OBDDs.

Lemma 3.1 [6] *Given OBDDs of a theory $\Sigma(f)$ and query α, deciding whether $\Sigma(f) \models \alpha$ holds can be done in $O(|OBDD(f)| \cdot |OBDD(\alpha)|)$ time.*

Let us consider the case when a query α is given as a CNF formula. We may apply Lemma 3.1 by constructing the OBDD of α from its CNF formula. However, the naive algorithm is intractable even when knowledge-bases are restricted to be Horn. This is because there exists a Horn theory for which CNF requires linear size with respect to the number of variables, while the smallest OBDD requires exponential size [3]. We however show that deduction can be done in linear time without explicitly constructing the OBDD of α.

Algorithm FIND-MODEL

Input: An OBDD of theory $\Sigma(f)$ with a variable ordering $(x_n, x_{n-1} \ldots, x_1)$.

Output: A model $a \in \{0,1\}^n$ of $\Sigma(f)$ if $\Sigma(f) \neq 0$; otherwise, "empty".

Step 1 (check emptiness). If v_{root} is the 0-node, where v_{root} is the root node of the given OBDD, then output "empty" and halt.

Step 2 (initialize). Set $v := v_{\text{root}}$ and $\ell := n$.

Step 3 (assign a_ℓ in level ℓ). If $(level(v) < \ell)$ then set $a_\ell := 0$; else if $0\text{-}succ(v)$ is the 0-node then set $a_\ell := 1$ and $v := 1\text{-}succ(v)$; else set $a_\ell := 0$ and $v := 0\text{-}succ(v)$.

Step 4 (iterate). If $\ell = 1$ then output a and halt. Otherwise set $\ell := \ell - 1$ and return to Step 3.

Figure 1: Algorithm FIND-MODEL to find a model of an OBDD.

Theorem 3.2 *Given an OBDD of a theory $\Sigma(f)$ and a CNF formula α, deciding whether $\Sigma(f) \models \alpha$ holds can be done in $O(|OBDD(f)| \cdot |CNF(\alpha)|)$ time.*

The above result says that OBDD-based representation can replace traditional knowledge-base systems, in which queries are given in CNF formulae. A strong point of our approach is that once knowledge is represented by an OBDD, any deductive query can be answered in linear time, even if the knowledge is not Horn.

3.2 Abduction with General OBDDs

We consider the computational cost of abduction from general theories. It is known that enumerating all possible explanations may require exponential time. The following theorem says that finding only one explanation is intractable.

Theorem 3.3 *Given an OBDD of a theory $\Sigma(f)$ on n variables $X = \{x_1, x_2, \ldots, x_n\}$, an assumption set $A \subseteq X$ and a query letter $x_q \in X$, deciding whether x_q has a non-trivial explanation is NP-complete.*

Proof: The problem is NP, since we can guess a set $E \subseteq A$ and check whether E is an explanation for x_q in polynomial time. The proof of the NP-hardness is based on a reduction from the problem of testing the non-tautology of DNF formulae (NON-TAUTOLOGY). Given a DNF formula $\varphi = \bigvee_{i=1}^{m} T_i$ on n variables $x_1, x_2, \ldots, x_n$, where $T_i = \left(\bigwedge_{j \in P(i)} x_j \right) \wedge \left(\bigwedge_{k \in N(i)} \overline{x}_k \right)$ and $P(i) \cap N(i) = \emptyset$ for $i = 1, 2, \ldots, m$, we construct an OBDD of theory $\Sigma(f_A)$ on $2n + m + 1$ variables $y_1, y_2, \ldots, y_{2n+m+1}$ with variable ordering $(y_{2n+m+1}, y_{2n+m}, \ldots, y_1)$; f_A is defined as

$$f_A = \overline{y}_{2n+m+1} g_A \vee y_{2n+m+1} h_A,$$

where $h_A = \left(\bigwedge_{i=1}^{n} \left(\overline{y}_{2i-1} y_{2i} \vee y_{2i-1} \overline{y}_{2i} \right) \right) \wedge \left(\bigwedge_{j=1}^{m} \overline{y}_{2n+j} \right)$, $g_A = \bigvee_{i=1}^{m} \left(\left(\bigwedge_{j=i+1}^{m} y_{2n+j} \right) \wedge \overline{y}_{2n+i} \wedge g_i \right)$ and $g_i = \left(\bigwedge_{j \in P(i)} \overline{y}_{2j-1} y_{2j} \right) \wedge \left(\bigwedge_{j \in N(i)} y_{2j-1} \overline{y}_{2j} \right) \wedge \left(\bigwedge_{j \in \{1,2,\ldots,n\} - P(i) - N(i)} y_{2j-1} y_{2j} \right)$. We can prove that f_A is represented by an OBDD of polynomial size, and that there exists a non-trivial explanation for $(f_A, \{y_1, y_2, \ldots, y_{2n}\}, y_{2n+m+1})$ if and only if $\varphi \equiv 1$ does not hold. $\square$

In many cases, polynomial time algorithms can be obtained by considering some specific but proper constraints. We consider the case when the assumption set is fixed to be the set of all propositional literals. The following corollary, however, gives a negative result under such condition.

Corollary 3.1 *Given an OBDD of a theory $\Sigma(f)$ on n variables $X = \{x_1, x_2, \ldots, x_n\}$, an assumption set $A = X$ and a query letter $x_q \in X$, deciding whether x_q has a non-trivial explanation is NP-complete.*

4 Reasoning with Horn OBDDs

4.1 Abduction with Horn OBDDs

In this section, we consider the complexity of reasoning with Horn knowledge-bases. Since the previous section tells that the problems of finding a model of any OBDD and deducing from any OBDD can be done in linear time, we only consider the problem of finding an abductive explanation from Horn OBDDs.

By restricting a given knowledge-base to be Horn, the computational cost for abduction may be reduced. For example, in the case of CNFs, it becomes NP-complete from $\sum_2^P$-complete. In the model-based case, the linear time adbuction is accomplished by making use of the properties of Horness. However, the following theorem says that abduction is intractable even for Horn OBDDs.

Theorem 4.1 *Given an OBDD of a Horn theory $\Sigma(f)$ on n variables $X = \{x_1, x_2, \ldots, x_n\}$, an assumption set $A \subseteq X$ and a query letter $x_q \in X$, deciding whether x_q has a non-trivial explanation is NP-complete.*

Proof: This theorem is proved in a similar way to Theorem 3.3. The problem is obviously NP. NP-hardness is proved by a reduction from NON-TAUTOLOGY. The difference is that we have to construct an Horn OBDD. □

Now, we consider the case when the assumption set is fixed to be the set of all propositional literals. Although abduction under such condition is NP-complete for the general case, it can be done in quadratic time for the Horn case. We may find an explanation in the following manner; For each model $a \in \Sigma(f)$, check whether both (I) $a \models x_q$ (i.e., $a_q = 1$) and (II) $f \wedge E(a) \models x_q$ hold or not, where $E(a)$ denotes the set $\chi^{-1}(a) - \{x_q\}$. If there exists a model a which passes both checks successfully, $E(a)$ is an explanation. Otherwise, there are no non-trivial explanations. Note that the checks (I) and (II) correspond to the constraints (ii) and (i) in the definition of abduction, respectively. Although this naive algorithm may require exponential computation time, it tells a key of our algorithm.

Theorem 4.2 *Given an OBDD of a Horn theory $\Sigma(f)$ on n variables $X = \{x_1, x_2, \ldots x_n\}$, an assumption set $A = X$ and a query letter x_q, a non-trivial explanation for $\Sigma(f)$ can be obtained in $O(|OBDD(f)|^2)$ time.*

Proof: We first show that $f|_{x_q=1} \wedge \overline{f|_{x_q=0}}$ is the characteristic function of E satisfying constraints (I) and (II) in the naive algorithm. The set satisfying constraint (I) has the characteristic function $f|_{x_q=1}$. Since the property $f \wedge (\chi^{-1}(a) - \{x_q\}) \models x_q$ is equivalent to $f \wedge (\chi^{-1}(a) - \{x_q\}) \wedge \overline{x}_q \not\equiv 0$, constraint (II) says that there exists no model b satisfying both $b_q = 0$ and $(\chi^{-1}(a) - \{x_q\}) \subseteq (\chi^{-1}(b) - \{x_q\})$, i.e., $a_A \leq b_A$, where a_A denotes the vector composed by all the components of a but a_q. Assume that there exists a model b satisfying $a_A \leq b_A$. Since $a = (a_A, 1)$ and $b = (b_A, 0)$ are the models of a Horn theory $\Sigma(f)$, $a \wedge_{bit} b = (a_A, 0)$ is also a model, i.e., a_A is a model of $f|_{x_q=0}$. Conversely, if a_A is a model of $f|_{x_q=0}$, there exists a model $b = (a_A, 0)$ satisfying $a_A = b_A$. Thus, the set satisfying constraint (II) has the characteristic function $\overline{f|_{x_q=0}}$, and hence we have $f|_{x_q=1} \wedge \overline{f|_{x_q=0}}$.

As noted in subsection 2.2, an OBDD representing $f|_{x_i=0}$ (resp., $f|_{x_i=1}$) can be obtained in $O(|f|)$ time from the OBDD representing f, where $|f|$ denotes its size. The size does not increase by a restriction $f|_{x_i=0}$ or $f|_{x_i=1}$. Negation can be done in constant time. Since the OBDD of $g \wedge h$ can be obtained in $O(|g| \cdot |h|)$ time, the OBDD of the characteristic function can be obtained in $O(|f|^2)$ time. If it represents 0, there are no non-trivial explanations. Otherwise, we obtain the characteristic vector of an explanation by applying Algorithm FIND-MODEL to the OBDD. □

Once a non-trivial explanation is obtained, a minimal non-trivial explanation can be found by eliminating unnecessary assumptions.

4.2 Polynomial Time Abduction with Horn OBDDs

Theorem 4.2 gives a non-trivial explanation in polynomial time. In this subsection, we relax the constraints of the theorem. First, we show that a given assumption set may not be fixed to the set of all propositional

literals, even though abduction for an arbitrary assumption set is proved to be intractable by Theorem 4.1. The following lemmas gives a key for this goal.

Lemma 4.1 *Given a Horn theory $\Sigma(f)$ on n variables $X = \{x_1, x_2, \ldots, x_n\}$, an assumption set $A \subseteq X$ and a positive literal x_q, an explanation $E \subseteq A$ for f is also an explanation for $\Sigma(\exists_{X-A-\{x_q\}}f)$.*

Lemma 4.2 *Let $\Sigma(f)$ be a Horn theory on n variables $X = \{x_1, x_2, \ldots, x_n\}$. Given a set S $(\subseteq X)$, $\Sigma(\exists_S f)$ is also Horn.*

By combining Theorem 4.2 and Lemmas 4.1 and 4.2, we obtain the following theorem which improves our algorithm.

Theorem 4.3 *Given an OBDD of a Horn theory $\Sigma(f)$ on n variables $X = \{x_1, x_2, \ldots x_n\}$, an assumption set $A \subseteq X$ and a querry letter x_q, a non-trivial explanation for $\Sigma(f)$ can be obtained in $O(|f|^{2^{k+1}})$ time, where $|f|$ is the size of the given OBDD, and k is the size of the set $\{x_i \mid x_i \in (X - A - \{x_q\})$ and $\exists x_j \in A$ s.t. $\pi(x_i) < \pi(x_j)\}$.*

Now, we consider another approach for abduction; we specify a set of variables which must be included in the obtained explanations. This operation is NP-conmplete for the CNF-based case, while it can be done in polynomial time for the model-based case. When a theory is given as a Horn

Theorem 4.4 *Given a Horn theory $\Sigma(f)$ on n variables $X = \{x_1, x_2, \ldots x_n\}$, an assumption set $A = X$, a query letter x_q and a set $S \subseteq A$, a non-trivial explanation can be obtained in $O(|OBDD(f)|^2)$ time.*

5   Conclusion

In this paper, we have considered the problems of reasoning with general and Horn knowledge-bases, which are represented by OBDDs. We have shown that both finding a model of a knowledge-base and deducing from a knowledge-base can be done in linear time for general case, but that abduction is NP-complete even if the knowledge-base is restricted to be Horn. However, by introducing some constraints on a assumption set, abduction from Horn OBDDs can be done in polynomial time. The first case was that the assumption set contains all propositional literals. Moreover, the constraint on the variable ordering was relaxed to the case that there exist constant number of variables which are not in the assumption set, and which appear in higher levels than some variables in the assumption set. The second case was that a set of variables is given as a constraint that they should be included in the explanation.

References

[1] R.E. Bryant, "Graph-Based Algorithms for Boolean Function Manipulation," *IEEE Trans. Comput.*, C-35, no.8, pp.677–691, 1986.

[2] T. Eiter and G. Gottlob, "The Complexity of Logic-Based Abduction," *J. ACM*, 42(1), pp.3–42, 1995.

[3] T. Horiyama and T. Ibaraki, "Ordered Binary Decision Diagrams as Knowledge-Bases," *Proc. of 10th ISAAC*, eds. A. Aggarwal et al., LNCS 1741, Springer, pp.83–92, 1999.

[4] H.A. Kautz, M.J. Kearns, and B. Selman, "Reasoning with Characteristic Models," *Proc. of AAAI-93*, pp.34–39, 1993.

[5] R. Khardon and D. Roth, "Reasoning with Models," *Artificial Intelligence*, 87, pp.187–213, 1996.

[6] J.C. Madre and O. Coudert, "A Logically Complete Reasoning Maintenance System Based on a Logical Constraint Solver," *Proc. of IJCAI-91*, pp.294–299, 1991.

[7] S. Minato, N. Ishiura and S. Yajima, "Shared Binary Decision Diagram with Attributed Edges for Efficient Boolean Function Manipulation," *Proc. of 27th ACM/IEEE DAC*, pp.52–57, 1990.