ISCAS-LCS-10-24 August, 2010

中国科学院软件研究所
计算机科学实验室报告

Ternary Boolean Diagrams

by

Wenhui Zhang

State key Laboratory of Computer Science
Institute of Software
Chinese Academy of Sciences
Beijing 100190. China

Ternary Boolean Diagrams

Wenhui Zhang
State Key Laboratory of Computer Science
Institute of Software, Chinese Academy of Sciences
P.O.Box 8718, Beijing 100190, China

18 August 2010

1 Introduction

Binary decision diagrams (BDDs) were introduced for representation of switching circuits in [5] in 1959. The full potential for efficient algorithms based on the BDDs was investigated in [1] in 1986, in which a fixed variable ordering and shared sub-graphs are used for compressed canonical representation. Efficient boolean function manipulation has made BDDs been extensively used in logic synthesis and formal verification, in particular, for combating the state explosion problem of model checking, symbolic algorithms based on BDDs were introduced in [2] in 1990, and since then, BDDs have dominated the research on efficient symbolic algorithms for model checking. It is therefore of great importance to search for new bases for boolean function manipulation and formal verification. This document proposes ternary boolean diagrams (TBDs). The rest of his paper is organized as follows. The basic idea is presented in section 2, in which TBDs are introduced and their relation to boolean formulas is established. The technical development is presented in Section 3, in which ordered TBDs are introduced and operations on ordered TBDs are defined. Further refinement of the idea for efficient TBD manipulation is presented in Section 4, in which reduced ordered TBDs are introduced. An application of TBDs is presented in Section 5, in which boolean diagram model checking is introduced, which in this context refers to model checking based on boolean diagrams, in particular, reduced ordered TBDs. Concluding remarks are presented in Section 6 with a short summary of an experimental evaluation on boolean diagram model checking.

2 Ternary Boolean Diagrams

Let $\mathcal{L}$ be a set of labels. Let $\mathcal{L}^- = \{-x \mid x \in \mathcal{L}\}$. A boolean diagram over $\mathcal{L}$ is a graph with a root node and each node is assigned a label of $\mathcal{L} \cup \mathcal{L}^-$. A ternary boolean diagram is such a graph where the out degree of a node is either 3 or 0, and the out edges of a node is ordered such that the left, the middle and the right out edges of a node can be identified. Formally, ternary boolean diagrams are defined as follows.

Definition 1 (Ternary Boolean Diagram). *Let $\mathcal{L}$ be a set of labels. A ternary boolean diagram (TBD) over $\mathcal{L}$ is a quadruple*

$$(N, n_0, E, L)$$

where N is a set of nodes, $n_0 \in N$ is the root node, $E : N \to N^3$ is a partial function that defines the three out edges of a node, $L : N \to \mathcal{L} \cup \mathcal{L}^-$ is a labeling function which assigns each node a label of $\mathcal{L}$ or its negation.

The set of ternary boolean diagrams over $\mathcal{L}$ is denoted $\mathcal{D}(\mathcal{L})$. For simplicity, we fix an $\mathcal{L}$ and write $\mathcal{D}$ for $\mathcal{D}(\mathcal{L})$.

Notations When $E(n)$ is not defined for a node n, the value of $E(n)$ is denoted by a special triple $(\epsilon, \epsilon, \epsilon)$. A TBD t with n_0 as the root node may be represented by (x, a, b, c) when $L(n_0) = x, E(n_0) = (a, b, c)$. For the clearness of the presentation, additional notations to be used are as follows.

notation	meaning	condition
$x > 0$	$x \in \mathcal{L}$	
$x < 0$	$x \in \mathcal{L}^-$	
$L(t)$	x	where $t = (x, a, b, c)$
$--x$	x	where $x > 0$
$\lvert x \rvert$	x	where $x > 0$
$\lvert x \rvert$	$-x$	where $x < 0$
$-(x, a, b, c)$	$(-x, a, b, c)$	where $x > 0$ or $x < 0$
$x \cdot t$	t	where $x > 0$
$x \cdot t$	$-t$	where $x < 0$
τ_x	$(x, \epsilon, \epsilon, \epsilon)$	where $x > 0$
$\tau_x^\sim$	$(x, \epsilon, \epsilon, \epsilon)$	where $x > 0$ or $x < 0$

2.1 Langauge, Model and Equivalence

Let $\Sigma = (\mathcal{L} \cup \mathcal{L}^-)^*$.

Definition 2. *Let $r \in \mathcal{D}$ and $\sigma = x_1 \cdots x_k \in \Sigma$. σ is accepted by r, denoted $\sigma \models r$, iff there is an $x \in \mathcal{L}$ such that $r\lceil_{x_1} \cdots \lceil_{x_k} = \tau_x$ in which $r\lceil_n$ is defined as follows:*

	$r\lceil_n$ *where* $n < 0$	$r\lceil_n$ *where* $n > 0$
$r = \tau_x^\sim$	r	r
$r = (v, \tau_x^\sim, \tau_y^\sim, \tau_z) \wedge \lvert v \rvert = \lvert n \rvert$	$v \cdot \tau_x^\sim$	$v \cdot \tau_y^\sim$
$r = (v, \tau_x^\sim, \tau_y^\sim, -\tau_z) \wedge \lvert v \rvert = \lvert n \rvert$	$v \cdot -\tau_z$	$v \cdot -\tau_z$
$r = (v, s, t, u)$	$(v, s\lceil_n, t\lceil_n, u\lceil_n)$	$(v, s\lceil_n, t\lceil_n, u\lceil_n)$

such that the last rule is applied only when the other rules are not applicable.

Definition 3. *Let $r \in \mathcal{D}$ and $m \subseteq \mathcal{L}$. m is a model of r, denoted $m \models r$, iff there is a sequence $\sigma \in \Sigma$ such that $x \in \sigma$ implies $x > 0 \wedge x \in m$ or $x < 0 \wedge -x \notin m$, and $\sigma \models r$.*

r is complete iff for all $m \subseteq \mathcal{L}$, m is a model of r.

Definition 4. *Let $s, t \in \mathcal{D}$ be TBDs. s is equivalent to t, denoted $s \equiv t$, iff for all $m \subseteq \mathcal{L}$, $m \models s$ iff $m \models t$.*

2.2 Boolean Formulas as TBDs

Let $\Phi(S)$ denote the set of boolean formulas with variables in S. Let $\mathcal{L}_I \subseteq \mathcal{L}$ be a subset of $\mathcal{L}$.

Definition 5. *Let $\varphi \in \Phi(\mathcal{L}_I)$ and $m \subseteq \mathcal{L}$. $m \models \varphi$ iff φ evaluates to true under the assignment such that $x_i = 1$ iff $x_i \in m$.*

Definition 6. *Let $\varphi \in \Phi(\mathcal{L}_I)$ and $r \in \mathcal{D}$. φ is equivalent to r, denoted $\varphi \equiv r$, iff for all $m \subseteq \mathcal{L}$, $m \models \varphi$ iff $m \models r$.*

3 Ordered TBDs

Let $\leq$ be a linear order on $\mathcal{L}$. $x < y$ iff $x \leq y$ and $x \neq y$. Let $\mathcal{L} = \{x_1, ..., x_n, x_{n+1}\}$ such that $x_i < x_{i+1}$ for $i = 1, ..., n$. Let $\tau = \tau_{x_{n+1}} = (x_{n+1}, \epsilon, \epsilon, \epsilon)$.

Definition 7. *The set $\mathcal{D}^{\mathcal{O}} \subseteq \mathcal{D}$ of ordered TBDs is defined as follows. $s \in \mathcal{D}^{\mathcal{O}}$ iff $s = \tau$, $s = -\tau$, or $s = (x, a_0, a_1, a_2) \in \mathcal{D}$ where a_0, a_1, a_2 are ordered and for all $i \in \{0, 1, 2\}$, $|x| < |L(a_i)|$.*

3.1 Operations

Negation: Let $s = (x, a, b, c) \in \mathcal{D}^{\mathcal{O}}$. The negation of s, denoted $\neg s$, is computed by $\neg s = (-x, a, b, c)$.

Conjunction: Let $s \in \{\tau, -\tau, (x, a, b, c)\}$, $t \in \{\tau, -\tau, (x', a', b', c')\}$ be elements of $\mathcal{D}^{\mathcal{O}}$. The conjunction of s and t, a commutative operation, denoted $s \wedge t$, is computed as follows.

	$s \wedge t$
$t = \tau$	s
$t = -\tau$	t
$x > 0 \wedge x' > 0 \wedge x = x'$	$(x, a \wedge a', b \wedge b', c \wedge c')$
$x < 0 \wedge x' < 0 \wedge x = x'$	$(x, \neg(\neg(a \wedge c) \wedge \neg(a' \wedge c')), \neg(\neg(b \wedge c) \wedge \neg(b' \wedge c')), \tau)$
$x > 0 \wedge x' < 0 \wedge \lvert x\rvert = \lvert x'\rvert$	$(x, a \wedge \neg(a' \wedge c'), b \wedge \neg(b' \wedge c'), c)$
$x > 0 \wedge \lvert x\rvert < \lvert x'\rvert$	$(x, a, b, c \wedge t)$
$x < 0 \wedge \lvert x\rvert < \lvert x'\rvert$	$(x, \neg(\neg a \wedge t), \neg(\neg b \wedge t), \neg(\neg c \wedge t))$

The cases not covered above, including $s \in \{\tau, -\tau\}$, $x < 0 \wedge x' > 0 \wedge |x| = |x'|$, and $|x| > |x'|$, are computed by $s \wedge t = t \wedge s$, since $\wedge$ is defined to be a commutative operation.

Abstraction: Let $r \in \{\tau, -\tau, (v, s, t, u)\}$ be an element of $\mathcal{D}^{\mathcal{O}}$. The abstraction of r on x, denoted $abs(x)(r)$, is computed by $abs(x)(r) = r|_{-x} \wedge r|_x$ in which $r|_n$ is computed as follows:

	$r\|_n$ when $n < 0$	$r\|_n$ when $n > 0$
$r \in \{\tau, -\tau\}$	r	r
$\|v\| = \|n\|$	$v \cdot (s \wedge u)$	$v \cdot (t \wedge u)$
$\|v\| \neq \|n\|$	$(v, s\|_n, t\|_n, u\|_n)$	$(v, s\|_n, t\|_n, u\|_n)$

The existential abstraction of r on x, is then defined by $\neg abs(x)(\neg r)$.

Proposition 1. *let $s_1, s_2, t_1, t_2 \in \mathcal{D}^{\mathcal{O}}$ and $x \in \mathcal{L}$. If $s_1 \equiv s_2$ and $t_1 \equiv t_2$, then $\neg s_1 \equiv \neg s_2$, $s_1 \wedge t_1 \equiv s_2 \wedge t_2$ and $abs(x)(s_1) \equiv abs(x)(s_2)$.*

3.2 Observations

Completeness: The predicate *comp* on $\mathcal{D}^{\mathcal{O}}$ for observation of whether a TBD is complete is defined as follows: $comp(r) = (abs(x_1)abs(x_2) \cdots abs(x_n)(r) = \tau)$.

Proposition 2. *Let $r \in \mathcal{D}^{\mathcal{O}}$. r is complete iff $comp(r)$ holds.*

3.3 Boolean Formulas as Ordered TBDs

Let $\mathcal{L}_I$ defined in the previous section be restricted to a subset of $\mathcal{L} \setminus \{x_{n+1}\}$.

Proposition 3. *Let $x, \varphi_0, \varphi_1 \in \Phi(\mathcal{L}_I)$ and $s, t \in \mathcal{D}^{\mathcal{O}}$. If $x \in \mathcal{L}_I$ is an atomic formulas, then $x \equiv (x, -\tau, \tau, \tau)$, and if $\varphi_0 \equiv s$ and $\varphi_1 \equiv t$, then $\neg\varphi_0 \equiv \neg s$ and $\varphi_0 \wedge \varphi_1 \equiv s \wedge t$.*

Proposition 4. *Let $\varphi \in \Phi(\mathcal{L}_I)$ and $s \in \mathcal{D}^{\mathcal{O}}$. If $\varphi \equiv s$, then $\forall x.\varphi \equiv abs(x)(s)$.*

Proposition 5. *Let $\varphi \in \Phi(\mathcal{L}_I)$ and $s \in \mathcal{D}^{\mathcal{O}}$. If $\varphi \equiv s$, then φ is valid iff s is complete.*

4 Reduced Ordered TBDs

Definition 8. *The set of reduced ordered TBDs, denoted $\mathcal{D}^{\mathcal{R}}$, is defined as follows. $\tau \in \mathcal{D}^{\mathcal{R}}$, $-\tau \in \mathcal{D}^{\mathcal{R}}$, and $(x, a, b, c) \in \mathcal{D}^{\mathcal{R}}$ iff the following conditions hold:*

$c \neq -\tau$	
$b = -\tau \wedge c = \tau$	$\Rightarrow a = \tau$
$a = -\tau \wedge c = \tau$	$\Rightarrow b = \tau$
$a = b$	$\Rightarrow a, c \notin \{\tau, -\tau\}$
$c \in \{a, b\}$	$\Rightarrow c = \tau \wedge x > 0$

Let $x \in \mathcal{L} \cup \mathcal{L}^-$ and $y \in \mathcal{L}$. Let R be a rewrite system with the following set of rules.

$(x, -\tau, -\tau, c) \to (x \cdot -\tau)$	$(x, a, b, -\tau) \to (x \cdot -\tau)$
$(x, \tau, \tau, c) \to (x \cdot c)$	$(x, a, a, \tau) \to (x \cdot a)$
$(x, a, -\tau, \tau) \to (x, \tau, -\tau, a)$	$(x, -\tau, b, \tau) \to (x, -\tau, \tau, b)$
$(x, a, c, c) \to (x, a, \tau, c)$	$(x, c, b, c) \to (x, \tau, b, c)$
$(-y, \tau, b, \tau) \to (y, -\tau, -b, \tau)$	$(-y, a, \tau, \tau) \to (y, -a, -\tau, \tau)$

Let $s \xrightarrow{*} t$ denotes that s rewrites to t in 0 or more steps.

Proposition 6. *For every $s \in \mathcal{D}^{\mathcal{O}}$, there exists $t \in \mathcal{D}^{\mathcal{R}}$, such that $s \xrightarrow{*} t$ and $s \equiv t$, and in addition, if $s \xrightarrow{*} t'$ and $t' \in \mathcal{D}^{\mathcal{R}}$, then $t = t'$.*

5 Boolean Diagram Model Checking

A state in a finite state transition system can be represented by an assignment of a set of boolean variables. Suppose that $\{v_1, ..., v_m\}$ is such a set of variables. Let $\Phi(x_1, ..., x_k)$ denote the set of boolean formulas with variables in $\{x_1, ..., x_k\}$. Then a finite state transition system can be represented by a formula $I \in \Phi(v_1, ..., v_m)$ representing the set of initial states and a formula $R \in \Phi(v_1, ..., v_m, v'_1, ..., v'_m)$ representing the transition relation.

Let $\vec{v}$ and $\vec{v}'$ denote respectively $v_1, ..., v_m$ and $v'_1, ..., v'_m$. Let φ, ψ be CTL formulas with boolean variables in $\{v_1, ..., v_m\}$. By interpreting a formula as a set of states, the set of states of the transition system (I, R) satisfying a CTL formula can be computed as follows [4]:

$EX\varphi$	$= \exists \vec{v}'.(R \wedge \varphi^{\vec{v}'}_{\vec{v}})$
$EF\varphi$	$= \mu Z(\varphi \vee EXZ)$
$EG\varphi$	$= \nu Z(\varphi \wedge EXZ)$
$E(\varphi U \psi)$	$= \mu Z(\psi \vee (\varphi \wedge EXZ))$
$E(\varphi R \psi)$	$= \nu Z(\psi \wedge (\varphi \vee EXZ))$

Let $\mathcal{L} = \{v_1, ..., v_m, v'_1, ..., v'_m, z\}$. Let $\leq$ be a linear order on $\mathcal{L}$ such that z is the largest element of L. Let $\mathcal{L}_I = \{v_1, ..., v_m, v'_1, ..., v'_m\}$.

For transformation of the problem of checking whether the finite state transition system (I, R) satisfies a CTL formula into the problem of manipulation of ordered TBDs, Proposition 3 provides a way to represent a formula of $\Phi(v_1, ..., v_m, v'_1, ..., v'_m)$ by an equivalent ordered TBD of $\mathcal{D}(\mathcal{L})$; Proposition 4 establishes the relation between quantified boolean formulas and TBD abstraction; Proposition 5 together with Proposition 2 provides a way for checking whether the set of states represented by a formula is empty. This is sufficient for the transformation of the problem.

For efficient manipulation of TBDs, Proposition 6 established the relation between ordered TBDs and reduced ordered TBDs, and together with Proposition 1, it provides the possibility for the use of reduced ordered TBDs instead of ordered TBDs.

6 Evaluation and Concluding Remarks

Ternary boolean diagrams are introduced and their connection to boolean formulas has been established. A connection of ternary boolean diagrams to the problem of model checking finite state transition systems is also established.

Evaluation Experimental evaluation[1] of the efficiency of boolean function manipulation with TBDs and that with BDDs has been carried out based on an experimental comparison of the efficiency of a plain implementation of boolean diagram model checking in *verds* version 1.30 and the BDD based symbolic model checking implemented in NuSMV version 2.5.0 [3]. The evaluation was based on two types of random boolean programs and a set of 24 CTL formulas which includes formulas with nested CTL operators. The experimental data show that *verds* has advantages in all 24 cases with the first type of random programs, and advantages in 20 of 24 cases with the second type of random programs. Although NuSMV has advantages in 4 of 24 cases in the latter case, the advantages decrease as the size of the problem increases. On the other hand, in most cases where *verds* has advantages, the differences seem to yield a coefficient of an exponential factor in the number of boolean variables (when testing a property with a set of models of different sizes).

Note: The aforementioned implementation of boolean diagram model checking is a plain one in the sense that there are no use of cone of influence reduction, partitioned transition relation, or sophisticated relational product computation (the implemented computation is straightforward: first conjunction and then existential abstraction). It is expected that the integration of such techniques [4] will greatly increase the efficiency of boolean diagram model checking.

References

1. R. Bryant. Graph based algorithms for boolean function manipulation. IEEE Transaction on Computers 35(8):677-691. 1986.
2. J. R. Burch, E. M. Clarke, K. L. McMillan, D. L. Dill, and J. Hwang. Symbolic model checking: 10^{20} states and beyond. LICS 1990: 428-439.
3. A. Cimatti, E. M. Clarke, F. Giunchiglia, M. Roveri. NUSMV: A New Symbolic Model Verifier. CAV 1999: 495-499.
4. E. M. Clarke, O. Grumberg and D. Peled. Model Checking. The MIT Press. 1999.
5. C. Y. Lee. Representation of Switching Circuits by Binary-Decision Programs. Bell Systems Technical Journal 38: 985-999. 1959.

[1] Details are available at http://lcs.ios.ac.cn/∼zwh/verds/