Reliability Engineering and System Safety 126 (2014) 107–115

Contents lists available at ScienceDirect

Reliability Engineering and System Safety

journal homepage: www.elsevier.com/locate/ress

A novel decision diagrams extension method

Shumin Li, Shubin Si*, Hongyan Dui, Zhiqiang Cai, Shudong Sun

Department of Industrial Enginnering, School of Mechatronics Engineering, Northwestern Polytechnical University, Xi'an 710072, Shaanxi, PR China

A R T I C L E I N F O

Article history:
Received 7 May 2013
Received in revised form
17 January 2014
Accepted 20 January 2014
Available online 30 January 2014

Keywords:
Binary decision diagram
Multi-state multi-valued decision diagram
Decision diagrams extension method

A B S T R A C T

Binary decision diagram (BDD) is a graph-based representation of Boolean functions. It is a directed acyclic graph (DAG) based on Shannon's decomposition. Multi-state multi-valued decision diagram (MMDD) is a natural extension of BDD for the symbolic representation and manipulation of the multi-valued logic functions. This paper proposes a decision diagram extension method based on original BDD/MMDD while the scale of a reliability system is extended. Following a discussion of decomposition and physical meaning of BDD and MMDD, the modeling method of BDD/MMDD based on original BDD/MMDD is introduced. Three case studies are implemented to demonstrate the presented methods. Compared with traditional BDD and MMDD generation methods, the decision diagrams extension method is more computationally efficient as shown through the running time.

1. Introduction

Considerable research efforts have been developed in modeling and analyzing the binary-state systems. The theory and tools are well established, including fault tree, reliability block diagram, and binary decision diagram (BDD) [1–3]. Lee [4] first defined binary-decision program as a program that contains a string of two-address conditional transfer instructions; the corresponding processor of binary-decision programs was called binary decision machine (BDM) [5]; Akers [6] introduced BDD by representing Boolean functions as decision graphs. Based on references [4,6], Bryant investigated the full potential for efficient algorithms based on reduced ordered BDD (ROBDD) in 1986 [7], ROBDD first encodes the model in a compact data structure, and has time complexity proportional to the sizes of the graphs being operated on. As a well-known alternative to the minimal cut sets approach to assess the reliability Boolean models, BDDs has been investigated to solve large fault trees: Rauzy [3] introduced a BDD-based method for fault tree management, which supplies the efficient computation of both the minimal cut sets of a fault tree and the probability of its top event; Way and Hsia [8] developed a special method to build a BDD encoding fault tree by decomposing the fault tree into components; the size of a BDD structure depends critically on variable ordering and the determination of an appropriate variable ordering has a highly heuristic nature [9,10]; Jung, Han and Ha [11] presented a coherent BDD algorithm for coherent fault trees, the construction of the BDD structure was simplified by subsuming subset and truncating if-then-else (ITE) connectives with a probability or size limit in the intermediate BDD structure; Remenyte-Prescott and Andrews [12] conversed the fault tree to BDD by ordering the basic events and merging sub-BDDs to represent a parent gate; Ibáñez-Llano, Rauzy, Meléndez and et al. [13] reduced the BDD model through the set of the most relevant minimal cut sets. BDD has also been applied in the areas of reliability analysis of phased-mission systems (PMS): Xing and Levitin [14] used BDD in the reliability evaluation of non-repairable binary PMS with common-cause failures (CCF); Zhang, Sun and Trivedi [15] presented a new PMS-BDD algorithm, phase algebra was used to deal with the dependence across the phases, and the phase algebra was incorporated by a new BDD operation.

To apply BDD to the reliability analysis of multistate systems (MSS) subject to imperfect coverage behavior, Refs. [16–18] combined the BDD model with the multistate concept, Zhang, Sun and Trivedi [17] generated and stored minimum file spanning trees (MFST) via BDD manipulation using the multistate concept, BDDs were used for dependability analysis of distributed computer systems (DCS) with imperfect coverage. In this multistate BDD (MBDD)-based method, each state of the component is represented using a Boolean variable, which indicates whether the component is in the particular state or not, and the system BDD is generated with these Boolean variables.

BDD was extended to multiple-valued logic called a multiple-valued decision diagram (MDD) [19], Miller and Drechsler [20] described a matrix method for level interchange in MDDs. A wide range of research works have been done. In the area of compact Markov chain representation, Ciardo, Lüttgen and Siminiceanu [21] encoded the next-state function of a system as cross-products of integer functions; to minimize the required number of

* Corresponding author. Tel.: +86 29 8849 4701.
E-mail addresses: lsm@mail.nwpu.edu.cn (S. Li), sisb@nwpu.edu.cn (S. Si), duihongyan@mail.nwpu.edu.cn (H. Dui), caizhiqiang@nwpu.edu.cn (Z. Cai), sdsun@nwpu.edu.cn (S. Sun).

0951-8320/$ - see front matter © 2014 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.ress.2014.01.017

iterations of Markov chain state classification, Miner and Cheng [22] used multiway decision diagrams and edge-valued decision diagrams to represent sets of states and distance information respectively. In the area of Petri net reachability set generation and storage, Ciardo [23] described recently-introduced data structures and algorithms particularly targeted to Petri nets and similar asynchronous models; inspired by previous work on binary and multi-valued decision diagrams, a concept of locality for the effect of a transition's firing was used for the generation and storage of the reachability set of a Petri net [24]. In the area of symbolic model checking, Chechik, Gurfinkel, Devereux and et al. [25] proposed several choices for implementing multi-valued sets with decision diagrams. In the area of fault-tolerant systems, Xing and Dugan [26] used MDDs for dependability analysis of fault tolerant systems; ternary decision diagrams (TDD) [27] was used for the reliability evaluation of generalized phased-mission system (GPMS) with two-level modular imperfect coverage.

Recently, an algorithm based on multi-state multi-valued decision diagram (MMDD) [28] has been proposed to address the above state dependency problem [29]. Refs. [30,31] used MMDD in multi-state component importance analysis. Shrestha, Xing and Dai [32] did comprehensive complexity analysis, and performance comparisons among BDD, Logarithmically-encoded BDD (LBDD) and MDD. A MMDD is a logical extension of a BDD. Each MMDD is a direct acyclic graph (DAG) with two and only two sink nodes, representing the system being/not being in a specific state, respectively. Reference [29] has shown that the MMDD approach offered smaller model size using far fewer multi-valued variables which also offers less computational complexity in the model generation and evaluation than the BDD method.

BDDs and MMDDs for binary and multistate systems can be modeled with the traditional BDD and MMDD generation method respectively. However, the BDD/MMDD of the original system will no longer be applicable when the system structure changes and the new BDD/MMDD must be generated over the new system again. In this paper, we proposed an enhanced BDD/MMDD generation method based on original BDD/MMDD when a new component/system is serial-added or parallel-added to the original system. Case studies are presented to illustrate the application of the proposed method, and computational efficiency of the suggested method is compared with that of the traditional BDD and MMDD generation method.

The remainder of this paper is organized as follows: Section 2 describes the system discussed in this paper, and presents how the BDD responds to a component/system serial-added and parallel-added to the original system. Section 3 presents how the MMDD responds to a component/system serial-added and parallel-added to the original system. Section 4 gives a 4-step procedure for the decision diagrams extension method present in Sections 2 and 3. Section 5 analyses three case studies to validate the proposed method for responding to the simple expansion of original systems; and efficiency of the suggested method is compared with that of the traditional BDD and MMDD generation method. Finally, Section 6 presents conclusions.

2. Binary system and binary decision diagrams

2.1. Background

BDD is a compact encoding of the truth tables of Boolean formulae. The BDD is a reduced representation based on the Shannon decomposition: let F be a Boolean function that depends on the variable v; then F has the following equality representation.

$$F = v \cdot F(V \leftarrow 1) + \overline{v} \cdot F(V \leftarrow 0) \tag{1}$$

A formula can be represented as a binary tree graphically by choosing a total order over the variables of the truth table and applying recursively the Shannon decomposition. The nodes that are labeled with variables have two outgoing edges, the edge pointing to the node that encodes $F(V \leftarrow 1)$ is a then-outgoing edge; and, the other edge pointing to the node that encodes $F(V \leftarrow 0)$ is an else-outgoing edge. And the leaves are labeled with either 0 or 1. The value of the formula can be obtained by descending along the corresponding branch of the tree when a variable assignment given. The Shannon tree for the formula $F = ab + \overline{a}c$ is shown in Fig. 1 (solid lines represent then-outgoing edges while dashed lines represent else-outgoing edges).

Such a representation is very space consuming. We can shrink it by means of the following two reduction rules.

(1) Isomorphic subtrees merging. At least one isomorphic subtree is useless when two isomorphic subtrees encode the same formula.
(2) Useless nodes deletion. A node with two equal sons is useless since it is equal to its son ($F = v \cdot F + \overline{v} \cdot F$).

The BDD for the formula can be got by applying these two rules from top to bottom. This process is illustrated in Fig. 1.

Each BDD has two and only two sink nodes each labeled by a distinct logic value '0'/'1', representing the system not being/being in a particular state, respectively. Each non-sink node in BDD is associated with a Boolean variable, and has two outgoing edges as shown in Fig. 1, each edge corresponds to a certain state of the component.

BDDs of reliability systems can be modeled with this method. As researched before, the BDD must be modeled over the system again when the system changes. In this paper, a BDD/MMDD

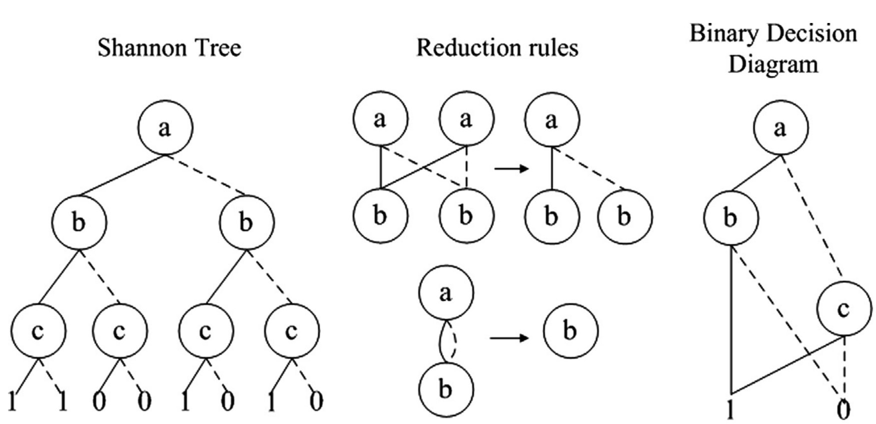

Fig. 1. From the Shannon tree to the BDD.