1

GRAPH-BASED REPRESENTATIONS OF DISCRETE FUNCTIONS
Shin-ichi MINATO

Abstract — BDDs are now commonly used for representing Boolean functions because of their efficiency in terms of time and space. There are many cases in which conventional algorithms can be significantly improved by using BDDs. Recently, several variants of BDDs have been developed to represent other kinds of discrete functions, such as multi-valued functions, cube sets, or arithmetic formulas. These techniques are useful not only for VLSI CAD but also for various areas in Computer Science.

In this chapter, we survey the techniques of BDD and its variants. We explain the basic method of BDD manipulation, and show the relationships between the different types of BDDs.

1.1 INTRODUCTION

Manipulation of discrete functions is fundamental to Computer Science. Many problems in digital system design and testing can be expressed as a sequence of operations on discrete functions. With the recent advance in VLSI technology, the problems have grown well beyond the scope of manual design, and VLSI CAD systems have become widely used. The performance of these systems greatly depends on the efficiency of the data structure for representing discrete functions.

Recently, BDDs (Binary Decision Diagrams) have attracted much attention because they enable us to manipulate Boolean functions efficiently in terms of time and space. There are many cases in which conventional algorithms can be significantly improved by using BDDs.
As our understanding of BDDs has deepened, the range of applications has broadened. Besides Boolean functions, we sometimes deal with other kinds of discrete functions, such as multi-valued functions, cube sets, or arithmetic formulas. Recently, several variants of BDDs have been devised to represent such data models. These techniques are useful not only for VLSI CAD but also for various areas in Computer Science.

In this chapter, we survey the techniques of BDD and its variants. We explain the basic method of BDD manipulation, and show the relationships between the different types of BDD. For detailed information on each method, refer to the original papers.

The rest of this chapter is organized as follows. In Section 1.2, we describe the basic concept of BDDs and the implementation techniques of the BDD manipulator. We then discuss the method of using attribute edges and variable ordering to reduce the size of BDDs. In Section 1.3, we present several variants of BDDs to represent multi-valued functions, that include MTBDDs, BDD vectors, EVBDDs and BMDs. Section 1.4 discusses the method of representing cube sets or combinatorial expressions. ZBDDs, FDDs, and KFDDs are described here. Section 1.5 is summary of this chapter.

1.2 BDDS

This section introduces the basic concept of BDDs, which is now commonly used for Boolean function representation. We discuss the techniques to manipulate BDDs on computers.

1.2.1 Data Structure

BDDs are graph representations of Boolean functions, as shown in Fig. 1.2.1(a). The basic concept was introduced by Akers [1], and an efficient manipulation method was developed by Bryant [2].

A BDD is a directed acyclic graph with two terminal nodes, which we call the 0-terminal and 1-terminal node. Every non-terminal node has an index to identify an input variable of the Boolean function, and has two outgoing edges, called the 0-edge and 1-edge.