IC/CAD Contest

Variable Ordering Optimization for Ordered Binary Decision Diagrams

Source: Springsoft Inc.

2003/01/06

I. Introduction

Ordered binary decision diagrams (OBDD's) [1] are a representation of Boolean function with application for verification, test pattern generation, synthesis and analysis of combinational and sequential circuits. A drawback of OBDD's is that the size of OBDD's depends on the order of variables for each specific application. It has been shown that improving the variable ordering of OBDD is NP-complete [2].

II. Input/Output Specification

The input is a combinational or sequential circuit in BLIF format [3]. You can implement a simple BLIF reader or use BLIF readers from other packages.

Here is an example of the input file which represents function f = (a+b)c.

```
.model example
.inputs a b c
.outputs f
.names a b c f
1-1   1
-11 1
.end
```

The output should contain an OBDD connection graph in DOT language, OBDD size, variable order, memory usage and CPU time. The DOT language and graph viewer can be found in [5]. In the document, you also need to mention "How to prove the correctness of your OBDD result".

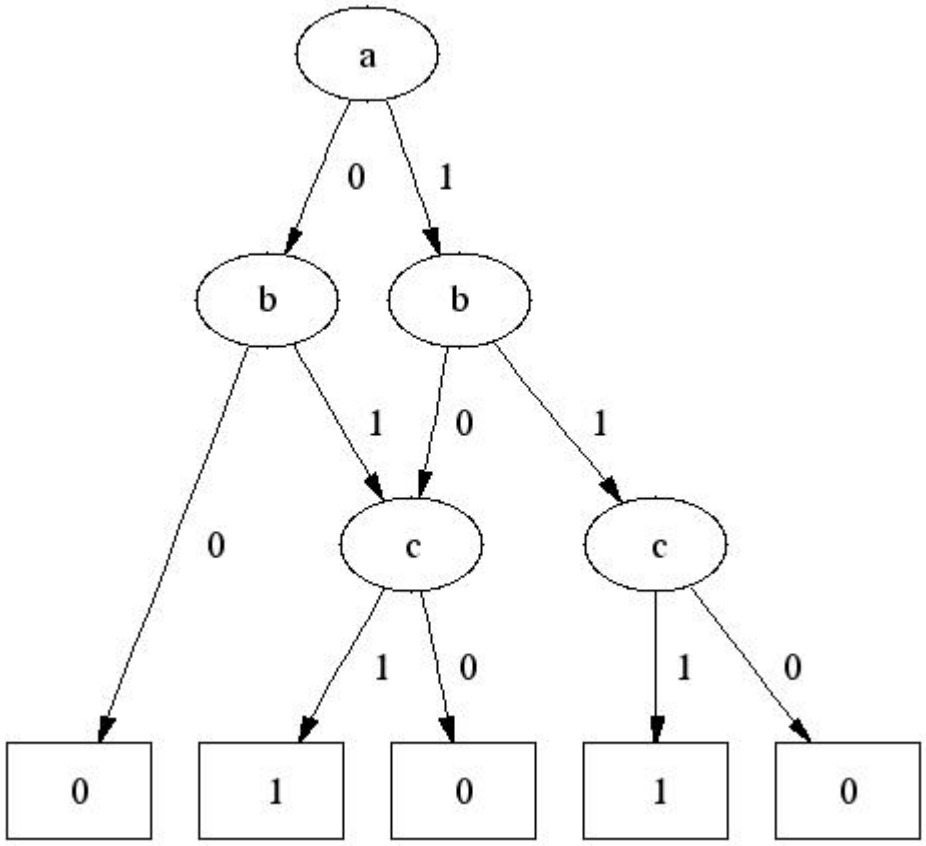
Here is an example of the OBDD in DOT language. The OBDD size is 5 and variable order is [a, b, c].

```
digraph OBDD_example {
        node [label = "a"];
        a_1;
        node [label = "b"];
        b_1; b_2;
        node [label = "c"];
```

```
        c_1; c_2;
        node [label = "0", shape = box];
        zero_1; zero_2; zero_3;
        node [label = "1", shape = box];
        one_1; one_2;

        {rank = source; a_1}
        {rank = same; b_1; b_2}
        {rank = same; c_1; c_2}
        {rank = same; zero_1; zero_2; zero_3;one_1; one_2}
        a_1 -> b_1 [label = "0"];
        a_1 -> b_2 [label = "1"];
        b_1 -> zero_1 [label = "0"];
        b_1 -> c_1 [label = "1"];
        c_1 -> zero_2 [label = "0"];
        c_1 -> one_1 [label = "1"];
        b_2 -> c_1 [label = "0"];
        b_2 -> c_2 [label = "1"];
        c_2 -> zero_3 [label = "0"];
        c_2 -> one_2 [label = "1"];
}
```

The diagram shown by dot is



III. Problem Statement

The goal is to find a variable order, which minimizes the size of OBDD. The results of BDD order can be obtained by **improving** existing BDD ordering or reordering packages. Note that all public domain packages which are used in the program must be clearly referenced in the final report.

IV. Advanced Features

1. Good GUI to view the result OBDD.

V. Language/Platform

1. Language: C or C++.
2. Platform: SUN OS/Solaris or PC DOS/Windows.

VI. Evaluation

The score will be given based on

1. Correctness, time and memory consumption;
2. The size of the OBDD;

Bonus will be rewarded if the advanced feature is done.

VII. Questions

Please report any questions regarding this problem to cad@cis.nctu.edu.tw with the email subject "CAD Contest: Problem 2." Your question(s) will be answered in two weeks, and the Q&A's will be posted at the contest web site

References

[1] R. E. Bryant, "Graph-based Algorithms for Boolean Function Manipulation," *IEEE Trans. Computers*, Vol. 35. No. 8, pp. 677-691, August 1986.

[2] B. Bollig and I. Wegener, "Improving the Variable Ordering of OBDDs is NP-Complete," *IEEE Trans. Computers*, Vol. 45, No. 9, September 1996.

[3] Berkeley Logic Interchange Format (BLIF), http://www-cad.eecs.berkeley.edu/Respep/Research/vis/blif.ps

[4] CMU BDD library http://www-2.cs.cmu.edu/~modelcheck/bdd.html.

[5] Graphviz - open source graph drawing software
http://www.research.att.com/sw/tools/graphviz/
http://www.research.att.com/sw/tools/graphviz/dotguide.pdf