Available online at www.sciencedirect.com
ScienceDirect
Progress in Natural Science 19 (2009) 489–494
Progress in Natural Science
www.elsevier.com/locate/pnsc

Deriving all minimal consistency-based diagnosis sets using SAT solvers

Xiangfu Zhao [a,b], Liming Zhang [a,b], Dantong Ouyang [a,b,*], Yu Jiao [a,b]

[a] Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University, Changchun 130012, China
[b] School of Computer Science and Technology, Jilin University, Changchun 130012, China

Received 27 April 2008; received in revised form 7 July 2008; accepted 7 July 2008

Abstract

In this paper, a novel method is proposed for judging whether a component set is a consistency-based diagnostic set, using SAT solvers. Firstly, the model of the system to be diagnosed and all the observations are described with conjunctive normal forms (CNF). Then, all the related clauses in the CNF files to the components other than the considered ones are extracted, to be used for satisfiability checking by SAT solvers. Next, all the minimal consistency-based diagnostic sets are derived by the CSSE-tree or by other similar algorithms. We have implemented four related algorithms, by calling the gold medal SAT solver in SAT07 competition – RSAT. Experimental results show that all the minimal consistency-based diagnostic sets can be quickly computed. Especially our CSSE-tree has the best efficiency for the single- or double-fault diagnosis.

Keywords: Conjunctive normal form; Consistency-based diagnosis; Model-based diagnosis; SAT solver

1. Introduction

Model-based diagnosis (MBD) is one of the active branches in artificial intelligence (AI). It plays an important role of test-bed of some approaches to knowledge representation and reasoning [1]. Its basic principle is to employ the model of a device to judge faults logically, according to the difference between the model's prediction and the actual observation.

Traditionally, conflict recognition, aiming at generating all minimal conflict sets, and candidate generation, aiming at generating all minimal hitting-sets, are of the two important steps towards the final diagnostic results. However, both of them are NP-complete problems. Therefore, the efficiency of each step greatly affects the final diagnosis.

In AI, most path-finding problems, notably AI planning [2] and model-checking [3], have been very successfully reduced to the propositional satisfiability problems (SAT). Recently, Grastien et al. [4,5] have also reduced diagnosis of discrete-event systems to SAT problems. Instead, in this paper, we consider determining a consistency-based diagnostic set and show how it can be reduced to SAT and efficiently solved by SAT solvers.

The structure of the paper is as follows: Some preliminaries of MBD and SAT are given in Section 2. Related algorithms are presented in Section 3. Some experimental results are given in Section 4. Some related work and comparisons are discussed in Section 5. Finally, conclusions and some future directions are put forward in Section 6.

2. Preliminaries

Firstly, let us introduce some definitions involved in model-based diagnosis.

Definition 1 [6]. A system is a triple ($SD, COMPS, OBS$), where SD (system description) is a set of first order sentences; $COMPS$ (system components) is a finite set of constants; OBS (system observation) is a finite set of first order sentences.

* Corresponding author. Tel.: +86 138 4300 6163.
E-mail address: ouyangdantong@163.com (D. Ouyang).

1002-0071/$ - see front matter
doi:10.1016/j.pnsc.2008.07.017

In the following, a unary predicate AB(·) is interpreted to mean "abnormal". AB(c) is true iff c is abnormal, where $c \in COMPS$.

Definition 2 ([6,7]). Given $\Delta \subseteq COMPS$, Δ is called a consistency-based diagnosis for $(SD, COMPS, OBS)$ if $SD \cup OBS \cup \{\neg AB(c) | c \in COMPS\text{-}\Delta\}$ is satisfiable.

A consistency-based diagnosis for $(SD, COMPS, OBS)$ Δ is a minimal one (MCBD), iff for no proper subset of Δ is a diagnosis for $(SD, COMPS, OBS)$.

In the following, let us have a look at SAT solvers. The purpose of a SAT solver is to accept a conjunctive normal formula (CNF) P, in clauses normal form, and then to evaluate whether P is true (satisfiable).

Recently, the field of SAT solving has advanced dramatically: a CNF containing hundreds of thousands or even millions of literals can now be handled by the state-of-the-art solvers, such as SATO [8] or zChaff [9] to name just two of the most popular solvers.

Any set of propositional forms can be transformed into a CNF. For instance, a set of propositional forms $\{A \to B, B \to C, \neg C, A\}$, can be described with a CNF file as follows (variables A, B, and C are denoted by 1, 2, and 3, respectively):

p cnf 3 4
−1 2 0
−2 3 0
−3 0
1 0

where in the first line, "p cnf" are keywords, "3" is the number of total variables, and "4" is the number of total clauses in this CNF file. Every negative number denotes that the corresponding variable is negative, for example, "−1" denotes "$\neg A$". "0" marks the end of a clause.

According to Definition 2 and SAT, now we can give the basic idea of judging whether a component set *SubCOMP* is a consistency-based diagnosis. Firstly, create a CNF file with all the component description clauses extracted from *SD* (in a CNF file) related to the component set (*COMPS* – *SubCOMP*), and all the corresponding clauses specifying the OK (i.e. ¬AB) mode for every related component. Then, call a SAT solver with this CNF. If "true" is returned, *SubCOMP* is a diagnosis.

3. Description of algorithms

In this section, we first show how a system model and observations can be described with CNF files. Next, an algorithm IsDiag is given to determine whether a subset of COMPS is a diagnosis. Then four variant CS-tree algorithms can derive all minimal diagnoses by calling IsDiag.

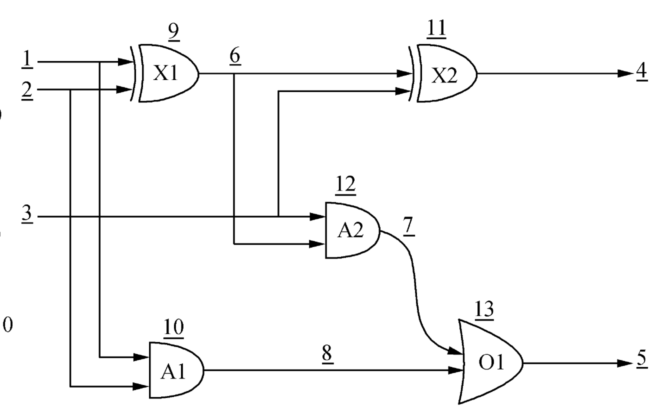
3.1. Modeling a system and observations with CNF

Given a system to be diagnosed, in contrast to the traditional approach [6], we use propositional logic for modeling. Not only the system components but also all the links between components are denoted by variables. For each component c, we use c and $\neg c$ to show that this component is in OK mode or in AB mode, respectively, for simplicity. Every component behavior is described with a propositional statement. All the component behavioral descriptions make up the system description: SD.cnf, a CNF file.

Let us take the fulladder shown in Fig. 1, for example, in which "1", "2" and "3" denote input variables; "4" and "5" denote output variables; "6", "7", and "8" denote all the internal link variables. All the gates, including XOR

SD.cnf

p cnf 13 17
−9 −1 −2 −6 0
−9 1 2 −6 0
−9 −1 2 6 0
−9 1 −2 6 0
−10 −1 −2 8 0
−10 1 −8 0
−10 2 −8 0
−11 −3 −6 −4 0
−11 3 6 −4 0
−11 −3 6 4 0
−11 3 −6 4 0
−12 −3 −6 7 0
−12 3 −7 0
−12 6 −7 0
−13 7 8 −5 0
−13 −7 5 0
−13 −8 5 0



Description of X1:	Description of A1:	Description of O1:
¬OK(X1)∨¬1∨¬2∨¬6	¬OK(A1)∨¬1∨¬2∨8	¬OK(O1)∨7∨8∨¬5
¬OK(X1)∨1∨2∨¬6	¬OK(A1)∨1∨¬8	¬OK(O1)∨¬7∨5
¬OK(X1)∨¬1∨2∨6	¬OK(A1)∨2∨¬8	¬OK(O1)∨¬8∨5
¬OK(X1)∨1∨¬2∨6		

Fig. 1. Fulladder and its SD.cnf.