Symbolic Models for Diagnosing Discrete-Event Systems

Anika Schumann[1], **Yannick Pencolé**[1] and **Sylvie Thiébaux**[1]

Abstract. We improve the efficiency of Sampath's diagnoser approach by exploiting compact symbolic representations of the system and diagnoser in terms of BDDs. We show promising results on test cases derived from a telecommunication application.

1 INTRODUCTION

A well-known approach to the diagnosis of discrete-event systems compiles, off-line, a centralised system model into another finite state machine, called diagnoser, which efficiently maps observations to possible failures [2]. While this approach exhibits excellent on-line performance, the space required by the centralised model, let alone that required by the diagnoser, constitutes a major problem. We aim to avoid this problem by representing and computing Sampath's diagnoser [2] symbolically, using compact representations of boolean functions $\mathcal{B}^n \mapsto \mathcal{B}$ as binary decision diagrams (BDDs) [1]. Starting from a symbolic representation of the system components in terms of BDDs, we compute the corresponding global model, abstract this model and retrieve the symbolic diagnoser using symbolic algorithms. The next 3 sections formally define these models and give their symbolic representation. Their symbolic computation presented in the full paper [4], to which we refer for details.

2 COMPONENT AND GLOBAL MODELS

Let $G_i = \langle X_i, \Sigma_o, \Sigma_u, \Sigma_f, x_{0_i}, T_i\rangle$, $i \in \{1, \ldots, n\}$, be a component model characterised by its states $X_i = \{x_{1_i}, \ldots, x_{m_i}\}$, its events $\Sigma = \{\sigma_1, \ldots, \sigma_p\}$ which are observable (Σ_o) or unobservable (Σ_u), its failure events $\Sigma_f \subseteq \Sigma_u$, its initial state x_{0_i}, and its transition relation $T_i \subseteq X_i \times \Sigma \times X_i$. For simplicity of the presentation and without loss of generality, we assume that all components share the same event set Σ. The thesis [3] shows how to handle events that are local to components. Symbolically, each component is encoded as[2] $G_i = \langle b_i^X, b_i^{X'}, b^\Sigma, X_i, \Sigma_o, \Sigma_u, \Sigma_f, x_{0_i}, T_i\rangle$, where $b_i^X = \{b_{1_i}^x, \ldots, b_{\lceil \log_2 m_i \rceil_i}^x\}$ are the state variables, $b_i^{X'} = \{b_{1_i}^{x'}, \ldots, b_{\lceil \log_2 m_i \rceil_i}^{x'}\}$ are the target variables, $b^\Sigma = \{b_1^\sigma, \ldots, b_{\lceil \log_2 p \rceil}^\sigma\}$ are the event variables, X_i is the boolean function over b_i^X characterising the states, Σ_o, Σ_f, and Σ_u are the boolean functions over b^Σ characterising the observable, unobservable, and failure events, x_{0_i} is the boolean function over b_i^X characterising the initial state, and T_i is the boolean function over $b_i^X \cup b^\Sigma \cup b_i^{X'}$ characterising the transition relation.

The global model $G = \langle X, \Sigma_o, \Sigma_u, \Sigma_f, x_0, T\rangle$ with its state set X, its sets of observable, unobservable and failure events Σ_o, Σ_u and Σ_f, its initial state x_0 and its transition set $T = \{(x_1, \ldots, x_n) \xrightarrow{\sigma} (x'_1, \ldots, x'_n) \mid \forall i \in 1 \ldots n, x_i \xrightarrow{\sigma} x'_i \in T_i\}$, is defined as the synchronous composition of the component models. Its symbolic representation is $G = \langle b^X, b^{X'}, b^\Sigma, X, \Sigma_o, \Sigma_u, \Sigma_f, x_0, T\rangle$, where $b^X = \cup_{i=1}^n b_i^X$, $b^{X'} = \cup_{i=1}^n b_i^{X'}$, $X = \wedge_{i=1}^n X_i$, $x_0 = \wedge_{i=1}^n x_{0_i}$ and $T = \wedge_{i=1}^n T_i$. The latter conjunction implements strong synchronisation by guaranteeing that the event variables, which are shared across components, have consistent values.

For example, consider the global model depicted in Figure 1: let $enc(u_1) = \overline{b_1^\sigma} \wedge \overline{b_2^\sigma} \wedge \overline{b_3^\sigma}$, $enc(x_1) = \overline{b_1^x} \wedge \overline{b_2^x} \wedge \overline{b_3^x}$ and $enc(x'_2) = \overline{b_1^{x'}} \wedge \overline{b_2^{x'}} \wedge b_3^{x'}$ denote the encoding of event u_1, and states x_1 and x'_2 respectively. The transition $x_1 \xrightarrow{u_1} x'_2$ can then be encoded as $enc(x_1) \wedge enc(u_1) \wedge enc(x'_2)$, and BDD can be generated from the disjunction of all transitions which represents the component.

3 ABSTRACTED MODEL

To speed up the computation of the diagnoser, we first abstract the global model, noting that the diagnoser does not depend on unobservable events which are not failures, on the order of successive failures, and on the global states encountered within a sequence of unobservable events. The abstracted model consists only of those states that are the origin or target of an observable transition (and of the initial state). It has two types of transitions: (1) the observable transitions of the global model, and (2) the failure transitions, each of which is labelled with a *set* of failure events that has occurred on some path from the transition's origin state to its target state (see Fig. 1).

The abstracted model is $\widetilde{G} = \langle \widetilde{X}, \Sigma_o, \widetilde{F}, x_0, \widetilde{T}_o, \widetilde{T}_F\rangle$: the states are $\widetilde{X} = \{x_0\} \cup \{x \in X \mid \exists \sigma \in \Sigma_o, \exists x' \in X \text{ s.t. } x \xrightarrow{\sigma} x' \in T \text{ or } x' \xrightarrow{\sigma} x \in T\}$, the failure labels are $\widetilde{F} = 2^{\Sigma_f}$, the observable transitions are $\widetilde{T}_o = \{x \xrightarrow{\sigma} x' \in T \mid \sigma \in \Sigma_o \wedge x, x' \in \widetilde{X}\}$, and the failure transitions $\widetilde{T}_F \subseteq \widetilde{X} \times \widetilde{F} \times \widetilde{X}$ are defined as follows:

$$\{x_1 \xrightarrow{l} x_k \mid (\exists x, x' \in \widetilde{X}, \exists \sigma, \sigma' \in \widetilde{\Sigma}_o \text{ such that } (x_1 = \widetilde{x}_0 \text{ or } x \xrightarrow{\sigma} x_1 \in \widetilde{T}_o) \text{ and } x_k \xrightarrow{\sigma'} x' \in \widetilde{T}_o), \text{ and } (\exists \sigma_1 \ldots \sigma_k \in \Sigma_u, \exists x_2 \ldots x_{k-1} \in X \text{ such that } \forall j = 1 \ldots k, x_j \xrightarrow{\sigma_j} x_{j+1} \in T \text{ and } \sigma_j \in l \text{ iff } \sigma_j \in \Sigma_f)\}$$

Symbolically, $\widetilde{G} = \langle b^X, b^{X'}, b^\Sigma, b^F, \widetilde{X}, \Sigma_o, \widetilde{F}, x_0, \widetilde{T}_o, \widetilde{T}_F\rangle$ is encoded using the same boolean variables as the global model and an additional $|\Sigma_f|$ variables $b^F = \{b_1^f, \ldots, b_{|\Sigma_f|}^f\}$ needed for the failure transition labels in $\widetilde{F}$. There is a one to one correspondence between failure events and these variables. A failure transition label is encoded as a conjunction of literals over b^F whose signs depend on whether the corresponding failure belongs to the label. For instance, the failure label $\{f_1\}$ of the abstracted model shown in Figure 1 is encoded as $enc(\{f_1\}) = b_1^f \wedge \overline{b_2^f}$.

[1] Computer Sciences Laboratory, The Australian National University National ICT Australia. email: {anika, pencole, thiebaux}@csl.anu.edu.au.

[2] We give identical names to sets and to the corresponding boolean functions, e.g. Σ_o, X, etc. This should not cause confusion.

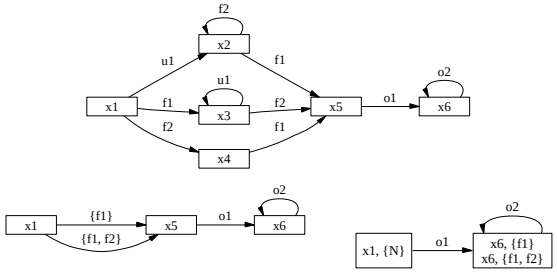

Figure 1. Global model (top) and its abstraction (bottom).
$\Sigma_u = \{u_1\}, \Sigma_f = \{f_1, f_2\}, \Sigma_o = \{o_1, o_2\}$

4 DIAGNOSER

A diagnoser is a deterministic finite state machine whose transitions correspond to observations and whose states correspond to the system states and failures that are consistent with the observations. More explicitly, its transitions are labelled with observable events, and its states are labelled with sets of pairs (v, l) denoting a state and a failure label of the abstracted model. On-line, the diagnoser efficiently maps observations to the possible system states and failures: it suffices to follow the path labelled by the actual observations and look up the label of the resulting diagnoser state.

Let $V = \{\widetilde{x_0}\} \cup \{x' \in \widetilde{X} \mid \exists \sigma \in \Sigma_o, \exists x \in \widetilde{X} \text{ s.t. } x \xrightarrow{\sigma} x' \in \widetilde{T}_o\}$ be the set of target states of observable transitions. The diagnoser is $\hat{G} = \langle \hat{X}, \hat{\Pi}, \Sigma_o, \hat{x}_o, \hat{R}, \hat{T} \rangle$, where $\hat{X} = \{\hat{x}_1, \ldots, \hat{x}_q\}$ is the set of diagnoser states, $\hat{\Pi} = V \times \widetilde{F}$ is the set of pairs that can belong to diagnoser state labels, $\hat{x}_0$ is the initial diagnoser state, $\hat{R} \subseteq \hat{X} \times \hat{\Pi}$ is the diagnoser state labelling relation which associates a state to the pairs in its label and verifies $\hat{R}(\hat{x}_0) = \{(\widetilde{x_0}, \emptyset)\}$ (by abuse of notation, we use the function notation $\hat{R}(\hat{x})$ for the label of state $\hat{x}$), and $\hat{T} \subseteq \hat{X} \times \Sigma_o \times \hat{X}$ is the set of diagnoser transitions, which verify: $\hat{x} \xrightarrow{\sigma} \hat{x}' \in \hat{T}$ iff

$$\begin{aligned}\hat{R}(\hat{x}') = \{(v', l') \mid \exists (v, l) \in \hat{R}(\hat{x}) \text{ such that} \\ \text{either } v \xrightarrow{\sigma} v' \in \widetilde{T}_o \text{ and } l' = l, \\ \text{or } \exists v \xrightarrow{l''} v'' \in \widetilde{T}_F \text{ and } \exists v'' \xrightarrow{\sigma} v' \in \widetilde{T}_o \text{ and } l' = l \cup l''\}\end{aligned}$$

$\hat{G} = \langle b^S, b^{S'}, b^X, b^{X'}, b^\Sigma, b^F, \hat{X}, \hat{\Pi}, \Sigma_o, \hat{x}_o, \hat{R}, \hat{T} \rangle$ is the symbolic diagnoser. It is encoded using the same boolean variables as the abstracted model, with an additional $2\lceil \log_2 q \rceil$ variables $b^S = \{b^s_1, \ldots, b^s_{\lceil \log_2 q \rceil}\}$ and $b^{S'} = \{b^{s'}_1, \ldots, b^{s'}_{\lceil \log_2 q \rceil}\}$ needed to encode the q diagnoser states. A complication here is that the number q of diagnoser states, and therefore the number of variables needed, is *a priori* unknown. In the worst case $q = 2^{|\hat{\Pi}|}$ and therefore $2|V||\widetilde{F}|$ new variables are theoretically needed. However, in practice, q will be much smaller and introducing that many variables will lead to an unnecessarily costly representation. To remedy this, we start with one single variable to encode the initial diagnoser state, and continually increase the number of variables, as needed during execution. Every time a new variable b^s_j is needed, we update all BDDs containing variables in b^S (resp. $b^{S'}$) by conjoining them with $\neg b^s_j$ (resp. $\neg b^{s'}_j$).

The diagnoser is described using two BDDs: one to represent the transitions over the variables $b^S \cup b^{S'} \cup b^\Sigma$ and one to represent the information of the individual states using the variables $b^S \cup b^X \cup b^F$.

5 RESULTS

Our approach has been implemented on top of the CUDD BDD package [5]. We present empirical evidence that our diagnoser representation yields important gains in space, taking a system consisting of a switch and two different control stations of a telecommunication network as example.

In this example, there are 9 observable events, 11 failure types, and 8 other unobservable events. The switch model has 12 states and 18 transitions, the primary control station 13 states and 15 transitions and the backup control station 19 states and 28 transitions. This yields a global model of 1062 states and 2911 transitions. To observe how the two approaches scale, we considered "lighter" versions of the example, where groups of failure types are fusioned. This yields 5 versions $V_1 \ldots V_5$, with a number of failure types ranging from 3 to 11. Table 1 depicts the version's diagnoser properties and its size.

Table 1 Diagnoser properties

	$\mathbf{V}_1$	$\mathbf{V}_2$	$\mathbf{V}_3$	$\mathbf{V}_4$	$\mathbf{V}_5$
States	353	921	2500	4355	18474
Transitions	2183	5774	16530	31024	120698
space symb. (Kb)	126	322	903	1696	7916
space enum. (Kb)	451	1531	7054	15851	172089

The superiority of the symbolic method increases with the model size, and exceeds an order of magnitude for the largest version. From the results, it can be conjectured that the space requirements of the symbolic approach for large models will often only represent a negligible portion of those of the enumerative setting. In [4], we make a similar observation about the time needed to generate the diagnoser.

It is also worth mentioning that the symbolic representation appears to preserve the real-time property of the diagnoser for on-line diagnosis: there was little variation in diagnosis time across the different example versions and the time taken to treat 1000 observations never exceeded 100 ms.

6 CONCLUSION AND FUTURE WORK

We have presented a symbolic framework based on BDDs for the diagnosis of discrete-event systems. It enables the synthesis of a symbolic version of Sampath's diagnoser [2], while requiring considerably lower space and time than the enumerative approach. This results from the fact that BDDs are suitable to compactly represent the large sets of system states and failures labelling the diagnoser states.

Our framework is not limited to fault diagnosis using diagnosers. In [3], we also give algorithms for checking diagnosability, as well as fault diagnosis algorithms based on all the models presented here. Our research agenda includes improving our results by experimenting with alternative encodings described in [3], dedicated heuristics for variable ordering. We also plan to extend our framework to stochastic system diagnosis using algebraic decision diagrams (see e.g. [5]). Finally, we shall investigate the use of symbolic representation in the context of decentralised diagnosis.

REFERENCES

[1] R. E. Bryant, 'Graph-based algorithms for boolean function manipulation', *IEEE Transactions on Computers*, **C-35**(8), 677–691, (1986).
[2] M. Sampath, R. Sengupta, S. Lafortune, K. Sinnamohideen, and D. Teneketzis, 'Failure diagnosis using discrete event models', *IEEE Transactions on Control Systems Technology*, **4**(2), 105–124, (1996).
[3] A. Schumann, *Is symbolic technology applicable to the diagnosis of discrete event systems?*, Master's thesis, University of Rostock, 2003.
[4] A. Schumann, Y. Pencolé, and S. Thiébaux, 'Diagnosis of discrete event systems using ordered binary decision diagrams', in *Fifteenth International Workshop on Principles of Diagnosis*, (2004).
[5] F. Somenzi. CUDD: CU Decision Diagram Package Release 2.3.0. University of Colorado at Boulder, 1998.