Diagnosability Testing with Satisfiability Algorithms

Jussi Rintanen and Alban Grastien
National ICT Australia Ltd and Australian National University
Canberra, Australia

Abstract

We show how testing whether a system is diagnosable can be reduced to the satisfiability problem and how satisfiability algorithms yield a very efficient approach to testing diagnosability.

Diagnosability is the question whether it is always possible to know whether a given system has exhibited a failure behavior. This is a basic question that underlies diagnosis, and it is also closely related to more general questions about the possibility to know given facts about system behavior.

The work combines the twin plant construct of Jiang et al., which is the basis of diagnosability testing of systems with an enumerative representation, and SAT-based techniques to AI planning which form a very promising approach to finding paths in very large transition graphs.

1 Introduction

Faults in dynamic systems can be diagnosed by observing a sequence of events taking place in the system and inferring the occurrence of unobservable failure events [Sampath *et al.*, 1995]. A main question arising in this setting is whether it is always possible to infer that a failure has occurred. A system that has this property is *diagnosable*. The diagnosability question for transition systems with a graph representation can be solved in polynomial time [Jiang *et al.*, 2001].

Many systems exhibit regularities best captured by representing them in terms of state variables, which also makes it possible to represent systems with very large state spaces succinctly without representing each state explicitly. The number of states of a succinctly represented system may be exponential in the size of its representation, which makes the diagnosability problem PSPACE-complete [Rintanen, 2007]. The diagnosability problem is similar to other PSPACE-complete problems like AI planning and LTL model-checking in that it reduces to finding paths in a graph. An efficient approach to solving AI planning and model-checking problems is to reduce them to the satisfiability problem of the classical propositional logic [Kautz and Selman, 1996; Biere *et al.*, 1999]. This suggests a similar approach to diagnosability testing, which is what we pursue in this work.

The structure of the paper is as follows. In Section 2 we present the transition system framework and formally define when a system is diagnosable. Section 3 contains the main contribution of the work, an encoding of the diagnosability problem as a formula in the classical propositional logic. In Section 4 we show how these formulae can be used to demonstrate that a system indeed is diagnosable, as opposed to showing that it is not. Section 5 contains a demonstration of the scalability of the approach to transition systems with very large state spaces, and Sections 6 and 7 conclude the paper by discussing related work and pointing out future research directions.

2 Preliminaries

We define transition systems following Sampath et al. [1995].

Definition 2.1 (Transition systems) *A transition system is a tuple $T = \langle X, \Sigma_o, \Sigma_u, \Sigma_f, \delta, s_0\rangle$ where*

- *X is a set of states,*
- *Σ_o is a set of* observable events,
- *Σ_u is a set of* unobservable events,
- *Σ_f is a set of* failure events,
- *$\delta \subseteq X \times (\Sigma_o \cup \Sigma_u \cup \Sigma_f) \times X$ is the transition relation,*
- *$s_0 \in X$ is an initial state.*

The transition system is initially in the state s_0, and an event sequence $e_0, \ldots, e_{n-1}$ takes the system through a sequence $s_0, s_1, \ldots, s_n$ of states such that $(s_i, e_i, s_{i+1}) \in \delta$ for all $i \in \{0, \ldots, n-1\}$. Note that a state s_i and an event e_i do not necessarily determine the successor state s_{i+1} uniquely.

Let $T = \langle X, \Sigma_o, \Sigma_u, \Sigma_f, \delta, s_0\rangle$ be a transition system. We say that $e_0, \ldots, e_{n-1}$ is *a sequence of events in T* if there are states $s_1, \ldots, s_n$ such that $(s_i, e_i, s_{i+1}) \in \delta$ for all $i \in \{0, \ldots, n-1\}$.

The state sequence nor the unobservable or the failure events can be observed. In deciding whether a failure has occurred only the sequence of observable events is available.

Let $\sigma \in (\Sigma_o \cup \Sigma_u \cup \Sigma_f)*$ be a sequence of events. We define its *projection* $\pi(\sigma)$ to observable events recursively as follows.

$$\begin{array}{rcl} \pi(\epsilon) & = & \epsilon \\ \pi(e\sigma) & = & \pi(\sigma) \text{ if } e \notin \Sigma_o \\ \pi(e\sigma) & = & e\pi(\sigma) \text{ if } e \in \Sigma_o \end{array}$$

Sampath et al. [1995] give a definition of diagnosability which we adapt to our notation.

Definition 2.2 (Diagnosability) *A transition system* $T = \langle X, \Sigma_o, \Sigma_u, \Sigma_f, \delta, s_0\rangle$ *is* diagnosable *if there is* d *such that for any sequence* σ *of events in* T *that ends in a failure event and for all sequences* $\sigma_1 = \sigma\sigma'$ *and* σ_2 *in* T *such that* $|\pi(\sigma')| \geq d$ *and* $\pi(\sigma_1) = \pi(\sigma_2)$, σ_2 *includes a failure event.*

Hence a system is diagnosable if and only if there are no two infinite event sequences which have the same observable events and one of them contains a failure event and the other one not. In other words, every infinite continuation of an event sequence with a failure is distinguishable from every infinite event sequence without a failure.

The constant d is called *the delay*. Not all failures can be detected immediately after they have taken place, and the delay expresses how many further events have to be observed before being certain that a failure has taken place.

2.1 Succinct System Representation

The structure of many systems is highly regular and it may be more practical to represent states in terms of state variables and the relations corresponding to events in terms of changes to the values of the state variables. This often makes it possible to represent very large systems compactly.

The set of states of a system consists of all the valuations of the state variables in a finite set A. In this paper we restrict to two-valued (Boolean) state variables. Hence a state $s : A \to \{0, 1\}$ is a total function from the state variables to the constants 1 (true) and 0 (false). A *literal* is a state variable or its negation, and the set of all literals is $L = A \cup \{\neg a | a \in A\}$. The language $\mathcal{L}$ over A consists of all formulae that can be formed from A and the connectives $\vee$ and $\neg$. We use the standard definitions of further connectives $\phi \wedge \psi \equiv \neg(\neg\phi \vee \neg\psi)$, $\phi \to \psi \equiv \neg\phi \vee \psi$ and $\phi \leftrightarrow \psi \equiv (\phi \to \psi) \wedge (\psi \to \phi)$.

The main design decision for succinct transition systems is how to represent transition relations. It would be possible to use arbitrary propositional formulae which is a powerful and general representation, but as our intention is to utilize independence of events for obtaining more efficient diagnosability testing, we have decided to use a more restricted representation that makes it possible to define what does it mean that two or more events take place simultaneously.

Definition 2.3 (Succinct transition systems) *A succinct transition system is a tuple* $\langle A, \Sigma_o, \Sigma_u, \Sigma_f, \delta, s_0\rangle$ *where*

- A *is a finite set of state variables,*
- Σ_o *is a set of* observable events,
- Σ_u *is a set of* unobservable events,
- Σ_f *is a set of* failure events,
- $\delta : \Sigma_o \cup \Sigma_u \cup \Sigma_f \to 2^{\mathcal{L}\times 2^L}$ *assigns each event a set of pairs* $\langle\phi, c\rangle$, *and*
- s_0 *is an initial state (a valuation of* A*).*

An event $e \in \Sigma_o \cup \Sigma_u \cup \Sigma_f$ is described by a set of pairs $\langle\phi, c\rangle$ which indicate that the event can be associated with changes c in states that satisfy the condition ϕ. More formally, an event $e \in \Sigma_o \cup \Sigma_u \cup \Sigma_f$ is possible in any state s such that $s \models \phi$ for some $\langle\phi, c\rangle \in \delta(e)$. When e takes place in s, one of the pairs $\langle\phi, c\rangle \in \delta(e)$ satisfying $s \models \phi$ is chosen and the effect of the event is that the literals in c become true.

Let s be a state and c a consistent set of literals. Then define the successor state $s' = \text{succ}(s, c)$ of s with respect to c by

1. $s'(a) = 1$ for all $a \in A$ such that $a \in c$,
2. $s'(a) = 0$ for all $a \in A$ such that $\neg a \in c$, and
3. $s'(a) = s(a)$ for all $a \in A$ that do not occur in c.

A succinct transition system can be mapped to a transition system as follows.

Definition 2.4 *Let* $T = \langle A, \Sigma_o, \Sigma_u, \Sigma_f, \delta, s_0\rangle$ *be a succinct transition system. Then define the transition system* $T' = R(T)$ *by* $R(T) = \langle X, \Sigma_o, \Sigma_u, \Sigma_f, \delta', s_0\rangle$ *where*

1. X *is the set of all valuations of* A *and*
2. $\delta' = \{(s, e, \mathit{succ}(s, c)) \in X \times (\Sigma_o \cup \Sigma_u \cup \Sigma_f) \times X | \langle\phi, c\rangle \in \delta(e), s \models \phi\}$.

2.2 Simultaneous Events

An important factor of efficient SAT-based planning [Kautz and Selman, 1996] is the notion of parallel or partially ordered plans. This means that several independent actions can be taken simultaneously, and it has the advantage that it is unnecessary to consider all $n!$ total orderings of n independent events as their mutual ordering does not matter.

Dependence is defined through the notion of *interference*. The pairs $\langle\phi_1, c_1\rangle$ and $\langle\phi_2, c_2\rangle$ *interfere* if there is $a \in A$ that occurs positively/negatively in c_1 and negatively/positively in ϕ_2 or in c_2, or positively/negatively in c_2 and negatively/positively in ϕ_1.

Events $e_1, \ldots, e_n$ can take place simultaneously with $o_1 \in \delta(e_1), \ldots, o_n \in \delta(e_n)$ if o and o' do not interfere for any $\{o, o'\} \subseteq \{o_1, \ldots, o_n\}$ such that $o \neq o'$.

We will consider diagnosability testing with sequences $E_1, \ldots, E_n$ of (possibly empty) sets E_i of event occurrences such that all members of E_i are mutually non-interfering. We define the successor state s' of s with respect to a set E of non-interfering event occurrences by $s' = \text{succ}(s, \bigcup_{\langle\phi,c\rangle\in E} c)$.

Let $\sigma \in (2^{\Sigma_o\cup\Sigma_u\cup\Sigma_f})*$ be a sequence of sets of events. Its *projection* $\pi(\sigma)$ to observable events is defined as follows.

$$\begin{aligned} \pi(\epsilon) &= \epsilon \\ \pi(E\sigma) &= (E \cap \Sigma_o)\pi(\sigma) \end{aligned}$$

For a succinct transition system we say that $E_0, \ldots, E_{n-1}$ is *a sequence of parallel events in* T if there are states $s_0, \ldots, s_n$ such that for all $i \in \{0, \ldots, n-1\}$ $s_{i+1} = \text{succ}(s_i, E)$ for some $E = \{o_1, \ldots, o_k\}$ such that there are $o_1 \in \delta(e_1), \ldots, o_k \in \delta(e_k)$ where $E_i = \{e_1, \ldots, e_k\}$ and o_h and o_j do not interfere for any $h \in \{1, \ldots, k\}$ and $j \in \{1, \ldots, k\}\backslash\{h\}$.

Diagnosability of succinct transition systems with simultaneous events is defined analogously to Definition 2.2. The

length $|(E_1, \ldots, E_n)|$ of sequences of sets of parallel events is defined as the sum $\sum_{i=1}^{n} |E_i|$ of the cardinalities of the sets. A small technical difference is caused by the fact that projection $\pi(E_1, \ldots, E_n)$ for sequences of parallel events always results in a sequence of the same length n (often with empty event sets.) It can be shown that the definitions are equivalent by interpreting a set of parallel events as any total ordering of the events.

3 Diagnosability as a Satisfiability Problem

Jiang et al. [2001] have shown how the diagnosability test can be reduced to finding a path in a graph. Their test is based on making the definition of diagnosability (Definition 2.2) finite by constructing a product transition system, sometimes called *the twin plant*, in which states are pairs $(s, \hat{s})$ of states of the original transition system, and events represent unobservable events in one or both of the components of these state pairs, or observable events shared by both components. If in this system there is an event sequence from (s_0, s_0) to some $(s, \hat{s})$ which includes a failure event in the first component but not in the second, and there is a non-empty event sequence back to $(s, \hat{s})$ with no failures in the second component, then a pair of infinite event sequences witnessing non-diagnosability exists. This reformulation of Definition 2.2 reduces infinite event sequences to cycles in a graph.

This diagnosability test can be formulated as a satisfiability problem in the classical propositional logic, similarly to other path finding problems in AI planning [Kautz and Selman, 1996]. We construct a formula for which the satisfiable valuations correspond to pairs $[(s_0, \ldots, s_n), (\hat{s}_0, \ldots, \hat{s}_n)]$ of state sequences with $s_0 = \hat{s}_0$ that correspond to pairs $[(E_0, \ldots, E_{n-1}), (\hat{E}_0, \ldots, \hat{E}_{n-1})]$ of event sequences such that $\pi(E_0, \ldots, E_{n-1}) = \pi(\hat{E}_0, \ldots, \hat{E}_{n-1})$ and of which one contains a failure event and the other does not, and which loop back to $(s_i, \hat{s}_i)$, that is, $s_n = s_i$ and $\hat{s}_n = \hat{s}_i$ for some $i \in \{0, \ldots, n-1\}$. The formula for a given event sequence length is satisfiable if and only if it is not possible to detect the occurrence of a failure event.

The encoding of state and event sequences is similar to the encoding of planning in the propositional logic [Kautz and Selman, 1996]. That the projections of both sequences to observable events coincide is guaranteed by forcing each observable event to take place in both sequences simultaneously. Essentially, each observable event is a joint event of both sequences.

Next we define the formula for diagnosability testing of a succinct transition system $T = \langle A, \Sigma_o, \Sigma_u, \Sigma_f, \delta, s_0 \rangle$. The events at each time point t are described by a formula $\mathcal{T}(t, t+1)$ parameterized with t. The propositional variables occurring in the formula, with superscripts t referring to different times points, are the following.

- a^t and $\hat{a}^t$ for all $a \in A$ and $t \in \{0, \ldots, n\}$.
- e_o^t for all $e \in \Sigma_o \cup \Sigma_u \cup \Sigma_f$ and $o \in \delta(e)$ and $t \in \{0, \ldots, n-1\}$.
- $\hat{e}_o^t$ for all $e \in \Sigma_o \cup \Sigma_u$, $o \in \delta(e)$ and $t \in \{0, \ldots, n-1\}$.
- e^t for all $e \in \Sigma_o$ and $t \in \{0, \ldots, n-1\}$.

Propositional variables with a hat $\hat{a}$ represent event occurrences and values of state variables in the second event sequence that does not contain failure events. The propositional variables e^t describe the occurrence of observable events simultaneously in both event sequences.

Next we describe $\mathcal{T}(t, t+1)$ for a given t. When an event occurs, the event must be possible in the current state and it has some effects.

$$e_o^t \rightarrow \phi^t \text{ for every } o = \langle \phi, c \rangle \in \delta(e)$$
$$e_o^t \rightarrow \textstyle\bigwedge_{l \in c} l^{t+1} \text{ for every } o = \langle \phi, c \rangle \in \delta(e)$$

The value of a state variable changes only if there is a reason for the change.

$$(a^t \wedge \neg a^{t+1}) \rightarrow \left(e_{1o_1}^t \vee \cdots \vee e_{ko_k}^t \right)$$

for all $a \in A$ where $o_1 = \langle \phi_1, c_1 \rangle, \ldots, o_k = \langle \phi_k, c_k \rangle$ are all event occurrences with $\neg a \in c_i$ and $e_1, \ldots, e_k$ are the respective events with $o_i \in \delta(e_i)$. For the change from false to true the formulae are defined similarly by interchanging a and $\neg a$.

An event can occur in only one way, and two events cannot be simultaneous if they interfere.

$$\neg(e_o^t \wedge e_{o'}^t) \quad \text{for all } e \in \Sigma_o \cup \Sigma_u \cup \Sigma_f \text{ and } \{o, o'\} \subseteq \delta(e) \text{ such that } o \neq o'$$
$$\neg(e_o^t \wedge e_{o'}'^t) \quad \text{for all } \{e, e'\} \in \Sigma_o \cup \Sigma_u \cup \Sigma_f \text{ and } o \in \delta(e) \text{ and } o' \in \delta(e') \text{ such that } o \text{ and } o' \text{ interfere}$$

The above formulae describe one step of an event sequence. We need to represent two event sequences, so we have a copy of all of the above formulae in which all propositional variables a^t and e_o^t have been replaced by $\hat{a}^t$ and $\hat{e}_o^t$ respectively. The second sequence is restricted to events in $\Sigma_o \cup \Sigma_u$. This is the key idea in the diagnosability test which makes it possible to identify two sequences of events that have the same observable events and only the first of which contains a failure event.

The formulae that connect the two event sequences require that observable events take place in both sequences whenever they take place.

$$(\textstyle\bigvee_{o \in \delta(e)} e_o^t) \leftrightarrow e^t \text{ for all } e \in \Sigma_o$$
$$(\textstyle\bigvee_{o \in \delta(e)} \hat{e}_o^t) \leftrightarrow e^t \text{ for all } e \in \Sigma_o$$

To avoid trivial cycles we require that at every time point at least one event takes place.

$$\bigvee_{e \in \Sigma_o} e^t \vee \bigvee_{e \in \Sigma_u \cup \Sigma_f, o \in \delta(e)} e_o^t \vee \bigvee_{e \in \Sigma_u, o \in \delta(e)} \hat{e}_o^t$$

The conjunction of all the above formulae for a given t is denoted by $\mathcal{T}(t, t+1)$. A formula for the initial state s_0 is

$$\begin{aligned} \mathcal{I}_0 \quad = \quad & \textstyle\bigwedge(\{a^0 \wedge \hat{a}^0 | a \in A, s_0(a) = 1\} \\ & \wedge \textstyle\bigwedge\{\neg a^0 \wedge \neg \hat{a}^0 | a \in A, s_0(a) = 0\}). \end{aligned}$$

We define a formula that finds a pair of infinite executions (in the form of a cycle) with the same observable events and a failure in one execution but not in the other.

$$\begin{aligned} \Phi_n^T \quad = \quad & \mathcal{I}_0 \wedge \mathcal{T}(0,1) \wedge \cdots \wedge \mathcal{T}(n-1, n) \wedge \\ & \textstyle\bigvee_{t=0}^{n-1} \bigvee_{e \in \Sigma_f} \bigvee_{o \in \delta(e)} e_o^t \wedge \\ & \textstyle\bigvee_{m=0}^{n-1} (\bigwedge_{a \in A} ((a^n \leftrightarrow a^m) \wedge (\hat{a}^n \leftrightarrow \hat{a}^m))) \end{aligned}$$

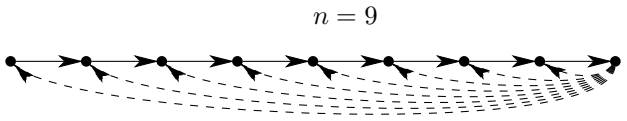

Figure 1: Parameter n in the diagnosability test with several potential cycle lengths

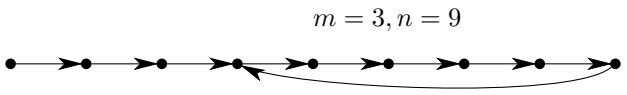

Figure 2: Parameters m and n in the diagnosability test

The possible cycles represented by Φ_n^T are illustrated in Figure 1. The parameter n controls the number of states in the sequence, and the last state must equal one of the preceding states so that a cycle is formed.

The diagnosability question can also be formalized with a fixed starting point m for the cycle (Figure 2.)

$$\begin{aligned}\Phi_{m,n}^T \quad = \quad & \mathcal{I}_0 \wedge \mathcal{T}(0,1) \wedge \cdots \wedge \mathcal{T}(n-1,n) \wedge \\ & \bigvee_{t=0}^{n-1} \bigvee_{e \in \Sigma_f} \bigvee_{o \in \delta(e)} e_o^t \wedge \\ & \bigwedge_{a \in A} ((a^n \leftrightarrow a^m) \wedge (\hat{a}^n \leftrightarrow \hat{a}^m))\end{aligned}$$

We show that the formulae Φ_n^T perform the diagnosability test correctly. Proofs for $\Phi_{m,n}^T$ are analogous.

Lemma 3.1 *Let $T = \langle A, \Sigma_o, \Sigma_u, \Sigma_f, \delta, s_0 \rangle$ be a succinct transition system and n a positive integer. The formula Φ_n^T is satisfiable if and only if there are sequences $\sigma = (E_0, \ldots, E_{n-1})$ and $\hat{\sigma} = (\hat{E}_0, \ldots, \hat{E}_{n-1})$ of parallel events in T such that*

1. *$(E_0 \cup \cdots \cup E_{n-1}) \cap \Sigma_f \neq \emptyset$,*
2. *$(\hat{E}_0 \cup \cdots \cup \hat{E}_{n-1}) \cap \Sigma_f = \emptyset$,*
3. *$E_i \cup \hat{E}_i \neq \emptyset$ for all $i \in \{0, \ldots, n-1\}$,*
4. *$\pi(\sigma) = \pi(\hat{\sigma})$, and*
5. *there are executions $s_0, \ldots, s_n$ of σ and $\hat{s}_0, \ldots, \hat{s}_n$ of $\hat{\sigma}$ in T such that $s_0 = \hat{s}_0$ and $s_m = s_n$ and $\hat{s}_m = \hat{s}_n$ for some $m \in \{0, \ldots, n-1\}$.*

Proof: Sketch: Proof of the equivalence from left to right is a demonstration that a valuation that satisfies Φ_n^T can be mapped to σ, $\hat{\sigma}$ and executions $s_0, \ldots, s_n$ and $\hat{s}_0, \ldots, \hat{s}_n$ that satisfy the required properties.

Proof from right to left is by constructing a valuation of all propositional variables in Φ_n^T based on σ, $\hat{\sigma}$ and the executions $s_0, \ldots, s_n$ and $\hat{s}_0, \ldots, \hat{s}_n$, and then showing that each conjunct of Φ_n^T is satisfied by that valuation. □

Theorem 3.2 *For a succinct transition system T, Φ_n^T is satisfiable for some $n \geq 1$ if and only if T is not diagnosable.*

Proof: Sketch: We show that T is not diagnosable iff for some n the right hand side of the equivalence in Lemma 3.1 holds.

Assume that T is not diagnosable. Hence there are infinite sequences $\sigma = (E_0, E_1, E_2, \ldots)$ and $\hat{\sigma} = (\hat{E}_0, \hat{E}_1, \hat{E}_2, \ldots)$ of parallel events in T such that $\pi(\sigma) = \pi(\hat{\sigma})$ and σ contains a failure event and $\hat{\sigma}$ does not. Let $s_0, s_1, s_2, \ldots$ and $\hat{s}_0, \hat{s}_1, \hat{s}_2, \ldots$ be the corresponding state sequences.

Since T has only a finite number of states, there are some m and n such that $m < n$ and $s_n = s_m$ and $\hat{s}_n = \hat{s}_m$ and $\Sigma_f \cap E_k \neq \emptyset$ for some $k \in \{0, \ldots, n-1\}$. Now $(E_0, \ldots, E_{n-1})$ contains a failure event and $(\hat{E}_0, \ldots, \hat{E}_{n-1})$ does not, and the sequences satisfy the requirements in the right hand side of the equivalence in Lemma 3.1 except for condition 3, which can be satisfied by deleting those E_i and $\hat{E}_i$ for which $E_i \cup \hat{E}_i = \emptyset$.

For the implication from right to left assume that the right hand side of Lemma 3.1 holds. Now the two sequences $(E_0, \ldots, E_{n-1})$ and $(\hat{E}_0, \ldots, \hat{E}_{n-1})$ yield the infinite sequences $\sigma = E_0, \ldots, E_{n-1}, E_m, \ldots, E_{n-1}, \ldots$ and $\hat{\sigma} = \hat{E}_0, \ldots, \hat{E}_{n-1}, \hat{E}_m, \ldots, \hat{E}_{n-1}, \ldots$ with $\pi(\sigma) = \pi(\hat{\sigma})$ so that σ contains a failure event and $\hat{\sigma}$ does not. Hence T is not diagnosable. □

Search for the proof of non-diagnosability with formulae Φ_n^T leads to a one-dimensional search problem for the value of n, and with $\Phi_{m,n}^T$ the problem is two-dimensional. The standard solution method in the one-dimensional case is to test the satisfiability of Φ_1^T, then Φ_2^T and so on, until a satisfiable formula is found. There are also parallelized algorithms that may find a satisfiable formula much faster than this sequential method [Rintanen *et al.*, 2006].

4 Showing Diagnosability

For a system that is actually diagnosable, the formulae in the previous section are not very practical for showing diagnosability because there is no simple way to guarantee that the parameter n is high enough so that all reachable states with a failure event have been covered. The most obvious upper bound for n is the cardinality of the set of all states, which is often impractically high: for q state variables one would be forced to consider $n = 2^{2q}$. Solutions to this problem have been proposed [Sheeran *et al.*, 2000] but they work for specific types of transition systems only or they step outside the SAT framework [McMillan, 2003].

However, in many cases it is not necessary to include a complete reachability test. It may be sufficient to use a formula ρ which gives an *upper bound* to the set of state pairs $(s, \hat{s})$ in the twin plant that could be reached by event sequences $\sigma = (E_0, \ldots, E_k)$ and $\hat{\sigma} = (\hat{E}_0, \ldots, \hat{E}_k)$ such that $\pi(\sigma) = \pi(\hat{\sigma})$ and $(\hat{E}_0 \cup \cdots \cup \hat{E}_k) \cap \Sigma_f = \emptyset$.

It may be possible to show that for no state pair satisfying ρ there is an event sequence starting with a failure event in the first component and another event sequence without failure events starting in the second component if both sequences must have the same observable events. This can be tested by using the following formula with increasing values of n.

$$\Psi_{n'}^{T,\rho} = \rho \wedge \mathcal{T}(0,1) \wedge \cdots \wedge \mathcal{T}(n-1,n) \wedge \bigvee_{e \in \Sigma_f, o \in \delta(e)} e_o^0$$

The formula is satisfiable iff after a failure in the first component of the twin plant (but not in the second) an event sequence of length n with the same observable events in both

components can take place. Hence, if the formula is unsatisfiable, the behavior after a failure necessarily differs from the behavior without a failure, and the system is diagnosable.

The incompleteness of this test is caused by the approximate nature of ρ: if ρ represents too many unreachable states of the twin plant, then it may appear that distinguishing between behavior with and without failure is not possible. Hence the unsatisfiability of the formula is a sufficient but not a necessary condition for diagnosability.

A complete positive and negative test for diagnosability is now obtained by testing the satisfiability of Φ_n^T for increasing n interleaved with testing the satisfiability of $\Psi_{n'}^{T,\rho}$ for increasing n' and increasingly strong upper approximations ρ of the reachable state pairs in the twin plant.

There are alternative ways how the formulae ρ for approximating reachability in the twin plant could be derived. One approach is by polynomial time algorithms for computing sets V of 2-literal clauses that are true in all reachable states [Rintanen, 1998]. If a state in the twin plant does not satisfy $\rho = \bigwedge V$ it is unreachable from the initial state. Other approaches to deriving tighter upper bounds ρ could be based on techniques for knowledge compilation [Selman and Kautz, 1996] and OBDDs [Bryant, 1992] or DNNF [Darwiche, 2001].

5 Experiments

The algorithm of Jiang et al. [2001] is impractical because it relies on the explicit enumeration of all pairs of states. For a transition system with n states, the twin plant consists of n^2 states. Starting from $n = 10000$ the size of the twin plant is too high for practically running their algorithm.

We demonstrate the much better scalability of our approach (Section 3) by using a system that consists of m identical interconnected components. Each component has 6 states. The number of states of the whole system with C components is therefore 6^C. There are some dependencies between the states of neighboring components which means that not all 6^C state combinations are actually possible. The twin plant in the diagnosability test has a quadratic number 6^{2C} of states, which becomes infeasible for explicit state enumeration starting from about $C = 8$ components.

Each component has 22 events of which 4 are observable. We tested the diagnosability of a faulty event in one component. To obtain bigger and bigger problems we increased the number of components. The parameter n for proving non-diagnosability was 4 irrespective of C, meaning that the shortest path with a cycle has length 4.

Statistics on formulae representing the diagnosability problem are given in Table 1. The formulae for $n = 3$ are unsatisfiable and for $n = 4$ they are satisfiable. After the first two columns for C and n we give the value of the formula (whether it is satisfiable or not), the number of propositional variables and clauses in the formula, and finally the time it took a SAT solver to test satisfiability.

The experiments were run with a 1.73 GHz Pentium 4 computer and the Siege SAT solver (version 4) [Ryan, 2003]. Since Siege is a randomizing SAT solver and the runtimes vary across different runs, we ran it 10 times with each formula and computed 95 per cent confidence intervals for the mean by using a standard boot-strapping method. These are shown in the last column of Table 1 next to the runtime.

C	n	val	vars	clauses	runtime
40	3	F	31683	427136	$0.01\,^{0.01}_{0.01}$
40	4	T	41764	568154	$0.35\,^{0.33}_{0.38}$
60	3	F	47523	640736	$0.01\,^{0.01}_{0.01}$
60	4	T	62644	852274	$1.94\,^{1.54}_{2.35}$
80	3	F	63363	854336	$0.01\,^{0.01}_{0.01}$
80	4	T	83524	1136394	$1.93\,^{1.75}_{2.08}$
100	3	F	79203	1067936	$0.01\,^{0.01}_{0.01}$
100	4	T	104404	1420514	$3.48\,^{3.15}_{3.82}$
120	3	F	95043	1281536	$0.03\,^{0.03}_{0.03}$
120	4	T	125284	1704634	$5.55\,^{5.26}_{5.83}$
140	3	F	110883	1495136	$0.03\,^{0.03}_{0.03}$
140	4	T	146164	1988754	$7.78\,^{7.32}_{8.22}$
160	3	F	126723	1708736	$0.03\,^{0.03}_{0.04}$
160	4	T	167044	2272874	$10.81\,^{9.90}_{11.69}$
180	3	F	142563	1922336	$0.06\,^{0.06}_{0.06}$
180	4	T	187924	2556994	$13.45\,^{12.22}_{14.57}$
200	3	F	158403	2135936	$0.01\,^{0.01}_{0.01}$
200	4	T	208804	2841114	$20.48\,^{18.56}_{22.90}$

Table 1: Runtimes for diagnosability testing. The number of components is C and the path length is n.

Unsurprisingly, there appears to be an exponential growth in the runtimes, but for our system non-diagnosability can be detected rather easily for systems with 160 or 200 components and an astronomic number of states. The propositional formulae are big, with one or two hundred thousand propositional variables and one or two million clauses, but efficient SAT solvers can utilize the regularity of the transition systems to find the cyclic path in the twin plant efficiently. For systems in which the components are more complicated or in which the cycles witnessing non-diagnosability are longer, the approach probably does not scale quite as far.

6 Related Work

Another approach to finding paths in graphs compactly represented in terms of state variables is based on ordered binary decision diagrams (OBDDs) [Bryant, 1992]. OBDDs were very popular in computer-aided verification, especially model-checking [Burch *et al.*, 1994], until the introduction of SAT based techniques [Biere *et al.*, 1999] following their success in AI planning [Kautz and Selman, 1996]. The main disadvantage of OBDDs is their fast growth when the number of state variables increases. Their main advantages are generality and flexibility. For instance, it is easy to detect that an OBDD represents all reachable states, which is more difficult for SAT as discussed in Section 4.

Cimatti et al. [2003] have expressed a narrow diagnosabil-

ity test, with only delays $d = 1$, as a model-checking problem in a temporal logic, and used a general-purpose model-checker NuSMV for testing diagnosability.

7 Conclusions

We have presented a SAT based approach to diagnosability testing, and demonstrated its scalability to systems with billions of states. Earlier works, most notably by Jiang et al. [2001], have relied on explicit enumeration of the states and are feasible only for systems with a much smaller number of states. The success of SAT for the diagnosability problems parallels its successes in areas like model-checking and AI planning, and has been made possible by the fast progress in the development of efficient algorithms for the satisfiability problem of the propositional logic.

A weakness of the SAT approach in comparison to for example OBDD based techniques is that the diagnosability test is better suited to detecting non-diagnosability than diagnosability. However, the same weakness is present in other related SAT based techniques, including AI planning and bounded model-checking, in which the presence of (short) paths in transition graphs can be often easily detected, but the absence of paths having a given property, without length restrictions, is often much more difficult to detect. We intend to further investigate approximate reachability techniques for proving diagnosability.

Acknowledgements

We thank the members of the Knowledge Representation and Reasoning group as well as Sophie Pinchinat for interesting discussions and comments.

This research was supported by National ICT Australia (NICTA) in the framework of the SuperCom project. NICTA is funded through the Australian Government's *Backing Australia's Ability* initiative, in part through the Australian National Research Council.

References

[Biere *et al.*, 1999] Armin Biere, Alessandro Cimatti, Edmund M. Clarke, and Yunshan Zhu. Symbolic model checking without BDDs. In W. R. Cleaveland, editor, *Tools and Algorithms for the Construction and Analysis of Systems, Proceedings of 5th International Conference, TACAS'99*, volume 1579 of *Lecture Notes in Computer Science*, pages 193–207. Springer-Verlag, 1999.

[Bryant, 1992] R. E. Bryant. Symbolic Boolean manipulation with ordered binary decision diagrams. *ACM Computing Surveys*, 24(3):293–318, September 1992.

[Burch *et al.*, 1994] J. R. Burch, E. M. Clarke, D. E. Long, K. L. MacMillan, and D. L. Dill. Symbolic model checking for sequential circuit verification. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 13(4):401–424, 1994.

[Cimatti *et al.*, 2003] Alessandro Cimatti, Charles Pecheur, and Roberto Cavada. Formal verification of diagnosability with via symbolic model-checking. In Georg Gottlob, editor, *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, pages 363–369. Morgan Kaufmann Publishers, 2003.

[Darwiche, 2001] Adnan Darwiche. Decomposable negation normal form. *Journal of the ACM*, 48(4):1–42, 2001.

[Jiang *et al.*, 2001] Shengbing Jiang, Zhongdong Huang, Vigyan Chandra, and Ratnesh Kumar. A polynomial algorithm for diagnosability of discrete-event systems. *IEEE Transactions on Automatic Control*, 46:1318–1321, 2001.

[Kautz and Selman, 1996] Henry Kautz and Bart Selman. Pushing the envelope: planning, propositional logic, and stochastic search. In *Proceedings of the 13th National Conference on Artificial Intelligence and the 8th Innovative Applications of Artificial Intelligence Conference*, pages 1194–1201. AAAI Press, August 1996.

[McMillan, 2003] Kenneth L. McMillan. Interpolation and SAT-based model checking. In Warren A. Hunt Jr. and Fabio Somenzi, editors, *Proceedings of the 15th International Conference on Computer Aided Verification (CAV 2003)*, number 2725 in Lecture Notes in Computer Science, pages 1–13, 2003.

[Rintanen *et al.*, 2006] Jussi Rintanen, Keijo Heljanko, and Ilkka Niemelä. Planning as satisfiability: parallel plans and algorithms for plan search. *Artificial Intelligence*, 170(12-13):1031–1080, 2006.

[Rintanen, 1998] Jussi Rintanen. A planning algorithm not based on directional search. In A. G. Cohn, L. K. Schubert, and S. C. Shapiro, editors, *Principles of Knowledge Representation and Reasoning: Proceedings of the Sixth International Conference (KR '98)*, pages 617–624. Morgan Kaufmann Publishers, June 1998.

[Rintanen, 2007] Jussi Rintanen. Diagnosers and diagnosability of succinct transition systems. In Manuela Veloso, editor, *Proceedings of the 20th International Joint Conference on Artificial Intelligence*. AAAI Press, 2007.

[Ryan, 2003] Lawrence Ryan. Efficient algorithms for clause-learning SAT solvers. Masters thesis, Simon Fraser University, September 2003.

[Sampath *et al.*, 1995] Meera Sampath, Raja Sengupta, Stéphane Lafortune, Kasim Sinnamohideen, and Demosthenis Teneketzis. Diagnosability of discrete-event systems. *IEEE Transactions on Automatic Control*, 40(9):1555–1575, 1995.

[Selman and Kautz, 1996] Bart Selman and Henry Kautz. Knowledge compilation and theory approximation. *Journal of the ACM*, 43:193–224, 1996.

[Sheeran *et al.*, 2000] Mary Sheeran, Satnam Singh, and Gunnar Stålmarck. Checking safety properties using induction and a SAT-solver. In W. A. Hunt and S. D. Johnson, editors, *Formal Methods in Computer-Aided Design, Third International Conference, FMCAD 2000, Austin, Texas, USA, November 1-3, 2000, Proceedings*, volume 1954 of *Lecture Notes in Computer Science*, pages 108–125. Springer-Verlag, 2000.