Diagnosers and Diagnosability of Succinct Transition Systems

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

Abstract

Reasoning about the knowledge of an agent is an important problem in many areas of AI. For example in diagnosis a basic question about a system is whether it is possible to diagnose it, that is, whether it is always possible to know whether a faulty behavior has occurred. In this paper we investigate the complexity of this diagnosability problem and the size of automata that perform diagnosis.

There are algorithms for testing diagnosability in polynomial time in the number of states in the system. For succinct system representations, which may be exponentially smaller than the state space of the system, the diagnosability problem is consequently in EXPTIME. We show that this upper bound is not tight and that the decision problem is in fact PSPACE-complete.

On-line diagnosis can be carried out by diagnosers which are automata that recognize faulty behavior. We show that diagnosers in the worst case have a size that is exponential in the number of states, both for explicit and succinct system representations. This is a consequence of the diagnoser having to maintain beliefs about the state of the system.

1 Introduction

Faults in dynamic systems can be diagnosed by an external diagnoser that observes the system behavior and infers the occurrence of failure events [Sampath *et al.*, 1995]. A main problem in on-line diagnosis is the construction of diagnosers which are deterministic finite automata that keep track of the possible state of the system on the basis of the observations and detect the occurrence of faults. Existing algorithms for constructing diagnosers have exponential running times and construct diagnosers of exponential size. In this work we investigate the inherent complexity of diagnosers and the complexity of constructing them, giving tight upper and lower bounds for the sizes of smallest diagnosers and the time it takes to construct diagnosers. Our results show that the asymptotic worst-case time and memory consumption of existing algorithms cannot be improved.

The work is based on the framework proposed by Sampath et al. [1995]. Since many systems exhibit regularities best captured by representing them in terms of state variables, we also consider a more compact representation with state variables. Our results show that the exponentially more compact representation leads in the worst case to a corresponding increase of complexity.

The structure of the paper is as follows. In Section 2 we define transition systems and succinct transition systems. Section 3 defines the framework of diagnosers and diagnosability of Sampath et al. [1995]. In Section 4 we present the main results which establish the complexity of diagnosability testing. In Section 5 we derive tight upper and lower bounds for the worst-case size of smallest diagnosers. Section 6 discusses related work and concludes the paper.

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 a transition relation,*
- $s_0 \in X$ *is an initial state.*

Initially, s_0 is the state of the system. A sequence of events $e_0, \ldots, e_{n-1}$ takes place and the system goes through some states $s_0, s_1, \ldots, s_n$ such that $(s_i, e_i, s_{i+1}) \in \delta$ for all $i \in \{0, \ldots, n-1\}$.

For $T = \langle X, \Sigma_o, \Sigma_u, \Sigma_f, \delta, s_0 \rangle$, $e_0, \ldots, e_{n-1}$ 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. Detection of failures is based on the sequence of observable events only.

A basic assumption made by Sampath et al. [1995] is that there are no infinite event sequences exclusively consisting of unobservable events.

Assumption 2.2 *There is no cycle in the transition graph consisting of unobservable events only.*

2.1 Succinct System Representation

The state spaces of many systems are highly regular, and the transition relations of events can be represented more compactly in terms of changes to the values of state variables. This also makes it possible to practically represent large systems. Given a set A of state variables, a state is defined as a valuation of A. We will restrict to two-valued (Boolean) state variables but in general there may be several different values. The transition relation associated with an event can be expressed as a propositional formula on A and A'. The set A' consists of propositional variables a' for every $a \in A$. When an event takes place, the variables in A represent the old values of the state variables and the variables in A' represent the corresponding new values. For example, given state variables $A = \{a, b\}$, an event that makes a *true* and does not change the value of b can be expressed as the formula $a' \wedge (b \leftrightarrow b')$. An arbitrary binary relation on the valuations of A can be expressed as a propositional formula, which suggests a succinct representation of transition systems in terms of state variables and formulae.

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,
- *$\delta : \Sigma_o \cup \Sigma_u \cup \Sigma_f \to \mathcal{L}$ assigns each event a propositional formula over $A \cup A'$ which represents a binary relation on the set of states, and*
- *s_0 is an initial state (a valuation of A).*

There are other types of succinct representations, for example based on Petri nets [Holloway *et al.*, 1997]. However, the Petri net representation can be translated into the logic representation quite easily (assuming finite state Petri nets with an n-safety property), but not vice versa, so the logic representation is more general.

Any 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 $R(T) = \langle X, \Sigma_o, \Sigma_u, \Sigma_f, \delta', s_0\rangle$ where*

1. *X is the set of all valuations of A, and*
2. *$\delta' = \{(s, e, s') \in X \times (\Sigma_o \cup \Sigma_u \cup \Sigma_f) \times X | s \cup \{(v', w) \in A' \times \{0, 1\} | (v, w) \in s'\} \models \delta(e)\}$.*

3 Diagnosability

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

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

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

Diagnosability is defined as the possibility to detect every occurrence of a failure event: the continuation of every event sequence that includes a failure event will at some future time point be observationally distinguishable from every event sequence that does not include a failure event.

Definition 3.1 (Diagnosability) *A transition system $T = \langle X, \Sigma_o, \Sigma_u, \Sigma_f, \delta, s_0\rangle$ is* diagnosable *iff there is an integer $d \geq 0$ such that for any sequence σ of events in T that ends in a failure event, 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.*

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.

Whether a system is diagnosable or not is important when trying to construct a diagnoser for it. A *diagnoser* is a deterministic finite automaton that recognizes sequences of observable events that correspond to sequences of events that include at least one failure event. If a system is diagnosable then a diagnoser exists.

Definition 3.2 *A* diagnoser *is $\langle Q, \Sigma_o, \gamma, z_0, Y\rangle$ where*

- *Q is a set of states (unrelated to the transition system),*
- *Σ_o is a set of observable events,*
- *$\gamma : Q \times \Sigma_o \to Q$ is a partial function,*
- *$z_0 \in Q$ is the initial state of the diagnoser, and*
- *$Y \subseteq Q$ is the set of* accepting states.

A sequence $e_0, \ldots, e_{n-1}$ of observable events takes a diagnoser through a sequence $z_0, \ldots, z_n$ of states such that $\gamma(z_i, e_i) = z_{i+1}$. An execution $z_0, \ldots, z_n$ is *accepting* if $z_n \in Y$. To accept means to detect a failure event.

Definition 3.3 *$\langle Q, \Sigma_o, \gamma, z_0, Y\rangle$ is* diagnoser for *$T = \langle X, \Sigma_o, \Sigma_u, \Sigma_f, \delta, s_0\rangle$ (with delay d) if*

1. *for every sequence $e_0, \ldots, e_n, e_{n+1}, \ldots, e_m$ of events in T such that $\{e_0, \ldots, e_{n-1}\} \subseteq \Sigma_o \cup \Sigma_u$ and $e_n \in \Sigma_f$ and $|\pi(e_{n+1}, \ldots, e_m)| \leq d$, there is a prefix of $\pi(e_0, \ldots, e_n, e_{n+1}, \ldots, e_m)$ which the diagnoser accepts, and if*
2. *for a sequence $e_0, \ldots, e_m$ of events in T the diagnoser accepts $\pi(e_0, \ldots, e_m)$ then for some $n \in \{0, \ldots, m-1\}$, $e_n \in \Sigma_f$, $\{e_0, \ldots, e_{n-1}\} \subseteq \Sigma_o \cup \Sigma_u$ and $|\pi(e_{n+1}, \ldots, e_m)| \leq d$.*

We can define diagnosers for succinct transition systems analogously through the reduction to transition systems.

In some cases there is no finite upper bound on how long it may take before a failure is detected. For a system to be diagnosable the delay must be bounded.

Example 3.4 (Unbounded delay) Consider the transition system in Figure 1. In the starting state two events are possible, of which the leftmost one is a failure event. After this an

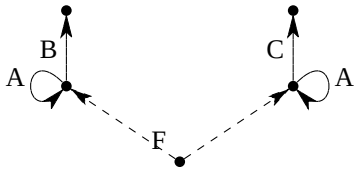

Figure 1: Delay may be arbitrarily long

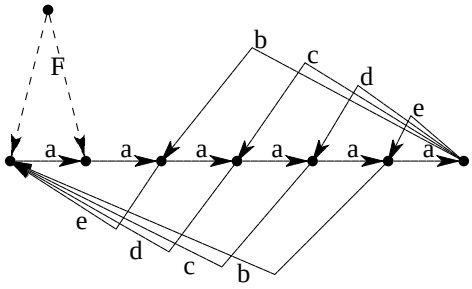

Figure 2: Delay may be quadratic in the number of states

unbounded number of unobservable events A may take place before the failure is diagnosed upon the occurrence of the observable event B. ■

A bounded delay may be quadratic in the number of states.

Example 3.5 (Quadratic delay) There are transition systems with a quadratic delay as depicted in Figure 2 [Grastien, 2006]. Of two sequences σ and σ' such that $\pi(\sigma) = \pi(\sigma') = aaaaabaaaacaaadaaea$ the first begins with a failure event but the second does not. The sequence σ' can continue with a but σ cannot. This can be generalized to any number n of states. The delay is $(n-2)+(n-3)+\cdots+3+2 = \frac{(n-1)(n-2)}{2} - 1$ which is $\mathcal{O}(n^2)$ in the number of states. ■

This is also an upper bound on bounded delays.

Theorem 3.6 *For a transition system with n states, if the system is diagnosable, then the delay is $\leq \frac{|X|^2-|X|}{2}$ where $|X|$ is the number of states.*

Proof: Assume $s_0 e_0 s_1 e_1 \cdots\cdots s_{m-1} e_{m_1} s_m e_m$ and $\hat{s}_0 e_0 \hat{s}_1 e_1 \cdots\cdots \hat{s}_{m-1} e_{m_1} \hat{s}_m \hat{e}_m$ are two equally long sequences of observable events interleaved with states that are distinguished at the last step (and not before) and of which the first contains an unobservable failure event and the second does not. All unobservable events and states other than s_0 and those immediately preceding an observable event are not included in the above sequences.

For the sake of argument assume that there are pairs of states $s_i, \hat{s}_i$ and $s_j, \hat{s}_j$ such that $j > i$, $s_i = s_j$ and $\hat{s}_i = \hat{s}_j$. Now the sequence $e_i, \ldots, e_{j-1}$ of events between these pairs of states could be repeated arbitrarily many times to have a pair of much longer sequences of events that are distinguishable only after the last event. This would violate our assumption that the delay is bounded.

Hence there are no i and j such that $s_i = s_j$ and $\hat{s}_i = \hat{s}_j$, which entails that every pair $s_i, \hat{s}_i$ occurs only once in the sequence. This entails that the length m of the sequences, and therefore the delay, is at most $|X|^2 - |X|$.

This bound can be tightened to $\frac{|X|^2-|X|}{2}$ by noticing that there cannot be i and j such that $s_i = \hat{s}_j$ and $\hat{s}_i = s_j$, or in other words, the order of the states in the pairs does not matter. □

4 Complexity of Testing Diagnosability

Jiang et al. [2001] have shown that diagnosability testing is in P. In this section we will show that this complexity upper bound is not tight. Our results also show that the EXPTIME upper bound for the diagnosability problem of succinct transition systems implied by the result of Jiang et al. is not tight.

A system is not diagnosable if there are two infinite event sequences which have exactly the same observable events and of which one contains a failure event and the other not. If this condition is not fulfilled, then every continuation of an event sequence with a failure is distinguishable from all event sequences without a failure.

Jiang et al. [2001] showed how this test can be made finitary for finite state systems, and how it can be done in polynomial time. The basic construction is a product transition system, sometimes called *the twin plant*, in which states are pairs $(s, \hat{s})$ of states of the original system, and events represent unobservable events in one or both of the components of these state pairs, or observable events shared by both components. Failure events take place only in the first component of each state pair. 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, and a non-empty event sequence back to $(s, \hat{s})$ then a pair of infinite event sequences witnessing non-diagnosability exists.

Candidate states $(s, \hat{s})$ are found by Tarjan's strongly connected components (SCC) algorithm [Tarjan, 1972]: every node contained in a nontrivial SCC is in a cycle contained in the SCC. After finding the SCCs the diagnosability test reduces to finding a path through a failure event from (s_0, s_0) to some $(s, \hat{s})$ in a non-trivial SCC. If such a path exists the system is not diagnosable.

Constructing the twin plant can be done in quadratic time in the number of states. Finding the SCCs is linear time in the number of states in the twin plant, and finding a path to a nontrivial SCC through a failure event can be done in polynomial time. Hence the diagnosability problem is in the complexity class P. However, this complexity upper bound is not tight.

We define the language NONDIAG which consists of all non-diagnosable transition systems in some suitable representation.

Our first result shows that determining non-diagnosability can be done by using only logarithmic space by a nondeterministic Turing machine, and hence the decision problem associated with NONDIAG is in NLOGSPACE. This complexity class is included in P but it is not known whether the inclusion is proper [Johnson, 1990].

Theorem 4.1 *NONDIAG is in NLOGSPACE.*

Proof: The proof is similar to the NLOGSPACE membership proof of testing the existence of a path between two nodes in a graph. We give a nondeterministic algorithm that takes logarithmic space. The alorithm's only data structure that does not have a constant size is the binary counter for counting up to n^2 where n is the number of states. A cycle is detected as in the P-time proof of Jiang et al. [2001]. The algorithm starts from the pair (s_0, s_0) of initial states and nondeterministically chooses an event and a successor state for each member of the pair. The event for the second state may not be a failure, and an observable event must always be shared by both components of a state pair. The counter counts the steps. After a number of steps, including at least one failure event, the algorithm reaches a starting state $(s, \hat{s})$ for a cycle. After this a sequence of events leading back to $(s, \hat{s})$ is nondeterministically chosen. The Turing machine accepts if a cycle is completed before the counter reaches n^2. The Turing machine rejects if the counter reaches n^2 before completing the cycle and encountering a failure event. □

We also have a simple proof of NLOGSPACE-hardness.

Theorem 4.2 *NONDIAG is NLOGSPACE-hard.*

Proof: We sketch a deterministic logarithmic space reduction from the NLOGSPACE-complete problem of testing whether a node in a directed graph is reachable from another node [Johnson, 1990].

Let $G = \langle V, E\rangle$ be a directed graph for which the existence of a path from $s \in V$ to $t \in V$ is tested. The reduction with only logarithmic space proceeds by reading the input graph G and outputting a corresponding transition system where states are the nodes of G and edges of G correspond to an observable event. The initial state of the transition system is s. The transition system additionally has a failure event F and an unobservable event NF which may take place in the state t. Both events lead to the same (arbitrary) state in V. Now the transition system is not diagnosable if and only if t is reachable from s. Construction of the transition system takes logarithmic space (actually only constant space.) □

Similarly to NONDIAG we define the language SUCCINCT-NONDIAG which consists of succinct transition systems that are not diagnosable.

Theorem 4.3 *SUCCINCT-NONDIAG is in PSPACE.*

Proof: Sketch: We could prove NPSPACE membership similarly to Theorem 4.1 and then use the equality PSPACE=NPSPACE. Alternatively, we could give a deterministic algorithm for finding a path in a graph with polynomial recursion depth and space consumption following Bylander [1994]. □

The PSPACE upper bound can be shown to be tight.

Theorem 4.4 *SUCCINCT-NONDIAG is PSPACE-hard.*

Proof: We reduce the halting problem of deterministic Turing machines with a polynomial space bound to the succinct diagnosability problem, showing the latter PSPACE-hard.

Part of the reduction is like in the reductions to some other succinct graph reachability problems [Bylander, 1994]. We can represent the Turing machine configurations, including the contents of a polynomially long tape, in terms of a polynomial number of state variables, and we can represent the transitions of the Turing machine in terms of a formula $\delta(e)$ which represents an unobservable event.

We construct a transition system that is diagnosable if and only if the Turing machine accepts. First a failure event F or an unobservable non-failure event NF takes place. In the non-failure case this is followed by an infinite sequence of events 1. In the failure case a simulation of the Turing machine follows. If the Turing machine accepts then an event 2 takes place. This makes it possible to distinguish between the failure and non-failure cases. So the system is diagnosable if and only if the Turing machine accepts.

Given a Turing machine, we define the succinct transition system $T = \langle A, \Sigma_o, \Sigma_u, \Sigma_f, \delta, s_0\rangle$ where

- A consists of all the state variables needed for encoding executions of the Turing machine with a bound on the number of used tape cells (see for example Bylander [1994]) and the state variable f that indicates that a failure has taken place and I that indicates that the execution has not started yet.
- $\Sigma_o = \{1, 2\}$, $\Sigma_u = \{\text{NF}\}$, $\Sigma_f = \{\text{F}\}$, and
- $$\delta(\text{NF}) = I \wedge \neg f' \wedge \neg I' \wedge \bigwedge_{a \in A \setminus \{f, I\}} (a \leftrightarrow a')$$

  $$\delta(\text{F}) = I \wedge f' \wedge \neg I' \wedge \bigwedge_{a \in A \setminus \{f, I\}} (a \leftrightarrow a')$$

  $$\delta(1) = \neg I \wedge \neg I' \wedge ((f \wedge f' \wedge \neg a \wedge \text{TM}) \vee (\neg f \wedge \neg f'))$$

  where TM is a formula for simulating the change in the Turing machine configuration in one execution step and a is a state variable that represents the accepting internal state of the Turing machine. If the Turing machine rejects it continues with event 1 indefinitely.

  When the Turing machine accepts at least one event 2 takes place.

  $$\delta(2) = f \wedge a$$

- s_0 is a valuation that encodes the initial configuration of the Turing machine and sets $s_0(I) = 1$.

□

Hence the succinct diagnosability problem is PSPACE-complete, like some other path-finding problems in succinctly represented graphs, the succinct s-t-reachability problem [Lozano and Balcázar, 1990] and the plan existence problem of AI planning [Bylander, 1994]. This further suggests similar approaches to solving the diagnosability problem, for example reduction to the propositional satisfiability problem SAT [Rintanen and Grastien, 2007].

5 Size of Diagnosers

In this section we show that, assuming a sufficiently restricted definition of diagnosers, sometimes diagnosers are necessarily quite large.

The size of diagnosers strongly depends on how complex computation they are allowed to perform. The more complex the computation may be the smaller the diagnoser. There are diagnosers that are Turing machines with only a constant size and that only need memory for storing the system description and the set of all states, but executing such a diagnoser may be too expensive for real-time on-line diagnosis.

In this section we focus on diagnosers that are finite automata which can process each observation in constant time.

5.1 Upper Bounds on Size

We use the notion of belief states to derive an upper bound on the size of smallest diagnosers. The basic construct is similar to that used by Sampath et al. [1995] and the size of smallest diagnosers follows from the construct by Sampath et al.

Let $T = \langle X, \Sigma_o, \Sigma_u, \Sigma_f, \delta, s_0\rangle$ be a transition system. Let $\mathrm{G}_0(T) = \langle V, E\rangle$ be a graph in which the set of nodes $V = 2^{X\times 2^{\Sigma_f}}$ is the set of all sets $B \subseteq X \times 2^{\Sigma_f}$ of pairs $(s, f) \in X \times 2^{\Sigma_f}$, and there is a labelled edge $(B, e, B') \in E$ from $B \in V$ to $B' \in V$ if and only if either

- $e \in \Sigma_f$ and $B' = \{(s', f \cup \{e\}) | (s, f) \in B, (s, e, s') \in \delta\}$, or
- $e \in \Sigma_o \cup \Sigma_u$ and $B' = \{(s', f) | (s, f) \in B, (s, e, s') \in \delta\}$,

A belief state B with $(s, f) \in B$ means that it is possible that s is the current state and its history includes exactly the failure events in f. Note that it is possible that $(s, f) \in B$ and $(s, f') \in B$ for some $f \neq f'$.

The *initial node* of the graph is $B_I = \{(s_0, \emptyset)\}$.

The number of belief states $B \subseteq X \times 2^{\Sigma_f}$ is $2^{|X|2^{|\Sigma_f|}}$. This is $\mathcal{O}(2^n)$ where n is the number of states.

This graph compactly encodes the states that could be reached by a given event sequence and the failure events in that sequence: starting in the initial node follow the edges corresponding to the events, and if $(s, f) \in B$ for the node B that is reached, then it is possible to reach s with that event sequence which includes the failure events in f.

A diagnoser cannot observe unobservable events, so we construct a second graph $\mathrm{G}(T)$ that represents the diagnoser's view of the possible current states and their failure histories in terms of observable events. This construction takes polynomial time in the size of $\mathrm{G}_0(T)$. Any path $e_1, \ldots, e_n, e$ in $\mathrm{G}_0(T)$ such that e is observable and $e_1, \ldots, e_n$ are unobservable is represented in $\mathrm{G}(T)$ by one edge with the label e.

Given $\mathrm{G}_0(T) = \langle V, E\rangle$ define $\mathrm{G}(T) = \langle V, E'\rangle$ as follows. There is an edge $(B, e, B') \in E'$ if and only if

- $e \in \Sigma_o$ and
- $B' = \bigcup_{i=1}^{m} B_i$ where $B_1, \ldots, B_m$ are all the nodes in $\mathrm{G}_0(T)$ such that for all $i \in \{1, \ldots, m\}$ there is a path $e_1, \ldots, e_n, e$ from B to B_i such that $n \geq 0$ and $\{e_1, \ldots, e_n\} \subseteq \Sigma_u \cup \Sigma_f$.

By Assumption 2.2 n is bounded by some constant $\leq |X|$.

The graph $\mathrm{G}(T)$ has the same nodes as $\mathrm{G}_0(T)$ and the number of edges is at most as high.

Lemma 5.1 *Let σ be a sequence of events in a system T, starting in its initial state s_0, in which exactly the failure events in f occur before the last observable event and s is the last state that is not immediately preceded by an unobservable event (it is preceded by an observable event or it is the first state s_0.) Then the sequence $\pi(\sigma)$ in $G(T)$ leads from $\{(s_0, \emptyset)\}$ to B such that $(s, f) \in B$.*

Lemma 5.2 *Let T be a system with the initial state s_0. Let σ be a sequence of observable events. If σ leads in $G(T)$ from $\{(s_0, \emptyset)\}$ to B such that there is some $(s, f) \in B$, then there is a sequence σ' of events in T such that $\sigma = \pi(\sigma')$ and σ' leads from s_0 to s in T and σ' includes exactly the failure events f.*

Finally, we show that $\mathrm{G}(T)$ is a diagnoser. The accepting states B of this diagnoser are those that include a known past failure: $\bigcap_{(s,f)\in B} f \neq \emptyset$.

Theorem 5.3 *Let $T = \langle X, \Sigma_o, \Sigma_u, \Sigma_f, \delta, s_0\rangle$ be a system that is diagnosable with delay d.*

1. *Let $\sigma = e_1, \ldots, e_n$ be a sequence of events in T starting in s_0 and assume $e_n \in \Sigma_f$. Then for every sequence of events $e'_1, \ldots, e'_m$ such that $|\pi(e'_1, \ldots, e'_m)| = d$ and $e_1, \ldots, e_n, e'_1, \ldots, e'_m$ is a sequence of events in T, there is $m' \leq m$ such that $\pi(e_1, \ldots, e_n, e'_1, \ldots, e'_{m'})$ leads in $G(T)$ from $\{(s_0, \emptyset)\}$ to B such that $e_n \in \bigcap_{(s,f)\in B} f$.*
2. *If for a sequence of events $e_1, \ldots, e_m$ in T the path $\pi(e_1, \ldots, e_m)$ in $G(T)$ leads from $\{(s_0, \emptyset)\}$ to B and there is $e \in \bigcap_{(s,f)\in B} f$, then $e \in \{e_1, \ldots, e_m\}$.*

Proof: For (1) take any $\sigma' = e'_1, \ldots, e'_m$ such that $|\pi(\sigma')| = d$ and $\sigma\sigma'$ is an event sequence in T. Let $B_0, \ldots, B_{n+m}$ be the belief states on the path $\pi(\sigma\sigma')$ in $\mathrm{G}(T)$.

Assume $e_n \notin \bigcap_{(s,f)\in B_{n+m}} f$. By Lemma 5.2 there is a sequence σ_2 of events such that $\pi(\sigma_2) = \sigma\sigma'$ and e_n does not occur in $\sigma\sigma'$. This contradicts the fact that the system is diagnosable with delay d: for all sequences $\sigma_1 = \sigma\sigma'$ and σ_2 of events in T such that $|\pi(\sigma')| \geq d$ and $\pi(\sigma_1) = \pi(\sigma_2)$, σ_2 includes the event e. Hence it must be that $e_n \in \bigcap_{(s,f)\in B_{n+m}} f$. Now $e_n \in \bigcap_{(s,f)\in B_{n+m'}} f$ for some $m' \leq n + m$.

For (2) assume that $e_1, \ldots, e_m$ is a sequence of events in T such that the path $\pi(e_1, \ldots, e_m)$ in $\mathrm{G}(T)$ leads from $\{(s_0, \emptyset)\}$ to B such that there is $e \in \bigcap_{(s,f)\in B} f$.

Let $\sigma = \pi(e_1, \ldots, e_m)$. Let m' be the least m' such that $e_{m'}$ is an observable event, and let s be the state immediately following $e_{m'}$. Let $\sigma = \pi(e_1, \ldots, e_{m'}) = \pi(e_1, \ldots, e_m)$. By Lemma 5.1 $(s, f) \in B$ where $f = \Sigma_f \cap \{e_1, \ldots, e_m\}$. Since $e \in \bigcap_{(s,f)\in B} f$, $e \in \{e_1, \ldots, e_{m'}\} \subseteq \{e_1, \ldots, e_m\}$. □

Let $\mathrm{G}(T) = \langle V, E'\rangle$. We define the diagnoser $\mathrm{CD}(T) = \langle V, \Sigma_o, \delta, \{(s_0, \emptyset)\}, A\rangle$ where

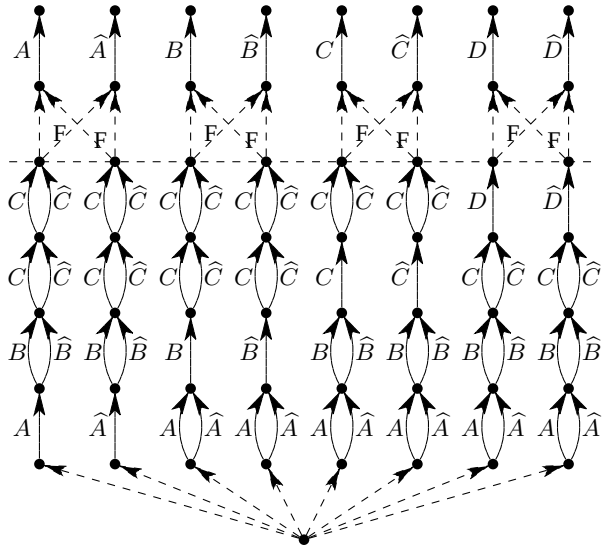

Figure 3: A system requiring a diagnoser of size $\Omega(2^n)$

- $\delta = \{((B, e), B') | (B, e, B') \in E'\}$ and
- $A = \{B \in V | \bigcap_{(s,f)\in B} f \neq \emptyset\}$.

Hence the graph $\mathrm{G}(T) = \langle V, E' \rangle$ corresponds to a diagnoser for T. The diagnoser starts from the node $\{(s_0, \emptyset)\}$. After observing an event $e \in \Sigma_o$ it follows the corresponding edge in the graph. Notice that unlike in the underlying system, each event in the graph $\langle V, E' \rangle$ leads to a unique successor node. When reaching a node $B = \{(s_1, f_1), \ldots, (s_n, f_n)\}$ with $f = \bigcap_{i=1}^{n} f_i \neq \emptyset$, the diagnoser has detected the failure events f.

The number of nodes in $\langle V, E' \rangle$ is $\mathcal{O}(2^n)$ in the number n of states in T. Some diagnosers may have more nodes than $\langle V, E' \rangle$ but those diagnosers are not the smallest ones. Hence the size of the smallest diagnoser is $\mathcal{O}(2^n)$.

Since a transition system may be exponentially bigger than its succinct representation, the $\mathcal{O}(2^n)$ bound on the diagnoser size for enumerative representations yields a $\mathcal{O}(2^{2^n})$ bound on the size of diagnosers of succinct transition systems.

5.2 Lower Bounds on Size

We derive an exponential lower bound on the worst-case size of smallest diagnosers by constructing a system for which the smallest diagnoser has an exponential number of states. The high number of states in the smallest diagnoser is an indication of the need for the diagnoser to remember the past events. The number of possible past histories may grow exponentially in the size of the system and the length of the histories. The past histories may have to be encoded in the diagnoser and this leads to an exponential size.

Theorem 5.4 *There are diagnosable systems for which the smallest diagnoser has size $\Theta(2^n)$ where n is the number of states.*

Proof: Consider the system in Figure 3. Let $\Sigma = \{A, B, C, D\}$. The events of the transition system are a and $\hat{a}$ for $a \in \Sigma$. In the initial state an unobservable event takes place. Then a sequence of $|\Sigma|$ events takes place, either a or $\hat{a}$ for every $a \in \Sigma$, followed by either an unobservable failure or an unobservable non-failure event. A failure event has taken place iff for some $a \in \Sigma$ both a and $\widehat{a}$ have taken place.

The number of different sequences of observable events before the failure events is $2^{|\Sigma|}$. No information on these sequences can be ignored before the last event has taken place. Hence the smallest diagnoser has $2^{|\Sigma|}$ states for representing the first $|\Sigma|$ events.

Clearly, similar systems can be constructed for arbitrary sets Σ, and the number of states of these systems grows quadratically in the cardinality of Σ and the number of states in the diagnosers grows exponentially. □

We adapt the construction in Theorem 5.4 to succinct transition systems by replacing the events A, B, C, D by sequences of events corresponding to binary numbers, and utilizing the regularity of the system to represent it more compactly so that the number of states is exponential in the size of the system description. This means that the number of nodes in the diagnoser is doubly exponential $\Omega(2^{2^n})$ in the size of the system description.

Theorem 5.5 *There are succinct systems for which the smallest diagnoser has size $\Theta(2^{2^n})$.*

Proof: Let n be a size parameter that characterizes the number 2^n of things to remember just like A, B, C, D in the proof of Theorem 5.4. Instead of producing a sequence of events A, $\widehat{A}$, B, $\widehat{B}$, C, $\widehat{C}$ D, $\widehat{D}$ indicating the different "values" of A, B, C and D, the succinct system produces a sequence $v_0, \ldots, v_{2^n-1}$ of 2^n events 0 or 1 followed by an unobservable failure or non-failure event, and then an event 0 or 1 for indicating a value v and a sequence of n events 0 or 1 that encodes an index $i \in \{0, \ldots, 2^n - 1\}$ to the first sequence. If the value v_i does not match v then a failure has taken place. The construction with $n = 2$ is illustrated in Figure 4.

Because of space restrictions we cannot present the system here. The number of states of the system is exponential in n, and the size of the succinct representation of the system is linear in n. The size of the smallest diagnoser is exponential in the number of states and doubly exponential in n. □

6 Related Work and Conclusions

We have shown that the diagnosability problem of succinct transition systems is PSPACE-complete. We also tightened an earlier known polynomial time upper bound of diagnosability testing of explicitly represented transition systems by showing that non-diagnosability is NLOGSPACE-complete.

The work by Sampath et al. [1995] underlies much of the work in the diagnoser framework, and also our belief state enumeration algorithm in Section 5.1 is based on it.

Other related works include Tripakis [2002] who shows that diagnosability of transition systems represented as timed automata is PSPACE-complete. This result can be contrasted

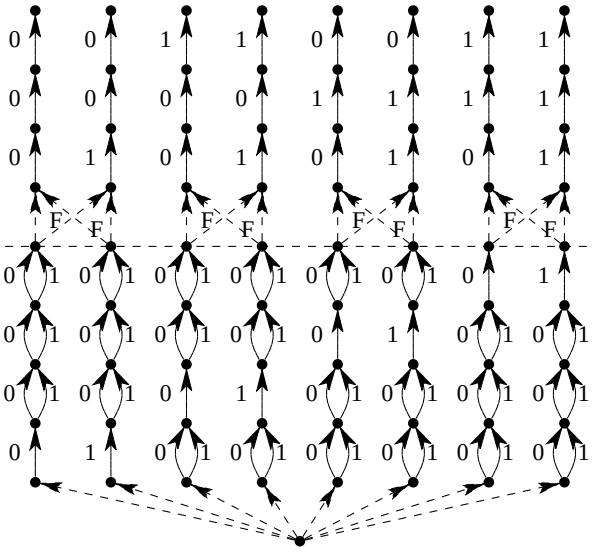

Figure 4: A system requiring a diagnoser of size $\Omega(2^{2^n})$

to the polynomial time result of Jiang et al. [2001] and our more tight NLOGSPACE-membership result in Theorem 4.1. More complex diagnosis problems for timed automata are investigated by Bouyer et al. [2005].

Wen et al. [2005] propose a polynomial time algorithm for testing diagnosability of transition systems represented as Petri nets. The algorithm tests for a sufficient but not necessary condition for diagnosability, and is therefore incomplete. A variant of our Theorem 4.4 shows that, assuming PSPACE≠P, the test cannot be made complete by strengthening it without losing the polynomial time property: the Turing machine simulation in the proof of Theorem 4.4 can be performed with Petri nets as well.

Acknowledgements

Many thanks to Sylvie Thiébaux for directing me to this research topic and to Alban Grastien for comments and tightening the upper bound in Theorem 3.6 by a factor of 2.

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

[Bouyer *et al.*, 2005] Patricia Bouyer, Fabrice Chevalier, and Deepak D'Souza. Fault diagnosis using timed automata. In Vladimiro Sassone, editor, *Foundations of Software Science and Computational Structures: 8th International Conference, FOSSACS 2005, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2005, Edinburgh, UK, April 4-8, 2005. Proceedings*, number 3441 in Lecture Notes in Computer Science, pages 219–233. Springer-Verlag, 2005.

[Bylander, 1994] Tom Bylander. The computational complexity of propositional STRIPS planning. *Artificial Intelligence*, 69(1-2):165–204, 1994.

[Grastien, 2006] Alban Grastien. personal communication, 2006.

[Holloway *et al.*, 1997] L. E. Holloway, B. H. Krogh, and A. Giua. A survey of Petri net methods for controlled discrete-event systems. *Discrete Event Dynamic Systems: Theory and Applications*, 7:151–190, 1997.

[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.

[Johnson, 1990] D. S. Johnson. A catalog of complexity classes. In J. Van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume A. Algorithms and Complexity, pages 67–161. Elsevier Science Publishers, 1990.

[Lozano and Balcázar, 1990] Antonio Lozano and José L. Balcázar. The complexity of graph problems for succinctly represented graphs. In Manfred Nagl, editor, *Graph-Theoretic Concepts in Computer Science, 15th International Workshop, WG'89*, number 411 in Lecture Notes in Computer Science, pages 277–286. Springer-Verlag, 1990.

[Rintanen and Grastien, 2007] Jussi Rintanen and Alban Grastien. Diagnosability testing with satisfiability algorithms. In Manuela Veloso, editor, *Proceedings of the 20th International Joint Conference on Artificial Intelligence*. AAAI Press, 2007.

[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.

[Tarjan, 1972] R. E. Tarjan. Depth first search and linear graph algorithms. *SIAM Journal on Computing*, 1(2):146–160, 1972.

[Tripakis, 2002] Stavros Tripakis. Fault diagnosis for timed automata. In W. Damm and E.-R. Olderog, editors, *Formal Techniques in Real-Time and Fault-Tolerant Systems: 7th International Symposium, FTRTFT 2002, Co-sponsored by IFIP WG 2.2, Oldenburg, Germany, September 9-12, 2002. Proceedings*, number 2469 in Lecture Notes in Computer Science, pages 205–221. Springer-Verlag, 2002.

[Wen *et al.*, 2005] YuanLin Wen, ChunHsi Li, and MuDer Jeng. A polynomial algorithm for checking diagnosability of Petri nets. In *IEEE International Conference on Systems, Man and Cybernetics*, volume 3, pages 2542–2547, 2005.