Canonical Finite State Machines For Distributed Systems

Robert M. Hierons [a]

[a] *School of Information Systems, and Computing Mathematics, Brunel University, Uxbridge, Middlesex, UB8 3PH*

Abstract

There has been much interest in testing from finite state machines (FSMs) as a result of their suitability for modelling or specifying state-based systems. Where there are multiple ports/interfaces a multi-port FSM is used and in testing a tester is placed at each port. If the testers cannot communicate with one another directly and there is no global clock then we are testing in the distributed test architecture. It is known that the use of the distributed test architecture can affect the power of testing and recent work has characterised this in terms of local s-equivalence: in the distributed test architecture we can distinguish two FSMs, such as an implementation and a specification, if and only if they are not locally s-equivalent. However, there may be many FSMs that are locally s-equivalent to a given FSM and the nature of these FSMs has not been explored. This paper examines the set of FSMs that are locally s-equivalent to a given FSM M. It shows that there is a unique smallest FSM $\chi_{min}(M)$ and a unique largest FSM $\chi_{max}(M)$ that are locally s-equivalent to M. Here smallest and largest refer to the set of traces defined by an FSM and thus to its semantics. We also show that for a given FSM M the set of FSMs that are locally s-equivalent to M defines a bounded lattice. Finally, we define an FSM that, amongst all FSMs locally s-equivalent to M, has fewest states. We thus give three alternative canonical FSMs that are locally s-equivalent to an FSM M: one that defines the smallest set of traces, one that defines the largest set of traces, and one with fewest states. All three provide valuable information and the first two can be produced in time that is polynomial in terms of the number of states of M. We prove that the problem of finding an s-equivalent FSM with fewest states is NP-hard in general but can be solved in polynomial time for the special case where there are two ports.

Key words: finite state machine, equivalence, distributed test architecture, canonical.

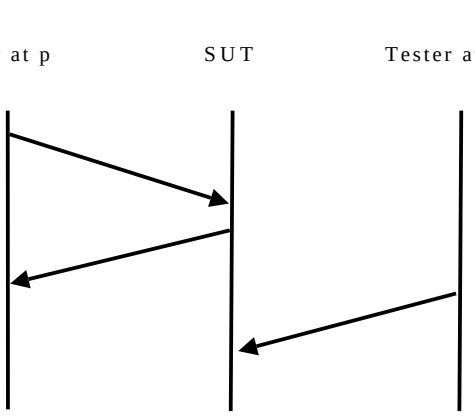

Fig. 1. A controllability problem

1 Introduction

Finite state machines (FSMs), and their extensions, are widely used to specify or model state-based systems. In addition, FSM based test techniques have been applied to systems specified in languages such as SDL [17,31] and Statecharts [3,15] and are used in Model Based Testing (see, for example, [1,2,11,12]). There has thus been much interest in testing from FSMs (see [16,25] for surveys).

A system with physically distributed ports or interfaces is said to be a multi-port system. When testing such a system it is usual to place a tester at each port and each tester sees only the interactions that occur at its port. If these testers cannot directly communicate with one another and there is no global clock then we are testing in the distributed test architecture and this can introduce controllability and observability problems (see, for example, [4–9,13,18,26,27,29,30]). Controllability problems occur when a tester at a port p is expected to apply an input but because it was not involved in the previous operation it does not know when to apply this input. For example, if a test involves input x_1 at port p, this should lead to output at p only and input x_2 should then be applied at a port $q \neq p$ then there is a controllability problem since the tester at port q does not know when to apply x_2 as it did not participate in the previous operation. This is illustrated in Figure 1 in which each vertical line represents a timeline, time progressing as we move down a line.

Observability problems occur if a tester at a port q is expecting an output in response to an input, possibly sent by another tester, but does not know when to start and stop waiting for this output. Let us suppose, for example, that input x_1 at port p should lead to output y_p at p and y_q at $q \neq p$, this is to be followed by input x_2 at p and this should lead to output y'_p at p only. Then the tester at port q expects to observe y_q only and the tester at p expects to observe $x_1 y_p x_2 y'_p$ and this is still the case if the response to x_1 is y_p and

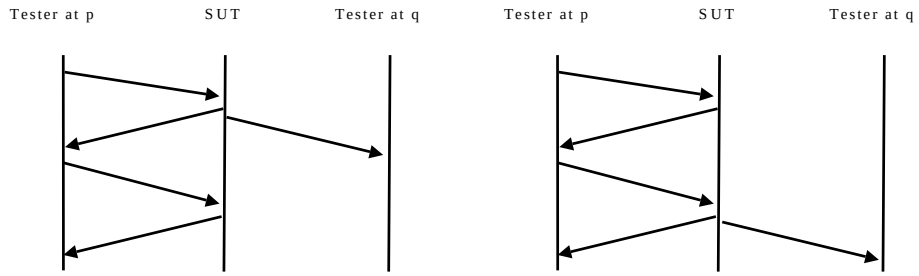

Fig. 2. An observability problem

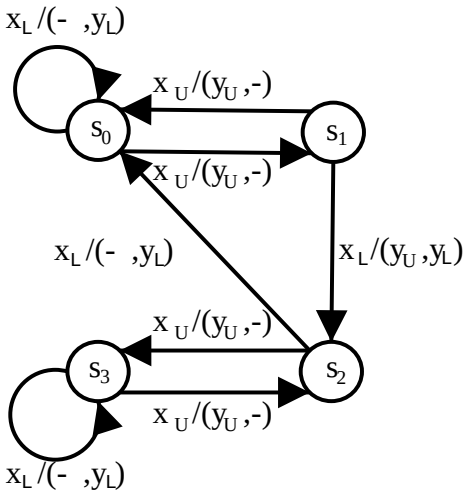

Fig. 3. The FSM M_0

the response to x_2 is y'_p at p and y_q at q. These two scenarios are illustrated in Figure 2. Here, two faults can mask one another in this test sequence but these faults may be observed in use if different sequences are used.

Consider, for example, the FSM M_0 shown in Figure 3, originally given in [19], in which x_U and x_L are inputs at U and L respectively and y_U and y_L are outputs at U and L respectively. Here, for example, there is a controllability problem if we apply input $x_L x_U$ in state s_0 since the first input is at L and leads to output at L only but the second input is at U. Similarly, there can be an observability problem if we apply $x_L x_L$ in state s_1 since the first input should lead to output (y_U, y_L) and the second should lead to output y_L at L only: this cannot be distinguished from the case where the first output is y_L at L and the second output is (y_U, y_L).

Sometimes it is possible to connect the testers using an external communications network and overcome controllability and observability problems through the exchange of coordination messages by the testers (see, for example, [5,29]). However, the introduction of such a network can increase the cost of testing and it may not be possible to overcome controllability and observability problems in this manner if there are timing constraints (see [24] for a discussion of timing issues). If the testers cannot exchange coordination messages and there

is no global clock then we are testing in the distributed test architecture [21].

The power of testing in the distributed test architecture has been characterised in terms of local s-equivalence and local s-distinguishability: it is possible for testing to distinguish a specification FSM M and an implementation FSM N in the distributed test architecture without introducing controllability problems if and only if M and N are locally s-distinguishable [19].

Previous work left open the question of whether, for a given FSM M, there is a sensible notion of a 'best' or 'canonical' FSM that is locally s-equivalent to M. This paper discusses three such possibilities. The first two are a smallest locally s-equivalent FSM and a largest locally s-equivalent FSM, where smallest and largest correspond to the set of traces defined by the FSM while the third is an FSM with fewest states. The smallest locally s-equivalent FSM defines the set of traces of the specification that must be implemented in order for the system under test (SUT) not to be distinguishable from the specification in the distributed test architecture when using input sequences that do not introduce controllability problems. If the use of the SUT corresponds to these conditions then the smallest locally s-equivalent FSM $\chi_{min}(M)$ defines exactly the traces that must be implemented. The largest locally s-equivalent FSM $\chi_{max}(M)$ defines the set of traces that the SUT can have while not being distinguishable from the specification when testing in the distributed test architecture without introducing controllability problems. By examining this largest locally s-equivalent FSM we can explore the potential consequences of testing in the distributed test architecture. There is a natural partial ordering on FSMs defined by their languages and it transpires that under this partial order the FSMs $\chi_{min}(M)$ and $\chi_{max}(M)$ give minimal and maximal elements of the bounded lattice defined by the set of FSMs that are locally s-equivalent to M. If in use the SUT will only ever receive input sequences that have no controllability problems and observations are made locally then it is sufficient to have an SUT that is locally s-equivalent to M. This paper thus also investigates the problem of finding a design with fewest states that is locally s-equivalent to M. The problems of finding $\chi_{min}(M)$ and $\chi_{max}(M)$ can be solved in time that is polynomial in the number of states of M. In addition, while we prove that the problem of finding an s-equivalent FSM with fewest states is NP-hard, this problem can be solved in polynomial time for the case often considered in the literature, in which there are only two ports

This paper is structured as follows. First background material is described and extended in Section 2. In Sections 3 and 4 we show how the FSMs $\chi_{min}(M)$ and $\chi_{max}(M)$ can be constructed. In Section 5 we prove that the set of FSMs that are locally s-equivalent to M defines a bounded lattice. In Section 6 we show how from an FSM M we can produce a locally s-equivalent FSM with fewest states if there are only two ports and prove that the general problem is NP-hard. Finally, in Section 7, conclusions are drawn.

2 Preliminaries

2.1 Basic notation

In this paper sequences are represented by listing their elements. For example, 01 denotes the sequence that contains two values, 0 followed by 1. Where a variable represents a sequence its name will have a bar above it, an example being $\bar{x}$, and ϵ denotes the empty sequence. Given a set X, $\mathcal{P}(X)$ denotes the powerset of X: the set of subsets of X. Given a set A of sequences, $Pre(A)$ denotes the set of prefixes of sequences from A.

2.2 Multi-port finite state machines

A multi-port FSM has $m > 1$ interfaces/ports at which it interacts with its environment. We label the ports with the integers 1 to m and so the ports are represented by $P = \{1, \ldots, m\}$. In this paper all examples have two ports called U and L and in an abuse of notation we use U and L in place of port names 1 and 2. Note, however, that the results are proved for the general case. The use of the names U and L for the ports is traditional since the original motivation for work in this area was protocol conformance testing, in which a protocol is tested through the use of an upper tester and a lower tester [21].

A multi-port FSM M with m ports is defined by a tuple (S, s_0, X, Y, T) in which:

- S is the finite *set of states* of M;
- $s_0 \in S$ is the *initial state* of M;
- $X = X_1 \cup \ldots \cup X_m$ is the finite *input alphabet* of M, where for $1 \leq i \leq m$, X_i is the input alphabet at port i and for all $1 \leq i < j \leq m$ we have that $X_i \cap X_j = \emptyset$;
- $Y = (Y_1 \cup \{-\}) \times \ldots \times (Y_m \cup \{-\})$ is the *output alphabet* of M, where for $1 \leq i \leq m$, Y_i is the output alphabet at port i, $-$ denotes no output, and for all $1 \leq i < j \leq m$ we have that $Y_i \cap Y_j = \emptyset$; and
- T is a set of *transitions* of the form $(s_i, s_j, x/y)$ for $s_i, s_j \in S$, $x \in X$, and $y \in Y$.

Multi-port FSMs are similar to transducers and were initially introduced for communications protocols. They have the property that a transition is triggered by a single input but may lead to multiple outputs. This may seem to preclude the specification of a system that has operations that receive inputs at different ports but such systems can be modelled by including transitions that produce no output. Some recent work [14] has looked at the testing of

distributed systems in which an operation can be triggered by multiple events at different ports and such models may well be more suitable for some systems. However, in this paper we focus on the type of model traditionally considered, the multi-port FSM, and we simply call these FSMs.

An FSM can be represented by a directed graph whose edges are labelled with the corresponding input/output pair. For example, the FSM M_0 in Figure 3 has the transition $(s_3, s_2, x_U/(y_U, -))$.

While we assume that the X_i are disjoint and so are the Y_i, this is not a restriction since if the same values can be received or sent at different ports then we can simply label these.

Throughout this paper $M = (S, s_0, X, Y, T)$ denotes an FSM with m ports and n states. A transition $t = (s_i, s_j, x/y) \in T$ should be interpreted in the following way: if M receives input x when in state s_i then it can output y and move to state s_j. The state s_i is said to be the *starting state* of t, the state s_j is the *ending state* of t, x/y is the *label* of t and x is the *input portion* of x/y.

An FSM M is *deterministic* if for every state $s \in S$ and input $x \in X$ there is at most one transition in T that has starting state s and whose label has input portion x. Further, M is *completely specified* if for every state $s \in S$ and input $x \in X$ there is at least one transition in T that has starting state s and whose label has input portion x. It is straightforward to see that M_0 is deterministic and completely specified.

A sequence $\bar{\rho}$ of consecutive transitions $(s_1, s_2, x_1/y_1) \dots (s_k, s_{k+1}, x_k/y_k)$ is a path of M that has *starting state* s_1, *ending state* $s_{k+1} = tail(\bar{\rho})$, and *label* $x_1/y_1 \dots x_k/y_k = label(\bar{\rho})$. An input/output sequence, or *trace*, $x_1/y_1 \dots x_k/y_k$ can also be represented as $\bar{x}/\bar{y}$ for input sequence $\bar{x} = x_1 \dots x_k$ and output sequence $\bar{y} = y_1 \dots y_k$. Here $\bar{x}$ is the *input portion* of $\bar{x}/\bar{y}$. For example, $(s_0, s_1, x_U/(y_U, -))(s_1, s_2, x_L/(y_U, y_L))$ is a path of M_0 with starting state s_0, ending state s_2 and label $x_U/(y_U, -)x_L/(y_U, y_L)$ and this label has input portion $x_U x_L$.

Given FSM M and state s of M, we let $L_M(s)$ denote the regular language formed from the labels of the paths of M that have starting state s and we let $L(M)$ denote $L_M(s_0)$. FSMs M_1 and M_2 are *globally equivalent* if $L(M_1) = L(M_2)$ and states s_i and s_j of FSM M are *globally equivalent* if $L_M(s_i) = L_M(s_j)$. Given completely specified FSMs M_1 and M_2 that have the same input alphabet, M_1 is a *reduction* of M_2 if $L(M_1) \subseteq L(M_2)$.

In testing when there is only one port it is common to use input sequences that distinguish states of the FSM from which tests are being generated. Given input sequence $\bar{x}$ let $L_M(s, \bar{x})$ denote the set of traces from $L_M(s)$ that have input portion $\bar{x}$. An input sequence $\bar{x}$ globally distinguishes state s_1 and s_2

of an FSM M if responses to $\bar{x}$ are defined from states s_1 and s_2 and there is no common response to $\bar{x}$ from s_1 and s_2. More formally, input sequence $\bar{x}$ *globally distinguishes* states s_1 and s_2 of an FSM M if $L_M(s_1, \bar{x})$ and $L_M(s_2, \bar{x})$ are non-empty and $L_M(s_1, \bar{x}) \cap L_M(s_2, \bar{x}) = \emptyset$.

The state s of an FSM M defines the regular language $L_M(s)$ and M defines the same language as its initial state. Thus, in order to compare FSMs M_1 and M_2 it is sufficient to compare the initial states of M_1 and M_2. Thus, it will transpire that an input sequence distinguishes two FSMs if and only if it distinguishes their initial states. As a result, it is usually possible to transfer results regarding distinguish states of an FSM to distinguishing FSMs. In order to do this formally, given FSMs M_1 and M_2, we define the FSM $M_1 \oplus M_2$ formed by taking the disjoint union of M_1 and M_2 (Definition 1). If the initial states of M_1 and M_2 are s_0 and q_0 respectively then we construct $M_1 \oplus M_2$ so that $L_{M_1 \oplus M_2}(s_0) = L(M_1)$ and $L_{M_1 \oplus M_2}(q_0) = L(M_2)$ and so an input sequence distinguishes M_1 and M_2 if and only if it distinguishes states s_0 and q_0 of $M_1 \oplus M_2$. We will define the initial state of $M_1 \oplus M_2$ to be the initial state of M_1 but in this paper the choice of initial state is not important. The following formally defines this for the case where the state sets of M_1 and M_2 are disjoint: If they are not disjoint then we simply relabel the states of one of the FSMs.

Definition 1 *Given FSMs* $M_1 = (S, s_0, X, Y, T_1)$ *and* $M_2 = (Q, q_0, X, Y, T_2)$ *with the same input and output alphabets and in which* $S \cap Q = \emptyset$ *we define* $M_1 \oplus M_2$ *to be the FSM* $(S \cup Q, s_0, X, Y, T_1 \cup T_2)$.

The key point is that $M_1 \oplus M_2$, M_1 and M_2 satisfy: $L(M_1) = L_{M_1 \oplus M_2}(s_0)$ and $L(M_2) = L_{M_1 \oplus M_2}(q_0)$. This allows us to transfer results regarding comparing states to problems in which we compare FSMs.

Given an input/output sequence $\bar{z}$ and a port i it is possible to define the projection $\pi_i(\bar{z})$ of $\bar{z}$ at i (see, for example, [19]).

$$
\begin{aligned}
\pi_i(\epsilon) &= \epsilon \\
\pi_i((x/(y_1, \ldots, y_m))\bar{z}) &= \pi_i(\bar{z}) \text{ if } x \notin X_i \wedge y_i = - \\
\pi_i((x/(y_1, \ldots, y_m))\bar{z}) &= x\pi_i(\bar{z}) \text{ if } x \in X_i \wedge y_i = - \\
\pi_i((x/(y_1, \ldots, y_m))\bar{z}) &= y_i\pi_i(\bar{z}) \text{ if } x \notin X_i \wedge y_i \neq - \\
\pi_i((x/(y_1, \ldots, y_m))\bar{z}) &= xy_i\pi_i(\bar{z}) \text{ if } x \in X_i \wedge y_i \neq -
\end{aligned}
$$

For example, $\pi_U(x_U/(y_U, -)x_L/(y_U, y_L)) = x_U y_U y_U$.

Given an input/output pair x/y, $ports(x/y)$ will denote the set of ports involved in x/y and so $ports(x/y) = \{i \in P | \pi_i(x/y) \neq \epsilon\}$. Given a transition

$t = (s_i, s_j, x/y)$, $ports(t) = ports(x/y)$ and $port(x)$ denotes the port i such that $x \in X_i$.

2.3 Controllability and observability problems

In the distributed test architecture, formalised by ISO [21], there are multiple ports/interfaces, a tester at each port, the testers cannot directly communicate with one another, and there is no global clock. Each tester is given a test script and is required to apply this test script. A controllability problem occurs if a tester is to apply an input and does not know when to apply this input since it was not involved in the previous transition. Let us suppose, for example, that input of x_U at U should lead to output y_U at U only and this is to be followed by input of x_L at L. Then the tester at L does not know whether the input x_U has been supplied and so does not know when to apply input x_L. If there are no controllability problems in a path then it and its label are said to be synchronisable (Definition 2).

There are no controllability problems in a path of the FSM with label x_1/y_1, $\ldots, x_k/y_k$ if this global trace has the property that for all $1 < i \leq k$ the tester to apply input x_i knows *when* to send x_i. The tester can only know when to send input x_i if it knows that x_{i-1} has already been sent and it can only know this if either it sent x_{i-1} or if it should receive an output produced by the SUT in response to x_{i-1}. If for all $1 < i \leq k$ the tester to apply x_i knows when to send x_i then $x_1/y_1, \ldots, x_k/y_k$ is synchronisable.

Definition 2 *Let us suppose that $\bar{\rho}$ is a path in an FSM with starting state s and label $\bar{z} = x_1/y_1, \ldots, x_k/y_k$ that has input portion $\bar{x}$. Then $\bar{\rho}$ and $\bar{z}$ are* synchronisable *if for all $1 < i \leq k$ we have that $port(x_i) \in ports(x_{i-1}/y_{i-1})$. In addition, we say that $\bar{x}$ is* synchronisable from *s.*

If a path with label $x_1/y_1, \ldots, x_k/y_k$ and starting state s_0 is not synchronisable and we attempt to apply input sequence $x_1, \ldots, x_k$ then we cannot know whether the SUT actually received the inputs in this order. This is a result of controllability problems and since we wish to avoid such controllability problems it is normal to aim to test with input sequences that correspond to synchronisable paths.

Note that by definition, all sequences and paths of length 0 and 1 are synchronisable. In the distributed test architecture each tester observes only the behaviour at its port and not the entire global behaviour. The tester thus compares the behaviour observed at its port with the expected behaviour and detects a failure if these are different. Observability problems occur when there is a difference in the global behaviour and yet no tester detects a failure: fault

masking has occurred. Let us suppose, for example, that input x_U is to be applied at port U, this should lead to output y_U at U only, and we then apply input x_U at U that in turn should lead to output y_U at U and y_L at L. Then no tester observes a failure if the first input leads to output y_U and y_L and the second leads to output y_U only: the tester at U observes the expected trace $x_U y_U x_U y_U$ and the tester at L observes the expected trace x_L. Two faulty transitions have masked one another in this test sequence but may lead to failures observed in use if the transitions are included in different sequences.

When we are testing in the distributed test architecture, we can only apply an input sequence without introducing controllability problems if the corresponding trace in M is synchronisable. Since we only consider input sequences that do not cause controllability problems in M, we can relax the usual restriction that an FSM considered is deterministic and completely specified and this will prove to be useful. Essentially, we can allow an FSM to be incompletely specified or nondeterministic in response to input sequences that we will not apply in testing since they cause controllability problems. This will give us scope to allow an FSM that we are comparing with M to be incompletely specified or nondeterministic as long as it is completely specified and deterministic for every input sequence that we might use in testing.

Definition 3 *Given FSMs M and N with the same input and output alphabets, N is* s_M-deterministic *if for every synchronisable path $\bar{\rho}$ of M with starting state s_0 and input portion $\bar{x}$ we have that N has exactly one path $\bar{\rho}'$ from its initial state such that label$(\bar{\rho}')$ has input portion $\bar{x}$.*

Throughout this paper we assume that M is a deterministic and completely specified FSM. We let Φ denote the set of s_M-deterministic FSMs with the same set of ports as M and the same input and output alphabets. Clearly, in discussing FSMs that are s-equivalent to M it is sufficient to only consider FSMs from Φ.

2.4 Locally s-distinguishing states and FSMs

This paper considers testing in the distributed test architecture. We wish to avoid controllability problems and thus, as usual, we assume that in testing we will only apply an input sequence if it is the input portion of the label of a synchronisable path of M. We also assume that observations are made locally. This scenario leads to the notion of locally s-distinguishing states introduced for deterministic FSMs [19]. The basic idea is that an input sequence $\bar{x}$ locally s-distinguishes two states s_1 and s_2 if it leads to no controllability problems when applied in states s_1 and s_2 and there is a port i such that the tester at i makes different observations when $\bar{x}$ is applied in states s_1 and s_2.

In this paper we allow a restricted form of nondeterminism: an FSM can be nondeterministic as long as it is s_M-deterministic. We now extend the notion of locally s-distinguishing two states to such FSMs, restricting testing to applying an input sequence for which there is only one corresponding path. This will allow us to compare an FSM M with FSMs that are s_M-deterministic.

Definition 4 *Input sequence* $\bar{x}$ locally s-distinguishes *states* s_1 *and* s_2 *of a possibly nondeterministic FSM* M_1 at port i *if* $\bar{x}$ *is the input portion of a unique path* $\bar{\rho}_1$ *from* s_1, $\bar{x}$ *is the input portion of a unique path* $\bar{\rho}_2$ *from* s_2, $\bar{\rho}_1$ *and* $\bar{\rho}_2$ *are synchronisable, and* $\pi_i(label(\bar{\rho}_1)) \neq \pi_i(label(\bar{\rho}_2))$. *Further,* $\bar{x}$ locally s-distinguishes *states* s_1 *and* s_2 *of* M *if there exists a port* $i \in P$ *such that* $\bar{x}$ *locally s-distinguishes* s_1 *and* s_2 *at* i. *If no input sequence locally s-distinguishes states* s_1 *and* s_2 *then they are locally s-equivalent.*

Consider again the FSM M_0 given in Figure 3. It is straightforward to see that no two states of M_0 are globally equivalent. However, we can observe that the only paths from states s_0 and s_3 that are synchronisable are paths whose label has an input portion of the form of either a sequence of zero or more instances of x_L or a sequence of zero or more instances of x_U. Further, for all such input sequences the traces from s_0 and s_3 are identical and so s_0 and s_3 are locally s-equivalent.

We can extend the definition from [19] to say what it means to locally s-distinguish two deterministic FSMs: it is sufficient to locally s-distinguish their initial states. However, for the purposes of this paper M is deterministic and we allow FSMs other than M to be nondeterministic as long as they are s_M-deterministic.

Definition 5 *Input sequence* $\bar{x}$ locally s-distinguishes *the FSM* M *and the* s_M*-deterministic FSM* $M_1 \in \Phi$ at port $i \in P$ *if* $\bar{x}$ *locally s-distinguishes the initial states of* M *and* M_1 *at* i *in the FSM* $M \oplus M_1$. *If there exists some such* $\bar{x}$ *and* i *then we say that* M *and* M_1 *are* locally s-distinguished by $\bar{x}$ *and that* M *and* M_1 *are* locally s-distinguishable. *If no input sequence locally s-distinguishes* M *and* M_1 *then they are said to be* locally s-equivalent.

Proposition 1 *Given FSM* M *and* s_M*-deterministic FSM* $M_1 \in \Phi$, *if* $\bar{x}$ *is the input portion of the label* $\bar{z}$ *of a synchronisable path* $\bar{\rho}$ *from the initial state of* M, $\bar{x}$ *is the input portion of the label* $\bar{z}_1$ *of a synchronisable path* $\bar{\rho}_1$ *from the initial state of* M_1 *and we have that* $\pi_i(\bar{z}) \neq \pi_i(\bar{z}_1)$ *then* $\bar{x}$ *locally s-distinguishes* M *and* M_1 *at port* i

Proof

This follows from the fact that the uniqueness of $\bar{\rho}$ and $\bar{\rho}_1$ is guaranteed by M being deterministic and M_1 being s_M-deterministic. □

3 A smallest locally s-equivalent FSM

This section describes how we can produce an s_M-deterministic FSM $\chi_{min}(M)$ that is locally s-equivalent to the completely specified deterministic FSM M and is minimal in the sense that for all $N \in \Phi$, if N is locally s-equivalent to M then $L(\chi_{min}(M)) \subseteq L(N)$. The motivation is that in order for an implementation N to be locally s-equivalent to M it must implement all of the traces in $L(\chi_{min}(M))$. Thus, these are the traces that must be included if we are building an implementation that should be indistinguishable from M when the use corresponds to the application of synchronisable input sequences and observations are made locally.

Previous work has shown how we can produce a rooted digraph G' in which there is a correspondence between the synchronisable paths in M and the paths from the root of G' [18]. However, this previous work only considers the case where there are two ports and in addition G' contains edges with no corresponding input or output and so cannot be directly converted into an FSM. In this section we use a related construction to generate an s_M-deterministic FSM $\chi_{min}(M)$ in which every path in $\chi_{min}(M)$ corresponds to a synchronisable path in M and every synchronisable path in M corresponds to a path in $\chi_{min}(M)$. We then prove that $\chi_{min}(M)$ is the FSM we are looking for.

For each state $s_i \in S$ and port $k \in P$ we define $Depart^k(s_i) = \{(s_i, s_j, x/y) \in T | x \in X_k\}$ which is the set of transitions of M whose starting state is s_i and whose input is at port k [18]. Similarly, for state s_i and set $\mathcal{P} \subseteq P$ of ports we define $Arrive^{\mathcal{P}}(s_i) = \{(s_j, s_i, x/y) \in T | ports(x/y) = \mathcal{P}\}$. $Arrive^{\mathcal{P}}(s_i)$ is the set of transitions of M whose ending state is s_i and that involve the set $\mathcal{P}$ of ports and so can *only* be followed by input at a port j if $j \in \mathcal{P}$; otherwise there will be controllability problems [18]. Thus, in a synchronisable path a transition from $Arrive^{\mathcal{P}}(s_i)$ can only be followed by a transition t if t is in $Depart^j(s_i)$ for some $j \in \mathcal{P}$.

We can now define $\chi_{min}(M) = (S', s'_0, X, Y, T')$. For each state $s_i \in S$ and $\mathcal{P} \subseteq P$ there can be a vertex $s_i^{\mathcal{P}}$ that represents the situation in which the next input must be at a port in $\mathcal{P}$. We define S' in the following way.

(1) For all $1 \leq i \leq n$ and $\mathcal{P} \subseteq P$ we include $s_i^{\mathcal{P}}$ in S' if $Arrive^{\mathcal{P}}(s_i) \neq \emptyset$.
(2) State s_0^P is in S' and $s'_0 = s_0^P$.

We include s_0^P in S' since we need to represent the situation in which we are in the initial state and have yet to apply any input; here we can apply input at any port. We can now define T' in the following way: for each transition $t = (s_i, s_j, x/y)$ and $s_i^{\mathcal{P}} \in S'$ with $port(x) \in \mathcal{P}$ we include in T' the transition $(s_i^{\mathcal{P}}, s_j^{\mathcal{P}_t}, x/y)$ where $\mathcal{P}_t = ports(x/y)$.

Transition	*Depart* sets	*Arrive* sets
$(s_0, s_0, x_L/(-, y_L))$	$Depart^L(s_0)$	$Arrive^L(s_0)$
$(s_0, s_1, x_U/(y_U, -)$	$Depart^U(s_0)$	$Arrive^U(s_1)$
$(s_1, s_2, x_L/(y_U, y_L))$	$Depart^L(s_1)$	$Arrive^{U,L}(s_2)$
$(s_1, s_0, x_U/(y_U, -))$	$Depart^U(s_1)$	$Arrive^U(s_0)$
$(s_2, s_0, x_L/(-, y_L))$	$Depart^L(s_2)$	$Arrive^L(s_0)$
$(s_2, s_3, x_U/(y_U, -))$	$Depart^U(s_2)$	$Arrive^U(s_3)$
$(s_3, s_3, x_L/(-, y_L)$	$Depart^L(s_3)$	$Arrive^L(s_3)$
$(s_3, s_2, x_U/(y_U, -)$	$Depart^U(s_3)$	$Arrive^U(s_2)$

Table 1
The $Depart^p$ and $Arrive^p$ sets for M_0

Naturally, any unreachable states can be removed from $\chi_{min}(M)$ but this will not affect the results since they do not contribute to $L(\chi_{min}(M))$.

The construction guarantees that for each transition $t \in T$ that occurs in a synchronisable path in M there is at least one corresponding transition in T'. Naturally, transitions that are not in synchronisable paths need not be included. Consider, for example, the FSM M_0 shown in Figure 3. The sets produced in the process of constructing $\chi_{min}(M_0)$ are shown in Table 1 and the resultant FSM is shown in Figure 4. Throughout this paper, when considering two ports U and L and state s_i we use the s_i^U, s_i^L, and $s_i^{U,L}$ to denote $s_i^{\{U\}}$, $s_i^{\{L\}}$ and $s_i^{\{U,L\}}$ respectively.

We now show how M and $\chi_{min}(M)$ relate.

Proposition 2 *For each synchronisable path $\bar{\rho}$ in M that starts at s_0, there is a unique synchronisable path $\bar{\rho}'$ in $\chi_{min}(M)$ that starts at s_0^P such that $label(\bar{\rho}) = label(\bar{\rho}')$.*

Proof

Proof will proceed by induction on the length of $\bar{\rho}$. Clearly the result holds for the base cases of paths of length 0 and 1.

Inductive case: let us suppose that $\bar{\rho} = \bar{\rho}_1 t$ for non-empty path $\bar{\rho}_1$ and transition t. Since $\bar{\rho}$ is a synchronisable path in M that starts at s_0, $\bar{\rho}_1$ must also be a synchronisable path in M that starts at s_0. Then, by the inductive hypothesis, there is a unique synchronisable path $\bar{\rho}'_1$ of $\chi_{min}(M)$ that starts at s_0^P such that $label(\bar{\rho}_1) = label(\bar{\rho}'_1)$.

Consider now the final transition t'_0 of $\bar{\rho}'_1$ and let the state of M reached by

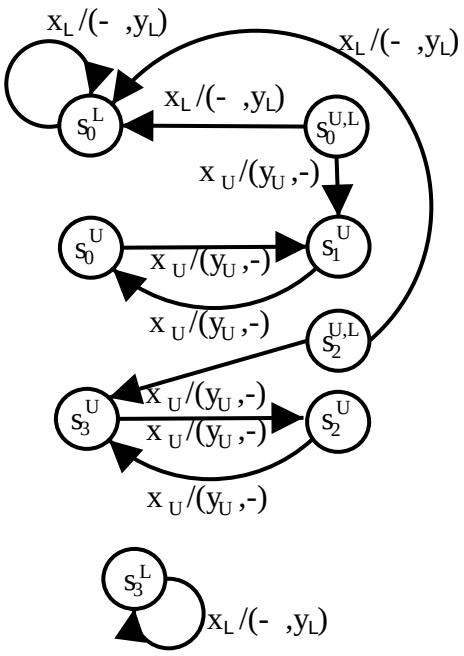

Fig. 4. The FSM $\chi_{min}(M_0)$

$\bar{\rho}_1$ be s_i (recall that $|\bar{\rho}'_1| \geq 1$). Let p denote the port such that the input from t is in X_p and so $p \in ports(t'_0)$ since $\bar{\rho}$ is synchronisable. By the definition of $\chi_{min}(M)$, the final vertex of $\bar{\rho}'_1$ is $s_i^{\mathcal{P}}$ for some $\mathcal{P}$ such that $p \in \mathcal{P}$. Thus, by the definition of $\chi_{min}(M)$, it is possible to follow $\bar{\rho}'_1$ by a transition t' with $label(t) = label(t')$ as required. By the definition of $\chi_{min}(M)$, t' is unique and so the result follows. □

Proposition 3 *For each path $\bar{\rho}'$ in $\chi_{min}(M)$ that starts at s_0^P, there is a unique synchronisable path $\bar{\rho}$ in M that starts at s_0 such that $label(\bar{\rho}) = label(\bar{\rho}')$.*

Proof

Proof will proceed by induction on the length of $\bar{\rho}'$. Clearly the result holds for the base cases of paths of length 0 and 1.

Inductive case: let us suppose that $\bar{\rho}' = \bar{\rho}'_1 t'$ for non-empty path $\bar{\rho}'_1$ and transition t'. Since $\bar{\rho}'$ is a path in $\chi_{min}(M)$ that starts at s_0^P, $\bar{\rho}'_1$ must also be a path in $\chi_{min}(M)$ that starts at s_0^P. So, by the inductive hypothesis, there is a unique synchronisable path $\bar{\rho}_1$ of M that starts at s_0 such that $label(\bar{\rho}_1) = label(\bar{\rho}'_1)$.

Consider the final transition t_0 of $\bar{\rho}_1$ and let s_i be the ending state of $\bar{\rho}_1$. Let p denote the port such that the input from t' is in X_p and so by the definition of $\chi_{min}(M)$ we have that $p \in ports(t_0)$. Since t' has input at port p and it is possible to follow $\bar{\rho}_1$ by input at p without causing a controllability problem, we have that there exists a transition t of M such that $\bar{\rho}_1 t$ is synchronisable and

$label(t) = label(t')$ as required. Clearly t is unique and so the result follows. □

Proposition 4 *If M is a deterministic FSM then all paths in $\chi_{min}(M)$ are synchronisable and $\chi_{min}(M)$ is s_M-deterministic.*

Proof

This result follows from Propositions 2 and 3 and the definition of $\chi_{min}(M)$. □

The following three results are similar to results proved in [19]. In contrast to [19] they allow some nondeterminism in the FSMs considered but the results contain hypotheses that essentially insist that the behaviour along the relevant paths is deterministic.

Proposition 5 *Let us suppose that s_1 and s_2 are states of an FSM M_1 and $\bar{x}$ is an input sequence such that for $i \in \{1, 2\}$ there is exactly one path from state s_i with a label whose input portion is $\bar{x}$. If $\bar{x}$ locally s-distinguishes states s_1 and s_2 then $\bar{x}$ globally distinguishes s_1 and s_2.*

Proof

For $i \in \{1, 2\}$ let $\bar{\rho}_i$ denote the path from state s_i that has label with input portion $\bar{x}$. Then the set of possible responses to $\bar{x}$ from s_1 is $\{label(\bar{\rho}_1)\}$ and the set of possible responses to $\bar{x}$ from s_2 is $\{label(\bar{\rho}_2)\}$. Since $\bar{x}$ locally s-distinguishes s_1 and s_2 these sets are disjoint and so, by definition, $\bar{x}$ globally distinguishes states s_1 and s_2. □

Proposition 6 *Let us suppose that s_1 and s_2 are states of an FSM M_1 and $\bar{x}$ is an input sequence such that for $i \in \{1, 2\}$ there is exactly one path from state s_i with a label whose input portion is $\bar{x}$ and this path is synchronisable. If $\bar{x}$ globally distinguishes s_1 and s_2 and $\bar{x}'$ is a minimal prefix of $\bar{x}$ that globally distinguishes s_1 and s_2 then $\bar{x}'$ locally s-distinguishes s_1 and s_2.*

Proof

For s_i, $i \in \{1, 2\}$, let $\bar{\rho}_i$ denote the unique path with starting state s_i that has a label with input portion $\bar{x}'$. For $i \in \{1, 2\}$ let $\bar{\rho}'_i$ denote the path formed by deleting the last element of $\bar{\rho}_i$. By the minimality of $\bar{x}$, $label(\bar{\rho}'_1) = label(\bar{\rho}'_2)$ and so for all $i \in P$ we must have that $\pi_i(label(\bar{\rho}'_1)) = \pi_i(label(\bar{\rho}'_2))$. Thus, since $label(\bar{\rho}'_1) \neq label(\bar{\rho}'_2)$, there must be a port i such that the output of the last transitions of $\bar{\rho}'_1$ and $\bar{\rho}'_1$ are different and so $\pi_i(label(\bar{\rho}_1)) \neq \pi_i(label(\bar{\rho}_2))$ as required. □

Proposition 7 *Let us suppose that s_1 and s_2 are states of an FSM M_1 and $\bar{x}$ is an input sequence such that for every $i \in \{1, 2\}$ there is exactly one path*

from state s_i with a label whose input portion is $\bar{x}$. If $\bar{x}$ is synchronisable from s_1 but not from s_2 then there is a prefix of $\bar{x}$ that locally s-distinguishes s_1 and s_2.

Proof

Let $\bar{x}_1$ denote the longest prefix of $\bar{x}$ such that there are synchronisable paths from both s_1 and s_2 whose labels have input portion $\bar{x}_1$. Let $\bar{x}_1 = \bar{x}'_1 x$ for some $\bar{x}'_1$ and x. If $\bar{x}'_1$ locally s-distinguishes states s_1 and s_2 then the result follows and so we assume that $\bar{x}'_1$ does not locally s-distinguish s_1 and s_2.

Let s'_1 and s'_2 be the states reached from s_1 and s_2 respectively using input sequence $\bar{x}'_1$. The responses to x in s'_1 and s'_2 must differ at some port because the input in $\bar{x}$ after x causes a controllability problem from one of these states but not the other. Thus $\bar{x}$ locally s-distinguishes s_1 and s_2 and so the result follows. □

We can combine these to get the following result.

Proposition 8 *Let us suppose that N is an s_M-deterministic FSMs in Φ. Then N and M are locally s-distinguishable if and only if there exists an input sequence $\bar{x}$ such that $\bar{x}$ is synchronisable from the initial states of N and M and $\bar{x}$ globally distinguishes N and M.*

Proof

We will consider the initial states of N and M in the FSM $M \oplus N$ formed by taking the disjoint union of M and N.

First assume that N and M are locally s-distinguishable and that $\bar{x}$ locally s-distinguishes them. By definition, $\bar{x}$ is synchronisable from the initial states of N and M. Since M is deterministic and N is s_M-deterministic, by Proposition 5 we have that $\bar{x}$ globally distinguishes N and M as required.

Now assume that there exists an input sequence $\bar{x}$ such that $\bar{x}$ is synchronisable from the initial states of N and M and $\bar{x}$ globally distinguishes N and M. Then by Proposition 6 we have that N and M are locally s-distinguishable as required. □

We can now prove the main result of this section.

Theorem 1 *For a deterministic and completely specified FSM M, if $N \in \Phi$ then N is locally s-equivalent to M if and only if $L(\chi_{min}(M)) \subseteq L(N)$.*

Proof

First assume that N is locally s-equivalent to M and let $\bar{x}/\bar{y}$ be an element of $L(\chi_{min}(M))$. Thus, there is a path $\bar{\rho}'$ of $\chi_{min}(M)$ that has starting state s_0^P and label $\bar{x}/\bar{y}$. By Proposition 3 we know that there is a synchronisable path $\bar{\rho}$ of M that has starting state s_0 and label $\bar{x}/\bar{y}$. Thus, since N is s_M-deterministic, N and M are locally s-equivalent and $\bar{\rho}$ is a synchronisable path of M, by Propositions 6 and 7 there must be a path from the initial state of N that has label $\bar{x}/\bar{y}$ and thus $\bar{x}/\bar{y} \in L(N)$. Since this holds for an arbitrary element of $L(\chi_{min}(M))$ we must have $L(\chi_{min}(M)) \subseteq L(N)$ as required.

Now assume that $L(\chi_{min}(M)) \subseteq L(N)$; we require to prove that N is locally s-equivalent to M. Proof by contradiction: assume that N is not locally s-equivalent to M. By Proposition 8 there are synchronisable paths from the initial states of M and N whose labels have input portion $\bar{x}$ for an input sequence $\bar{x}$ that globally distinguishes M and N. Then, since M is deterministic and N is s_M-deterministic, there is exactly one output sequence $\bar{y}$ such that $\bar{x}/\bar{y} \in L(M)$ and there is exactly one output sequence $\bar{y}'$ such that $\bar{x}/\bar{y}' \in L(N)$ and we must have that $\bar{y} \neq \bar{y}'$. By Proposition 2 we have that $\bar{x}/\bar{y} \in L(\chi_{min}(M))$ and so $\bar{x}/\bar{y} \in L(N)$. This provides a contradiction as required. □

The FSM $\chi_{min}(M)$ thus defines those traces from M that must be implemented in order for an s_M-deterministic FSM to be locally s-equivalent to M. As a result, the other traces from M can be seen as optional and further traces can be added as long as they do not stop the implementation being s_M-deterministic. In the next section we show how we can complete $\chi_{min}(M)$ in a maximal manner.

The FSM $\chi_{min}(M)$ can be constructed in time that is polynomial in the number of states of M.

Proposition 9 *Given a completely specified deterministic FSM M with transition set T and input alphabet X we have that $\chi_{min}(M)$ has at most $|T|+1$ states and at most $|X|(|T|+1)$ transitions.*

Proof

We only include the state $s_i^{\mathcal{P}}$ if $Arrive^{\mathcal{P}}(s_i)$ is non-empty and this requires there to be a transition t with ending state s_i such that $ports(t) = \mathcal{P}$. As a result, in the worst case we obtain one state in $\chi_{min}(M)$ for every transition of M in addition to s_0^P and so $\chi_{min}(M)$ has at most $|T|+1$ states. In addition, since $\chi_{min}(M)$ is deterministic it has at most $|X|$ transitions leaving each state and so no more than $|X|(|T|+1)$ transitions. □

4 A largest locally s-equivalent FSM

The use of the distributed test architecture reduces the ability of testing to distinguish between FSMs. A natural question is: For a given FSM specification M, what traces that are not in $L(M)$ might be contained in an implementation despite the implementation being locally s-equivalent to M? This section shows how we can answer this question by producing an s_M-deterministic FSM $\chi_{max}(M)$ that is locally s-equivalent to M and that has the property that for an FSM $N \in \Phi$ we have that N is locally s-equivalent to M if and only if $L(N) \subseteq L(\chi_{max}(M))$. This result has the following practical ramifications:

(1) Let us suppose that the use of the SUT N reflects the constraints placed on testing by the distributed test architecture: in use only synchronisable input sequences will be applied and observations can only be made locally at individual ports. Then N is acceptable if and only if N is a reduction of $\chi_{max}(M)$ and N does not have to be a reduction of M. Even if we can overcome controllability and observability problems through the use of coordination messages when testing the SUT N, we should not test to check that N is a reduction of M since N may be indistinguishable from M in use but still not be a reduction of M: we may get a false negative. Instead we should test to check that N is a reduction of $\chi_{max}(M)$.
(2) The traces in $L(\chi_{max}(M)) \setminus L(M)$ are the traces that are not in the specification and that can occur in machines indistinguishable from M if we are testing in the distributed test architecture. Thus, we can explore properties of $L(\chi_{max}(M))$ in order to investigate the potential impact of the limitations placed on testing by the distributed test architecture and this might be used to help decide whether it is worth introducing an external network through which coordination messages can be sent.

We will produce $\chi_{max}(M)$ by completing $\chi_{min}(M)$. We will want to be able to include multiple possible outputs in response to an input and so will introduce the symbol $*$ whose use as an output represents all outputs from Y. Thus a transition of the form $(s, s', x/*)$ in $\chi_{max}(M)$ will represent the situation where if x is received when $\chi_{max}(M)$ is in state s then $\chi_{max}(M)$ can move to state s' and produce any output from Y. The following is the algorithm for generating $\chi_{max}(M)$.

(1) Input FSM M.
(2) Produce $\chi_{min}(M)$.
(3) If $\chi_{min}(M)$ is completely specified then return $\chi_{min}(M)$ and stop.
(4) Form an FSM M_1 from $\chi_{min}(M)$ by adding a state s_c and for all $x \in X$ adding a transition $(s_c, s_c, x/*)$.
(5) Form $\chi_{max}(M)$ from M_1 in the following way: For every state s of M_1 and input $x \in X$ such that M_1 has no transition from s with input x,

add the transition $(s, s_c, x/*)$.

(6) Return $\chi_{max}(M)$.

Proposition 10 *Given deterministic and completely specified FSM M, $\chi_{max}(M)$ is s_M-deterministic.*

Proof

By Proposition 2, for every synchronisable path $\bar{\rho}$ in M from s_1, there is a unique synchronisable path $\bar{\rho}'$ in $\chi_{min}(M)$ from s_0^P such that $label(\bar{\rho}) = label(\bar{\rho}')$ and corresponding paths must exist in $\chi_{max}(M)$. Further, by Proposition 4 we know that $\chi_{min}(M)$ is s_M-deterministic. The result now follows from observing that if $\bar{\rho}$ is a synchronisable path in $\chi_{min}(M)$ from s_0^P that can be followed by input at $p \in P$ without causing a controllability problem and $x \in X_p$ then there is a transition in $\chi_{min}(M)$ from $tail(\bar{\rho})$ that has input x and thus the addition of transitions in Step 5 does not introduce nondeterminism in such situations. □

Proposition 11 *Given deterministic and completely specified FSM M we have that M is locally s-equivalent to $\chi_{max}(M)$.*

Proof

Clearly $L(\chi_{min}(M)) \subseteq L(\chi_{max}(M))$. By Proposition 10, $\chi_{max}(M)$ is s_M-deterministic and so the result follows from Theorem 1. □

Proposition 12 *Given deterministic and completely specified FSM M and FSM $N \in \Phi$, if $L(N) \subseteq L(\chi_{max}(M))$ then N is locally s-equivalent to M.*

Proof

Proof by contradiction: let us suppose that N is not locally s-equivalent to M. Then there exist input sequences that locally s-distinguish N and M and let $\bar{x}$ denote a minimal such input sequence. Let $\bar{x}/\bar{y}$ and $\bar{x}/\bar{y}'$ denote the labels of the synchronisable paths from the initial states of M and N respectively. Since M is deterministic and N is s_M-deterministic the sequences $\bar{x}/\bar{y}$ and $\bar{x}/\bar{y}'$ are uniquely defined and so $\bar{y} \neq \bar{y}'$. Clearly $\bar{x}/\bar{y} \in L(\chi_{max}(M))$. Since $L(N) \subseteq L(\chi_{max}(M))$ we have that $\bar{x}/\bar{y}' \in L(\chi_{max}(M))$ but this gives a contradiction since, by Proposition 10 we know that $\chi_{max}(M)$ is s_M-deterministic. □

Proposition 13 *Given deterministic and completely specified FSM M and $N \in \Phi$, we have that if N is locally s-equivalent to M then $L(N) \subseteq L(\chi_{max}(M))$.*

Proof

Assume that N is locally s-equivalent to M and let $\bar{x}/\bar{y}$ be some element of $L(N)$ and so it is sufficient to prove that $\bar{x}/\bar{y} \in L(\chi_{max}(M))$. We will use proof by induction on the length of $\bar{x}/\bar{y}$. The result clearly holds for the base case of sequences of length 0 or 1.

Inductive case: let $\bar{x}/\bar{y} = \bar{x}_1 x/\bar{y}_1 y$ where $x \in X$ and $y \in Y$. By the inductive hypothesis $\bar{x}_1/\bar{y}_1 \in L(\chi_{max}(M))$. If $\bar{x}_1/\bar{y}_1$ is not the label of a synchronisable path of M from s_0 then by the definition of $\chi_{max}(M)$ we know that for all $y' \in Y$ we have that $\bar{x}_1 x/\bar{y}_1 y' \in L(\chi_{max}(M))$ and so the result follows. Similarly, if $\bar{x}_1 x/\bar{y}_1 y$ is not synchronisable then $\chi_{max}(M)$ can produce all possible output in response to x after $\bar{x}_1/\bar{y}_1$ and so the result follows. Finally, consider the case where $\bar{x}_1/\bar{y}_1$ is the label of a synchronisable path of M from s_0 and $\bar{x}_1 x/\bar{y}_1 y$ is synchronisable. Since N is locally s-equivalent to M we have that, by Definition 5, $\bar{x}_1 x/\bar{y}_1 y$ is the label of a synchronisable path of M from s_0. By Proposition 2 we have that $\bar{x}_1 x/\bar{y}_1 y \in L(\chi_{min}(M)) \subseteq L(\chi_{max}(M))$ as required. □

Theorem 2 *Given deterministic and completely specified FSM M, for every s_M-deterministic FSM N we have that N is locally s-equivalent to M if and only if $L(N) \subseteq L(\chi_{max}(M))$.*

Proof

The result follows from Propositions 12 and 13. □

It is clear that the complexity of producing $\chi_{max}(M)$ is dominated by the step that devises $\chi_{min}(M)$.

Proposition 14 *Given a completely specified deterministic FSM M with transition set T and input alphabet X we have that $\chi_{max}(M)$ has at most $|T|+2$ states and at most $(|T|+2)|X|$ transitions.*

Proof

This follows from $\chi_{max}(M)$ having at most one more state than $\chi_{min}(M)$ and the fact that for each state s it has $|X|$ transitions that leave s. □

5 The set of locally s-equivalent FSMs

We have seen that there exist minimal and maximal elements of the set of FSMs that are locally s-equivalent to M. This section proves that the set of s_M-deterministic FSMs that are locally s-equivalent to M defines a bounded lattice. This will be achieved by, for two FSMs M_1 and M_2, defining an

FSM $Int(M_1, M_2)$ such that $L(Int(M_1, M_2)) = L(M_1) \cap L(M_2)$ and an FSM $U(M_1, M_2)$ such that $L(U(M_1, M_2)) = L(M_1) \cup L(M_2)$.

Definition 6 *Given FSMs $M_1 = (S, s_0, X, Y, T_1)$ and $M_2 = (Q, q_0, X, Y, T_2)$ with the same input and output alphabets we define*

(1) The FSM $Int(M_1, M_2)$ is $(S \times Q, (s_0, q_0), X, Y, T_{Int})$ where T_{Int} is defined by: $((s, q), (s', q'), x/y) \in T_{Int}$ if and only if $(s, s', x/y) \in T_1 \wedge (q, q', x/y) \in T_2$.
(2) The FSM $U(M_1, M_2)$ is $((S \cup \{\perp_1\}) \times (Q \cup \{\perp_2\}) \setminus \{(\perp_1, \perp_2)\}, (s_0, q_0), X, Y, T_U)$ where T_U is defined by: $((s, q), (s', q'), x/y) \in T_U$ if and only if either $(s, s', x/y) \in T_1 \wedge (q, q', x/y) \in T_2$ or $(s, s', x/y) \in T_1 \wedge (\neg\exists q'' \in Q.(q, q'', x/y) \in T_2) \wedge q' = \perp_2$ or $(\neg\exists s'' \in S.(s, s'', x/y) \in T_1) \wedge s' = \perp_1 \wedge (q, q', x/y) \in T_2$.

The following are important properties of $Int(M_1, M_2)$ and $U(M_1, M_2)$ and follow directly from the definitions.

Proposition 15 *If $M_1 = (S, s_0, X, Y, T_1)$ and $M_2 = (Q, q_0, X, Y, T_2)$ are s_M-deterministic and locally s-equivalent to M then the following hold:*

(1) $Int(M_1, M_2)$ is s_M-deterministic;
(2) $L(Int(M_1, M_2)) = L(M_1) \cap L(M_2)$;
(3) $U(M_1, M_2)$ is s_M-deterministic; and
(4) $L(U(M_1, M_2)) = L(M_1) \cup L(M_2)$.

Proposition 16 *Let us suppose that M_1 and M_2 are s_M-deterministic FSMs that are locally s-equivalent to M. Then there exists an s_M-deterministic FSM $M' \in \Phi$ such that M' is locally s-equivalent to M and $L(M') = L(M_1) \cap L(M_2)$.*

Proof

Let $M' = Int(M_1, M_2)$. By Theorem 1 we know that $L(\chi_{min}(M)) \subseteq L(M_1)$ and $L(\chi_{min}(M)) \subseteq L(M_2)$ and so $L(\chi_{min}(M)) \subseteq (L(M_1) \cap L(M_2))$. By Proposition 15, $L(M') = L(M_1) \cap L(M_2)$ and so we have that $L(\chi_{min}(M)) \subseteq L(M')$. By Proposition 15, M' is s_M-deterministic and so $M' \in \Phi$. Thus, by Theorem 1, M' is locally s-equivalent to M. □

Proposition 17 *Let us suppose that M_1 and M_2 are s_M-deterministic FSMs that are locally s-equivalent to M. Then there exists an s_M-deterministic FSM $M' \in \Phi$ such that M' is locally s-equivalent to M and $L(M') = L(M_1) \cup L(M_2)$.*

Proof

Let $M' = U(M_1, M_2)$. By Theorem 1 we know that $L(\chi_{min}(M)) \subseteq L(M_1)$ and $L(\chi_{min}(M)) \subseteq L(M_2)$ and so $L(\chi_{min}(M)) \subseteq (L(M_1) \cup L(M_2))$. By Proposition

15, $L(M') = L(M_1) \cup L(M_2)$ and so we have that $L(\chi_{min}(M)) \subseteq L(M')$. By Proposition 15, M' is s_M-deterministic and so $M' \in \Phi$. Thus, by Theorem 1, M' is locally s-equivalent to M. □

We let Φ_M denote the set of s_M-deterministic FSMs that are locally s-equivalent to M: these are the FSMs we consider in this section. There is a natural partial order on the languages defined by FSMs in Φ_M. This is not a partial order on the set of FSMs in Φ_M since two such FSMs may define the same languages. However, it becomes a partial order once we quotient out FSM equivalence.

Definition 7 *If two FSMs M_1 and M_2 are globally equivalent ($L(M_1) = L(M_2)$) then we write $M_1 \sim M_2$. We let $\tilde{\Phi}_M$ denote the set of equivalence classes of Φ_M under $\sim$ and given an FSM $M_1 \in \Phi_M$ we let $\|M_1\|$ denote the set of FSMs from Φ_M that are globally equivalent to M_1 and thus $\|M_1\|$ is in $\tilde{\Phi}_M$. For $||M_1||, ||M_2|| \in \tilde{\Phi}_M$ we write $\|M_1\| \sqsubseteq \|M_2\|$ if and only if $L(M_1) \subseteq L(M_2)$.*

For set A and partial order $\leq$ on A, $(A, \leq)$ is a *lattice* if for each pair $a_1, a_2 \in A$ we have that: there exists an element a^+, called the join of a_1 and a_2, that is the least upper bound of a_1 and a_2; and there exists an element a^-, called the meet of a_1 and a_2, that is the greatest lower bound of a_1 and a_2. A lattice $(A, \leq)$ is a *bounded lattice* if it contains a greatest element and a least element. We know from Propositions 16 and 17, that $(\tilde{\Phi}_M, \sqsubseteq)$ is a lattice. In addition, from Theorems 1 and 2, we know that $(\tilde{\Phi}_M, \sqsubseteq)$ contains minimal and maximal elements $||\chi_{min}(M)||$ and $||\chi_{max}(M)||$ respectively.

Theorem 3 *Given deterministic completely specified FSM M, $(\tilde{\Phi}_M, \sqsubseteq)$ is a bounded lattice.*

6 A locally s-equivalent FSM with fewest states

So far we have shown that there are unique minimal and maximal members of the set of FSMs that are locally s-equivalent to M. However, the notions of minimal and maximal were defined in terms of the language specified by an FSM, not by the size of its representation. If we intend to produce an implementation of M and the restrictions imposed by the distributed test architecture are also imposed in use (only synchronisable sequences are used and behaviour is observed locally) then we may want to implement a smallest deterministic complete design that is locally s-equivalent to M. In this section we therefore investigate the problem of producing a completely-specified deterministic FSM M' that has fewest states amongst all completely-specified deterministic FSMs that are locally s-equivalent to M.

The first observation that can be made is that we are looking for a completely-specified deterministic FSM that contains the behaviour of $\chi_{min}(M)$ and has fewest states amongst all completely-specified deterministic FSMs whose behaviour contains $\chi_{min}(M)$. This problem can be seen as that of minimising the partially specified FSM $\chi_{min}(M)$. The general problem of minimising a partially specified FSM is known to be NP-hard [28]. However, in this section we show that $\chi_{min}(M)$ can be minimised in polynomial time in the special case often considered in the literature in which there are two ports. We then consider the general case.

6.1 FSMs with two ports

In this section we only consider FSMs that have two ports U and L. Two states s_1 and s_2 of an FSM M_1 are globally equivalent if they define the same language: $L_{M_1}(s_1) = L_{M_1}(s_2)$. However, it is sometimes possible to merge two states that are not globally equivalent when minimising an incompletely specified FSM: we just require that the two states produce the same output for every input sequence $\bar{x}$ such that the response to $\bar{x}$ is defined from both states. More formally, states s_1 and s_2 of an FSM M_1 are *compatible* if for every input sequence $\bar{x}$ such that there is a path $\bar{\rho}_i$ from s_i whose label has input portion $\bar{x}$, $i \in \{1, 2\}$, we have that the labels of $\bar{\rho}_1$ and $\bar{\rho}_2$ are identical.

The process of minimising $\chi_{min}(M)$ will proceed via two phases: merging states that are globally equivalent and then merging states that are compatible. The approach described in this section is based on the following observations regarding $\chi_{min}(M)$.

(1) All paths from the initial state of $\chi_{min}(M)$ are synchronisable.
(2) For state s_i^α, $\alpha \in \{U, L\}$, for all $x \in X_\alpha$ we have that there is a transition from s_i^α with input x and for all $x \in X_\beta$, $\beta \neq \alpha$, we have that there is no transition from s_i^α with input x.
(3) States $s_i^{U,L}$ and $s_j^{U,L}$ are compatible if and only if $s_i^{U,L}$ and $s_j^{U,L}$ are globally equivalent and for $\alpha \in \{U, L\}$, states s_i^α and s_j^α are compatible if and only if s_i^α and s_j^α are globally equivalent.
(4) For any two states s_i and s_j, by definition we have that s_i^L and s_j^U are compatible.

We start by removing unreachable states, then merge globally equivalent states, and finally merge compatible states. The algorithm for generating the FSM $\chi_s(M)$ is given in Figure 5.

It is known that for an FSM with n states it is possible to decide whether two states are globally equivalent in $O(n \log n)$ time [20]. It has also been proved that the problem of deciding whether two states of an n state FSM are locally

(1) Input $\chi_{min}(M)$
(2) Delete all states that cannot be reached from $s_0^{U,L}$ and all transitions that start or end at such states.
(3) Produce an FSM $\chi_s(M)$ with state set S' and transition set T' by merging globally equivalent states of $\chi_{min}(M)$.
(4) While there exists $\alpha \in \{U, L\}$ and states s_i^α and $s_j^{U,L}$ of $\chi_s(M)$ that are locally s-equivalent at port α do
   (a) For every transition from a state s to s_i^α with label l add to T' a transition from s to $s_j^{U,L}$ with label l.
   (b) Delete state s_i^α from S' and delete all transitions from T' that have s_i^α as a starting state or ending state.
(5) Endwhile
(6) While there exists states s_i^L and s_j^U in S' do
   (a) Add a new state $s(i, j)$ to S'.
   (b) For every transition from a state $s \notin \{s_i^L, s_j^U\}$ to a state $s' \in \{s_i^L, s_j^U\}$ with label l add to T' a transition from s to $s(i, j)$ with label l.
   (c) For every transition from $s' \in \{s_i^L, s_j^U\}$ to a state $s \notin \{s_i^L, s_j^U\}$ with label l add to T' a transition from $s(i, j)$ to s with label l.
   (d) For every transition from $s' \in \{s_i^L, s_j^U\}$ to a state $s \in \{s_i^L, s_j^U\}$ with label l add to T' a transition from $s(i, j)$ to $s(i, j)$ with label l.
   (e) Delete states s_i^L and s_j^U from S' and delete from T' all transition that have either s_i^L or s_j^U as a starting state or an ending state.
(7) Endwhile
(8) If $\chi_s(M)$ is not completely specified then add arbitrary transitions to complete it.
(9) Output $\chi_s(M)$.

Fig. 5. Producing an FSM from Φ_M with fewest states

s-equivalent can be solved in time of $O(n^2)$. It is therefore clear that $\chi_s(M)$ can be produced in time that is polynomial in the number of states of M.

Now consider the generation of $\chi_s(M_0)$ for the FSM M_0 shown in Figure 3. The FSM $\chi_{min}(M_0)$ is given in Figure 4 and we can see that the only states that are reachable from $s_0^{U,L}$ are $s_0^{U,L}$, s_0^U, s_0^L, and s_1^U and so all other states are deleted in Step 2. In the next step the globally equivalent states s_0^U and s_1^U are merged and we can assume that the state s_1^U is eliminated. We now have an FSM with state set $\{s_0^{U,L}, s_0^U, s_0^L\}$. It is now sufficient to observe that s_0^U is locally s-equivalent to $s_0^{U,L}$ at U and s_0^L is locally s-equivalent to $s_0^{U,L}$ at L and so the states s_0^U and s_0^L can be eliminated in Step 4. This leaves us with an FSM with one state shown in Figure 6.

We are now in the position to prove the main results of this section.

Theorem 4 *Given deterministic and completely specified FSM M, the FSM $\chi_s(M)$ is locally s-equivalent to M.*

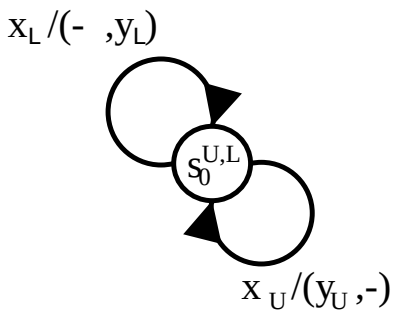

Fig. 6. A smallest FSM locally s-equivalent to M_0

Proof

By construction $\chi_s(M)$ is s_M-deterministic and we must have that $L(\chi_{min}(M)) \subseteq L(\chi_s(M))$. The result thus follows from Theorem 1. □

Theorem 5 *If FSMs M and M' are deterministic, completely specified and locally s-equivalent then the number of states of $\chi_s(M)$ is less than or equal to the number of states of M'.*

Proof

First observe that every transition from a state of the form $s_i^{U,L}$ in $\chi_{min}(M)$ can be included in a synchronisable path and thus that if $s_i^{U,L}$ and $s_j^{U,L}$ are not globally equivalent then they are not locally s-equivalent. As a result of this, for any such pair of states $s_i^{U,L}$ and $s_j^{U,L}$ of $\chi_{min}(M)$, if the end states of paths $\bar{\rho}_1$ and $\bar{\rho}_2$ in $\chi_{min}(M)$ are $s_i^{U,L}$ and $s_j^{U,L}$ respectively then in M' the paths with the same labels as $\bar{\rho}_1$ and $\bar{\rho}_2$ must reach distinct states that are locally s-equivalent to $s_i^{U,L}$ and $s_j^{U,L}$ respectively. Let k denote the number of pairwise locally s-distinguishable states of the form $s_i^{U,L}$ in $\chi_{min}(M)$. Clearly, $\chi_s(M)$ has k states of the form $s_i^{U,L}$ and M' must have at least k states that are locally s-equivalent to these states.

For port $\alpha \in P$ let k_α denote the number of pairwise globally distinguishable states of the form s_i^α in $\chi_{min}(M)$ that are not locally s-equivalent to any state of the form $s_j^{U,L}$ at α. Then clearly, for $\alpha \in P$, M' must have at least k_α states in addition to the k states that are locally s-equivalent to states of the form $s_j^{U,L}$ from $\chi_{min}(M)$. Thus, M' has at least $k + \max\{k_U, k_L\}$ states. But this is the number of states of $\chi_s(M)$ and so the result follows. □

6.2 General multi-port FSMs

We now consider the general case in which there are $m > 2$ ports. This problem is similar to minimising a partially specified FSM, a problem that is known to

be NP-hard in general. Pfleeger [28] proves that this is NP-hard by reducing an NP-hard graph colouring problem to it. A graph G is defined by a pair (V, E) in which V is a set of vertices and E is a set of unordered pairs of vertices, each element of E being an edge. An edge between vertices v and v' is represented by the unordered pair (v, v'), which is equal to (v', v). Let $G = (V, E)$ be a graph with vertices $V = \{v_1, \ldots, v_m\}$. The function $f : V \to \{1, \ldots, c\}$ *colours* G if for all $(v, v') \in E$ we have that $f(v) \neq f(v')$. Then the following *graph colouring problem* is NP-hard [23]: given G and c, does such a colouring function f exist?

We now adapt the proof of Pfleeger. We define an FSM $M(G, c)$ that is similar to a finite automaton used by Pfleeger. However, we require $M(G, c)$ to be completely-specified, so we introduce the opportunity for there to be many locally s-equivalent FSMs by including transitions that are not in any synchronisable path.

Definition 8 *Given graph $G = (V, E)$ with m vertices that has no loops and has no isolated vertices and c, we define the finite state machine $M(G, c) = (S, s_0, X, Y, T)$ with ports $1, \ldots, m$ where:*

(1) $S = V \cup \{S_0, S_N, S_F\}$ *($S_0 \notin V$, $S_N \notin V$, $S_F \notin V$)*
(2) $X = \{a_i | v_i \in V\}$ *in which a_i is input at i*
(3) $Y = \{0, 1, 2, -\} \times \ldots \times \{0, 1, 2, -\} = \{0, 1, 2, -\}^m$
(4) The set T of transitions is defined by:
- *For all $a_i \in \Sigma$, $(S_0, v_i, a_i/\bar{y}(i))$ is in T, where in $\bar{y}(i)$ the value 2 is sent to port j if $(v_i, v_j) \in E$ and $-$ is sent to all other ports*
- *For all $a_i \in \Sigma$, $(S_N, S_N, a_i/(0, \ldots, 0))$ is in T*
- *For all $a_i \in \Sigma$, $(S_F, S_F, a_i/(1, \ldots, 1))$ is in T*
- *For all $a_i \in \Sigma$, $(v_i, S_F, a_i/(1, \ldots, 1))$ is in T*
- *For all $a_i \in \Sigma, v_j \in V$, if $i \neq j$ then $(v_j, S_N, a_i/(0, \ldots, 0))$ is in T*

Note that every transition with ending state S_F has output $(1, \ldots, 1)$ and every transition with ending state S_N has output $(0, \ldots, 0)$. It should be clear that the transitions in $M(G, c)$ that are not contained in any synchronisable paths are those from a state v_j with input a_i such that $i \neq j$ and $(v_i, v_j) \notin E$ since the edge from S_0 to v_j has input a_j at j and has output at port i if and only if $(v_i, v_j) \in E$. Let T_1 denote the set of transitions of $M(G, c)$ that are contained in synchronisable paths and so a transition from v_i with input a_k is in T_1 if and only if either $i = k$ or $(v_i, v_k) \in E$. Then the FSM $\chi_{min}(M(G, c))$ is equivalent to the FSM formed by removing from $M(G, c)$ all transitions not in T_1.

Proposition 18 *Given $M(G, c) = (S, s_0, X, Y, T)$, in which T_1 is the set of transitions contained in synchronisable paths that start at s_0, the FSM $\chi_{min}(M(G, c))$ is globally equivalent to (S, s_0, X, Y, T_1).*

Proof

First observe that a path of $M(G, C)$ is synchronised if and only if it only contains transitions from T_1. The result thus follows from Proposition 2 and 3. □

Pfleeger considers two approaches to minimising an incompletely specified FSM: completing the FSM or state splitting. Here we only investigate the process of completing $\chi_{min}(M)$ in order to produce a completely-specified and deterministic FSM with fewest states that is locally s-equivalent to M and we prove that this problem is NP-hard. The proof that using state splitting is NP-hard is similar.

Lemma 1 *Let us suppose that $M(G, c) = (S, s_0, X, Y, T)$ and T_1 is the set of transitions contained in synchronisable paths of $M(G, c)$ that start at s_0. If G is colourable with c colours then we can complete (S, s_0, X, Y, T_1) by adding transitions to produce a completely-specified deterministic FSM that is locally s-equivalent to $M(G, c)$ and has at most $c + 3$ equivalence classes of states.*

Proof

Let f be a colouring of G. First note that a transition from v_i with input a_k is in T_1 if and only if either $i = k$ or $(v_i, v_k) \in E$. We define a set T_1' of transitions by: all the transitions that have starting state S_0, S_N, or S_F are the same as those in T_1. In addition, for all v_i and a_k we have that:

(1) If there is some edge between v_i' and v_k in E for some v_i' such that $f(v_i) = f(v_i')$ then $(v_i, S_N, a_k/(0, \ldots, 0))$ is in T_1';
(2) otherwise $(v_i, S_F, a_k/(1, \ldots, 1))$ is in T_1'.

Clearly FSM $M'(G, c) = (S, s_0, X, Y, T_1')$ is deterministic and completely-specified. We now prove that $T_1 \subseteq T_1'$ and so $M'(G, c)$ can be produced from $M(G, c)$ by deleting the transitions that are in no synchronisable path and then adding transitions. First consider a transition $(v_i, S_N, a_k/(0, \ldots, 0)) \in T_1$. We must have that E contains an edge between v_i and v_k and so by definition, $(v_i, S_N, a_k/(0, \ldots, 0)) \in T_1'$. Now consider a transition $(v_i, S_F, a_k/(1, \ldots, 1)) \in T_1$ and so $i = k$. If we have v_i' such that $f(v_i) = f(v_i')$ and there is an edge in E between v_k and v_i' then we would contradict f being a colouring since this would imply that $f(v_k) = f(v_i')$. Thus, $(v_i, S_F, a_k/(1, \ldots, 1)) \in T_1'$ as required.

Consider $v_i, v_j \in V$ such that $f(v_i) = f(v_j)$ and the FSM $M'(G, c)$. There is a transition with starting state v_i, input a_k and ending state S_N if and only if $(v_i', v_k) \in E$ for some v_i' such that $f(v_i) = f(v_i')$. Similarly, there is a transition with starting state v_j, input a_k and ending state S_N if and only if $(v_j', v_k) \in E$ for some v_j' such that $f(v_j) = f(v_j')$. Thus, for all $v_i, v_j \in V$, if $f(v_i) = f(v_j)$ then we must have that for all a_k, there is a transition with starting state v_i,

input a_k and ending state S_N if and only if there is a transition with starting state v_j, input a_k and ending state S_N. Further, these transitions have the same output. As a result, for all v_i, v_j if $f(v_i) = f(v_j)$ then v_i and v_j are globally equivalent in $M'(G,c)$ and so $M'(G,c)$ has at most $c+3$ equivalence classes. Clearly, $M'(G,c)$ is locally s-equivalent to $M(G,c)$ because the transitions added to T_1 to form T_1' are not contained in any synchronisable paths and so the result follows. □

Lemma 2 *Let us suppose that $M(G,c) = (S, s_0, X, Y, T)$ and T_1 is the set of transitions contained in synchronisable paths that start at s_0. If (S, s_0, X, Y, T_1) may be completed so that the resulting reduced deterministic finite state machine has k states and is locally s-equivalent to $M(G,c)$ then G may be coloured with $k-3$ colours.*

Proof

Let $M'(G,c)$ denote a deterministic FSM that can be produced by completing (S, s_0, X, Y, T_1) and so is locally s-equivalent to $M(G,c)$. Clearly, each of the states S_0, S_F, S_N of $M'(G,c)$ are not compatible with each other or with any other state of $M'(G,c)$. Let $\{S_0\}$, $\{S_N\}$, $\{S_F\}$, $C_1, \ldots, C_{k-3}$ be the classes of states of $M'(G,c)$ that are combined in forming an FSM with k states.

We define a function f by: $f(v_i) = p$ if $v_i \in C_p$ and so it is sufficient to prove that f colours G. Let us suppose that E contains an edge between v_i and v_j. We can note that:

(1) $M(G,c)$ contains the edge $(S_0, v_i, a_i/\bar{y}(i))$ and this can be followed in a synchronisable path by the edge $(v_i, S_F, a_i/(1, \ldots, 1))$
(2) $M(G,c)$ contains the edge $(S_0, v_j, a_j/\bar{y}(j))$ that has output at port i and thus this can be followed in a synchronisable path by the edge $(v_j, S_N, a_i/(0, \ldots, 0))$.

Thus v_i and v_j lie in different C_l and so the result follows. □

Theorem 6 *The following problem is NP-complete. Given a completely specified and deterministic FSM M and $k > 0$ is it possible to complete $\chi_{min}(M)$ to produce a completely specified and deterministic FSM M' that is locally s-equivalent to M and that has at most k states?*

Proof

This follows from Proposition 18, Lemmas 1 and 2 and the fact that the graph colouring problem is NP-hard. □

This shows that the problem of producing a smallest FSM M' that is locally equivalent to M, by completing $\chi_{min}(M)$, is NP-hard. However, it is worth

noting that this is an instance of the problem of minimising an incompletely specified FSM for which heuristics have been developed (see, for example [10,22]).

7 Conclusions

A system under test (SUT) with multiple interfaces/ports can be tested in the distributed test architecture in which a tester is placed at each interface/port, these testers cannot directly communicate with one another and there is no global clock. It is known that the use of the distributed test architecture introduces limits in testing and recent work has characterised the effectiveness of testing a finite state machine (FSM) in the distributed test architecture in terms of local s-equivalence: it is possible to distinguish two FSMs in the distributed test architecture if and only if they are not locally s-equivalent [19]. Previous work has studied deterministic and completely-specified FSMs but for an FSM M we have considered s_M-deterministic FSMs, which are completely-specified and deterministic for each input sequence $\bar{x}$ that causes no controllability problems in M. This paper has explored the set of s_M-deterministic FSMs that are locally s-equivalent to a given deterministic and completely specified FSM M.

We have shown that it is possible to construct an FSM $\chi_{min}(M)$ that, amongst the FSMs that are locally s-equivalent to M, defines the smallest set of traces. Let us suppose that for an FSM M' we use $L(M')$ to denote the set of traces defined by M'. Then an s_M-deterministic FSM is locally s-equivalent to M if and only if $L(\chi_{min}(M)) \subseteq L(M')$. Thus, $\chi_{min}(M)$ defines the set of traces that *must be* included in an implementation in order for it to be locally s-equivalent to M. As a result, if we are building an implementation of M and this is to be placed in a context in which its use will correspond to the restrictions imposed by the distributed test architecture then $\chi_{min}(M)$ defines the set of behaviours that we have to implement.

As well as defining an FSM with a minimal language, we have defined an FSM $\chi_{max}(M)$ that, amongst the FSMs that are locally s-equivalent to M, has the largest language. An s_M-deterministic FSM M' is locally s-equivalent to M if and only if $L(M') \subseteq \chi_{max}(M)$. The FSM $\chi_{max}(M)$ thus defines the set of behaviours that can be contained in an SUT without it being possible to distinguish between the SUT and M in testing in the distributed test architecture. Thus $\chi_{max}(M)$ can be used to explore the consequences of the limitations introduced by using the distributed test architecture and thus potentially to inform the decision as to whether it is worth incurring the additional expense of introducing an external network through which the testers can communicate in order to overcome these problems (see, for example, [5,29] for a description

of such an external network).

Given an FSM M with multiple ports there is a set of locally s-equivalent FSMs. If we use set inclusion on the languages defined by the FSMs then we get a natural partial order between these FSMs. In this paper we proved that this defines a bounded lattice, with minimal element $L(\chi_{min}(M))$ and maximal element $L(\chi_{max}(M))$.

The definitions of $\chi_{min}(M)$ and $\chi_{max}(M)$ refer to the semantics of the FSMs and not the size of their representation. Let us suppose that we are developing a system and its use will correspond to the restrictions imposed by the distributed test architecture: only input sequences corresponding to synchronisable paths are applied and observations are made locally. Then we may want a smallest design that is locally s-equivalent to M: a deterministic and completely specified FSM M that has fewest states. The problem of producing such an FSM corresponds to minimising the incompletely specified FSM $\chi_{min}(M)$ and we have proved that in general this problem is NP-hard. However, we have also proved that the problem can be solved in polynomial time for the special case, often considered in the literature, in which there are two ports.

This paper has considered three alternative notions of a canonical FSM that is locally s-equivalent to M. The FSMs $\chi_{min}(M)$ and $\chi_{max}(M)$ can be constructed in time that is polynomial in terms of the number of states of M and a locally s-equivalent FSM with fewest states can be constructed in polynomial time if M has two ports. Recent work [14] has looked at the testing of distributed systems in which an operation can be triggered by the SUT receiving multiple events at different ports and it would be interesting to extend the work described in this paper to such a situation.

References

[1] M. Barnett, W. Grieskamp, L. Nachmanson, W. Schulte, N. Tillmann, and M. Veanes. Towards a tool environment for model-based testing with AsmL. In *Formal Approaches to Testing*, volume 2931 of *Lecture Notes in Computer Science*, pages 252–266, Montreal, Canada, 2003. Springer-Verlag.

[2] E. Bernard, F. Bouquet, A. Charbonnier, B. Legeard, F. Peureux, M. Utting, and E. Torreborre. Model-based testing from UML models. In *Informatik 2006 - Informatik für Menschen, Band 2, Beiträge der 36. Jahrestagung der Gesellschaft für Informatik e.V. (GI)*, volume 94 of *LNI*, pages 223–230, 2006.

[3] K. Bogdanov and M. Holcombe. Statechart testing method for aircraft control systems. *Journal of Software Testing, Verification and Reliability*, 11(1):39–54, 2001.

[4] S. Boyd and H. Ural. The synchronization problem in protocol testing and its complexity. *Information Processing Letters*, 40(3):131–136, 1991.

[5] L. Cacciari and O. Rafiq. Controllability and observability in distributed testing. *Information and Software Technology*, 41(11–12):767–780, 1999.

[6] J. Chen, R. M. Hierons, and H. Ural. Conditions for resolving observability problems in distributed testing. In *24rd IFIP International Conference on Formal Techniques for Networked and Distributed Systems (FORTE 2004)*, volume 3235 of *Lecture Notes in Computer Science*, pages 229–242. Springer-Verlag, 2004.

[7] W. Chen and H. Ural. Synchronizable checking sequences based on multiple UIO sequences. *IEEE/ACM Transactions on Networking*, 3:152–157, 1995.

[8] R. Dssouli and G. von Bochmann. Error detection with multiple observers. In *Protocol Specification, Testing and Verification V*, pages 483–494. Elsevier Science (North Holland), 1985.

[9] R. Dssouli and G. von Bochmann. Conformance testing with multiple observers. In *Protocol Specification, Testing and Verification VI*, pages 217–229. Elsevier Science (North Holland), 1986.

[10] S. Gören and F. J. Ferguson. On state reduction of incompletely specified finite state machines. *Computers & Electrical Engineering*, 33(1):58–69, 2007.

[11] W. Grieskamp. Multi-paradigmatic model-based testing. In *Formal Approaches to Software Testing and Runtime Verification, First Combined International Workshops, (FATES 2006 and RV 2006)*, volume 4262 of *Lecture Notes in Computer Science*, pages 1–19. Springer, 2006.

[12] W. Grieskamp, Y. Gurevich, W. Schulte, and M. Veanes. Generating finite state machines from abstract state machines. In *Proceedings of the ACM SIGSOFT Symposium on Software Testing and Analysis*, pages 112–122, 2002.

[13] S. Guyot and H. Ural. Synchronizable checking sequences based on UIO sequences. In *Protocol Test Systems, VIII*, pages 385–397, Evry, France, September 1995. Chapman and Hall.

[14] S. Haar, C. Jard, and G.-V. Jourdan. Testing input/output partial order automata. In *19th IFIP TC6/WG6.1 International Conference on The Testing of Software and Communicating Systems and the 7th International Workshop on Formal Approaches to Software Testing (TestCom/FATES 2007)*, volume 4581 of *Lecture Notes in Computer Science*, pages 171–185. Springer, 2007.

[15] D. Harel and M. Politi. *Modeling reactive systems with statecharts: the STATEMATE approach.* McGraw-Hill, New York, 1998.

[16] R. M. Hierons, K. Bogdanov, J. P. Bowen, R. Cleaveland, J. Derrick, J. Dick, M. Gheorghe, M. Harman, K. Kapoor, P. Krause, G. Lüttgen, A. J. H. Simons, S. A. Vilkomir, M. R. Woodward, and H. Zedan. Using formal specifications to support testing. *ACM Compututing Surveys*, 41(2), 2009.

[17] R. M. Hierons, T.-H. Kim, and H. Ural. On the testability of SDL specifications. *Computer Networks*, 44(5):681–700, 2004.

[18] R. M. Hierons and H. Ural. Synchronized checking sequences based on UIO sequences. *Information and Software Technology*, 45(12):793–803, 2003.

[19] R. M. Hierons and H. Ural. The effect of the distributed test architecture on the power of testing. *The Computer Journal*, 51(4):497–510, 2008.

[20] J. E. Hopcroft. An n log n algorithm for minimizing the states in a finite automaton. In Z. Kohavi, editor, *The theory of Machines and Computation*, pages 189–196. Academic Press, 1971.

[21] Joint Technical Committee ISO/IEC JTC 1. *International Standard ISO/IEC 9646-1. Information Technology - Open Systems Interconnection - Conformance testing methodology and framework - Part 1: General concepts*. ISO/IEC, 1994.

[22] T. Kam, T. Villa, R. K. Brayton, and A. L. Sangiovanni-Vincentelli. *Synthesis of Finite State Machines: Functional Optimization*. Kluwer Academic Press, London, 1996.

[23] R. M. Karp. Reducibility among combinatorial problems. In R. E. Miller and J. W. Thatcher, editors, *Complexity of Computer Computations*. Plenum Press, New York-London, 1972. 85–103.

[24] A. Khoumsi. A temporal approach for testing distributed systems. *IEEE Transactions on Software Engineering*, 28(11):1085–1103, 2002.

[25] D. Lee and M. Yannakakis. Principles and methods of testing finite-state machines - a survey. *Proceedings of the IEEE*, 84(8):1089–1123, 1996.

[26] G. Luo, R. Dssouli, and G. v. Bochmann. Generating synchronizable test sequences based on finite state machine with distributed ports. In *The 6th IFIP Workshop on Protocol Test Systems*, pages 139–153. Elsevier (North-Holland), 1993.

[27] G. Luo, R. Dssouli, G. v. Bochmann, P. Venkataram, and A. Ghedamsi. Test generation with respect to distributed interfaces. *Computer Standards and Interfaces*, 16:119–132, 1994.

[28] C. P. Pfleeger. State reduction in incompletely specified finite-state machines. *IEEE Transactions on Computers*, 22(12):1099–1102, 1973.

[29] O. Rafiq and L. Cacciari. Coordination algorithm for distributed testing. *The Journal of Supercomputing*, 24(2):203–211, 2003.

[30] H. Ural and Z. Wang. Synchronizable test sequence generation using UIO sequences. *Computer Communications*, 16(10):653–661, 1993.

[31] G. v. Bochmann, A. Petrenko, O. Bellal, and S. Maguiraga. Automating the process of test derivation from SDL specifications. In *SDL Forum'97*, Paris, France, 1997.