New state-recognition patterns for conformance testing of finite state machine implementations

Monika Kapus-Kolar

Jožef Stefan Institute, Department of Communication Systems, Jamova 39, SI-1111 Ljubljana, Slovenia

Abstract

Standards are of little value without conformance testing. Systematic testing relies on a formal model of the system under test. In black-box conformance testing of reactive systems, the system is often assumed to be an implementation of a given finite state machine and to possess no more states than the machine. The key activity in the interpretation of an input/output sequence observed (or, in test construction, planned to be observed) on the system is then to recognize the visited states as states of the specification machine. In the interpretation, one applies various state-recognition patterns (SRPs). The stronger the SRPs available, the shorter the test can be. In the paper, three traditional SRPs are generalized to two much stronger, but still relatively easy to apply SRPs. The SRPs are then generalized even further, to an extremely strong and general SRP interesting also as a template from which further practically interesting SRPs can be developed simply by specialization.

Keywords: Conformance testing, Deterministic finite state machine, State recognition.

1. Introduction

1.1. Preface

A standard is a document specifying the properties which an object must possess if it is to *conform* to the standard. Such a specification, however, is of little value if one does not know how to *test* whether an object conforms to it or not. Conformance testing is so central to the standardization process that it is itself subject of intensive standardization. The International Organization for Standardization has so far published 79 standards on the subject, of which as many as 66 have been issued by its Joint Technical Committee for Information Technology (IT). Particularly important is conformance testing in the field of communication protocols [1], for which the development of public services started already in the eighties [2], with utmost importance especially for safety-critical applications, e.g. railway signalling [3], and security-critical application, e.g. firewalls [4].

IT standards specify various kinds of hardware, software and hybrid objects. Regardless of whether such an object is self-standing or a part of another, one is actually interested only in its behaviour towards its environment, e.g. towards its users and peers. When testing an IT system or a part of it for conformance with a standard, it is, hence, appropriate to regard it as a *reactive system* [5] and desirable to know how to test it as a *black box*. In such testing, the topic of our paper, one assumes that the only way to affect the behaviour of the *system under test* (SUT) is to offer it signals from its *input alphabet* and that the only way to observe its behaviour is to observe which signals from its *output alphabet* it produces in response.

1.2. Reactive systems as DFSM implementations

In conformance testing, one assumes that the SUT should ideally be a specific machine M, the *specification machine* (SM), but is actually one of the machines from the adopted set of the *expected implementations* of M. The SM can be the *model* which the pertinent standard defines for the SUT, but more often, it is an *abstraction* of the model, for one does not want it to be too complex. In any case, it is desirable that the SM and its expected implementations are given *formally*, for only then the testing process can profit from advanced formal methods and tools [6].

In the paper, we adopt the very common assumption that the SM and its implementations are *deterministic finite state machines* (DFSMs), all defined on the same input alphabet and the same output alphabet.

Email address: monika.kapus-kolar@ijs.si (Monika Kapus-Kolar)

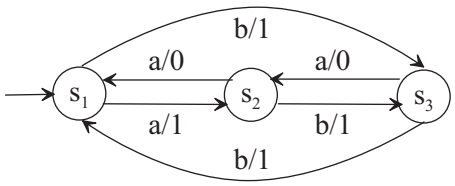

Figure 1: An example DFSM

A DFSM is a machine possessing a finite number of *states* in which it might reside, among them its *initial state*. Upon receiving a specific input when in a specific state, it executes the corresponding *transition*, i.e., generates the corresponding *output* and then enters the corresponding *next state* (possibly the same as the state before the transition). The example DFSM [7] in Fig. 1 has states s_1, s_2 and s_3, with s_1 the initial state, accepts inputs a and b and responds with outputs 0 and 1.

When a specific sequence of inputs is applied to a DFSM in a specific state, the DFSM responds with the corresponding sequence of outputs and thereby executes the corresponding *input/output sequence* (IOS). For example, when the input sequence *aba* is applied to the state s_2 of the DFSM in Fig. 1, the DFSM executes the IOS $a/0b/1a/0$. The IOSs which a DFSM is able to execute from a specific state constitute the *language* of the state.

1.3. Checking sequences

IOSs are the central concept of black-box testing. One provokes and observes them on the SUT and then tries to *interpret their observation as a proof that the SUT is non-faulty*, i.e., that its initial state has the same language as the initial state of the SM. If for a set of IOSs, observation of its members on the SUT allows construction of such a proof, the set is a *complete test suite* (CTS). The members of a CTS are by definition IOSs executable on the SM and supposed to be executable *from the initial state* of the SUT. One favours CTSs which are a small set of short IOSs. Of particular interest are CTSs comprising a single IOS, a so called *checking sequence* (CS), for they are applicable also if the SUT does not possess the reliable reset capability or its reset into the initial state is an undesirable (e.g. time consuming) operation.

The difficulty of CS construction highly depends on the nature of the SM and its expected implementations. One, hence, typically makes the following assumptions, without which the existence of a short CS is highly improbable:

1. The SM possesses a *distinguishing set* (DS), i.e., a set which for every state s of the SM comprises such an IOS $\bar{z}_s$ in the language of s, the response to an input sequence $\bar{x}_s$, that for every two different states s and s' of the SM, $\bar{x}_s$ and $\bar{x}_{s'}$ have a common prefix to which SM in the two states responds with two different output sequences. For example, the SM in Fig. 1 has a DS $\{a/1, a/0a/1, a/0a/0\}$.
2. For every two different states s and s' of the SM, there is an input sequence leading the SM from s to s'.
3. The SUT has at most as many states as the SM.

Under the assumptions, adopted also for the rest of the paper, *the SUT is non-faulty exactly if it is an instance of the SM*. Moreover, for such an SUT, the construction of a correctness proof from the IOS observed as a proper response to a CS amounts to *recognizing the states which the SUT visits during the IOS execution as states of the SM and thereby finding an instance of execution for every SM transition*. As an example, Fig. 2 shows an increasingly finer response interpretation for a CS which we constructed [8] for the SM in Fig. 1. In the figure, for each point of the CS execution, an inserted set lists the indices of the SM states not yet eliminated as the possible current state of the SUT. In the last line, all the sets are singleton, so that one can see the initial state and the final state of every transition executed, the transitions including every transition of the SUT.

1.4. Contributions and organization of the paper

In CS construction, an interpretation plan for the corresponding IOS is virtually conceived in parallel with the CS. The key decision in CS construction is, hence, on which *state-recognition patterns* (SRPs) to rely for the interpretation. The traditional methods supporting (also) DS-based CS construction [9–23] rely for it on just four SRPs. We present them in Section 3, as SRPs 1 to 4, after in Section 2 introducing the employed notation and definitions. In Section 4, we then propose four new SRPs, SRPs 5 to 8, of which particularly SRP 8, a generalization of the SRPs 2-7, is extremely strong. With stronger SRPs, more input sequences can be recognized as CSs, meaning that with such SRPs, one can potentially construct shorter CSs. In Section 5, we identify two possibilities for employing additional SRPs, of any kind, in the existing CS-construction methods. Section 6 comprises a discussion and conclusions.

The most innovative concept in the paper is that of a *distinguisher*. The author first tried to publish it in a wider paper submitted for review in April 2009. Unfortunately, the paper was rejected as too formal for the typical reader, and subsequently archived as a technical report [24]. In the paper, distinguishers were de-

(1) $a/1b/1b/1a/1b/1a/0a/0a/1a/0b/1a/0a/0$
(2) $\{1\}a/1\{1,2,3\}b/1\{1,2,3\}b/1\{1\}a/1\{1,2,3\}b/1\{3\}a/0\{2\}a/0\{1\}a/1\{1,2,3\}a/0\{1,2,3\}b/1\{3\}a/0\{1,2,3\}a/0\{1,2,3\}$
(3) $\{1\}a/1\{1,2,3\}b/1\{1,2,3\}b/1\{1\}a/1\{1,2,3\}b/1\{3\}a/0\{2\}a/0\{1\}a/1\{2,3\}a/0\{1,2,3\}b/1\{3\}a/0\{2,3\}a/0\{1,2,3\}$
(4) $\{1\}a/1\{1,2,3\}b/1\{3\}b/1\{1\}a/1\{1,2,3\}b/1\{3\}a/0\{2\}a/0\{1\}a/1\{2,3\}a/0\{1,2,3\}b/1\{3\}a/0\{2,3\}a/0\{1,2,3\}$
(5) $\{1\}a/1\{2,3\}b/1\{3\}b/1\{1\}a/1\{2,3\}b/1\{3\}a/0\{2\}a/0\{1\}a/1\{2,3\}a/0\{1,2,3\}b/1\{3\}a/0\{2\}a/0\{1,2,3\}$
(6) $\{1\}a/1\{2\}b/1\{3\}b/1\{1\}a/1\{2\}b/1\{3\}a/0\{2\}a/0\{1\}a/1\{2,3\}a/0\{1,2\}b/1\{3\}a/0\{2\}a/0\{1,2,3\}$
(7) $\{1\}a/1\{2\}b/1\{3\}b/1\{1\}a/1\{2\}b/1\{3\}a/0\{2\}a/0\{1\}a/1\{2\}a/0\{1,2\}b/1\{3\}a/0\{2\}a/0\{1\}$
(8) $\{1\}a/1\{2\}b/1\{3\}b/1\{1\}a/1\{2\}b/1\{3\}a/0\{2\}a/0\{1\}a/1\{2\}a/0\{1\}b/1\{3\}a/0\{2\}a/0\{1\}$

Figure 2: An increasingly finer interpretation of the IOS corresponding to the CS *abbabaaaabaa* of the SM in Fig. 1

fined for multi-port systems with no coordination between the testers controlling and observing individual ports, such as the systems which Hierons targets in a DS-based method for constructing synchronizable CSs [25]. Along with the generalized distinguishers, the paper proposed an SRP semantically equivalent to a slight specialization of SRP 8 generalized to multi-port systems and context-dependent state recognition. While the paper was in review, distinguishers of the restricted kind defined below in Section 4.1 and SRP 5 were independently discovered also by Dincturk [26] (more on his work in Section 6). As he didn't publish the results either, the author was first made aware of his M.Sc. thesis in October 2010, after inventing the SRP 8 in its present form and reinventing SRP 5 as its specialization.

2. Notation and definitions

A transition of a DFSM is an $(s, x/y, s')$ with s its initial state, x the applied input, y the executed output and s' its final state, in the following also called $\delta(s, x)$. A *transition sequence* (TS) of a DFSM is a sequence $(s_1, x_1/y_1, s_2) \dots (s_m, x_m/y_m, s_{m+1})$ of its consecutive transitions. For such a sequence $\bar{\tau}$, let $st(\bar{\tau})$ denote its state sequence $s_1 \dots s_{m+1}$, $in(\bar{\tau})$ its input sequence $x_1 \dots x_m$, $out(\bar{\tau})$ its output sequence $y_1 \dots y_m$, $ios(\bar{\tau})$ its IOS $x_1/y_1 \dots x_m/y_m$, $init(\bar{\tau})$ its initial state s_1 and $fin(\bar{\tau})$ its final state s_{m+1}.

If for a DFSM M and an IOS $\bar{z}$ which it can execute, $init(\bar{\tau})$ is for every TS $\bar{\tau}$ of M with $ios(\bar{\tau}) = \bar{z}$ the same state, an s, $\bar{z}$ is in M a *unique IOS* (UIO), of s. If for a DFSM M and an IOS $\bar{z}$ which it can execute, $fin(\bar{\tau})$ is for every TS $\bar{\tau}$ of M with $ios(\bar{\tau}) = \bar{z}$ the same state, an s, $\bar{z}$ is in M a *backward UIO* (BUIO), of s.

In the following, let $\{s_1, \dots, s_n\}$ denote the state set of the SM, with s_1 the initial state, and N the state index set $\{1, \dots, n\}$. An *interpreted IOS* (IIOS), such as those in Fig. 2, is then a sequence $I_1 x_1/y_1 I_2 x_2/y_2 \dots I_m x_m/y_m I_{m+1}$ with $x_1/y_1 \dots x_m/y_m$ a non-empty IOS (the interpreted IOS) and with $I_i \subseteq N$ for every $1 \leq i \leq m+1$. For such an IIOS ω, let $in(\omega)$ denote the input sequence $x_1 \dots x_m$, $out(\omega)$ the output sequence $y_1 \dots y_m$, $ios(\omega)$ the IOS $x_1/y_1 \dots x_m/y_m$, $init(\omega)$ the state set $\{s_i | i \in I_1\}$, $fin(\omega)$ the state set $\{s_i | i \in I_{m+1}\}$ and $st(\omega)$ the set of all state sequences $s_{i_1} \dots s_{i_{m+1}}$ with $i_j \in I_j$ for every $1 \leq j \leq m+1$.

We say that a TS $\bar{\tau}$ *matches* an IIOS ω, denoted as $\mu(\bar{\tau}, \omega)$, if $(ios(\bar{\tau}) = ios(\omega)) \wedge (st(\bar{\tau}) \in st(\omega))$. For an IIOS ω, let $\mu(\omega)$ denote that the SUT has a TS $\bar{\tau}$ with $\mu(\bar{\tau}, \omega)$.

3. The traditional state-recognition patterns

In DS-based CS construction, one starts by adopting a DS D of the SM as the *primary state recognizer*, i.e., by deciding that an SUT state s will be recognized as a specific SM state s_i, with the corresponding IOS in D in the following called d_i, only if D is a DS also of the SUT (so that one knows that the SUT has exactly n states) and the SUT can execute d_i from s. This is below formalized in SRP 1, with which one can establish the required one-to-one correspondence between the SM states and the SUT states. The SRP, in combination with the SRP 2, explains how in Fig. 2, with $D = \{a/1, a/0a/1, a/0a/0\}$, line (2) was devised from line (1).

SRP 1. *If the SUT for every i in N has a TS $\bar{\tau}_i$ with $ios(\bar{\tau}_i) = d_i$, then*

1) for every i in N, with d_i an $x_1^i/y_1^i \dots x_{m_i}^i/y_{m_i}^i$, $\mu(\{i\}$ $x_1^i/y_1^i N \dots N x_{m_i}^i/y_{m_i}^i N)$, and

2) for every TS $\bar{\tau}$ of the SUT, with $ios(\bar{\tau})$ an $x_1/y_1 \dots x_m/y_m$, $\mu(\bar{\tau}, N x_1/y_1 N \dots N x_m/y_m N)$.

The remaining traditional SRPs are intended for recognizing the initial state or the final state of an SUT transition as a specific previously recognized SUT state. The second SRP formalizes recognition of a state by an outgoing IOS. Informally, it says that if in a specific

state s_i, the SUT once responded to a specific input sequence $\bar{x}$ with a specific output sequence $\bar{o}$, then whenever the SUT responds to $\bar{x}$ with an output sequence different from $\bar{o}$, one knows that the state just before the application of $\bar{x}$ was not s_i. The SRP explains, for example, how in Fig. 2, line (3) was devised from line (2), after in line (2) observing that in s_1, the SUT once responded to a with 1.

SRP 2. *If*

1) $\mu(\bar{\tau}, \omega)$ *for a TS* $\bar{\tau}$ *of the SUT and an IIOS* $\omega = I_1x_1/y_1I_2 \dots I_mx_m/y_mI_{m+1}$ *and*

2) $\mu(\omega_1)$ *for an IIOS* $\omega_1 = \{i\}x_1/y'_1I'_2 \dots I'_mx_m/y'_mI'_{m+1}$ *with* $y'_1 \dots y'_m \neq y_1 \dots y_m$,

then $\mu(\bar{\tau}, (I_1 \setminus \{i\})x_1/y_1I_2 \dots I_mx_m/y_mI_{m+1})$.

The third SRP formalizes recognition of a state by an incoming IOS and its source state. Informally, it says that if from a specific state s_i, a specific input sequence $\bar{x}$ once took the SUT to a state in a set S, then whenever the SUT executes $\bar{x}$ from s_i, the resulting state is in S. The SRP explains, for example, how in Fig. 2, line (4) was devised from line (3), after in line (3) observing that from s_1, ab once took the SUT to s_3.

SRP 3. *If*

1) $\mu(\bar{\tau}, \omega)$ *for a TS* $\bar{\tau}$ *of the SUT and an IIOS* $\omega = \{i\}x_1/y_1I_2 \dots I_mx_m/y_mI_{m+1}$ *and*

2) $\mu(\omega_1)$ *for an* $\omega_1 = \{i\}x_1/y_1I'_2 \dots I'_mx_m/y_mI'_{m+1}$,

then $\mu(\bar{\tau}, \{i\}x_1/y_1I_2 \dots I_mx_m/y_m(I_{m+1} \cap I'_{m+1}))$.

The last SRP, employed in [16, 22], formalizes recognition of a state by an outgoing IOS and its destination state. Informally, it says that if from a specific state s_i, a specific input sequence $\bar{x}$ once took the SUT to a state in a set S, then whenever the SUT executes $\bar{x}$ and thereby enters a state outside S, one knows that the state just before the application of $\bar{x}$ was not s_i. The SRP explains, for example, how in Fig. 2, line (6) was devised from line (5), after in line (5) observing that from s_3, b once took the SUT to s_1.

SRP 4. *If*

1) $\mu(\bar{\tau}, \omega)$ *for a TS* $\bar{\tau}$ *of the SUT and an IIOS* $\omega = I_1x_1/y_1I_2 \dots I_mx_m/y_mI_{m+1}$ *and*

2) $\mu(\omega_1)$ *for an* $\omega_1 = \{i\}x_1/y'_1I'_2 \dots I'_mx_m/y'_mI'_{m+1}$ *with* $I'_{m+1} \cap I_{m+1} = \emptyset$,

then $\mu(\bar{\tau}, (I_1 \setminus \{i\})x_1/y_1I_2 \dots I_mx_m/y_mI_{m+1})$.

4. Four new state-recognition patterns

4.1. Distinguishers

For two IIOS ω_1 and ω_2, let $dif(\omega_1, \omega_2)$ denote that there exist two IIOSs ω'_1 and ω'_2 with ω'_1 a prefix of ω_1, ω'_2 a prefix of ω_2, $in(\omega'_1) = in(\omega'_2)$ and $(out(\omega'_1) \neq out(\omega'_2)) \vee (fin(\omega'_1) \cap fin(\omega'_2) = \emptyset)$. The convenience of a $dif(\omega, \omega')$ is that it for every two TSs $\bar{\tau}$ and $\bar{\tau}'$ of the SUT with $\mu(\bar{\tau}, \omega)$ and $\mu(\bar{\tau}', \omega')$ implies $init(\bar{\tau}) \neq init(\bar{\tau}')$ and, hence, $\bar{\tau} \neq \bar{\tau}'$.

The SRPs 2-4 are all of the general form $(\mu(\bar{\tau}, \omega) \wedge (\wedge_{1 \leq i \leq k} \mu(\omega_i))) \Rightarrow \mu(\bar{\tau}, \omega')$ with $ios(\omega') = ios(\omega)$ and $st(\omega') \subseteq st(\omega)$, but restrict k to 1 and $init(\omega_1)$ to singleton sets. The second restriction means that *one cannot profit from an observed IOS unless the state from which it has been executed is precisely recognized.* As a motivation for developing SRPs which work without the restrictions and are, hence, potentially stronger, we give two examples of interpretation which follows the same template, but with $k = 3$ and $init(\omega_1)$, $init(\omega_2)$ and $init(\omega_3)$ not singleton.

Example 1. If $\mu(\bar{\tau}, \omega)$, $\mu(\omega_1)$, $\mu(\omega_2)$ and $\mu(\omega_3)$ for a TS $\bar{\tau}$ of the SUT and the IIOSs
$\omega = \{1, 2, 3, 4\}a/0\{1, 2, 3, 4\}b/2\{1, 2, 3, 4\}$,
$\omega_1 = \{1, 2, 3\}a/0\{1, 2, 3, 4\}b/0\{1, 2\}$,
$\omega_2 = \{1, 2, 3\}a/0\{1, 2, 3, 4\}b/0\{4\}$ and
$\omega_3 = \{1, 2, 3\}a/0\{1, 2, 3, 4\}b/1\{4\}$,
then $\mu(\bar{\tau}, \{4\}a/0\{1, 2, 3, 4\}b/2\{1, 2, 3, 4\})$. □

Example 2. Take the same ω_1, ω_2 and ω_3 as in Example 1. If $\mu(\omega_i)$ for $1 \leq i \leq 3$ and $\mu(\bar{\tau}, \omega)$ for a TS $\bar{\tau}$ of the SUT and the IIOS
$\omega = \{1, 2, 3\}a/0\{1, 2, 3, 4\}b/1\{1, 2, 3, 4\}$,
then $\mu(\bar{\tau}, \{1, 2, 3\}a/0\{1, 2, 3, 4\}b/1\{4\})$. □

In both examples, the implication results from the fact that the IIOSs ω_1, ω_2 and ω_3 match three different TSs $\bar{\tau}'$ of the SUT with $init(\bar{\tau}')$ in the set $S = \{s_1, s_2, s_3\}$ and $in(\bar{\tau}') = ab$, meaning that the SUT has no other such TS and that each of the TSs starts in an *unknown, but different* state in S. So if $in(\bar{\tau}) = ab$ and $out(\bar{\tau}) = 02$, $init(\bar{\tau})$ is not in S (Example 1). Besides, if $init(\bar{\tau}) \in S$ and $in(\bar{\tau}) = ab$, $\bar{\tau}$ is one of the three TSs, with $out(\bar{\tau}) = 01$ implying that it is the one matching ω_3, so that $fin(\bar{\tau}) = s_4$ (Example 2).

The set $\{\omega_1, \omega_2, \omega_3\}$ is what we call a *distinguisher*, which we define as a non-empty IIOS set Δ satisfying the following:

1) The set $\cup_{\omega \in \Delta} init(\omega)$, call it $init(\Delta)$, is of the size $|\Delta|$.

2) For every ω in Δ, $dif(\omega, \omega')$ for every other IIOS ω' in Δ.

Lemma 1. *If $\mu(\omega)$ for every ω in a distinguisher Δ, the SUT has for every ω in Δ exactly one TS $\bar{\tau}$ with $\mu(\bar{\tau}, \omega)$ and among the TSs, there is for every state in $init(\Delta)$ exactly one starting in the state.*

Proof. As $\mu(\omega)$ for every ω in Δ, one can for every ω in Δ select a TS $\bar{\tau}_\omega$ with $\mu(\bar{\tau}_\omega, \omega)$. For every ω in Δ, $\mu(\bar{\tau}_\omega, \omega)$ implies $init(\bar{\tau}_\omega) \in init(\omega)$. For every two different ω and ω' in Δ, we have, by $dif(\omega, \omega')$, $init(\bar{\tau}_\omega) \neq init(\bar{\tau}_{\omega'})$, implying $|\{init(\bar{\tau}_\omega) | \omega \in \Delta\}| = |\Delta|$. With $\{init(\bar{\tau}_\omega) | \omega \in \Delta\} \subseteq init(\Delta)$ and $|init(\Delta)| = |\Delta|$, this implies $\{init(\bar{\tau}_\omega) | \omega \in \Delta\} = init(\Delta)$, which implies that for every ω in Δ, exactly one state qualifies for $init(\bar{\tau}_\omega)$, and for every state s in $init(\Delta)$, Δ comprises an ω with $init(\bar{\tau}_\omega) = s$. So if for an ω in Δ also $\mu(\bar{\tau}, \omega)$, $init(\bar{\tau}) = init(\bar{\tau}_\omega)$ and, by $in(\bar{\tau}) = in(\bar{\tau}_\omega)$, $\bar{\tau} = \bar{\tau}_s$. Besides, if for an s in $init(\Delta)$, with $init(\bar{\tau}_\omega) = s$ for an ω in Δ, also $init(\bar{\tau}_{\omega'}) = s$ for an ω' in Δ, then $\omega = \omega'$. □

4.2. Interpretation With General Distinguishers

The SRPs 2-4 rely on the distinguisher $\{\omega_1\}$. The below defined SRPs 5 and 6 generalize the SRPs 2 and 4 and the SRP 3, respectively, to general distinguishers. They are, respectively, the SRPs on which the Examples 1 and 2 rely.

SRP 5. *If*

1) $\mu(\bar{\tau}, \omega)$ for a TS $\bar{\tau}$ of the SUT and an IIOS $\omega = I_1x_1/y_1I_2 \dots I_mx_m/y_mI_{m+1}$ and

2) $\mu(\omega')$ for every ω' in a distinguisher Δ with $dif(\omega, \omega')$ for every $\omega' \in \Delta$,

then $\mu(\bar{\tau}, (I_1 \setminus \{i | s_i \in init(\Delta)\})x_1/y_1I_2 \dots I_mx_m/y_m\ I_{m+1})$.

SRP 6. *If*

1) $\mu(\bar{\tau}, \omega)$ for a TS $\bar{\tau}$ of the SUT and an IIOS $\omega = I_1x_1/y_1I_2 \dots I_mx_m/y_mI_{m+1}$ and

2) $\mu(\omega')$ for every ω' in a distinguisher Δ with $init(\omega) \subseteq init(\Delta)$,

then $\mu(\bar{\tau}, I_1x_1/y_1I_2 \dots I_mx_m/y_m(I_{m+1} \cap I))$ with $I = \cup_{(\omega' \in \Delta) \wedge \neg dif(\omega, \omega')} I_{\omega'}$ with $I_{\omega'}$ for ω' in Δ defined as I'_{m+1} if ω' is an $I'_1x_1/y_1I'_2 \dots I'_mx_m/y_mI'_{m+1} \dots$ and as N otherwise.

The SRPs 5 and 6 are reasonably simple to apply, because when they on the basis of a distinguisher refine a $\mu(\bar{\tau}, \omega)$ into a $\mu(\bar{\tau}, \omega')$, they let ω' differ from ω only in the first or the last element, respectively. The following example indicates that interpretation can go beyond that, even without distinguishers.

Example 3. For a TS $\bar{\tau}$ of the SUT, $\mu(\bar{\tau}, \{1\}a/0\{1, 2, 3\}\ a/0\{2\})$ implies $\mu(\bar{\tau}, \{1\}a/0\{2, 3\}a/0\{2\})$ simply because $\bar{\tau} = (s_1, a/0, s_1)(s_1, a/0, s_2)$ would mean that the SUT is in s_1 able to react to a in more than one way, which is, by the assumption that it is deterministic, not the case. □

The reasoning in the example is an instance of the following SRP:

SRP 7. *If*

1) $\mu(\bar{\tau}, \omega)$ for a TS $\bar{\tau}$ of the SUT and an IIOS $\omega = I_1x_1/y_1I_2 \dots I_mx_m/y_mI_{m+1}$ and

2) for a $1 \leq p \leq m + 1$ and a $q \in I_p$, there are for every state index sequence $i_1 \dots i_{m+1}$ in $st(\omega)$ with $i_p = q$ some $1 \leq j < k \leq m$ with $((i_j, x_j) = (i_k, x_k)) \wedge ((y_j, i_{j+1}) \neq (y_k, i_{k+1}))$,

then $\mu(\bar{\tau}, I_1x_1/y_1I_2 \dots (I_p \setminus \{q\}) \dots I_mx_m/y_mI_{m+1})$.

The next example indicates the need for an even stronger SRP.

Example 4. If $\mu(\bar{\tau}, \omega)$, $\mu(\omega_1)$ and $\mu(\omega_2)$ for a TS $\bar{\tau}$ of the SUT and the IIOSs
$\omega = \{1\}a/0\{2, 3\}a/0\{2, 3\}a/0\{1, 2, 3\}$,
$\omega_1 = \{2\}a/0\{1, 2\}$ and
$\omega_2 = \{3\}a/0\{1, 3\}$,
one can, as $\{\omega_1\}$ and $\{\omega_2\}$ are distinguishers, deduce $\mu(\bar{\tau}, \omega')$ for the IIOS
$\omega' = \{1\}a/0\{2, 3\}a/0\{2, 3\}a/0\{2, 3\}$,
but only with the following reasoning beyond the SRPs 2-7: If $fin(\bar{\tau}) = s_1$, $st(\bar{\tau}) \in st(\omega)$ implies that $st(\bar{\tau})$ is $s_1s_2s_2s_1$ or $s_1s_2s_3s_1$ or $s_1s_3s_2s_1$ or $s_1s_3s_3s_1$. But it cannot be $s_1s_2s_2s_1$ or $s_1s_3s_3s_1$, for this would imply that the SUT is not deterministic, and it cannot be $s_1s_2s_3s_1$, for this would contradict $\mu(\omega_1)$, and it cannot be $s_1s_3s_2s_1$, for this would contradict $\mu(\omega_2)$. $fin(\bar{\tau}) = s_1$ is, hence, impossible. □

The reasoning in the example is an instance of the below defined SRP 8, which allows *synergetic exploitation of the SUT determinism and any number of distinguishers and simultaneous reduction of multiple elements of the IIOS to which the considered TS is matched*. To understand the ideas behind the SRP, please, read its proof, which is also a proof of its *specializations* SRPs 2-7.

SRP 8. *For a TS $\bar{\tau}$ of the SUT, an IIOS ω with $ios(\omega)$ an $x_1/y_1 \dots x_m/y_m$ and an IIOS ω' with $ios(\omega') = ios(\omega)$ and $st(\omega') \subset st(\omega)$, $\mu(\bar{\tau}, \omega)$ implies $\mu(\bar{\tau}, \omega')$ provided that for every state sequence $s_{i_1} \dots s_{i_{m+1}}$ in $st(\omega) \setminus st(\omega')$,*

1) there are some $1 \leq j < k \leq m$ *with* $((i_j, x_j) = (i_k, x_k)) \wedge ((y_j, i_{j+1}) \neq (y_k, i_{k+1}))$ *or*

2) for a $1 \leq j \leq m$ *and a distinguisher* Δ *with* $s_{i_j} \in init(\Delta)$, $\mu(\omega'') \wedge dif(\{i_j\}x_j/y_j\{i_{j+1}\}\ldots\{i_m\}x_m/y_m\{i_{m+1}\}, \omega'')$ *for every* ω'' *in* Δ.

Proof. If $\neg\mu(\bar{\tau}, \omega')$, $\mu(\bar{\tau}, \omega)$ implies that $st(\bar{\tau})$ is a sequence $s_{i_1} \ldots s_{i_{m+1}}$ in $st(\omega) \setminus st(\omega')$ and, hence, satisfies one of the two conditions stated in the SRP.

1) If $st(\bar{\tau})$ satisfies the first condition, $\bar{\tau}$ is a TS contradicting the assumption that the SUT is deterministic.

2) If $st(\bar{\tau})$ satisfies the second condition, $s_{i_j} \in init(\Delta)$ by Lemma 1 implies that for an $\omega'' \in \Delta$, the SUT has a TS $\bar{\tau}_1$ with $\mu(\bar{\tau}_1, \omega'')$ and $init(\bar{\tau}_1) = s_{i_j}$. On the other hand, $(ios(\bar{\tau}) = x_1/y_1 \ldots x_m/y_m) \wedge (st(\bar{\tau}) = s_{i_1} \ldots s_{i_{m+1}})$ implies that the SUT has a TS $\bar{\tau}_2$ with $init(\bar{\tau}_2) = s_{i_j}$ and $\mu(\bar{\tau}_2, \omega''')$ for the IIOS $\omega''' = \{i_j\}x_j/y_j\{i_{j+1}\}\ldots\{i_m\}x_m/y_m\{i_{m+1}\}$. We, hence, have $init(\bar{\tau}_1) = init(\bar{\tau}_2)$, but this by $\mu(\bar{\tau}_1, \omega'') \wedge \mu(\bar{\tau}_2, \omega''')$ contradicts the assumed $dif(\omega''', \omega'')$. □

In the proof, we recognized the two conditions as two *alternative reasons for dismissing a candidate state sequence*. A simple way for *further generalization* of the SRP is, hence, to define *additional such reasons*.

5. Two possibilities for employing additional SRPs in the existing CS-construction methods

If Fig. 2, we interpreted an IOS corresponding to an entire CS. In the construction of such an IOS, one gradually constructs, interprets and finally connects its special-purpose segments, in the following called *tests*.

Example 5. For the SM in Fig. 1, we repeat our systematic construction [8] of the IOS interpreted in Fig. 2. First we choose $d_1 = a/1$, $d_2 = a/0a/1$ and $d_3 = a/0a/0$. We adopt the IOSs as tests and interpret them as follows, concluding that the states of the SUT are s_1, s_2 and s_3:

$\{1\}a/1\{1,2,3\}$
$\{2\}a/0\{1\}a/1\{1,2,3\}$
$\{3\}a/0\{2,3\}a/0\{1,2,3\}$

Hence, the testing of $\delta(s_2, a) = s_1$ in the SUT is secured. We introduce the test $a/0a/0a/1$ for $\delta(s_3, a) = s_2$ in the SUT, because one can then delete its subtests $a/0a/0$ and $a/0a/1$. The test $a/1$ is preserved, as the only one which is in the SM an UIO of s_1 and, hence, appropriate for recognizing the initial state of the SUT. Our tests, interpreted, are then:

$\{1\}a/1\{1,2,3\}$
$\{3\}a/0\{2\}a/0\{1\}a/1\{1,2,3\}$

We introduce the test $a/1b/1a/0a/0$ for verifying that $a/1b/1$ is in the SUT a BUIO of s_3, because then one can easily construct a test for $\delta(s_3, b) = s_1$ in the SUT, which in turn makes $a/1b/1a/0a/0$ a test for $\delta(s_1, a) \neq s_3$ in the SUT. Consequently, the test $a/1$ can be deleted. Our tests, interpreted, are then:

$\{3\}a/0\{2\}a/0\{1\}a/1\{1,2,3\}$
$\{1\}a/1b/1\{3\}a/0\{2\}a/0\{1\}$

We introduce the test $a/1b/1b/1a/1$ for $\delta(s_3, b) = s_1$ in the SUT. Our tests, interpreted, are then:

$\{3\}a/0\{2\}a/0\{1\}a/1\{1,2,3\}$
$\{1\}a/1\{1,2\}b/1\{3\}a/0\{2\}a/0\{1\}$
$\{1\}a/1\{1,2\}b/1\{3\}b/1\{1\}a/1\{1,2,3\}$

We introduce the test $a/1a/0$ for $\delta(s_1, a) \neq s_1$ in the SUT, thereby completing the testing of $\delta(s_1, a) = s_2$ in the SUT. Our tests, interpreted, are then:

$\{3\}a/0\{2\}a/0\{1\}a/1\{2\}$
$\{1\}a/1\{2\}b/1\{3\}a/0\{2\}a/0\{1\}$
$\{1\}a/1\{2\}b/1\{3\}b/1\{1\}a/1\{2\}$
$\{1\}a/1\{2\}a/0\{1\}$

Hence, the testing of $\delta(s_2, b) = s_3$ in the SUT is secured. Besides, it is safe to assume that $a/1a/0$ is in the SUT a BUIO of s_1. So we introduce $a/1a/0b/1a/0a/0$ as a test for $\delta(s_1, b) = s_3$ in the SUT, consequently deleting the subtest $a/1a/0$. Our tests, interpreted, are then:

$\{3\}a/0\{2\}a/0\{1\}a/1\{2\}$
$\{1\}a/1\{2\}b/1\{3\}a/0\{2\}a/0\{1\}$
$\{1\}a/1\{2\}b/1\{3\}b/1\{1\}a/1\{2\}$
$\{1\}a/1\{2\}a/0\{1\}b/1\{3\}a/0\{2\}a/0\{1\}$

It remains to find a short IOS which is in the SM an UIO of s_1 and comprises each of the tests as a segment. One of such IOSs is the one interpreted in Fig. 2. □

In the example, only one sufficient set of tests was constructed. More advanced CS-construction methods [11–19, 21, 22] construct multiple sets and then use global optimization to decide which of them to employ and how to pack its members into an IOS which the SM can execute from s_1. Although in the methods, test set construction is mainly implicit and strongly entangled with the activity of test selection and packing, it is not impossible to see what the considered candidate test sets are [8]. They should better be constructed and checked through interpretation explicitly, because in some cases, they are not just non-optimal, but also insufficient [8, 27], meaning that the constructed input sequence might fail to be a CS. Through disentangling test set construction and test selection and packing, one also opens possibilities for further optimization of the latter [8].

Once candidate test set construction is made explicit, there are at least two possibilities for employing additional SRPs, of any kind. The first is for *answering the question whether a specific test set covers a specific test goal*, either a direct one, such as e.g. verification of a specific transition, or an indirect one, such as e.g. verification that a specific IOS is an UIO or a BUIO of the SUT and, hence, able to play the role in a subsequently constructed test. To a limited extent, the considered CS-construction methods already pose such questions explicitly. For example, Chen and Tekle [15, 16] ask whether the tests primarily introduced for DS and BUIO verification verify also a specific set of transitions. Actually, asking such questions explicitly is recently becoming a trend [8, 24, 26, 28], although so far predominantly in methods without global optimization [20, 23, 26].

Another possibility for employing additional SRPs is in the *construction of tests covering specific test goals not yet covered by a specific candidate test set under construction*. One could, for example, check whether a test proposed by an existing method really has to be introduced in its full length. There is plenty of space also for generalizing the *template* to which the current (even the more advanced [15, 16, 18, 19, 22]) CS-construction methods stick in the construction or recognition of transition tests [8]. Every such template must be proven to cover the target goal and this is where additional SRPs can help.

6. Discussion and conclusions

When interpreting IOSs observed or (in test construction) considered for observation on a DFSM implementation with no extra states, the central activity is recognition of the visited states. The traditional DS-based CS-construction methods rely on just four SRPs, of which three, the SRPs 2-4, are applicable also when the SM is reduced, but has no DS.

The three SRPs all rely on a single IIOS as a witness. In this paper, we proposed the concept of distinguishers, IIOS sets which can be employed as collective witnesses. Generalizing the SRPs 2 and 4 to general distinguishers, we obtained SRP 5. The concept and the new SRP were discovered, concurrently and independently, also by Dincturk [26], who calls the SRP candidate elimination using incompatible sets. In his M.Sc. thesis, he experimentally demonstrated that by relying on the more general SRP, one can typically construct a shorter CS. The reduction which he obtained for the CS part constructed with the help of SRP 5 was about 30%.

In principle, any interpretation pattern facilitating more general and, hence, finer state recognition potentially facilitates construction of shorter CSs. Unlike Dincturk, we generalized to general distinguishers also SRP 3, obtaining the SRP 6. The application of SRP 6 is expected to be approximately as time consuming for larger distinguishers as that of SRP 5, for both SRPs rely on a single distinguisher. For the application of SRP 5, Dincturk experimentally demonstrated that consideration of distinguishers of the size up to 10 for larger SMs requires approximately 20 times as much time as consideration of singleton distinguishers only. This seems a lot, but not for the cases where the size of the constructed CS really matters, e.g. because every test step is somehow expensive and/or the CS is intended for extensive application.

After in SRP 7 formalizing state recognition through forcing a contradiction with the SUT determinism, we generalized the SRPs 2-7 to SRP 8, which formalizes *synergetic exploitation of the SUT determinism and any number of distinguishers*. Dincturk suggested such state recognition implicitly, by suggesting trial and error assignment of SM states to states visited during the interpreted experiments. SRP 8 excels in *abstractness*, *obviousness of semantics* and *ease of further generalization*. It is therefore interesting primarily as a *template* for developing, simply by specialization, further relatively easy to apply SRPs complementing the currently available.

An item for further study is also generalization of SRP 8 to multi-port systems with no coordination between the testers controlling and observing individual ports, for which we have already developed a partial solution [24]. The ultimate challenge, however, is to extend the ideas to adaptive testing strategies, which are in distributed testing or for a non-deterministic system often the only option [29, 30]. For a conclusion, we note that the SRPs 5-8 are, like the SRPs 1-4, useful also in the construction of non-singleton CTSs.

References

[1] ISO/IEC 9646: Information technology - Open Systems Interconnection - Conformance testing methodology and framework.

[2] J. Tenckhoff, Establishment of conformance test services in Europe, Computer Standards and Interfaces 7(1-2) (1988) 33-36.

[3] J.-H. Lee, J.-G. Hwang, D. Shin, K.-M. Lee, S. U. Kim, Development of verification and conformance testing tools for a railway signaling communication protocol, Computer Standards and Interfaces 31(2) (Feb. 2009) 362-371.

[4] A. X. Liu, M. G. Gouda, Firewall policy queries, IEEE Trans. Parallel and Distributed Systems 20(6) (June 2009) 766-777.

[5] D. Harel, A. Pnueli, On the development of reactive systems, in Logics and Models of Concurrent Systems (La Colle-sur-Loup,

1984), vol. 13 of NATO Adv. Sci. Inst. Ser. F Comput. Systems Sci., Springer-Verlag, Berlin, 1985, pp. 477-498.
[6] R. M. Hierons, K. Bogdanov, J. P. Bowen, R. Cleaveland, J. Derrick, J. Dick, M. Gheorghe, M. Harman, K. Kapoor, P. Krause, G. Luettgen, A. J. H. Simons, S. Vilkomir, M. R. Woodward, H. Zedan, Using formal methods to support testing, ACM Computing Surveys 41(2) (2009).
[7] R. Anido, A. Cavalli, Guaranteeing full fault coverage for UIO-based testing methods, Proc. IFIP Int'l Workshop Protocol Test Systems, pp. 221-236, Sept. 1995.
[8] M. Kapus-Kolar, A Better Procedure and a Stronger State-Recognition Pattern for Checking Sequence Construction, Jožef Stefan Institute Technical Report #10574, 2010.
[9] F. C. Hennie, Fault detecting experiments for sequential circuits, Proc. Fifth Ann. Symp. Switching Circuit Theory and Logical Design, pp. 95-110, Nov. 1964.
[10] G. Gönenc, A method for the design of fault detection experiments, IEEE Trans. Computers, 19(6) (June 1970) 551-558.
[11] A. Rezaki, H. Ural, Construction of checking sequences based on characterization sets, Computer Communications 18(12) (Dec. 1995) 911-920.
[12] H. Ural, X. Wu, F. Zhang, On minimizing the length of checking sequences, IEEE Trans. Computers 46(1) (Jan. 1997) 93-99.
[13] K. Inan, H. Ural, Efficient checking sequences for testing finite state machines, Information & Software Technology 41(11-12) (Sept. 1999) 799-812.
[14] R. M. Hierons, H. Ural, Reduced length checking sequences, IEEE Trans. Computers 51(9) (Sept. 2002) 1111-1117.
[15] J. Chen, R. M. Hierons, H. Ural, H. Yeningün, Eliminating redundant tests in a checking sequence, Proc. IFIP Int'l Conf. Testing of Communicating Systems, pp. 146-158, May-June 2005.
[16] K. T. Tekle, H. Ural, M. C. Yalcin, H. Yeningün, Generalizing redundancy elimination in checking sequences, Proc. Int'l Symp. Computer and Information Sciences, pp. 915-925, Oct. 2005.
[17] R. M. Hierons, H. Ural, Optimizing the length of checking sequences, IEEE Trans. Computers 55(5) (May 2006) 618-629.
[18] H. Ural, F. Zhang, Reducing the length of checking sequences by overlapping, Proc. IFIP Int'l Conf. Testing of Communicating Systems, pp. 274-288, May 2006.
[19] M. C. Yalcin, H. Yeningün, Using distinguishing and UIO sequences together in a checking sequence, Proc. IFIP Int'l Conf. Testing for Communicating Systems, pp. 259-273, May 2006.
[20] A. Simão, A. Petrenko, Generating checking sequences for partial reduced finite state machines. Proc. IFIP Int'l Con. Testing of Software and Communicating Systems, pp. 153-168, June 2008.
[21] R. M. Hierons, G.-V. Jourdan, H. Ural, H. Yenigün, Using adaptive distinguishing sequences in checking sequences, Proc. ACM Symp. Applied Computing, pp. 682-687, March 2008.
[22] L. Duan, J. Chen, Exploring alternatives for transition verification, J. Syst. Software 82(9) (Sept. 2009) 1388-1402.
[23] A. Simão, A. Petrenko, Checking sequence generation using state distinguishing subsequences, Proc. IEEE Int'l Workshops Software Testing, Verification, and Validation, pp. 48-56, April 2009.
[24] M. Kapus-Kolar, A Pragmatic Generic Test Sequence Construction Method With a Specialization for Checking Sequence Construction, Jožef Stefan Institute Technical Report #10333, 2009.
[25] R.M. Hierons, Checking sequences for distributed test architectures, Distributed Computing 21(3) (Sept. 2008) 223-238.
[26] M. E. Dincturk, A Two Phase Approach for Checking Sequence Generation, M.Sc. Thesis, Sabancı University, August 2009.
[27] M. Kapus-Kolar, On "Exploring alternatives for transition verification", submitted for publication, 2011.
[28] A. Simão, A. Petrenko, Checking completeness of tests for finite state machines, IEEE Trans. Computers 59(8) (Aug. 2010) 1023-1032.
[29] R. M. Hierons, Reaching and distinguishing states of distributed systems, SIAM J. Comput. 39(8) (Aug. 2010) 3480-3500.
[30] R. M. Hierons, Applying adaptive test cases to nondeterministic implementations, Information Processing Letters 98(2) (April 2006) 56-60.

Monika Kapus-Kolar received her B.Sc. degree in electrical engineering from the University of Maribor, Slovenia, and her M.Sc. and Ph.D. degrees in computer science from the University of Ljubljana, Slovenia. Since 1981 she has been with the Jožef Stefan Institute, Ljubljana, where she is currently a researcher at the Department of Communication Systems. Her current research interests include formal specification techniques and methods for the development of real-time, concurrent and reactive systems [5].