Journal of Universal Computer Science, vol. 23, no. 6 (2017), 560-585
submitted: 20/4/16, accepted: 23/5/17, appeared: 28/6/17 © *J.UCS*

A Generalization of a Popular Fault-Coverage-Preserving Test Set Transformation

Monika Kapus-Kolar
(Jožef Stefan Institute, Ljubljana, Slovenia
monika.kapus-kolar@ijs.si)

Abstract: In the optimization of test sets for black-box conformance testing of objects specified and modelled as a finite state machine (FSM), a popular transformation is that under a certain precondition, a tail of a test is removed and appended to some other test. We propose a weaker precondition under which the transformation remains fault-coverage-preserving. Along with a weaker precondition, we propose some weaker sufficient conditions for its satisfaction. To demonstrate the usefulness of the relaxations, we employ them for generalizing the checking sequence (CS) construction method of Inan and Ural (1999), to incomplete FSMs and with additional dimensions for CS optimization. The method and its generalized version are exceptional in that they can handle also the case where the upper bound, call it m, assumed for the size of the state set of the FSM under test is not less than twice the size, call it n, of the state set of the specification FSM. We prove that for complete FSMs, the additional optimization dimensions facilitate that in the limit for increasingly large (m/n) and (a/m), with a the number of the defined inputs, the factor of CS length reduction is of the order $\mathcal{O}(a^{m-n+1})$.
Key Words: black-box conformance testing, model-based testing, finite state machine, quasi-equivalence, test set optimization, fault-coverage-preserving transformation, checking sequence
Category: B.4.5., D.2.5.

1 Introduction

In the era of ubiquitous computing, we are embedded in a sea of devices of which we can see only the interface towards the environment. To establish whether such a device complies with its specification, one would perform black-box conformance testing. Formal construction of a test set for the purpose usually starts by adoption of a simple formal model of what the object under test could be. With such a model, it is usually easy to construct a test set that has the required fault coverage, but to give such a test set also the desired form, for example the desired trade-off between the number of tests and their length, is a more difficult task. Here, fault-coverage-preserving test set transformations can help. In most of the current test set construction methods with optimization, the exploitation of such transformations is only implicit, but nevertheless noticeable.

The paper generalizes a popular fault-coverage-preserving test set transformation intended for the cases where the specification is an observable finite state machine (OFSM), whereas the object under test presumably belongs to a given set of OFSMs and is supposed to be quasi-equivalent to the specification. OFSMs and related concepts and notations are presented in Section 2. OFSM-based test sets and related concepts and notations are presented in Section 3. The transformation and its generalization are presented in Section 4. In the transformation, a tail of a test is under a certain precondition removed and appended to some other test. The test set optimization method in which the use of the transformation is most explicit is that of [Simão et al. 2012], the paper in which the reader can find also a very illustrative example. Our generalization of the transformation is by weakening the precondition, in a way securing that the transformation remains fault-coverage-preserving. Along with a weaker precondition, we propose, in Section 5, some weaker sufficient conditions for its satisfaction.

A common application of the considered transformation is, for example, the trading of the number of tests for their length. There are at least four possible reasons why one would want to do this:

1. Because experiments reported in [Endo and Simão 2013] indicate the following: Test sets trading the number of tests for their length have better chances that errors will not go undetected even if the OFSM under test actually has more states than the upper bound assumed during the test set construction. The latter might well be the case, for in the wish to make both the test set construction process and its result acceptably cheap, one often deliberately makes over-optimistic assumptions.
2. To reduce the cumulative length of the test set members, as, for example, in the methods of [Simão and Petrenko 2010, Simão et al. 2012, Cutigi et al. 2016].
3. To reduce the number of the necessary resets of the OFSM under test (see, for example, [Hierons 2004, Hierons and Ural 2006]), for they usually take a lot of time or are undesirable in some other way.
4. To completely avoid the need for resets, for it might well be the case that reliable reset is not possible. To avoid resets, one needs a test set consisting of a single test. Such a test can be generated only for some kinds of the specification OFSM, provided that one also makes sufficiently strong assumptions on the OFSM under test. Particularly challenging is the construction of so-called checking sequences (CSs), individual tests that fail exactly on the faulty members of the target set of candidates for the OFSM under test.

To demonstrate the usefulness of the relaxations proposed in the Sections 4 and 5, we in Section 6 employ them for generalizing (after correcting a tiny error in it) the CS construction method of [Inan and Ural 1999], to incomplete OFSMs and with additional dimensions for CS optimization. In comparison to the other currently available CS construction methods [Hennie 1964, Hsieh 1971, Farmer 1973, Braun and Givone 1979, Braun and Givone 1981, Yao et al. 1993, Rezaki and Ural 1995, Ural et al. 1997, Hierons and Ural 2002, Hierons and Ural 2003, Chen et al. 2005, Tekle et al. 2005, Hierons and Ural 2006, Ural and Zhang 2006, Ural and Williams 2006, Yalcin and Yenigün 2006, Hierons et al. 2008, Hierons and Ural 2008, Simão and Petrenko 2008, Dincturk 2009, Duan and Chen 2009, Simão and Petrenko 2009, Hierons and Ural 2010, Kapus-Kolar 2012b, Petrenko et al. 2012, Porto et. al. 2013, Kapus-Kolar 2014, Jourdan 2015], the method of [Inan and Ural 1999] and its generalized version are exceptional in that they can handle also the case where the upper bound, call it m, assumed for the size of the state set of the OFSM under test is not less than twice the size, call it n, of the state set of the specification OFSM. In Section 7 we prove that for complete FSMs, the additional optimization dimensions facilitate that in the limit for increasingly large (m/n) and (a/m), with a the number of the defined inputs, the factor of CS length reduction is of the order $\mathcal{O}(a^{m-n+1})$. Section 8 concludes the paper.

2 OFSMs and related concepts and notations

An OFSM is an automaton that at any moment either idles in one of its candidate states, of which one is its initial state, or is executing a transition. The latter involves accepting an input x defined in the current state s, issuing an output defined for the pair (s, x) and entering the state defined as the one resulting from executing the input/output (IO) xy from s. By executing from a state a sequence of transitions, an OFSM executes an IO sequence (IOS) of the state. The state of an OFSM can presumably be changed also with the reset operation, which returns the OFSM into its initial state.

In OFSM-based black-box testing, what one directly observes are IOSs. States are only a secondary concept, with each state name regarded mainly as a synonym for the set of all IOSs executable from the state. OFSMs are a tertiary concept, with each OFSM name regarded mainly as a synonym for the initial state of the OFSM or for the IOS set of the state. In the rest of the paper, we also reason primarily in terms of IOSs. This is evident already from the rest of this section, in which we present our basic notations.

Definition 1 Q, s, x, y, z and Z, possibly decorated, denote, respectively, an OFSM, a state, an input, an output, an IOS and an IOS set.

Definition 2 For each IOS $z = x_1y_1x_2y_2 \ldots x_ky_k$:
- $ln(z)$ denotes its length k, with ε denoting an IOS of the length 0.
- $is(z)$ denotes its input sequence $x_1 \ldots x_k$.
- $pf(z)$ denotes the set of all IOSs that are its prefix.
- For each IOS z', $z' < z$ denotes that $z' \in (pf(z) \setminus \{z\})$.
- For each natural k, $(z)^k$ denotes the IOS $z_1z_2 \ldots z_k$ with $z_i = z$ for each $1 \leq i \leq k$.

Definition 3 For each OFSM Q:

- $st(Q)$ denotes the set of the defined states.
- $init(Q)$ denotes the member of $st(Q)$ that is its initial state.
- $in(Q)$ denotes the set of the defined inputs.
- $out(Q)$ denotes the set of the defined outputs.
- $tr(Q)$ denotes the set of the defined transitions.
- Each transition in $tr(Q)$ is an $(s, x/y, s')$ with $(\{s, s'\} \subseteq st(Q)) \wedge (x \in in(Q)) \wedge (y \in out(Q))$. For such a transition, s is the starting state, x is the input, y is the output and s' is the ending state.
- There is no transition pair $\{(s, x/y, s'), (s, x/y, s'')\} \subseteq tr(Q)$ with $s' \neq s''$, which is the reason why Q is called observable.
- We assume $in(Q) = \{x | \exists (s, x/y, s') \in tr(Q)\}$.

Definition 4 For each OFSM Q and each state $s \in st(Q)$:

- $ios(s)$ denotes the set of all IOSs executable from s.
- A unique IOS of s is such an IOS $z \in ios(Q)$ that $z \notin ios(s')$ for each state $s' \in (st(Q) \setminus \{s'\})$.
- For each natural i, $ios_{\leq i}(s)$ denotes the IOS set $\{z | (z \in ios(s)) \wedge (ln(z) \leq i)\}$.
- For each IOS $z \in ios(s)$:
  - $ts(s, z)$ denotes the set of all those transitions in $tr(Q)$ of which at least one instance is executed during its execution from s.
  - $st(s, z)$, also called 's-after-z', denotes the state resulting from its execution from s.
  - $ios(s, z)$ denotes the IOS set $\{z' | zz' \in ios(s)\}$.
  - $in(s, z)$ denotes the input set $\{x | \exists y : (zxy \in ios(s))\}$.
- For each IOS set $Z \subseteq ios(s)$, $end(s, Z)$ denotes the state set $\{st(s, z) | z \in Z\}$.
- For each IOS pair $\{z, z'\} \subseteq ios(s)$, 'divergence of z and z' in s' denotes that $st(s, z) \neq st(s, z')$.

Definition 5 The concepts introduced in Definition 4 naturally extend to OFSMs, through the following shorthand notation: In each of the there defined expressions, including those in quotes, it is allowed that a name of an OFSM Q occurs in the place of s. In such a case, the name denotes the state $init(Q)$.

Definition 6 A given OFSM Q is

- complete if for each state $s \in st(Q)$ and input $x \in in(Q)$, there is an $xy \in ios(s)$;
- deterministic if there is no transition pair $\{(s, x/y, s'), (s, x/y', s'')\} \subseteq tr(Q)$ with $(y, s') \neq (y', s'')$;
- initially connected if for each state $s \in st(Q)$, there is an IOS $z \in ios(Q)$ with $st(Q, z) = s$;

- strongly connected if for each state pair $\{s, s'\} \subseteq st(Q)$, there is an IOS $z \in ios(s)$ with $st(s, z) = s'$;
- minimal if for each state pair $\{s, s'\} \subseteq st(Q)$ with $s \neq s'$, $ios(s) \neq ios(s')$;
- input-equivalent to a given OFSM Q' if for each IOS $z \in (ios(Q) \cap ios(Q'))$, $in(Q, z) = in(Q', z)$.

Definition 7 For each IOS set Z:

- $pf(Z)$ denotes the IOS set $\{z|\exists z' \in Z : (z \in pf(z'))\}$.
- $min(Z)$ denotes the set $\{z|(z \in Z) \wedge \neg\exists z' \in Z : (z' < z)\}$ of all its minimal members.
- $max(Z)$ denotes the set $\{z|(z \in Z) \wedge \neg\exists z' \in Z : (z < z')\}$ of all its maximal members.
- For each IOS z, $z \otimes Z$ denotes the IOS set $\{zz'|z' \in Z\}$.
- For each IOS set Z', $Z \otimes Z'$ denotes the IOS set $\{zz'|(z \in Z) \wedge (z' \in Z')\}$.
- For each IOS $z \in pf(Z)$, $ios(Z, z)$, also called 'Z-after-z', denotes the IOS set $\{z'|zz' \in pf(Z)\}$.
- For each IOS $z \in pf(Z)$, $in(Z, z)$ denotes the input set $\{x|\exists y : (zxy \in pf(Z))\}$.

3 OFSM-based test sets and related concepts and notations

Definition 8 In the rest of the paper:

- M denotes the specification OFSM.
- S and n denote $st(M)$ and $|st(M)|$, respectively.
- $\mathcal{I}$ denotes the set of all OFSMs possibly in the role of the OFSM under test, i.e., satisfying all assumptions made on the OFSM under test.
- Every OFSM $Q \in (\mathcal{I} \cup \{M\})$ is presumably initially connected and minimal.

We assume that a given OFSM Q is non-faulty exactly if it is quasi-equivalent, i.e., $ios(M)$-equivalent to M (see the Definitions 9-11 below). Accordingly, test sets are defined as subsets of $ios(M)$, whereas to pass a given test set Z means to be Z-equivalent to M. Informally, to pass an individual test z means to implement it exactly like M. The implementation of a given IOS z in a given OFSM Q is the set of all IOSs in $ios(Q)$ that are a prefix of z or differ from one of its prefixes only in the last output. A deterministic OFSM Q passes a test set Z of a deterministic M exactly if $Z \subseteq ios(Q)$.

Definition 9 For each IOS set Z:

- For each IOS $z = x_1y_1x_2y_2 \ldots x_ky_k$, $imp(Z, z)$, also called 'the implementation of z in Z', denotes the set of all IOSs in $pf(Z)$ that are of the form $z'x_iy'_i$ with $z'x_iy_i \in pf(z)$ and y'_i an output.

- For each IOS set Z', $imp(Z, Z')$, also called 'the implementation of Z' in Z', denotes the IOS set $\{z | \exists z' \in Z' : (z \in imp(Z, z'))\}$.

Definition 10 For each IOS set pair (Z, Z'):

- For each IOS z, $Z \equiv_z Z'$, also called 'z-equivalence of Z and Z'', denotes that $imp(Z, z) = imp(Z', z)$.
- For each IOS set Z'', $Z \equiv_{Z''} Z'$, also called 'Z''-equivalence of Z and Z'', denotes that $\forall z \in Z'' : (Z \equiv_z Z')$.
- $Z \sqsupseteq Z'$, also called 'quasi-equivalence of Z to Z'', denotes that $Z \equiv_{Z'} Z'$.
- For each IOS set Z'' and each IOS z, $Z \approx_{Z'',z} Z'$ denotes that $(Z \equiv_z Z'') \Leftrightarrow (Z' \equiv_z Z'')$, i.e., that Z is z-equivalent to Z'' exactly if Z' is.
- For each IOS set Z'', $Z \approx_{Z''} Z'$ denotes that $\forall z \in pf(Z'') : (Z \approx_{Z'',z} Z')$, i.e., that Z and Z' are z-equivalent to Z'' for the same IOSs z in the prefix set of Z''.

Definition 11 The concepts introduced in the Definitions 9 and 10 naturally extend to states and OFSMs, through the following shorthand notation: In each of the there defined expressions, including those in quotes, it is allowed that a name of a state s or a name of an OFSM Q occurs in the place of Z, Z' or Z''. In such a case, the name denotes the IOS set $ios(s)$ or $ios(Q)$, respectively.

The fault coverage of a given test set is defined by the set of those OFSMs in $\mathcal{I}$ that do not pass it. If the set comprises all faulty members $\mathcal{I}$, the test set is complete, but there are also situations where one wants a test set with some other fault coverage. A CS is a test z with $\{z\}$ a complete test set. Test sets Z with the same $imp(M, Z)$ have the same fault coverage.

For experimental confirmation that a given OFSM Q passes a given test set Z, the main requirement is that Q from its initial state executes every member of the IOS set $max(imp(M, Z))$. This requires at least $(|max(imp(M, Z))| - 1)$ resets. Accordingly, the default cost of a given test set Z is $(\Sigma_{z \in max(imp(M,Z))} ln(z) + |max(imp(M, Z))| - 1)$. Test sets Z with the same $imp(M, Z)$ have the same default cost.

4 The popular transformation and a weaker precondition

The considered fault-coverage-preserving test set transformation goes as follow:

Transformation 1 In the given test set Z, replace a test pair $\{zz', z''\}$ with the test pair $\{z, z''z'\}$ provided that for each OFSM $Q \in (\mathcal{I} \cup \{M\})$ with $Q \equiv_{(Z \setminus \{zz'\}) \cup \{z\}} M$, $st(Q, z'') \sqsupseteq st(Q, z)$.

The precondition of the transformation requires that in each OFSM in $\mathcal{I} \cup \{M\}$ that passes the test set $(Z \setminus \{zz'\}) \cup \{z\}$, the state $st(Q, z'')$ is quasi-equivalent to the state $st(Q, z)$. This secures that the implementation of z' in $st(Q, z'')$ is the same as the implementation of z' in $st(Q, z)$, meaning that in the case of $Q \in \mathcal{I}$, it does not matter which of them is established experimentally, by pursuing z' after z'' or after z, respectively, for in both cases, one obtains the same information on Q.

Now recall that we are discussing conformance testing. This means that one is not interested in the details of the implementation of z' in $st(Q, z)$. All that matters is whether $st(Q, z)$ is z'-equivalent to the state $st(M, z)$, for this tells whether the implementation of z' in $st(Q, z)$ can make Q fail the test set Z. If

(1) the states $st(M, z)$ and $st(M, z'')$ are z'-equivalent and

(2) the state $st(Q, z)$ is z'-equivalent to $st(M, z)$ exactly if the state $st(Q, z'')$ is z'-equivalent to $st(M, z'')$,

it does not matter whether one checks z'-equivalence of $st(Q, z)$ to $st(M, z)$ or z'-equivalence of $st(Q, z'')$ to $st(M, z'')$. This suggests that the following generalization of Transformation 1 is also fault-coverage-preserving:

Transformation 2 In the given test set Z, replace a test pair $\{zz', z''\}$ with the test pair $\{z, z''z'\}$ provided that for each OFSM $Q \in (\mathcal{I} \cup \{M\})$ with $Q \equiv_{(Z\setminus\{zz'\})\cup\{z\}} M$, $st(Q, z'') \approx_{st(M,z),z'} st(Q, z)$.

Theorem 1 (Proof in Appendix)
Transformation 2 is fault-coverage-preserving.

5 Some old and some new sufficient conditions for satisfying the weaker precondition

5.1 Two simple sufficient conditions for direct satisfaction

For proving the $st(Q, z'') \approx_{st(M,z),z'} st(Q, z)$ required in Transformation 2, a possible method is to prove $st(Q, z'') = st(Q, z)$, which proves also the $st(Q, z'') \sqsupseteq st(Q, z)$ required in Transformation 1. For proving an $st(Q, z) = st(Q, z')$, a possible method is to rely on the following proposition, which we newly prove also for incomplete M (for complete M, it has been justified in [Bonifácio et al. 2012]):

Proposition 1 (Proof in Appendix) *If M and a given OFSM $Q \in \mathcal{I}$ are deterministic and input-equivalent, then for each natural $m \geq n$ with $|st(Q)| \leq m$, state $s \in S$ and IOS pair $\{z, z'\} \subseteq ios(M)$ with $st(M, z) = st(M, z') = s$, a sufficient condition for $st(Q, z) = st(Q, z')$ is the existence of an IOS set $Z \subseteq (ios(M) \cap ios(Q))$ satisfying all the following:*

(1) $\{z, z'\} \otimes ios_{\leq m-|end(M,Z)|}(s) \subseteq Z$
(2) $st(Q, z_1) \neq st(Q, z_2)$ *for each IOS pair* $\{z_1, z_2\} \subseteq Z$ *with* $st(M, z_1) \neq st(M, z_2)$.

Informally, the condition claimed sufficient requires that Q passes a test set Z satisfying all the following:

(1) For each IOS $z_1 \in \{z, z'\}$, Z comprises every IOS $z_1z_2 \in ios(M)$ with $ln(z_2) \leq m - |end(M, Z)|$.
(2) Any two IOSs in Z that diverge in M diverge also in Q.

By proving an $st(Q, z) = st(Q, z')$, one proves $(z'' \in ios(Q, z)) \Leftrightarrow (z'' \in ios(Q, z'))$ for every IOS z''. This is interesting because for a deterministic Q, the $st(Q, z'') \approx_{st(M,z),z'} st(Q, z)$ required in Transformation 2 translates to $(z' \in ios(Q, z'')) \Leftrightarrow (z' \in ios(Q, z))$.

For proving a $(z'' \in ios(Q, z)) \Leftrightarrow (z'' \in ios(Q, z'))$ without proving $st(Q, z) = st(Q, z')$, one can rely on the following new proposition:

Proposition 2 (Proof in Appendix) *If* M *and a given OFSM* $Q \in \mathcal{I}$ *are deterministic, then for each natural* $m \geq n$ *with* $|st(Q)| \leq m$*, state* $s \in S$*, IOS pair* $\{z, z'\} \subseteq ios(M)$ *with* $st(M, z) = st(M, z') = s$ *and IOS* $z'' \in ios(s)$*, a sufficient condition for* $(z'' \in ios(Q, z)) \Leftrightarrow (z'' \in ios(Q, z'))$ *is the existence of an IOS set* $Z \subseteq (ios(M) \cap ios(Q))$ *satisfying all the following:*

(1) $\{z, z'\} \otimes \{z'''|(z''' \in ios_{\leq m-|end(M,Z)|}(s)) \wedge (ts(s, z''') \subseteq ts(s, z''))\} \subseteq Z$
(2) $st(Q, z_1) \neq st(Q, z_2)$ *for each IOS pair* $\{z_1, z_2\} \subseteq Z$ *with* $st(M, z_1) \neq st(M, z_2)$.

Proposition 2 strongly resembles Proposition 1, but defines for $(z'' \in ios(Q, z)) \Leftrightarrow (z'' \in ios(Q, z'))$ a sufficient condition that is weaker. Namely, M and Q need not be input-equivalent and (1) requires the presence in Z only for those IOSs z_1z_2 with $(z_1 \in \{z, z'\}) \wedge (z_2 \in ios_{\leq m-|end(M,Z)|}(s))$ that satisfy $ts(s, z_2) \subseteq ts(s, z'')$.

5.2 Two advanced sufficient conditions for direct satisfaction

This section is intended only for advanced readers and, with the exception of the first of the two additional definitions below, contains no material necessary for understanding the subsequent sections.

Definition 12 For each number set N, $inf(N)$ denotes the smallest element and $sup(N)$ denotes the largest element.

Definition 13 For each IOS set pair (Z, Z'), $notall(Z, Z')$ denotes the IOS set $min(\{z|(z \in (pf(Z) \cap pf(Z'))) \wedge (in(Z', z) \not\subseteq in(Z, z))\})$.

Informally, $notall(Z, Z')$ is the set of the minimal ones among those IOSs z for which there is an input x with the following property: The prefix set of Z' comprises an IOS starting with zx, whereas the prefix set of Z does not.

For proving the $st(Q, z'') \approx_{st(M,z),z'} st(Q, z)$ required in Transformation 2, the Propositions 1 and 2 can be helpful only if M is deterministic and state distinguishing is not a problem. In this section, we propose two new theorems which are conceptually their generalizations and applicable in the general case.

The first of the new theorems is conceptually a generalization of Proposition 1 and of the ideas of [Bonifácio et al. 2012]:

Theorem 2 (Proof in Appendix) *For each OFSM $Q \in \mathcal{I}$, each non-empty IOS set $Z \subseteq ios(Q)$ and each natural k with $0 < k \leq |Z|$, a sufficient condition for $|end(Q, Z)| \leq k$ is that for the IOSs $z \in ios(Q)$, there exists a function $f(z)$ satisfying all the following:*

(1) For each IOS pair $\{z, z'\} \subseteq ios(Q)$ with $st(Q, z) = st(Q, z')$, $f(z) = f(z')$.

(2) Let Δ denote $sup(\{0, |st(Q)| - |\{f(z)|z \in ios(Q)\}| - k + 1\})$. For each IOS pair $\{z, z'\} \subseteq Z$:

(i) $ios_{\leq\Delta}(st(Q, z)) = ios_{\leq\Delta}(st(Q, z'))$

(ii) For each IOS $z'' \in ios_{\leq\Delta}(st(Q, z))$, $f(zz'') = f(z'z'')$.

Informally, Theorem 2 defines a sufficient condition for $|end(Q, Z)| \leq k$, which in the case of $k = 1$ and Z a pair $\{z, z'\}$ means that $st(Q, z) = st(Q, z')$. For $|end(Q, Z)| \leq k$, Theorem 2 requires the existence of a function f by which the IOSs in $ios(Q)$ are partitioned in such a way that those with the same resulting state in Q are in the same group. The remaining constraints for f are defined in terms of Δ, a natural reflecting the number of the groups into which $ios(Q)$ is partitioned by f. It is required that for each IOS $z \in Z$:

1. The set of those IOSs z with $ln(z) \leq \Delta$ which Q can execute after individual IOSs $z' \in Z$ is the same for every IOS $z' \in Z$.
2. For each of the continuations z, $f(z'z)$ is the same for every IOS $z \in Z$.

To prove the $st(Q, z'') \approx_{st(M,z),z'} st(Q, z)$ required in Transformation 2 without proving $st(Q, z'') = st(Q, z)$, it suffices to prove $st(Q, z'') \approx_{imp(st(M,z),Z)} st(Q, z)$ for an IOS set Z satisfying $\{z'\} \subseteq Z \subseteq ios(M, z)$. For this, one can rely on the following new theorem, conceptually a generalization of Proposition 2:

Theorem 3 (Proof in Appendix) *For each OFSM Q, subset $K = \{z, z'\}$ of $ios(Q)$ and IOS set Z, a sufficient condition for $st(Q, z) \approx_Z st(Q, z')$ is the existence of an IOS set $Z' \subseteq pf(Z)$ satisfying all the following:*

(1) $st(Q, z) \equiv_{Z'} st(Q, z') \equiv_{Z'} Z$

(2) For each IOS $\beta \in notall(imp(Z, Z'), Z)$, there is a pair $(L_\beta, [H_{\beta,h}]_{1\leq h\leq k_\beta})$ satisfying all the following:

(a) $(L_\beta \subseteq ios(Q)) \wedge (k_\beta > 0)$
(b) *For each* $1 \leq h \leq k_\beta$, $\emptyset \subset H_{\beta,h} \subseteq pf(\beta)$. *Let* H_β *denote the IOS set* $H_{\beta,1} \cup \ldots \cup H_{\beta,k_\beta}$.
(c) *For each* $1 \leq h \leq k_\beta$, $ios(Z, z_i) \sqsupseteq ios(Z, z_j)$ *for each IOS pair* $\{z_i, z_j\} \subseteq H_{\beta,h}$ *with* $z_i < z_j$.
(d) *For each* $1 \leq i < j \leq k_\beta$, $st(Q, z_i) \neq st(Q, z_j)$ *for each IOS pair* $\{z_i, z_j\}$ *with* $(z_i \in (K \otimes H_{\beta,i})) \wedge (z_j \in (K \otimes H_{\beta,j}))$.
(e) $|H_\beta| + k_\beta > |end(Q, L_\beta \cup (K \otimes H_\beta))| - |end(Q, L_\beta)|$
(f) $st(Q, z_i) \neq st(Q, z_j)$ *for each IOS pair* $\{z_i, z_j\}$ *with* $(z_i \in L_\beta) \wedge (z_j \in (K \otimes H_\beta))$.

Informally, Theorem 3 defines a sufficient condition for $st(Q, z) \approx_Z st(Q, z')$, which means that Q-after-z and Q-after-z' are z''-equivalent to Z for the same IOSs z'' in the prefix set of Z. For this, Theorem 3 requires the existence of an IOS set $Z' \subseteq pf(Z)$ satisfying specific constrains, of which the first is that Q-after-z and Q-after-z' must be Z'-equivalent to Z.

The contributors of the remaining constraints are the IOSs in the set $notall(imp(Z, Z'), Z)$, i.e., the IOSs in $pf(Z) \cap ios(Q, z) \cap ios(Q, z')$ after which the behaviour of Q-after-z and Q-after-z' becomes questionable for at least one next input of interest. Additional constraints for each such IOS β are necessary because the fact that Q-after-z and Q-after-z' are Z'-equivalent to Z does not secure that Q-after-$z\beta$ and Q-after-$z'\beta$ are z''-equivalent to Z-after-β for the same IOSs z'' in Z-after-β.

For each IOS $\beta \in notall(imp(Z, Z'), Z)$, Theorem 3 requires the existence of a non-empty set H_β of prefixes of β, of a partitioning $[H_{\beta,i}]_{1 \leq i \leq k_\beta}$ of H_β and of an IOS set $L_\beta \subseteq ios(Q)$ such that:

(1) For each $1 \leq h \leq k_\beta$ and IOS pair $\{z_i, z_j\} \subseteq H_{\beta,h}$ with z_i shorter than z_j, Z-after-z_i is quasi-equivalent to Z-after-z_j.
(2) For each $1 \leq i < j \leq k_\beta$, divergence in Q is secured for each pair of an IOS in $K \otimes H_{\beta,i}$ and an IOS in $K \otimes H_{\beta,j}$.
(3) The sum of k_β and the size of H_β is more than the difference between the degree to which the IOSs in $L_\beta \cup (K \otimes H_\beta)$ diverge in Q and the degree to which the IOSs in L_β diverge in Q.
(4) Divergence in Q is secured for each pair of an IOS in L_β and an IOS in $K \otimes H_\beta$.

5.3 Two sufficient conditions for indirect satisfaction

From the Sections 5.1 and 5.2, respectively, recall that to prove the $st(Q, z'') \approx_{st(M,z),z'} st(Q, z)$ required in Transformation 2, it suffices to prove $st(Q, z'') = st(Q, z)$ or $st(Q, z'') \approx_{imp(st(M,z),Z)} st(Q, z)$ for an IOS set Z satisfying $\{z'\} \subseteq Z \subseteq ios(M, z)$.

After an IOS pair $\{z, z'\} \subseteq ios(Q)$ with $st(Q, z) = st(Q, z')$ has been identified for a given OFSM $Q \in \mathcal{I}$, further such pairs can be identified with the Proposition 3 below. In the method of [Simão et al. 2012], for example, synergetic exploitation of Propositions 1 and 3 is employed for implicit exploitation of Transformation 1.

Proposition 3 *For each OFSM $Q \in \mathcal{I}$ and IOS pair $\{z, z'\} \subseteq ios(Q)$ with $st(Q, z) = st(Q, z')$, $st(Q, zz'') = st(Q, z'z'')$ for each IOS $z'' \in ios(Q, z')$.*

After an IOS pair $\{z, z'\} \subseteq ios(Q)$ with $st(Q, z) \approx_Z st(Q, z')$ for an IOS set Z has been identified for a given OFSM $Q \in \mathcal{I}$, further such pairs can be identified with help of the following new proposition:

Proposition 4 (Proof in Appendix) *For each OFSM $Q \in \mathcal{I}$, IOS Z and IOS pair $\{z, z'\} \subseteq ios(Q)$ with $st(Q, z) \approx_Z st(Q, z')$, $st(Q, zz'') \approx_{ios(Z,z'')} st(Q, z'z'')$ for each IOS $z'' \in pf(Z)$ with $st(Q, z) \equiv_{z''} st(Q, z') \equiv_{z''} Z$.*

Informally, Proposition 4 defines a sufficient condition for $st(Q, zz'') \approx_{ios(Z,z'')} st(Q, z'z'')$, which means that the Q-after-zz'' and Q-after-$z'z''$ are z'''-equivalent to Z-after-z'' for the same IOSs z''' in the prefix set of Z-after-z''. For this, Proposition 4 requires the following:

(1) That Q-after-z and Q-after-z' are z'''-equivalent to Z for the same IOSs z''' in the prefix set of Z.

(2) z''-equivalence of Q-after-z, Q-after-z' and Z.

6 The method of [Inan and Ural 1999], a correction and a generalization

6.1 Introduction

To be a CS, the generated test must check not only that the OFSM Q under test has a state quasi-equivalent to $init(M)$, but also that $init(Q)$ is such a test. To check the latter, the test must start with a special-purpose segment, whose feasibility is an additional assumption on M. Besides, one needs an additional assumption on Q. In the method of [Inan and Ural 1999], the special-purpose segment and the additional assumptions are missing. The method that we generalize is the method of [Inan and Ural 1999] without the error.

The new method works under the following assumptions, besides those given already in Definition 8:

(1) M and the OFSM Q under test are deterministic and input-equivalent.

(2) M is strongly connected.

(3) $|st(Q)|$ is not more than a given natural $m \geq n$.

1. If there is a state $s \in (S \setminus \{init(M)\})$ with $in(s, \varepsilon) = in(M, \varepsilon)$, choose a unique IOS w_0 of $init(M)$. Otherwise, let $w_0 \leftarrow \varepsilon$.
2. Choose a collection $[D_s]_{s \in S}$ of harmonized state identifiers.
3. For each state $s \in S$: $Z_s \leftarrow max(\{zz' | (z \in ios_{\leq m-n}(s)) \wedge (z' \in D_{st(s,z)})\})$
4. For each transition $t = (s, x/y, s')$ in $tr(M)$: $Z_t \leftarrow max(xy \otimes Z_{s'}) \setminus Z_s$
5. Choose a 4-tuple $(S', [Z'_t]_{t \in tr(M)}, \prec, [(q_{t,z}, v_{t,z})]_{t \in tr(M), z \in (Z_t \setminus Z'_t)})$ with the following properties (if the first two components of the 4-tuple are given their default values, the remaining components are unnecessary):
   (1) $S' \subseteq S$, where the default S' is S.
   (2) For each transition $t \in tr(M)$, $Z'_t \subseteq Z_t$, where the default Z'_t is Z_t.
   (3) $\prec$ is a strict order on $tr(M)$.
   (4) For each transition $t = (s, x/y, s')$ in $tr(M)$ and each IOS $z \in (Z_t \setminus Z'_t)$, $(q_{t,z}, v_{t,z})$ is a path with the following properties:
      (1) $(q_{t,z} \in (S \setminus S')) \wedge (st(q_{t,z}, v_{t,z}) = s)$
      (2) For each transition $t' \in ts(q_{t,z}, v_{t,z})$, $t' \prec t$.
6. For each transition $t \in tr(M)$ and each IOS $z \in Z'_t$:
   $Z_{t,z} \leftarrow max(\{z' | (z' \in ios_{\leq m-n}(s)) \wedge (ts(s, z') \subseteq ts(s, z))\})$
7. Choose (the default procedure for $m \geq 2n$ is in Figure 2) a path set
   $P = [(r_s, v_s v'_s)]_{s \in S'} \cup [(r_s, v_s z)]_{s \in (S \setminus S'), z \in Z_s} \cup [(r_{t,z}, v_{t,z} v'_{t,z})]_{t \in tr(M), z \in Z'_t} \cup$
   $[(r_{q_{t,z}}, v_{q_{t,z}} v_{t,z} z)]_{t \in tr(M), z \in (Z_t \setminus Z'_t)} \cup P'$
   with the following properties:
   (1) For each state $s \in S'$, (r_s, v_s, v'_s) is a (P, s, Z_s)-locator.
   (2) For each state $s \in (S \setminus S')$, (r_s, v_s) is a (P, s)-locator.
   (3) For each transition $t = (s, x/y, s')$ in $tr(M)$ and each IOS $z \in Z'_t$, $(r_{t,z}, v_{t,z}, v'_{t,z})$ is a $(P, s, max(\{z\} \cup Z_{t,z}))$-locator.
8. Choose (the default is to do it as suggested in the Section 5 of [Inan and Ural 1999]) a test w that starts with w_0 and covers P.

Figure 1: The new method.

(4) $m = n$ or Q is strongly connected.
(5) If there is a state $s \in (S \setminus \{init(M)\})$ with $in(s, \varepsilon) = in(M, \varepsilon)$, the initial state of M has a unique IOS.

The assumptions missing in the method of [Inan and Ural 1999] are the last two.

In Figure 1, we give for the new method a formal specification deliberately prescribing only *what* to choose, so that one is free to make the choices with any optimization method (ideally, all choices would be made in an integrated way). In [Inan and Ural 1999], one also finds such separation of concerns, though not without some guidelines on *how* to make the choices. The auxiliary procedures in the Figures 2 and 3 are also based on them. The corrected method of [Inan

1. For each state $s \in (S \setminus S')$, choose a state $q_s \in S$.
2. For each state $s \in (S' \cup \{q_{s'} | s' \in (S \setminus S')\})$, construct (the default procedure is in Figure 3) an $(\emptyset, s, Z_s)$-locator (q'_s, z_s, z'_s).
3. For each state $s \in S'$: $(r_s, v_s, v'_s) \leftarrow (q'_s, z_s, z'_s)$
4. For each state $s \in (S \setminus S')$:
   i. Choose an IOS $v'_s \in ios(q_s)$ with $st(q_s, z'_{q_s} v'_s) = s$.
   ii. $(r_s, v_s) \leftarrow (q'_{q_s}, z_{q_s} z'_{q_s} v'_s)$
5. For each transition $t = (s, x/y, s')$ in $tr(M)$ and each ios $z \in Z'_t$, construct (the default procedure is in Figure 3) an $(\emptyset, s, max(\{z\} \cup Z_{t,z}))$-locator $(r_{t,z}, v_{t,z}, v'_{t,z})$.
6. $P \leftarrow [(r_s, v_s v'_s)]_{s \in S'} \cup [(r_s, v_s z)]_{s \in (S \setminus S'), z \in Z_s} \cup [(r_{t,z}, v_{t,z} v'_{t,z})]_{t \in tr(M), z \in Z'_t} \cup$ $[(r_{q_{t,z}}, v_{q_{t,z}} v_{t,z} z)]_{t \in tr(M), z \in (Z_t \setminus Z'_t)}$

Figure 2: The default step 7 of the new method for $m \geq 2n$.

1. Order the IOSs in Z into a sequence $z_1 \ldots z_k$.
2. For each $1 \leq i < k$, choose an IOS z'_i with $(z_i z'_i \in ios(s)) \wedge (st(s, z_i z'_i) = s)$.
3. Let $v_1 \leftarrow \varepsilon$. For each $1 \leq i < k$, let $v_{i+1} \leftarrow v_i (z_i z'_i v_i)^m$.
4. Return the triplet (s, v_k, z_k).

Figure 3: The default procedure for constructing an $(\emptyset, s, Z)$-locator.

and Ural 1999] is conceptually that specialization of the new method in which M is complete and S' and all Z'_t are empty. In the original method of [Inan and Ural 1999], the starting part w_0 of w is virtually missing..

The procedures in the Figures 1-3 rely also on the following definitions:

Definition 14

- A path is a pair (s, z) with $(s \in S) \wedge (z \in ios(s))$.
- A given test z covers a given path set P if for each path $(s, z') \in P$, there is a test $z''z' \in pf(z)$ with $st(M, z'') = s$.
- A given OFSM Q covers a given path set P if $z \in ios(Q)$ for a test z which covers P.

Definition 15 A collection of harmonized state identifiers is an IOS set collection $[Z_s]_{s \in S}$ that for each state $s \in S$ satisfies all the following:

(1) $\emptyset \subset Z_s \subseteq ios(s)$
(2) For each state $s' \in (S \setminus \{s\})$, there is an IOS pair (z, z') with $(z \in Z_s) \wedge (z' \in Z_{s'}) \wedge (is(z) = is(z')) \wedge ((z, in(s, z)) \neq (z', in(s', z')))$.

Definition 16 For each path set P, each state $s \in S$ and each IOS set $Z \subseteq ios(s)$, a (P, s, Z)-locator is a triplet (s', z, z') satisfying all the following:

(1) $(s' \in S) \wedge (z \in ios(s')) \wedge (st(s', z) = s) \wedge (z' \in ios(s))$
(2) For each OFSM $Q \in \mathcal{I}$ covering P and each IOS $z''zz' \in (ios(M) \cap ios(Q))$ with $st(M, z'') = s'$, $Z \subseteq ios(Q, z''z)$.

Definition 17 For each path set P and each state $s \in S$, a (P, s)-locator is a pair (s', z) satisfying all the following:

(1) $(s' \in S) \wedge (z \in ios(s')) \wedge (st(s', z) = s)$
(2) For each OFSM $Q \in \mathcal{I}$ covering P, $|end(Q, \{z'z | (z'z \in ios(M)) \wedge (st(M, z') = s')\})| \leq 1$.

6.2 An explanatory proof of the new method

The plan behind the steps 1-4 is as follows:

Plan 1

1. For each state $s \in S$:
   i. The test w constructed in the step 8 will have a prefix w_s with $st(M, w_s) = s$.
   ii. w will check that in the OFSM Q under test, $Z_s \subseteq ios(Q, w_s)$.
   iii. By $D_s \subseteq pf(Z_s)$, w will, hence, check that $D_s \subseteq ios(Q, w_s)$.
2. By $[D_s]_{s \in S}$ a collection of harmonized state identifiers and by Proposition 1, w will, hence, check that for each state $s \in S$, all the following is true:
   i. $end(Q, ios(Q) \cap ios(M))$ comprises at most one state s' with $Z_s \subseteq ios(s')$, call it $f(s)$.
   ii. $st(M, w_s) = f(s)$. For each state $s' \in (S \setminus \{s\})$, $f(s') \neq f(s)$.
3. For each transition $(s, x/y, s') \in tr(M)$:
   i. w will check that $Z_t \subseteq ios(Q, w_s)$.
   ii. Hence, w will check not only that $st(Q, w_s) = f(s)$, but also that $st(Q, w_s xy) = f(s')$.
   iii. Hence, w will check that $tr(Q)$ comprises the transition $(f(s), x/y, f(s'))$.
4. By M and Q deterministic, strongly connected and input-equivalent, w will, hence, check that all the following is true:
   (1) $st(Q) = \{f(s) | s \in S\}$
   (2) $tr(Q) = \{(f(s), x/y, f(s')) | (s, x/y, s') \in tr(M)\}$
5. Hence, w will check that for each state $s \in S$, $ios(f(s)) = ios(s)$.
6. Hence if there is a state $s \in (S \setminus \{init(M)\})$ with $in(s, \varepsilon) = in(M, \varepsilon)$, w will check that w_0 is a unique IOS of $f(init(M))$.
7. w_0 will be a prefix of w.
8. Hence, by $in(Q, \varepsilon) = in(M, \varepsilon)$, w will check that $init(Q) = f(init(M))$.

9. Hence, w will check that $ios(Q) = ios(M)$, meaning that w will be a CS.

The plan behind the steps 5 and 6 refines Plan 1 and is as follows:

Plan 2

1. For each state $s \in (S \setminus S')$:
   i. For each IOS $z \in Z_s$, w will have a prefix $w_{s,z}z$ with $st(M, w_{s,z}) = s$.
   ii. w will check that the state $st(Q, w_{s,z})$ is the same for every IOS $z \in Z_s$.
   iii. For each IOS $z \in Z_s$, w will, hence, check that $Z_s \subseteq ios(Q, w_{s,z})$.
   iv. Hence, one safely assumes $w_s \in \{w_{s,z} | z \in Z_s\}$.

   In other words, for each extension $z \in Z_s$ with $w_{s,z} \neq w_s$ which Plan 1 virtually foresees for the test w_s, Plan 2 virtually foresees the following instance of Transformation 1: While securing the checking of $st(Q, w_{s,z}) = st(Q, w_s)$, z is detached from w_s and instead appended to the test $w_{s,z}$.
2. For each transition $t = (s, x/y, s')$ in $tr(M)$ and each IOS $z \in Z'_t$:
   i. w will have a prefix $w_{t,z}$ with $st(M, w_{t,z}) = s$.
   ii. w will check that $Z_{t,z} \subseteq ios(Q, w_{t,z})$.
   iii. By $Z_{t,z} \subseteq pf(Z_s)$, w will check $Z_{t,z} \subseteq ios(Q, w_s)$.
   iv. By $[D_s]_{s\in S}$ a collection of harmonized state identifiers and by Proposition 2, w will, hence, check that $(z \in ios(Q, w_s)) \Leftrightarrow (z \in ios(Q, w_{t,z}))$.
   v. w will check that $z \in ios(Q, w_{t,z})$. Hence, w will check that $z \in ios(Q, w_s)$.

   In other words, for each extension $z \in Z'_t$ which Plan 1 virtually foresees for the test w_s, Plan 2 virtually foresees the following instance of Transformation 2: While securing the checking of $(z \in ios(Q, w_s)) \Leftrightarrow (z \in ios(Q, w_{t,z}))$, z is detached from w_s and instead appended to the test $w_{t,z}$.
3. For each transition $t = (s, x/y, s')$ in $tr(M)$, assuming that w will check $(f(q), x'/y', f(q')) \in tr(Q)$ for every transition $t' = (q, x'/y', q')$ in $tr(M)$ with $t' \prec t$:
   i. For each IOS $z \in (Z_t \setminus Z'_t)$:
      I. w will have a prefix $w_{t,z}v_{t,z}$ with $st(M, w_{t,z}) = q_{t,z}$.
      II. w will check that $st(Q, w_{t,z}) = f(q_{t,z})$.
      III. By $t' \prec t$ for every transition $t' \in ts(q_{t,z}v_{t,z})$, w will, hence, check that $st(Q, w_{t,z}v_{t,z}) = f(s)$.
      IV. Hence, w will check that $(st(Q, w_{t,z}v_{t,z}) = st(Q, w_s)) \wedge (z \in ios(Q, w_{t,z}v_{t,z}))$.
      V. Hence, w will check that $z \in ios(Q, w_s)$.

      In other words, for each extension $z \in (Z_t \setminus Z'_t)$ which Plan 1 virtually foresees for the test w_s, Plan 2 virtually foresees the following instance of Transformation 1: While securing the checking of $st(Q, w_{t,z}v_{t,z}) = st(Q, w_s)$, z is detached from w_s and instead appended to the test $w_{t,z}v_{t,z}$.
   ii. Hence, w will check that $(Z_s \cup Z_t) \subseteq ios(Q, w_s)$.

   iii. Hence, w will check that $(f(s), x/y, f(s')) \in tr(M)$.
4. By $\prec$ a strict order on $tr(M)$, w will, hence, check that for each transition $t = (s, x/y, s')$ in $tr(M)$, $Z_t \subseteq ios(Q, w_s)$.

The plan behind the step 7 refines Plan 2 and is as follows:

Plan 3

1. w will cover P.
2. For each state $s \in S'$:
   i. w will have a prefix $u_s v_s v'_s$ with $st(M, u_s) = r_s$.
   ii. By (r_s, v_s, v'_s) a (P, s, Z_s)-locator, w will check that $Z_s \subseteq ios(Q, u_s v_s)$.
   iii. Hence, one safely assumes $w_s = u_s v_s$.
3. For each state $s \in (S \setminus S')$:
   i. For each IOS $z \in Z_s$, w will have a prefix $u_{s,z} v_s z$ with $st(M, u_{s,z}) = r_s$.
   ii. By (r_s, v_s) a (P, s)-locator, w will check that the state $st(Q, u_{s,z} v_s)$ is the same for every IOS $z \in Z_s$.
   iii. Hence, one safely assumes $w_{s,z} = u_{s,z} v_s$.
4. For each transition $t = (s, x/y, s')$ in $tr(M)$ and each IOS $z \in Z'_t$:
   i. w will have a prefix $u_{t,z} v_{t,z} v'_{t,z}$ with $st(M, u_{t,z}) = r_{t,z}$.
   ii. By $(r_{t,z}, v_{t,z}, v'_{t,z})$ a $(P, s, max(\{z\} \cup Z_{t,z}))$-locator, w will check that $(\{z\} \cup Z_{t,z}) \subseteq ios(Q, u_{t,z} v_{t,z})$.
   iii. Hence, one safely assumes $w_{t,z} = u_{t,z} v_{t,z}$.
5. For each transition $t = (s, x/y, s')$ in $tr(M)$ and each IOS $z \in (Z_t \setminus Z'_t)$:
   (i) w will have a prefix $u_{t,z} v_{q_{t,z}} v_{t,z} z$ with $st(M, u_{t,z}) = r_{q_{t,z}}$.
   (ii) By $(r_{q_{t,z}}, v_{q_{t,z}})$ the selected $(P, q_{t,z})$-locator, w will, hence, check that $st(Q, u_{t,z} v_{q_{t,z}}) = st(Q, w_{q_{t,z}})$.
   (iii) As w will check that $st(Q, w_{q_{t,z}}) = f(q_{t,z})$, it will, hence, check that $st(Q, u_{t,z} v_{q_{t,z}}) = f(st(Q, u_{t,z} v_{q_{t,z}}))$.
   (iv) Hence, one safely assumes $w_{t,z} = u_{t,z} v_{q_{t,z}}$.

Theorem 4 *In the new method, w is a CS,*

Proof. w starts with w_0 and covers P. Hence:
1. For each state $s \in S'$, it has a prefix qualifying for the u_s in Plan 3.
2. For each state $s \in (S \setminus S')$, it has a prefix qualifying for the $u_{s,z}$ in Plan 3.
3. For each transition $t \in tr(M)$ and each IOS $z \in Z_t$, it has a prefix qualifying for the $u_{t,z}$ in Plan 3. $\square$

The plan behind the default procedure for the step 7 in the case of $m \geq 2n$ (Figure 2) is as follows:

Plan 4

1. For each state $s \in (S' \cup \{q_{s'} | s' \in (S \setminus S')\})$:
   i. (q'_s, z_s, z'_s) will be an $(\emptyset, s, Z_s)$-locator and, hence, a (P, s, Z_s)-locator.
   ii. By $m \geq 2n$, $end(st(s, z) | z \in ios_{\leq m-n}(s)) = S$.
   iii. Hence if there is a test covering (q'_s, z_s, z'_s):
      I. For each state $s' \in S$, the test has a prefix $z''_{s'}$ for which it checks that in the OFSM Q under test, $D_{s'} \subseteq ios(Q, z''_s)$.
      II. By $[D_{s'}]_{s' \in S}$ a collection of harmonized state identifiers and by Proposition 1, the test, hence, checks that $end(Q, ios(Q) \cap ios(M))$ comprises at most one state s' with $Z_s \subseteq ios(s')$.
   iv. Hence, $(q'_s, z_s z'_s)$ will be an $(\emptyset, st(s, z'_s))$-locator.
2. For each state $s \in (S \setminus S')$, (r_s, v_s) will, hence, be an $(\emptyset, s)$-locator and, hence, a (P, s)-locator.
3. For each transition $t = (s, x/y, s')$ in $tr(M)$ and each ios $z \in Z'_t$, $(r_{t,z}, v_{t,z}, v'_{t,z})$ will be an $(\emptyset, s, max(\{z\} \cup Z_{t,z}))$-locator and, hence, a $(P, s, max(\{z\} \cup Z_{t,z}))$-locator.

The default procedure for constructing an $(\emptyset, s, Z)$-locator (Figure 3) is the one which [Inan and Ural 1999] proves for the purpose.

7 Estimation of the possible CS length reduction

In the new method, $ln(w)$ strongly depends on the extent to which M is complete and on the employed $[D_s]_{s \in S}$. In [Inan and Ural 1999], M is assumed to be complete and CS construction for the general m is presented in detail only for the case where the input sequence set $\{is(z) | z \in D_s\}$ is the same for every state $s \in S$. Assuming such an $\{is(z) | z \in D_s\}$ and completeness of M, we in the following assess the ratio between the $ln(w)$ obtained if the new method is executed with $(S \setminus S')$ and all $(Z_t \setminus Z'_t)$ empty and the $ln(w)$ obtained if the new method is executed, as originally the only option, with S' and all Z'_t empty.

For each state $s \in S$ and each non-empty IOS set $Z \subseteq ios(s)$, let $l(s, Z)$ denote $ln(v_k z_k)$ of the shortest IOS $v_k z_k$ which the procedure in Figure 3 can construct for them. In the rest of the section, we assume also the following:

(1) $m \geq 2n$
(2) The step 7 of the new method is executed as the default version of the procedure in Figure 2, i.e., as a procedure calling the one in Figure 3.
(3) In every in instance of the procedure in Figure 3, $ln(v_k z_k) = l(s, Z)$.
(4) For each state $s \in S$:
   i. As $|D_s|$ need not be more than $(n-1)$, this is its assumed upper bound.
   ii. As $ln(z)$ of individual IOSs $z \in D_s$ need not be more than $(n-1)$, this is its assumed upper bound.

(5) Because in the case of no neglectably short unique IOS for $init(M)$, one usually considers sufficient that w covers P, i.e., checks that each transition in $tr(M)$ is correctly implemented, we assume that the step 8 of the new method is, like in [Inan and Ural 1999], executed without securing that w starts with w_0.
(6) For each gap z between consecutive special-purpose segments combined in w, as $ln(z)$ need not be more than $(n-1)$, this is its assumed upper bound.
(7) Because for the possible overlapping of the segments, [Inan and Ural 1999] reports experiments showing that the resulting reduction of $ln(w)$ is 10-15%, this is its assumed range.

Under the above assumptions:

1. Let a denote $|in(M)|$. Let b denote $|Z_{init(M)}|$.
2. Let l_{min} denote $inf(\{l(s,Z_s)|s \in S\})$. Let l_{max} denote $sup(\{l(s,Z_s)|s \in S\})$.
3. Let l'_{max} denote $sup(\{l(s, max(\{z\} \cup Z_{t,z}))|(t \in tr(M)) \wedge (z \in Z'_t)\})$.
4. For each state $s \in S$:
   i. For each IOS $z \in Z_s$: $m-n \leq ln(z) \leq m-1$
   ii. $(|Z_s| = b) \wedge (a^{m-n} \leq b \leq a^{m-1})$
5. Hence:
   i. $l_{min} \geq (m-n) \cdot (m+1)^{a^{m-n}-1}$
   ii. $(m-n) \cdot (m+1)^{a^{m-n}-1} \leq l_{max} \leq l_{min} \cdot \frac{m+n}{m-n}$
6. For each transition $t \in tr(M)$ and each IOS $z \in Z'_t$:
   i. For each IOS $z' \in max(\{z\} \cup Z_{t,z})$: $ln(z') \leq m$
   ii. $|max(\{z\} \cup Z_{t,z})| \leq (1+m^{m-n}) \cdot (n-1)$
7. Hence: $l'_{max} \leq m \cdot (m+1)^{(1+m^{m-n})\cdot(n-1)-1} \leq (m+1)^{(1+m^{m-n})\cdot(n-1)}$
8. In the case of $(S \setminus S')$ and all $(Z_t \setminus Z'_t)$ empty:
   $ln(w) \leq 0.9 \cdot ((\Sigma_{s\in S'}(ln(v_s v'_s)+n)) + (\Sigma_{t\in tr(M), z\in Z'_t}(ln(v_{t,z}v'_{t,z})+n)))$
   $\leq 0.9 \cdot ((\Sigma_{s\in S'}(l_{max}+n)) + (\Sigma_{t\in tr(M), z\in Z'_t}(l'_{max}+n)))$
   $\leq 0.9 \cdot (n \cdot (l_{max}+n) + (l'_{max}+n) \cdot (\Sigma_{t\in tr(M)}|Z_t|))$
   $\leq 0.9 \cdot (n \cdot (l_{max}+n) + (l'_{max}+n) \cdot (\Sigma_{t\in tr(M)} b))$
   $\leq 0.9 \cdot n \cdot (l_{max}+n+a \cdot b \cdot (l'_{max}+n))$
9. In the case of S' and all Z'_t empty:
   $ln(w) \geq 0.85 \cdot ((\Sigma_{s\in(S\setminus S'), z\in Z_s} ln(v_s z)) + (\Sigma_{t\in tr(M), z\in(Z_t\setminus Z'_t)} ln(v_{q_{t,z}} v_{t,z} z)))$
   $\geq 0.85 \cdot ((\Sigma_{s\in S, z\in Z_s} l_{min}) + (\Sigma_{t\in tr(M), z\in Z_t} l_{min}))$
   $\geq 0.85 \cdot l_{min} \cdot ((\Sigma_{s\in S}|Z_s|) + (\Sigma_{t\in tr(M)}|Z_t|))$
10. Let R denote the ratio $\frac{0.9\cdot n\cdot(l_{max}+n+a\cdot b\cdot(l'_{max}+n))}{0.85\cdot l_{min}\cdot((\Sigma_{s\in S}|Z_s|)+(\Sigma_{t\in tr(M)}|Z_t|))}$
11. $R \leq \frac{0.9\cdot(l_{max}+n+a\cdot b\cdot(l'_{max}+n))}{0.85\cdot l_{min}\cdot a^{m-n+1}}$
12. $\frac{a\cdot b\cdot(l'_{max}+n)}{l_{max}+n} \leq \frac{a^m\cdot((m+1)^{(1+m^{m-n})\cdot(n-1)}+n)}{(m-n)\cdot(m+1)^{a^{m-n}-1}+n}$
13. In the limit for increasingly large (a/m) and (m/n):

i. $\frac{a^m\cdot((m+1)^{(1+m^{m-n})\cdot(n-1)}+n)}{(m-n)\cdot(m+1)^{a^{m-n}-1}+n} \approx \frac{a^m\cdot(m+1)^{(1+m^{m-n})\cdot(n-1)}}{(m-n)\cdot(m+1)^{a^{m-n}-1}}$
$\approx \frac{a^m\cdot(m+1)^{(1+m^{m-n})\cdot(n-1)}}{(m+1)^{a^{m-n}}} \approx 0$

ii. Hence: $\frac{a\cdot b\cdot(l'_{max}+n)}{l_{max}+n} \approx 0$

iii. Hence: $\frac{0.9\cdot(l_{max}+n+a\cdot b\cdot(l'_{max}+n))}{0.85\cdot l_{min}\cdot a^{m-n+1}} \approx \frac{0.9\cdot(l_{max}+n)}{0.85\cdot l_{min}\cdot a^{m-n+1}}$

iv. $\frac{n}{l_{max}} \leq \frac{n}{l_{min}} \leq \frac{n}{(m-n)\cdot(m+1)^{a^{m-n}-1}}$

v. $\frac{n}{(m-n)\cdot(m+1)^{a^{m-n}-1}} \approx 0$

vi. Hence: $\frac{0.9\cdot(l_{max}+n)}{0.85\cdot l_{min}\cdot a^{m-n+1}} \approx \frac{0.9\cdot l_{max}}{0.85\cdot l_{min}\cdot a^{m-n+1}}$

vii. $\frac{0.9\cdot l_{max}}{0.85\cdot l_{min}\cdot a^{m-n+1}} \leq \frac{0.9\cdot(m+n)}{0.85\cdot(m-n)\cdot a^{m-n+1}}$

viii. $\frac{0.9\cdot(m+n)}{0.85\cdot(m-n)\cdot a^{m-n+1}} \approx \frac{0.9}{0.85\cdot a^{m-n+1}}$

ix. Hence, the order of R is at most $\mathcal{O}(a^{-(m-n+1)})$.

x. Hence, w obtained with $(S \setminus S')$ and all $(Z_t \setminus Z'_t)$ empty is shorter than w obtained with S' and all Z'_t empty, by a factor whose order is at least $\mathcal{O}(a^{m-n+1})$.

8 Conclusions

For the optimization of OFSM-based test sets, we have discussed a popular transformation by which under a certain precondition, a tail of a test is removed and appended to some other test. We have proposed a weaker precondition under which the transformation remains fault-coverage-preserving. Along with a weaker precondition, we have proposed some weaker sufficient conditions for its satisfaction. For practice this means that for any planned move of a test tail, the tests introduced for checking that this is safe can now be simpler.

The latter can be useful particularly in the construction of CSs, which we have demonstrated on the CS construction method of [Inan and Ural 1999]. With the help of the new insights, we have generalized it to incomplete OFSMs and furnished it with new options which can sometimes reduce the CS length by a factor exponential in the assumed upper bound on the number of extra states in the implementation. For complete OFSMs and the upper bound at least the size of the state set of the specification OFSM (such a large upper bound is currently allowed only in the method of [Inan and Ural 1999] and its proposed generalization), this has been formally proven. For the general case, guidelines for optimally choosing the parameters of the generalized method are yet to be developed.

In the considered transformation and in most of the current OFSM-based test set construction methods with optimization, one reasons in terms of (quasi-) equivalence or equality of states. With our generalization of the transformation and one of the methods, we have introduced the reasoning in terms of a weaker and parameterized relation between states. It would be interesting to see how much (and at what cost) the switching to the new kind of reasoning can improve the other methods.

For the reasoning in terms of (quasi-)equivalence or equality of states, one has at disposal a formal apparatus consisting of items such as templates for deduction steps in the interpretation of IOSs observed or expected to be observed on the OFSM under test (see, for example, [Simão and Petrenko 2010, Bonifácio et al. 2012, Kapus-Kolar 2012a]) and sufficient conditions for various test set properties (see, for example, [Bonifácio et al. 2012, Simão et al. 2012]). Of the apparatus, we have for the new kind of reasoning adapted (in Theorem 3 and Proposition 4) only two deduction step templates. It would be interesting to adapt also the rest.

We have assumed that to pass a test, the OFSM under test has to implement it exactly like the specification OFSM. It would be interesting to generalize the new insights and their applications to a wider class of conformance relations, for example the one introduced in [Kapus-Kolar 2016].

Acknowledgement

The work was supported by the Slovenian Research Agency under the research programme P2-0095 Parallel and Distributed Systems.

References

[Bonifácio et al. 2012] Bonifácio, A.L., Vieira Moura, A., Simão, A.: "Model partitions and compact test case suites", Int. J. Found. Comput. Sci., 23, 1 (Jan 2012) 147-172.

[Braun and Givone 1979] Braun, R.D., Givone, D.D.: "An improved algorithm for deriving checking experiments", IEEE Trans. Comput., C-28, 2 (Feb 1979) 153-156.

[Braun and Givone 1981] Braun, R.D., Givone, D.D.: "A generalized algorithm for constructing checking sequences", IEEE Trans. Comput., C-30, 2 (Feb 1981) 141-144.

[Chen et al. 2005] Chen, J., Hierons, R.M., Ural, H., Yenigün, H.: "Eliminating redundant tests in a checking sequence", Proc. TestCom'05, LNCS 3502, Springer-Verlag, Berlin, Germany (2005), 146-158.

[Cutigi et al. 2016] Cutigi, J.F., Simão, A., Souza S.R.S.: "Reducing FSM-based test sets with guaranteed fault coverage", Comput. J., Advanced access, 2016.

[Dincturk 2009] Dincturk, M.E.: "A Two Phase Approach for Checking Sequence Generation", M.Sc. Thesis, Sabancı University (2009).

[Duan and Chen 2009] Duan, L., Chen, J.: "Exploring alternatives for transition verification", J. Syst. Soft., 82, 9 (Sept 2009) 1388-1402.

[Endo and Simão 2013] Endo, A.T., Simão, A.: "Evaluating test set characteristics, cost, and effectiveness of FSM-based testing methods", Inf. Softw. Technol., 55, 6 (June 2013) 1045-1062.

[Farmer 1973] Farmer, D.E.: "Algorithms for designing fault-detection experiments for sequential machines", IEEE Trans. Comput., C-22, 2 (Feb 1973) 159-167.

[Hennie 1964] Hennie, F.C.: "Fault detecting experiments for sequential circuits", Proc. SWCT'64, Princeton, NJ, USA (1964), 95-110.

[Hierons 2004] Hierons, R.M.: "Using a minimal number of resets when testing from a finite state machine", Inf. Proc. Let., 90, 6 (June 2004) 287-292.

[Hierons and Ural 2002] Hierons, R.M., Ural, H.: "Reduced length checking sequences", IEEE Trans. Comput., 51, 9 (Sept 2002) 1111-1117.

[Hierons and Ural 2003] Hierons, R.M., Ural, H.: "UIO sequence based checking sequences for distributed test architectures", Inf. Softw. Technol., 45, 12 (Sept 2003) 793-803.

[Hierons and Ural 2006] Hierons, R.M., Ural, H.: "Optimizing the length of checking sequences", IEEE Trans. Comput., 55, 5 (May 2006) 618-629.

[Hierons and Ural 2008] Hierons, R.M., Ural, H.: "Checking sequences for distributed architectures", Distrib. Comput., 21, 3 (Sept 2008) 223-238.

[Hierons and Ural 2010] Hierons, R.M., Ural, H.: "Generating a checking sequence with a minimum number of reset transitions", Aut. Soft. Eng., 17, 3 (Sept 2010) 217-250.

[Hierons et al. 2008] Hierons, R.M., Jourdan, G.-V., Ural, H., Yenigün, H.: "Using adaptive distinguishing sequences in checking sequences", Proc. SAC'08, ACM (2008), 682-687.

[Hsieh 1971] Hsieh, E.P.: "Checking experiments for sequential machines", IEEE Trans. Comput., C-20, 10 (Oct 1971) 1152-1166.

[Inan and Ural 1999] Inan, K., Ural, H.: "Efficient checking sequences for testing finite state machines", Inf. Softw. Technol., 41, 11-12 (Sept 1999) 799-812.

[Jourdan 2015] Jourdan, G.-V.: "Reduced checking sequences using unreliable reset", Inf. Process. Lett., 115, 5 (May 2015) 532-535.

[Kapus-Kolar 2012a] Kapus-Kolar, M.: "New state-recognition patterns for conformance testing of finite state machine implementations", Comput. Stand. Int., 34, 4 (June 2012) 390-395.

[Kapus-Kolar 2012b] Kapus-Kolar, M.: "On "Exploring alternatives for transition verification"", J. Syst. Soft., 85, 8 (Aug 2012) 1744-1748.

[Kapus-Kolar 2014] Kapus-Kolar, M.: "On the global optimization of checking sequences for finite state machine implementations", Microprocess. Microsys., 38, 3 (May 2014) 208-215.

[Kapus-Kolar 2016] Kapus-Kolar, M.: "Improved state-counting-based construction of complete test sets for FSM implementations", Elektrotehniški vestnik/ Electrotechnical Review, 83, 3 (2016).

[Petrenko et al. 2012] Petrenko, A., Simão, A., Yevtushenko, N.: "Generating checking sequences for nondeterministic finite state machines", Proc. ICST'12, IEEE CS (2012), 310-319.

[Porto et. al. 2013] Porto, F.R., Endo, A.T., Simão, A.: "Generation of checking sequences using identification sets", Proc. ICFEM'13, LNCS 8144, Springer-Verlag, Berlin, Germany (2013), 115-130.

[Rezaki and Ural 1995] Rezaki, A., Ural, H.: "Construction of checking sequences based on characterization sets", Comput. Commun., 18, 12 (Dec 1995) 911-920.

[Simão and Petrenko 2008] Simão, A., Petrenko, A.: Generating checking sequences for partial reduced finite state machines", Proc. TestCom/FATES'08, LNCS 5047, Springer-Verlag, Berlin, Germany (2008), 153-168.

[Simão and Petrenko 2009] Simão, A., Petrenko, A.: "Checking sequence generation using state distinguishing subsequences", Proc. ICST'09 Workshops, IEEE CS (2009), 48-56.

[Simão and Petrenko 2010] Simão A, Petrenko A.: "Fault coverage-driven incremental test generation", Comput. J., 53, 9 (Nov 2010) 1508-1522.

[Simão et al. 2012] Simão, A., Petrenko, A., Yevtushenko, N.: "On reducing test length for FSMs with extra states"; Softw. Test. Verif. Rel., 22, 6 (Sept 2012) 435-454.

[Tekle et al. 2005] Tekle, K.T., Ural, H., Yalcin, M.C., Yenigün, H.: "Generalizing redundancy elimination in checking sequences", Proc. ISCIS'05, LNCS 3733, Springer-Verlag, Berlin, Germany (2005), 915-925.

[Ural et al. 1997] Ural, H., Wu, X., Zhang, F.: "On minimizing the length of checking sequences", IEEE Trans. Comput., 46, 1 (Jan 1997) 93-99.

[Ural and Williams 2006] Ural, H., Williams, C.: "Constructing checking sequences for distributed testing", Form. Asp. Comput., 18, 1 (March 2006) 84-101.

[Ural and Zhang 2006] Ural, H., Zhang, F.: "Reducing the length of checking sequences by overlapping", Proc. TestCom'06, LNCS 3964, Springer-Verlag, Berlin, Germany (2006), 274-288.

[Yalcin and Yenigün 2006] Yalcin, M.C., Yenigün, H.: "Using distinguishing and UIO sequences together in a checking sequence", Proc. TestCom'06, LNCS 3964, Springer-Verlag, Berlin, Germany (2006), 259-273.

[Yao et al. 1993] Yao, M., Petrenko, A., von Bochmann, G.: "Conformance testing of protocol machines without reset", Proc. PSTV'93, IFIP Trans. C-16, North-Holland, Amsterdam (1993), 241-256.

Appendix

Proof of Theorem 1.

1. Let Z' denote the test set $(Z \setminus \{zz'\}) \cup \{z\}$.
2. Let Z'' denote the test set $Z' \cup \{z''z'\}$.
3. By $st(M, z'') \approx_{st(M,z),z'} st(M, z)$, $st(M, z'') \equiv_{z'} st(M, z)$.
4. For each OFSM $Q \in \mathcal{I}$ with $Q \equiv_Z M$:
   i. $(Q \equiv_{Z'} M) \wedge (st(Q, z) \equiv_{z'} st(M, z))$
   ii. Hence, $st(Q, z'') \approx_{st(M,z),z'} st(Q, z)$. Hence, $st(Q, z'') \equiv_{z'} st(M, z)$.
   iii. Hence, $st(Q, z'') \equiv_{z'} st(M, z'')$. Hence, $Q \equiv_{Z''} M$.
5. For each OFSM $Q \in \mathcal{I}$ with $Q \not\equiv_Z M$:
   i. If $Q \not\equiv_{Z'} M$ then $Q \not\equiv_{Z''} M$.
   ii. If $Q \equiv_{Z'} M$:
      I. $(st(Q, z) \not\equiv_{z'} st(M, z)) \wedge (st(Q, z'') \approx_{st(M,z),z'} st(Q, z))$.
      II. Hence, $st(Q, z'') \not\equiv_{z'} st(M, z)$.
      III. Hence, $st(Q, z'') \not\equiv_{z'} st(M, z'')$. Hence, $Q \not\equiv_{Z''} M$. □

Proof of Theorem 2.

1. For each IOS $z \in ios(Q)$ and natural i, let $f_i(z)$ denote the set $\{(z', f(zz'))|(z' \in ios(Q, z)) \wedge (ln(z') \leq i)\}$.
2. For each natural i, let Π_i denote the set of the sets into which the IOSs in $ios(Q)$ are partitioned according to f_i, with $|\Pi_0| = |\{f(z)|z \in ios(Q)\}|$.
3. Let k' denote $|st(Q)| - |\Pi_0| - \Delta + 1$.
4. If $\Delta = |st(Q)| - |\Pi_0| - k + 1$, then $k' = k$ and, hence, $k' > 0$.
5. If $\Delta = 0$, then $k' = |st(Q)| - |\Pi_0| + 1$ and, hence, $k' \leq k$ and, by (1), $k' > 0$.
6. By the Lemma 1 below, $|\Pi_i|$ strictly increases with i until it, as it can be at most $|st(Q)|$, stabilizes to a certain value v.

7. As for every z in $ios(Q)$, $f_\infty(z)$ provides precise information on $ios(Q,z)$, $v = |st(Q)|$ and, hence, $|\Pi_i| \geq inf(\{|st(Q)|, |\Pi_{i-1}|+1\})$ for every $i > 0$.
8. Hence, by $\Delta \geq 0$ and induction on i, $|\Pi_\Delta| \geq inf(\{|st(Q)|, |\Pi_0| + \Delta\})$.
9. Hence, $|\Pi_\Delta| \geq inf(\{|st(Q)|, |st(Q)| - k' + 1\})$.
10. Hence, by $k' > 0$, $|\Pi_\Delta| \geq |st(Q)| - k' + 1$.
11. By (2), $|\{f_\Delta(z)|z \in Z\}| = 1$ and, hence, Π_Δ comprises a set Z' with $Z \subseteq Z'$.
12. Suppose that $|end(Q,Z)| > k'$ and, hence, $|end(Q,Z')| > k'$.
13. Take any set Z'' consisting of the members of Z' and one member of each of the sets in $\Pi_\Delta \setminus \{Z'\}$.
14. By (1), for each IOS pair $\{z,z'\} \subseteq ios(Q)$ with z and z' belonging to different sets in Π_Δ, $st(Q,z) \neq st(Q,z')$.
15. Hence, by $|end(Q,Z')| > k'$, $|end(Q,Z'')| > k' + |\Pi_\Delta| - 1$.
16. Hence, by $|\Pi_\Delta| \geq |st(Q)| - k' + 1$, $|end(Q,Z'')| > |st(Q)|$, which is a contradiction. □

Lemma 1 *In the setting of the proof of Theorem 2, if $|\Pi_{i+1}| = |\Pi_i|$ for a natural i then $|\Pi_{i+2}| = |\Pi_{i+1}|$.*

Proof.

1. Suppose that $|\Pi_{i+1}| = |\Pi_i|$ and, hence, $\Pi_{i+1} = \Pi_i$.
2. Suppose that $|\Pi_{i+2}| \neq |\Pi_{i+1}|$ and, hence, $\Pi_{i+2} \neq \Pi_{i+1}$.
3. By $\Pi_{i+2} \neq \Pi_{i+1}$, $ios(Q)$ has a subset $\{zxyz'', z'xyz''\}$ with the following properties:
   i. z, z' and z'' are IOSs, whereas xy is an IO.
   ii. $(ln(z'') = i) \wedge (f_{i+1}(z) = f_{i+1}(z')) \wedge (f_1(zxyz'') \neq f_1(z'xyz''))$
   iii. Hence, $f_{i+1}(zxy) \neq f_{i+1}(z'xy)$.
   iv. By $\Pi_{i+1} = \Pi_i$, hence, $f_i(zxy) \neq f_i(z'xy)$.
   v. Hence, $f_{i+1}(z) \neq f_{i+1}(z')$, which is a contradiction. □

Proof of Proposition 1.

1. Take any IOS set $Z' \subseteq Z$ with $|Z'| = |end(M,Z)| = |end(M,Z')|$.
2. For each IOS $z'' \in ios(Q)$ and each IOS $z''' \in Z'$, let $f'(z'',z''')$ denote the Boolean telling whether there is an IOS $z_1 \in Z$ with $(st(Q,z_1) = st(Q,z'')) \wedge (st(M,z_1) \neq st(M,z''')) \wedge (st(Q,z_1) \neq st(Q,z'''))$.
3. For each IOS $z'' \in ios(Q)$, let $f(z'')$ denote the collection $[f'(z'',z''')]_{z''' \in Z'}$.
4. For each IOS pair $\{z'',z'''\} \subseteq ios(Q)$ with $st(Q,z'') = st(Q,z''')$, $f(z'') = f(z''')$.
5. Let Δ denote $(|st(Q)| - |\{f(z'')|z'' \in ios(Q)\}|)$.
6. Hence, by M and Q deterministic and input-equivalent and by $(st(M,z) = st(M,z')) \wedge (Z \subseteq ios(Q))$:

i. By (1), $ios_{\leq m-|end(M,Z)|}(st(Q,z)) = ios_{\leq m-|end(M,Z)|}(st(Q,z'))$.
ii. By (2), for each IOS $z'' \in ios_{\leq m-|end(M,Z)|}(st(Q,z))$, $f(zz'') = f(z'z'')$.
iii. By (2), $|\{f(z'')|z'' \in ios(Q)\}| \geq |end(M,Z)|$.
iv. Hence, by $|st(Q)| \leq m$, $m - |end(M,Z)| \geq \Delta$.

7. By Theorem 2 for $k = 1$, hence, $st(Q,z) = st(Q,z')$. □

Definition 18 A component of an undirected graph G is a connected subgraph of G that is not a subgraph of any larger such subgraph.

Proof of Theorem 3.

1. Let F denote the set of all IOSs $z''xy \in pf(Z)$ with $(st(Q,z) \equiv_{z''} st(Q,z') \equiv_{z''} Z) \wedge (st(Q,z) \not\approx_{Z,z''xy} st(Q,z'))$.
2. Suppose that the condition claimed sufficient is satisfied, but $st(Q,z) \not\approx_Z st(Q,z')$. Hence, $F \neq \emptyset$.
3. Let ϕ be one of the shortest IOSs in F.
4. Take any IOS $\beta \in notall(imp(Z,Z'),Z)$ with $H_\beta \subseteq pf(\phi)$.
5. By the Lemma 2 below, $|end(Q, K \otimes H_{\beta,h})| > |H_{\beta,h}|$ for every $1 \leq h \leq k_\beta$.
6. By (d), hence, $|end(Q, K \otimes H_\beta)| \geq |H_\beta| + k_\beta$.
7. By (e), hence, $|end(Q, L_\beta)| + |end(Q, K \otimes H_\beta)| > |end(Q, L_\beta \cup (K \otimes H_\beta))|$.
8. Hence, $end(Q, L_\beta) \cap end(Q, K \otimes H_\beta) \neq \emptyset$, which contradicts (f). □

Lemma 2 *In the setting of the proof of Theorem 3,* $|end(Q, K \otimes H_{\beta,h})| > |H_{\beta,h}|$ *for each* $1 \leq h \leq k_\beta$.

Proof. For each $1 \leq h \leq k_\beta$:

1. Start regarding the pairs (α, z_i) with $(\alpha \in K) \wedge (z_i \in H_{\beta,h})$ as the vertices of an undirected graph G in which any two vertices (α, z_i) and (α', z_j) are connected by an edge exactly if $((\alpha, z_i) \neq (\alpha', z_j)) \wedge (st(Q, \alpha z_i) = st(Q.\alpha' z_j)$.
2. Individual components of G, hence, represent individual states in $end(Q, K \otimes H_{\beta,h})$.
3. For given components G_u and G_v and IOS $z_i \in H_{\beta,h}$, let $G_u \sim_{z_i} G_v$ denote that there is an $\{\alpha, \alpha'\} = K$ with (α, z_i) a vertex of G_u and (α', z_i) a vertex of G_v.
4. For each IOS $z_i \in H_{\beta,h}$:
   i. By $st(Q,z) \not\approx_{Z,\phi} st(Q,z')$, the vertices (z, z_i) and (z', z_i) belong to two different components.
   ii. For this and only this pair of components, z_i is, hence, a kind of a connector.
5. Start regarding the connectors as the edges of an undirected graph G' in which individual vertices represent individual components of G.
6. By the Lemma 3 below, G' is acyclic. Hence, G' has more vertices than edges.
7. Hence, G has more than $|H_{\beta,h}|$ components.

8. Hence, $|end(Q, K \otimes H_{\beta,h})| > |H_{\beta,h}|$. $\square$

Lemma 3 *In the setting of the proof of Lemma 2, $\vee_{1 \leq i \leq k}(G_i \not\sim_{z_i} G_{i \bmod k+1})$ for each pair $([G_i]_{1 \leq i \leq k}, [z_i]_{1 \leq i \leq k})$ satisfying $k \geq 2$ and all the following:*

(1) $[G_i]_{1 \leq i \leq k}$ is a collection of k different components of G.

(2) $[z_i]_{1 \leq i \leq k}$ is a collection of k different IOSs in $H_{\beta,h}$.

Proof. For each such pair $([G_i]_{1 \leq i \leq k}, [z_i]_{1 \leq i \leq k})$:

1. For each $1 \leq i \leq k$, let i' denote $(i \bmod k + 1)$.
2. Suppose that $\forall 1 \leq i \leq k : (G_i \sim_{z_i} G_{i'})$.
3. Take the IOS z'' with $\phi = z_k z''$.
4. For each $1 \leq i \leq k$, there is, hence, an $\{\alpha_i, \alpha'_i\} = K$ with (α_i, z_i) a vertex of G_i and (α'_i, z_i) a vertex of $G_{i'}$.
5. Without loss of generality, assume $(\forall 1 \leq i < k : (z_i < z_k)) \wedge (st(Q, \alpha_k z_k) \not\equiv_{z''} ios(Z, z_k))$.
6. For each $1 \leq i \leq k$:
   i. (α'_i, z_i) and $(\alpha_{i'}, z_{i'})$ are vertices of $G_{i'}$.
   ii. Hence, $st(Q, \alpha'_i z_i) = st(Q, \alpha_{i'} z_{i'})$.
   iii. Hence, $(st(Q, \alpha_{i'} z_{i'}) \not\equiv_{z''} ios(Z, z_k)) \Rightarrow (st(Q, \alpha'_i z_i) \not\equiv_{z''} ios(Z, z_k))$.
7. For each $1 \leq i < k$:
   i. Suppose that $(st(Q, \alpha'_i z_i) \not\equiv_{z''} ios(Z, z_k)) \not\Rightarrow (st(Q, \alpha_i z_i) \not\equiv_{z''} ios(Z, z_k))$.
   ii. By (c) and $z_i < z_k$, hence, $ios(Z, z_i) \sqsupseteq ios(Z, z_k)$.
   iii. Hence, F comprises a prefix of $z_i z''$, which, by $ln(z_i z'') < ln(\phi)$, is a contradiction.
   iv. Hence, $(st(Q, \alpha'_i z_i) \not\equiv_{z''} ios(Z, z_k)) \Rightarrow (st(Q, \alpha_i z_i) \not\equiv_{z''} ios(Z, z_k))$.
8. By induction starting with $st(Q, \alpha_k z_k) \not\equiv_{z''} ios(Z, z_k)$, hence, $st(Q, \alpha'_k z_k) \not\equiv_{z''} ios(Z, z_k)$, which, by $\phi \in F$, is a contradiction. $\square$

Proof of Proposition 2.

1. Suppose that the premise is true and let K denote the IOS pair $\{z, z'\}$.
2. Let Z' denote the IOS set $\{z'''|(z''' \in ios(s)) \wedge (ts(s, z''') \subseteq ts(s, z''))\}$.
3. $(Z \subseteq Z') \wedge (Z \subseteq ios(Q, z)) \wedge (Z \subseteq ios(Q, z'))$
4. For each IOS $\beta \in notall(Z, Z')$:
   i. Take any IOS set $L_\beta \subseteq end(M, Z)$ with $(end(M, L_\beta) = (end(M, Z) \setminus \{st(s, z''')|z''' \in pf(\beta)\})) \wedge (|L_\beta| = |end(M, L_\beta)|)$.
   ii. Let H_β denote $pf(\beta)$. Hence, $|H_\beta| = m - |end(M, Z)| + 1$.
   iii. Order the state set $\{st(s, z''')|z''' \in pf(\beta)\}$ into a sequence $s_1 \ldots s_{k_\beta}$.
   iv. For each $1 \leq h \leq k_\beta$, let $H_{\beta,h}$ denote the IOS set $\{z'''|(z''' \in H_\beta) \wedge (st(s, z''') = s_h)\}$.

v. For each $1 \leq h \leq k_\beta$, $ios(Z', z_1) = ios(Z', z_2)$ for each IOS pair $\{z_1, z_2\} \subseteq H_{\beta,h}$.
vi. For each $1 \leq i < j \leq k_\beta$, $st(Q, z_i) \neq st(Q, z_j)$ for each IOS pair $\{z_i, z_j\}$ with $(z_i \in (K \otimes H_{\beta,i})) \wedge (z_j \in (K \otimes H_{\beta,j}))$.
vii. $(|end(Q, L_\beta)| = |end(M, Z)| - k_\beta) \wedge (|end(Q, L_\beta \cup (K \otimes H_\beta))| \leq m)$
viii. $|H_\beta| + k_\beta > |end(Q, L_\beta \cup (K \otimes H_\beta))| - |end(Q, L_\beta)|$
ix. $st(Q, z_1) \neq st(Q, z_2)$ for each IOS pair $\{z_1, z_2\}$ with $(z_1 \in L_\beta) \wedge (z_2 \in (K \otimes H_\beta))$.

5. By Theorem 3, hence, $st(Q, z_1) \approx_{Z'} st(Q, z_2)$.
6. By $z'' \in Z'$, hence, $(z'' \in ios(Q, z)) \Leftrightarrow (z'' \in ios(Q, z'))$. □

Proof of Proposition 4.

1. Suppose that an IOS z'' satisfies the condition claimed sufficient for $st(Q, zz'') \approx_{ios(Z,z'')} st(Q, z'z'')$, but the latter is not true.
2. Hence, there is an IOS $z''' \in ios(Z, z'')$ with $st(Q, zz'') \not\approx_{ios(Z,z''),z'''} st(Q, z'z'')$.
3. By $st(Q, z) \equiv_{z''} st(Q, z') \equiv_{z''} Z$, $st(Q, z) \not\approx_{Z,z''z'''} st(Q, z')$.
4. The latter contradicts $st(Q, z) \approx_Z st(Q, z')$. □