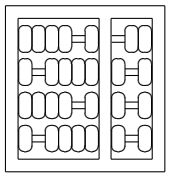

INSTITUTO DE COMPUTAÇÃO
UNIVERSIDADE ESTADUAL DE CAMPINAS

Test Suite Completeness and Weak Equivalence

Adilson Luiz Bonifacio *Arnaldo Vieira Moura*

Technical Report - IC-14-01 - Relatório Técnico

January - 2014 - Janeiro

Checking Test Suite Completeness with Partial FSMs and Weak Equivalence

Adilson Luiz Bonifacio* Arnaldo Vieira Moura†

Abstract

One of the important tasks in model-based testing is checking completeness of test suites. In this paper we first extend some known sufficient conditions for test suite completeness by also allowing partial implementations. We also study a new notion of equivalence, and show that the same conditions are still sufficient when treating only complete implementations, but not when we also allow partial implementation models.

1 Introduction

Automatic test suite generation for Finite State Machine (FSM) models has been widely investigated in the literature [3, 1, 4, 7]. Several approaches have proposed conditions and techniques to generate test suites based on FSMs with complete fault coverage [2, 5, 8, 9, 10]. Some of these works have shown sufficient conditions that guarantee the completeness of the test suites [4, 12].

Simao and Petrenko [11] proposed sufficient conditions for checking test suite completeness based on the notion of confirmed sets. Informally, a set of input sequences is confirmed if its sequences are such that two of them lead to a same state in the specification if and only if they also lead to a common state in the implementation. However, in that approach, implementation candidates are assumed to be complete models. Further, specifications and implementations are also required to be reduced and initially connected FSMs with n states.

In this work we first remove the restrictions on the implementation machines being complete models. We also show that the existence of confirmed sets is enough to guarantee test suite completeness when the implementations have any number os states. These relaxations seem natural, given that implementations are usually treated as black boxes. We also explore the new notion of equivalence, where we treat test cases that may not run to completion in implementation models. This further improves the treatment of implementations as real black boxes. It is shown that confirmed sets, under the new notion of equivalence, are also sufficient to guarantee test suite completeness when implementations are complete models. By contrast, the existence of confirmed sets is no longer sufficient to ascertain test suite completeness when implementations may be partial machines.

*Computing Department, University of Londrina, Londrina, Brazil, *email: bonifacio@uel.br*. Supported by FAPESP, process 2012/23500-6.

†Computing Institute, University of Campinas, Campinas, Brazil, *email: arnaldo@ic.unicamp.br*.

This paper is organized as follows. Section 2 gives some important definitions. In Section 3 we show that confirmed sets are sufficient for test suite completeness under the classical notion of equivalence, and even when implementations are partial machines. In Section 4 we extend this result under the new notion of equivalence, restricted to implementations being complete FSMs. Test suite completeness under the new notion of equivalence and allowing for partial implementations is studied in Section 5. Section 6 ends with some concluding remarks.

2 Definitions

Let $\mathcal{I}$ be an alphabet. We denote by $\mathcal{I}^\star$ the set of all finite sequences of symbols from $\mathcal{I}$. When we write $\sigma = x_1 x_2 \cdots x_n \in \mathcal{I}^\star$ $(n \geq 0)$ we mean $x_i \in \mathcal{I}$ $(1 \leq i \leq n)$, unless noted otherwise. The length of any finite sequence of symbols α over $\mathcal{I}$ is indicated by $|\alpha|$. The empty sequence will be indicated by ε, with $|\varepsilon| = 0$. Given any two sets of sequences $A, B \subseteq \mathcal{I}^\star$, their symmetric difference will be indicated by $A \ominus B$.

Remark 1 *$A \ominus B = \emptyset$ iff[1] $A = B$.*

Next, we define Finite State Machines (FSMs) [6, 11].

Definition 1 *A FSM is a tuple $(S, s_0, \mathcal{I}, \mathcal{O}, D, \delta, \lambda)$ where*

1. *S is a finite set of* states
2. *$s_0 \in S$ is the initial state*
3. *$\mathcal{I}$ is a finite set of* input actions *or* input events
4. *$\mathcal{O}$ is a finite set of* output actions *or* output events
5. *$D \subseteq S \times \mathcal{I}$ is a* specification domain
6. *$\delta : D \to S$ is the* transition function
7. *$\lambda : D \to \mathcal{O}$ is the* output function.

All FSMs treated here are deterministic, since δ and λ are functions.

In the sequel, M and N will always denote the FSMs $(S, s_0, \mathcal{I}, \mathcal{O}, D, \delta, \lambda)$ and $(Q, q_0, \mathcal{I}, \mathcal{O}', D', \mu, \tau)$, respectively. Let $\sigma = x_1 x_2 \cdots x_n \in \mathcal{I}^\star$, $\omega = a_1 a_2 \cdots a_n \in \mathcal{O}^\star$ $(n \geq 0)$. If there are states $r_i \in S$ $(0 \leq i \leq n)$ such that $(r_{i-1}, x_i) \in D$, with $\delta(r_{i-1}, x_i) = r_i$ and $\lambda(r_{i-1}, x_i) = a_i$ $(1 \leq i \leq n)$, we may write $r_0 \stackrel{\sigma/\omega}{\to} r_n$. When σ, or ω, or both, is not important, we may write $r_0 \stackrel{\sigma/}{\to} r_n$, or $r_0 \stackrel{/\omega}{\to} r_n$, or $r_0 \to r_n$, respectively. We can also drop the target state, when it is not important, *e.g.* $r_0 \stackrel{\sigma/\omega}{\to}$ or $r_0 \to$. It will be useful to extend the functions δ and λ to pairs $(s, \sigma) \in S \times \mathcal{I}^\star$. Let $\widehat{D} = \{(s, \sigma) \,|\, s \stackrel{\sigma/}{\to}\}$. Define the extensions $\widehat{\delta} : \widehat{D} \to S$

[1] Here, 'iff' is short for 'if and only if'.

and $\widehat{\lambda} : \widehat{D} \to \mathcal{O}^\star$ by letting $\widehat{\delta}(s,\sigma) = r$ and $\widehat{\lambda}(s,\sigma) = \omega$ whenever $s \xrightarrow{\sigma/\omega} r$. When there is no reason for confusion, we may write D, δ and λ instead of $\widehat{D}$, $\widehat{\delta}$ and $\widehat{\lambda}$, respectively. Also, let $U(s) = \{\sigma \mid (s,\sigma) \in \widehat{D}\}$ for all $s \in S$.

Remark 2 *We always have $\varepsilon \in U(s)$, for all $s \in S$.*

Next we define reachability and completeness for FSMs.

Definition 2 *Let M be a FSM, and $s, r \in S$. Then r is* reachable from s *iff $s \xrightarrow{\sigma/} r$ for some $\sigma \in \mathcal{I}^\star$. We say that r is* reachable *iff it is reachable from s_0. Also, M is* complete *iff for all reachable $s \in S$ and all $x \in \mathcal{I}$ we have $s \xrightarrow{x/}$.*

From Definition 1, FSMs need not be complete, that is, we may have $D \neq S \times \mathcal{I}$. On the other hand, if M is complete then $U(s) = \mathcal{I}^\star$ for all $s \in S$.

Observe that when M and N are not complete then there are cases when we can differentiate two states $s \in S$ and $q \in Q$ if there is $\sigma \in U(s) \ominus U(q)$. In this case, by running σ starting at s and q, one would observe a longer output sequence from either M or N. The notion of weak equivalence captures this effect: two FSMs are weakly equivalent when any input sequence that runs in one also runs in the other. The notion of equivalence is the classical one: any input sequence that runs on both machines must yield the same behavior.

Definition 3 *Let M and N be FSMs and $s \in S$, $q \in Q$. Let $C \subseteq \mathcal{I}^\star$. We say that s and q are*

1. C-weakly-distinguishable *iff $\big(U(s) \ominus U(q)\big) \cap C \neq \emptyset$, denoted $s \not\sim_C q$. Otherwise they are* C-weakly-equivalent*, denoted $s \sim_C q$.*
2. C-distinguishable *iff $\lambda(s,\sigma) \neq \tau(q,\sigma)$ for some $\sigma \in U(s) \cap U(q) \cap C$, denoted $s \not\approx_C q$. Otherwise, they are* C-equivalent*, denoted $s \approx_C q$.*

Two FSMs M and N are *C-weakly-distinguishable* or *C-distinguishable* iff $s_0 \not\sim q_0$ or $s_0 \not\approx q_0$, respectively. Otherwise M and N are *C-weakly-equivalent* or *C-equivalent*, respectively. We will use the same notation for FSM equivalence as we did for states, *e.g.*, $M \approx_C N$ when $s_0 \approx_C q_0$. When C is not important, or it is clear from the context, we might drop the index. When there is no mention to C, we mean $C = \mathcal{I}^\star$.

Remark 3 *Weak-distinguishability and weak-equivalence are purely structural notions. They do not mention the output sequence of the FSMs in any way.*

Next we say when FSMs are reduced.

Definition 4 *We say that a FSM M is* reduced *iff every state is reachable and every two distinct states are distinguishable.*

Now we define test cases and test suites.

Definition 5 *Let M be a FSM. A* test suite for M *is any finite subset of $\mathcal{I}^\star$. Any element of a test suite is a* test case*.*

Consider a specification M, a test suite T for M, and the notion of T-equivalence. In many situations, one considers only test suites whose test cases run in M, that is, with $T \subseteq U(s_0)$. It is also common to assume that implementations N are complete machines, that is, $U(q_0) = \mathcal{I}^\star$. Under these conditions we get $T \cap U(s_0) \cap U(q_0) = T$, and T-equivalence reduces to $\lambda(s_0, \alpha) = \tau(q_0, \alpha)$, for all $\alpha \in T$, that is M and N display the same behavior under all tests in T. Another common constraint is to restrict implementations N to have at most as many states as specifications M. However, when implementations N are treated as black boxes, usually one cannot guarantee completeness of N, nor does one control the number of states in N. Many of the results that follow will establish a contrast between this simpler state of affairs and other, more realistic, scenarios where test suites can be more general sets and implementations may be partial models with any number of states.

Next, we give the concept of n-completeness for test suites, under the two notions of equivalence.

Definition 6 *Let M be a FSM, let T be a test suite for M and take $n \geq 1$. Then T is*

1. weakly-n-complete for M *iff for any FSM N with $|Q| \leq n$, if $M \sim_T N$ then $M \sim N$.*
2. n-complete for M *iff for any FSM N with $|Q| \leq n$, if $M \approx_T N$ then $M \approx N$.*

3 Equivalence and Partial Implementations

Simao and Petrenko [11] proposed sufficient conditions for checking test suite n-completeness under a number of restrictions about the specifications and the implementations, such as reducibility of both machines and completeness of the implementations. In this section we extend that result, by showing that those conditions are also sufficient for checking test suite completeness even when implementations are not complete. We also will not require that implementations have no more states than the specifications, nor we will require that both models be reduced FSMs. We remark also that in this section we will be treating only the classical definition of equivalence.

For the ease of reference, we repeat part of [11]. If M is a FSM and T a test suite for M, let $\mathfrak{T}_T(M)$ be the set of all FSMs with the same number os states as M, and that are reduced, complete and T-equivalent to it, as in [11]. Since we will not need these restrictions, let $\mathcal{E}_T(M)$ be the set of all FSMs that are just T-equivalent to M.

Definition 7 ([11]) *Let M be a FSM, T a test suite for M and $K \subseteq T$. The set K is* confirmed *iff for any $s \in S$ there is some $\alpha \in K$ such that $\delta(s_0, \alpha) = s$ and, further, for each $N \in \mathcal{E}_T(M)$, it holds that*

1. *$K \subseteq U(s_0) \cap U(q_0)$;*
2. *For all α, $\beta \in K$, $\mu(q_0, \alpha) = \mu(q_0, \beta)$ if and only if $\delta(q_0, \alpha) = \delta(q_0, \beta)$.*

Remark 4 *We note that condition (1) is not explicitly stated in Definition 4 of [11]. Rather, in [11] it is implicitly assumed the slightly stronger condition that $T \subseteq U(s_0) \cap U(q_0)$ or, equivalently $T \subseteq U(s_0)$, given that implementations are always taken as complete models*

in [11]. Also, we required the weaker condition $N \in \mathcal{E}_T(M)$ in our Definition 7, and not $N \in \mathcal{T}_T(M)$ as in Definition 4 of [11], since we are not restricted to complete implementations only.

Theorem 1 ([11]) *Let M be a reduced FSM with n states. When restricted to reduced and complete implementations, T is n-complete for M if there exists a confirmed set $K \subseteq T$ such that $\varepsilon \in K$ and for each $(s, x) \in D$ there exist α, $\alpha x \in K$ such that $\delta(s_0, \alpha) = s$.*

We start by showing that confirmed sets induce certain injective functions between the states of a specification and an implementation, given that they are T-equivalent.

Lemma 1 *Let M, N be FSMs and T a test suite for M. Assume that $M \approx_T N$, and that $K \subseteq T$ is a confirmed set. Let $f = \{(\delta(s_0, \alpha), \mu(q_0, \alpha)) \,|\, \alpha \in K\}$. Then f is an injective function.*

Proof Assume that (s, q_1), $(s, q_2) \in f$. Then, we must have α_1, $\alpha_2 \in K$ with $\delta(s_0, \alpha_1) = s = \delta(s_0, \alpha_2)$ and $q_1 = \mu(q_0, \alpha_1)$, $q_2 = \mu(q_0, \alpha_2)$. But since K is confirmed, $M \approx_T N$ and α_1, $\alpha_2 \in K$ we get $\mu(q_0, \alpha_1) = \mu(q_0, \alpha_2)$. This gives $q_1 = q_2$, showing that f is a function. Let now (s_1, q), $(s_2, q) \in f$. This gives $\alpha_1, \alpha_2 \in K$ with $\delta(s_0, \alpha_1) = s_1$, $\delta(s_0, \alpha_2) = s_2$ and $\mu(q_0, \alpha_1) = q = \mu(q_0, \alpha_2)$. Again, since K is confirmed, $M \approx_T N$ and α_1, $\alpha_2 \in K$ we get $\delta(s_0, \alpha_1) = \delta(s_0, \alpha_2)$. Hence, $s_1 = s_2$ and f is injective. ∎

Remark 5 *The bi-directional requirement in Definition 7(2) was crucial to show that f is an injective function.*

If we take only complete implementations with no more states than the specification, then f, as in Lemma 1 is, in fact, a bijection.

Lemma 2 *Let M, N be FSMs and T a test suite for M. Assume that $M \approx_T N$ and that $K \subseteq T$ is a confirmed set. Let f be as in Lemma 1. Then, f is a bijection if N is complete and $|Q| \leq |S|$.*

Proof Let $s \in S$. Since K is confirmed, there is some $\alpha \in K$ such that $\delta(s_0, \alpha) = s$. Since N is complete, we know that $\alpha \in U(q_0)$, and so $\alpha \in K \cap U(s_0) \cap U(q_0)$. The definition of f gives $(s, \mu(q_0, \alpha)) \in f$, and so f is a total relation. Since, Lemma 1, $f \subseteq S \times Q$ is also an injective function, we get $|S| \leq |Q|$. Thus, $|S| = |Q|$ and so f is also onto. We conclude that f is a bijection. ∎

The next result says that f, defined in Lemma 1, can be extended to $U(s_0) \cap U(q_0)$, when the hypothesis of Theorem 1 are satisfied.

Lemma 3 *Let M, N be FSMs and T a test suite for M with $M \approx_T N$. Let $K \subseteq \mathcal{I}^\star$ be a confirmed set satisfying the hypothesis of Theorem 1. Assume that f, as defined in Lemma 1, is a function. Then $(\delta(s_0, \rho), \mu(q_0, \rho)) \in f$ for all $\rho \in U(s_0) \cap U(q_0)$.*

Proof We go by induction on $|\rho| \geq 0$. When $|\rho| = 0$ we get $\rho = \varepsilon$. Hence, by the hypothesis in Theorem 1, $\rho \in K$. Clearly, we also have $\rho \in U(s_0) \cap U(q_0)$ and the result follows. Inductively, assume the result for all ρ with $|\rho| \leq n$. Take ρx, with $x \in \mathcal{I}$, $|\rho| = n$ and such that $\rho x \in U(s_0) \cap U(q_0)$. Then $\rho \in U(s_0) \cap U(q_0)$ and the induction hypothesis gives $(s, q) \in f$, where $\delta(s_0, \rho) = s$ and $\mu(q_0, \rho) = q$. Since $\rho x \in U(s_0)$, we have $(s, x) \in D$, and the hypothesis of Theorem 1 give α, $\alpha x \in K$ with $\delta(s_0, \alpha) = s$. Because $\alpha x \in K$, the definition of f gives $(\delta(s_0, \alpha x), \mu(q_0, \alpha x)) \in f$. We conclude the argument by showing that $\delta(s_0, \alpha x) = \delta(s_0, \rho x)$ and $\mu(q_0, \alpha x) = \mu(q_0, \rho x)$, thus extending the induction. The former follows immediately since $\delta(s_0, \rho) = s = \delta(s_0, \alpha)$. For the latter, since $\alpha \in K$, from the definition of f we also get $(\delta(s_0, \alpha), \mu(q_0, \alpha)) \in f$. But we already know that $(s, q) = (\delta(s_0, \alpha), q)$ is also in f and so, because f is a function, we conclude that $\mu(q_0, \alpha) = q = \mu(q_0, \rho)$. This immediately gives $\mu(q_0, \alpha x) = \mu(q_0, \rho x)$, as desired. ∎

The next result essentially says that states mapped under f, as defined in Lemma 1, will display the same behavior.

Lemma 4 *Let M, N be FSMs, T a test suite for M and let $K \subseteq T$ be a confirmed set satisfying the sufficiency conditions of Theorem 1. Let $s \in S$, $x \in \mathcal{I}$ with $(s, x) \in D$, and assume that f, as in Lemma 1, is a function. Then $\lambda(s, x) = \tau(f(s), x)$ whenever $M \approx_T N$.*

Proof The sufficiency conditions in Theorem 1 give α, $\alpha x \in K$ such that $\delta(s_0, \alpha) = s$. From Definition 7 and $K \subseteq T$, we get $\alpha x \in U(s_0) \cap U(q_0) \cap T$ and so, because $M \approx_T N$, we get $\lambda(s_0, \alpha x) = \tau(q_0, \alpha x)$ using Definition 3. But

$$\lambda(s_0, \alpha x) = \lambda(s_0, \alpha)\lambda(\delta(s_0, \alpha), x) = \lambda(s_0, \alpha)\lambda(s, x) \quad \text{and}$$
$$\tau(q_0, \alpha x) = \tau(q_0, \alpha)\tau(\mu(q_0, \alpha), x).$$

Hence, $\lambda(s, x) = \tau(\mu(q_0, \alpha), x)$. Because $\alpha \in K$, the definition of f now yields $(\delta(s_0, \alpha), \mu(q_0, \alpha)) = (s, \mu(q_0, \alpha))$ is in f. Since f is a function, $f(s) = \mu(q_0, \alpha)$. Thus, $\lambda(s, x) = \tau(f(s), x)$. ∎

We are now in a position to extend Theorem 1. We show that the conditions there required are sufficient to guarantee test suite completeness even when implementations are not complete and have any number of states.

Theorem 2 *Let M be a FSM and T a test suite for M. Let $K \subseteq T$ be a confirmed set satisfying the sufficiency conditions of Theorem 1. Then T is n-complete for M, for all $n \geq 1$.*

Proof Assume that T is not n-complete for M, for some $n \geq 1$. Then, by Definition 6, there is a FSM N, with $|Q| \leq n$ and such that $M \not\approx N$ and $M \approx_T N$. Using Definition 3 we get $\rho \in \mathcal{I}^\star$, $x \in \mathcal{I}$ with $\rho x \in U(s_0) \cap U(q_0)$, and such that $\lambda(s_0, \rho) = \tau(q_0, \rho)$ and $\lambda(s_0, \rho x) \neq \tau(q_0, \rho x)$. Let $s = \delta(s_0, \rho)$ and $q = \mu(q_0, \rho)$, so that $\lambda(s, x) \neq \tau(q, x)$. Since $M \approx_T N$ and K is confirmed, the relation f, as defined in Lemma 1, is an injective function. Since K also satisfies the sufficiency conditions of Theorem 1, Lemma 3 says that $f(s) = f(\delta(s_0, \rho)) = \mu(q_0, \rho) = q$. Clearly. $(\delta(s_0, \rho), x) = (s, x) \in D$, and so Lemma 4 gives $\lambda(s, x) = \tau(f(s), x)$, that is, $\lambda(s, x) = \tau(q, x)$, which is a contradiction. ∎

Remark 6 *For this proof it is important that f is simply a* function*, and not necessarily a* bijection*, as required in [11]. Also, the fact that $|Q| \leq n$ was not important for the proof. This assumption is need in Lemma 2, but not in Lemma 1.*

4 Weak-equivalence and Complete Implementations

Let M be a FSM with n states. We denote by $\mathcal{W}(M)$ and by $\mathcal{WC}(M)$ the sets of all FSMs and all complete FSMs, respectively, with at most n states. Let T a test suite for M. We denote by $\mathcal{W}_T(M)$, and by $\mathcal{WC}_T(M)$, the sets of all FSMs in $\mathcal{W}(M)$, and in $\mathcal{WC}(M)$, respectively, and that are T-weakly-equivalent to M.

Definition 8 *Let M be a FSM, T a test suite for M and $K \subseteq T$. The set K is* weak-confirmed *for M and T iff for any $s \in S$ there is some $\alpha \in K$ such that $\delta(s_0, \alpha) = s$ and, further, for each $N \in \mathcal{W}_T(M)$, it holds that*

1. *$K \subseteq U(s_0) \cap U(q_0)$;*
2. *For all α, $\beta \in K$, $\mu(q_0, \alpha) = \mu(q_0, \beta)$ if and only if $\delta(q_0, \alpha) = \delta(q_0, \beta)$.*

Remark 7 *When using Definition 8 together with the sufficiency conditions in Theorem 1, it is redundant to require, in Definition 8, that for any $s \in S$ there is some $\alpha \in K$ such that $\delta(s_0, \alpha) = s$, when M is a reduced machine with at least two states. To see this, let $s \in S$. Then, $(s, x) \in D$ for some $x \in \mathcal{I}$ because M is reduced. The sufficiency conditions then give some $\alpha \in K$ such that $\delta(s_0, \alpha) = s$, as desired.*

Suppose now that we are treating only complete implementations. In particular, in Definition 8 we replace $\mathcal{W}_T(M)$ by $\mathcal{WC}_T(M)$.

Lemma 5 *Let M and N be FSMs, with N complete, and let T be a test suite for M. Then*

1. *$M \sim_T N$ iff $T \subseteq U(s_0)$.*
2. *$M \not\sim N$ iff $U(s_0) \neq \mathcal{I}^\star$.*

Proof Follow directly from the definitions. ∎

Now we have the following result.

Theorem 3 *Assume that we only allow complete implementations. Let M be a FSM with $n \geq 2$ states, and let T a test suite for M. If there is a weakly-confirmed set $K \subseteq T$ that also satisfies the sufficiency conditions of Theorem 1, then T is weak-n-complete for M.*

Proof First assume that $T \not\subseteq U(s_0)$. For the sake of contradiction, assume the result is false. Then we have a FSM M and a test suite T for M such that there is a weakly-confirmed $K \subseteq T$ that satisfies the sufficiency conditions of Theorem 1. Further, we know that $M \sim_T N$ and $M \not\sim N$. But Lemma 5 immediately gives $M \not\sim_T N$, a contradiction.

Now let $T \subseteq U(s_0)$. Lemma 5 immediately gives that $M \sim_T N$ for any complete FSM N. We construct a complete FSM N with $Q = S$ as follows. Since M has $n \geq 2$ states, let $s_0 \xrightarrow[M]{x/} s_1$, for some $s_1 \in S$. Fix some a, $b \in \mathcal{O}$ with $a \neq b$. Since we also want N to be reduced, construct the cycle $s_i \xrightarrow[N]{y/a} s_{i+1}$, $0 \leq i < n$, and close the cycle with $s_n \xrightarrow[N]{y/b} s_0$, for all $y \in \mathcal{I}$ and $y \neq x$. Terminate the construction with $s_i \xrightarrow[N]{x/a} s_0$, $0 \leq i \leq n$. Clearly, N is also complete. It is easy to see that N is reduced, since the input sequence y^n starting at state s_i gives $a^{n-1-i}ba^i$. Then, $N \sim_T M$ and so $N \in \mathcal{WC}_T(M)$. Let $K \subseteq T$ be a weakly-confirmed set that also satisfies the sufficiency conditions of Theorem 1. Then, since $(s_0, x) \in D$, there are α, $\alpha x \in K$ with $\delta(s_0, \alpha) = s_0$. We also have $\varepsilon \in K$. Then, ε, $\alpha x \in K$, with $\delta(s_0, \varepsilon) = s_0 \neq s_1 = \delta(s_0, \alpha x)$ in M. But the construction of N gives $\mu(s_0, \varepsilon) = s_0 = \mu(s_0, \alpha x)$. This contradicts the fact that K is weakly-confirmed. ∎

5 Weak-equivalence and Partial Implementations

In this section we show that Theorem 1 does not hold under weak equivalence and when we allow for partial implementations. We want a (partial) specification FSM M and a test suite T for M such that T contains a confirmed subset $K \subseteq T$ which also satisfies the sufficiency conditions of Theorem 1. We then construct a (partial) implementation FSM N such that $M \sim_T N$, but $M \not\sim N$, thus establishing that Theorem 1 does not hold under these conditions, as desired.

Remark 8 *Since, in this section, we are treating weak-equivalence only, we will drop output symbols in transitions. Output symbols could easily be injected so that all FSM considered here are reduced.*

We start with a specification M with $n + 1$ states, where $n \geq 2$, that is we let $S = \{s_0, s_1, \ldots, s_n\}$. The simpler cases when $n = 1$ or $n = 0$ will be dealt with later. We define machine M as in Figure 1. More formally, we let $\mathcal{I} = \{0, 1\}$, $S = \{s_0, s_1, \cdots, s_n\}$ and

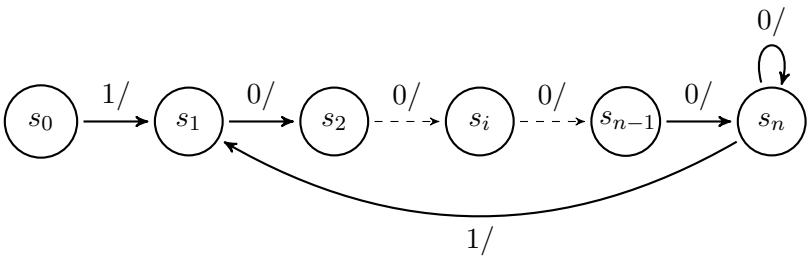

Figure 1: Speficication M

$$\begin{aligned} &s_0 \xrightarrow{1/} s_1 \xrightarrow{0/} s_2 \xrightarrow{0/} s_3 \xrightarrow{0/} \cdots \xrightarrow{0/} s_{n-1} \xrightarrow{0/} s_n \xrightarrow{1/} s_1 \\ &s_n \xrightarrow{0/} s_n. \end{aligned} \tag{1}$$

Next we define a test suite T and a subset $K \subseteq T$. For the ease of reference, we let $K = K_1 \cup K_2$ with

$$\begin{aligned} K_1 &= \{\varepsilon\} \cup \{10^p \,|\, 1 \leq p \leq n\}, \quad \text{and} \\ K_2 &= \{10^{n-1}1, 10^{n+1}1, 10^{n-1}10, 10^n10\}. \end{aligned} \tag{2}$$

The test suite T is now given by $T = K \cup T_1$, where

$$T_1 = \{0\} \cup \{10^p10, 10^{n-1}10^p10 \,|\, 0 \leq p \leq n-2\}. \tag{3}$$

Note that since $n \geq 2$, K and T are well defined. Further, we clearly have $T_1 \cap K = \emptyset$.

Using K_1 it is easy to see that for all states $s \in S$, there is some $\alpha \in K$ such that $\delta(s_0, \alpha) = s$, so that K satisfies the reachability condition in Definition 8. Using K_1 again and $10^{n-1}1 \in K_2$ it is also a simple matter to verify that K satisfies the sufficiency conditions of Theorem 1.

To ease the notation, given a FSM M, some $s \in S$ and some $\alpha \in \mathcal{I}^\star$, we will say that s *runs* α if $\alpha \in U(s)$, and we will say that M *runs* α if s_0 runs α.

Remark 9 *It is easy to verify that $K \subseteq U(s_0)$, that is M runs all $\alpha \in K$. It is also easy to verify that $T_1 \cap U(s_0) = \emptyset$, that is, M does not run any $\alpha \in T_1$.*

Next, in order to verify that K is indeed confirmed, according to Definition 8, we need to investigate which FSMs are in $\mathcal{W}_T(M)$. Note that, according to Definition 3 and Remark 9, for N to be in $\mathcal{W}_T(M)$ it is necessary and sufficient that N runs all $\alpha \in K$ and that N does not run any $\alpha \in T_1$. We will show that any FSM in $\mathcal{W}_T(M)$ aave $n+1$ states and it must also have, at least, all transitions that are in M. More specifically, if N is a FSM in $\mathcal{W}_T(M)$, then the states in N can be listed as $Q = \{q_0, q_1, \ldots, q_n\}$, with the property that if $s_i \xrightarrow{x/} s_j$ in M, then we also have $q_i \xrightarrow{x/} q_j$ in N, for all $x \in \mathcal{I}^\star$. We establish this result by proving a series of claims.

Let N be a FSM such that $N \sim_T M$.

Claim 1: We have $q_0 \xrightarrow{1/} q_1 \xrightarrow{0/} q_2$, with q_0, q_1, q_2 distinct.

Proof. First note that since M does not run 0, then there is no outgoing transition on 0 from q_0. That is, N does not run 0. Since M runs $100^{n-2}1$, then N will also run 1, but we cannot have $q_0 \xrightarrow{1/} q_0$ because, since N will also run 10, then it would also run 0, which is a contradiction. We conclude that we must have $q_0 \xrightarrow{1/} q_1$, for some $q_1 \in Q$, $q_0 \neq q_1$.

Now, since M runs $1000^{n-1}1$, so does N. This means that q_1 runs $000^{n-1}1$. Clearly, we cannot have $q_1 \xrightarrow{0/} q_0$ because this will make q_0 run 0, which cannot happen.

Next, we show that we cannot have a self-loop on 0 at q_1. Since M runs $100^{n-2}1$ so would N. Then, q_1 would run 1. There are three cases.

Assume that $q_1 \xrightarrow{1/} q_2$, for some $q_2 \in Q$. Since M runs $10^{n-1}10$, so does N. But then N would also run $\alpha = 110 = 10^p10$, with $p = 0$. Since $\alpha \in T_1$, we get a contradiction.

Assume that $q_1 \xrightarrow{1/} q_1$. Then, as before, N would also run 110, which cannot happen.

Finally, assume that $q_1 \xrightarrow{1/} q_0$. Then, since $\alpha = 10^{n-1}10 \in K_2$, N would run α, and so N would also run $0 \in T_1$, a contradiction.

We conclude that $q_1 \xrightarrow{0/} q_2$, for some $q_2 \in Q$, distinct from q_0 and q_1. △

From Claim 1, we have $q_0 \xrightarrow{1/} q_1 \xrightarrow{0/} q_2$ in N, with $|\{q_0, q_1, q_2\}| = 3$. Next, we show that this argument can be extended inductively.

Claim 2: Assume that we have q_j $(0 \leq j \leq k)$ as distinct states in Q, where $2 \leq k < n$, and such that $q_0 \xrightarrow{1/} q_1$ and

$$q_1 \xrightarrow{0/} q_2 \xrightarrow{0/} q_3 \xrightarrow{0/} \cdots \xrightarrow{0/} q_k.$$

Then there is a new distinct state $q_{k+1} \in Q$ with $q_k \xrightarrow{0/} q_{k+1}$.

Proof. Recall that $10^{n+1}1 \in K_2$, and so N runs $10^{n+1}1$. Since $n > k$, this means that q_k runs $000^{n-k}1$. We cannot have $q_k \xrightarrow{0/} q_0$ because then N would run $0 \in T_1$, which would be a contradiction. There are two other cases: either $q_k \xrightarrow{0/} q_j$, where $1 \leq j \leq k$, or $q_k \xrightarrow{0/} q_{k+1}$, where $q_{k+1} \in Q$ is a new distinct state. We show that the former cannot happen, and so the latter must be true, establishing the claim.

For the sake of contradiction, assume that $q_k \xrightarrow{0/} q_j$, where $1 \leq j \leq k$. Since $10^{n-1}1 \in K_2$, then N must run $10^{n-1}1$. We have $n-1-(j-1) = n-j \geq 1$ and so q_j must run $0^{n-j}1$. Also, $n-j \geq (k+1)-j = k-j+1 = \ell$, and so we have the cycle $\mu(q_j, 0^\ell) = q_j$. We can write $n-j = c\ell + r$, for some $c \geq 1$ and some $0 \leq r < \ell$. So, because of the cycle, we have

$$\begin{aligned}\mu(q_0, 10^{n-1}) &= \mu(q_j, 0^{n-j}) = \mu(q_j, 0^{c\ell+r}) \\ &= \mu(q_j, 0^r) = q_{j+r}.\end{aligned}$$

Note that $1 \leq r + j < \ell + j = k + 1$, so that $1 \leq j + r \leq k$. Since $\mu(q_0, 10^{n-1}) = q_{j+r}$ and $10^{n-1}10 \in K_2$, we know that $q_{j+r} \xrightarrow{1/} q$, for some $q \in Q$. If $q = q_0$ then N would run $0 \in T_1$, which cannot happen. Then, $q_{j+r} \xrightarrow{1/} q_i$, for some $1 \leq i \leq k+1$, where $q_{k+1} \in Q$ is a new distinct state. If this is the case, then we would get $\mu(q_0, 10^{n-1}10) = \mu(q_{j+r}, 10) = \mu(q_i, 0)$. So, q_i runs 0. Since $j + r \leq k$, we also have $\mu(q_0, 10^{j+r-1}10) = \mu(q_{j+r}, 10) = \mu(q_i, 0)$, and so N would run $10^{j+r-1}10 = 10^p10$, with $p = j + r - 1$. Since $1 \leq j + r \leq k$ we get $0 \leq p \leq k - 1 \leq n - 2$, because $k < n$. This shows that N runs $10^p10 \in T_1$, which is a contradiction. △

Now, using Claim 1 as the basis and Claim 2 as the induction step, we have

$$q_0 \xrightarrow{1/} q_1 \xrightarrow{0/} q_2 \xrightarrow{0/} q_3 \xrightarrow{0/} \cdots \xrightarrow{0/} q_{n-1} \xrightarrow{0/} q_n.$$

But $10^{n+1}1 \in K_2$, and we conclude that q_n runs 0^21.

Claim 3: We must have $q_n \xrightarrow{0/} q_n$.

PROOF. If $q_n \stackrel{0/}{\to} q_0$ we would immediately have N running $0 \in T_1$, which cannot happen. Now assume that $q_n \stackrel{0/}{\to} q_i$ with $1 \leq i < n$. Recall that because $10^n10 \in K_2$, then N runs 10^n10. Hence,

$$\mu(q_0, 10^{n-1}010) = \mu(q_n, 010) = \mu(q_i, 10),$$

and so q_i runs 1. We cannot have $q_i \stackrel{1/}{\to} q_0$ because q_0 would run $0 \in T_1$, which is not allowed. If $q_i \stackrel{1/}{\to} q_j$, with $2 \leq j \leq n$, then we would have

$$\mu(q_0, 10^n10) = \mu(q_i, 10) = \mu(q_j, 0),$$

and so q_j would run 0. But we also have $\mu(q_0, 10^{i-1}10) = \mu(q_i, 10) = \mu(q_j, 0)$, and so N would also run $10^{i-1}10$. Because $0 \leq i-1 \leq n-2$, we get $10^{i-1}10 \in T_1$, a contradiction. The only other possibility is to have $q_n \stackrel{0/}{\to} q_n$, thus establishing the claim. △

Using Claim 3, we now have

$$q_0 \stackrel{1/}{\to} q_1 \stackrel{0/}{\to} q_2 \stackrel{0/}{\to} q_3 \stackrel{0/}{\to} \cdots \stackrel{0/}{\to} q_{n-1} \stackrel{0/}{\to} q_n, \text{ and } q_n \stackrel{0/}{\to} q_n.$$

CLAIM 4: We must have $q_n \stackrel{1/}{\to} q_1$.

PROOF. Observe that $10^{n-1}10 \in K_2$, and so N must run $10^{n-1}10$. Since $\mu(q_0, 10^{n-1}10) = \mu(q_n, 10)$, we conclude that q_n runs 10. If $q_n \stackrel{1/}{\to} q_0$ we immediately get that q_0 runs $0 \in T_1$, which cannot happen.

Assume now that $q_n \stackrel{1/}{\to} q_j$, with $2 \leq j \leq n$. Then, q_j runs 0. Also,

$$\begin{aligned}\mu(q_0, 10^{n-1}10^{n-j}10) &= \mu(q_n, 10^{n-j}10) = \mu(q_j, 0^{n-j}10) \\ &= \mu(q_n, 10) = \mu(q_j, 0).\end{aligned}$$

Thus, N runs $\alpha = 10^{n-1}10^{n-j}10$. Since $2 \leq j \leq n$, we have $0 \leq n-j \leq n-2$ and so $\alpha \in T_1$, contradicting N running α.

Since N can have at most $n+1$ states, because $N \in \mathcal{W}_T(M)$, the only possibility left is $q_n \stackrel{1/}{\to} q_1$, thus establishing the claim. △

We now have reached the desired result.

CLAIM 5: Let N be a FSM. If $N \sim_T M$ then N has $n+1$ states $\{q_0, q_1, \ldots, q_n\}$ satisfying

$$\begin{aligned}&q_0 \stackrel{1/}{\to} q_1 \stackrel{0/}{\to} q_2 \stackrel{0/}{\to} q_3 \stackrel{0/}{\to} \cdots \stackrel{0/}{\to} q_{n-1} \stackrel{0/}{\to} q_n \stackrel{1/}{\to} q_1 \\ &q_n \stackrel{0/}{\to} q_n.\end{aligned} \tag{4}$$

PROOF. Use Claims 1–4. △

We can now argue that K is confirmed for M and T.

CLAIM 6: Let M, T and K be as given in (1)–(3). Then, K is confirmed for M and T.

PROOF. Let N be a FSM. From Remark 9, we get $K \subseteq U(s_0)$. Let $N \in \mathcal{W}_T(M)$. Then, by Claim 5, N satisfies (4), and we clearly get $U(s_0) \subseteq U(q_0)$. Hence, $K \subseteq U(s_0) \cap U(q_0)$, satisfying the first condition at Definition 8. It is also easily seen that we have $\delta(s_0, \alpha) = s_i$ if and only if $\mu(q_0, \alpha) = q_i$, for all $\alpha \in K$ and all $0 \leq i \leq n$. From this, it follows easily that $\delta(s_0, \alpha) = \delta(s_0, \beta)$ if and only if $\mu(q_0, \alpha) = \mu(q_0, \beta)$, for all α, $\beta \in K$. Thus, the second condition at Definition 8 is satisfied and so K is weakly-confirmed for M and T. △

Finally, we construct the counter-example to Theorem 1. Let N be a FSM with $n+1$ states and that satisfies (4). To conclude the construction of N, add the transition $q_1 \xrightarrow{1/} q_0$ as shown in Figure 2.

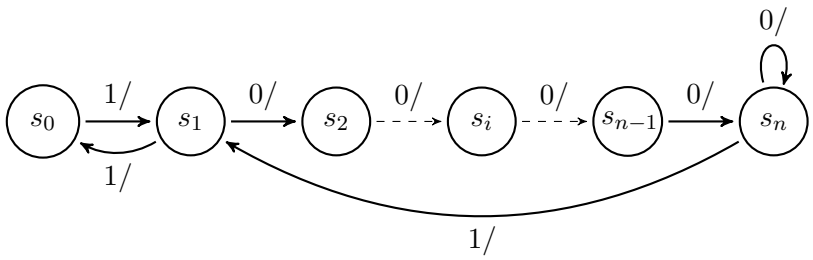

Figure 2: Implementation N

Now, we check that N runs any string in K.

CLAIM 7: N runs all $\alpha \in K$.

PROOF. From Remark 9, M runs all $\alpha \in K$. From (1) and (4), we conclude that N also runs all $\alpha \in K$. △

Next, we check that N does not run any string in T_1.

CLAIM 8: N does not run α, for any $\alpha \in K$.

PROOF. Clearly, N does not run 0.

Take $10^p10 \in T_1$, with $0 \leq p \leq n-2$. Then,

$$\mu(q_0, 10^p10) = \mu(q_1, 0^p10) = \mu(q_{p+1}, 10).$$

There are two cases. If $1 \leq p \leq n-2$, then $2 \leq p+1 \leq n-1$ and from (4) we get that q_{p+1} does not run 10, and so N does not run 10^p10 either. If $p = 0$ we get $\mu(q_0, 10^p10) = \mu(q_1, 10) = \mu(q_0, 0)$. Again, from (4), q_0 does not run 0 and so N does not run 10^p10.

Finally, take $10^{n-1}10^p10 \in T_1$, with $0 \leq p \leq n-2$. Now we have

$$\mu(q_0, 10^{n-1}10^p10) = \mu(q_1, 0^{n-1}10^p10) = \mu(q_n, 10^p10) = \mu(q_1, 0^p10),$$

and we can repeat the previous argument.

This exhausts all of T_1 and the claim holds. △

We have now our negative result.

Theorem 4 *Assume that specifications are partial FSMs with $n \geq 3$ states, and implementations are partial FSMs with at most n states. Then Theorem 1 does not hold under weak-equivalence.*

Proof From Claims 7 and 8 we know that $N \sim_T M$. But we can easily check that N runs 11 and M does not run 11. Hence $N \not\sim M$. Since, by Claim 6, K is confirmed for M and T, we conclude that Theorem 1 does not hold. ∎

The particular cases when specifications have 1 or 2 states are easy to treat separately. First, let M be a two-state FSM with $S = \{s_0, s_1\}$, $\mathcal{I} = \{0, 1\}$ and

$$s_0 \xrightarrow{1/} s_1 \xrightarrow{0/} s_1. \tag{5}$$

Let $T = K \cup T'$ where

$$K = \{\varepsilon, 1, 10, 100\}, \quad T' = \{0\}. \tag{6}$$

Clearly, any state in M is reachable using a string in K. Moreover, it is easy to check that K satisfies the sufficiency condition at Theorem 1. Let N be a FSM with at most 2 states and such that $N \sim_T M$. Since M runs 10, so does N. If $q_0 \xrightarrow{1/} q_0$ then q_0 would also run 0. But this is a contradiction since $0 \in T'$ and M does not run 0. Hence, we must have $q_0 \xrightarrow{1/} q_1$, with $q_0 \neq q_1$. Since M also runs 100, so does N. If we had $q_1 \xrightarrow{0/} q_0$, then q_0 would run 0, which can not happen. We conclude that in N we have

$$q_0 \xrightarrow{1/} q_1 \xrightarrow{0/} q_1. \tag{7}$$

But now it is immediate that $\delta(s_0, \alpha) = \delta(s_0, \beta)$ if and only if $\mu(q_0, \alpha) = \mu(q_0, \beta)$, for all α, $\beta \in K$. Since N was any machine in $\mathcal{W}_T(M)$, we thus conclude that K is confirmed for M and K. For the desired counter-example, let N' be as in (7) with the added transition $q_1 \xrightarrow{1/} q_1$. Clearly, N' runs all $\alpha \in K$ and does not run $0 \in T'$. Hence, $N' \sim_T M$. But N' runs 11 and M does not, so that $N' \not\sim M$. Because K was shown to be weakly-confirmed for M and T, we conclude that Theorem 1 also does not hold for two-state FSMs under weak-equivalence and when we allow for partial specifications and partial implementations.

For completeness, we also look at one-state machines. Take M with $S = \{s_0\}$ and no transitions. Let $T = K = \{\varepsilon\}$. It is easily checked that K satisfies the sufficiency conditions at Theorem 1. Since M is a one-state FSM and $T = \{\varepsilon\}$ we immediately conclude that any one-state FSM is T-weakly-equivalent to M. From this, and since $K = \{\varepsilon\}$, it follows easily that K is weakly-confirmed for M and T. Now take the on-state FSM N where we just let $q_0 \xrightarrow{0/} q_0$. Then, $N \sim_T M$ but $N \not\sim M$ because N runs 0 and M does not. Again, Theorem 1 does not hold for one-state FSMs.

We can now state a general result.

Corollary 1 *Assume that specifications are partial FSMs with n states, and implementations are partial FSMs with at most n states. Then Theorem 1 does not hold under weak-equivalence.*

Proof From Theorem 4 and the preceding discussion.

6 Conclusions

In this work we showed that confirmed sets can be applied to less restrictive FSM models, when we do not require implementations to be completely specified, an important relaxation when treating implementations as true black boxes. Also, we showed that specifications nor implementations need to be reduced models.

We also investigated the problem of checking test suite completeness under the notion of weak-equivalence, that is when we have test suites whose test cases that may not run to completion in specifications or in implementations. This further strengths the black box characteristic of implementations. We showed that confirmed sets, in the presence of weak-equivalence, are still sufficient for checking test suite completeness when we allow complete implementations only. In contrast, we also showed that confirmed sets are not any more sufficient for checking test suite completeness when we also consider partial implementations.

All statements were proved correct by rigorous arguments.

References

[1] Adilson L. Bonifacio, Arnaldo V. Moura, and Adenilso da S. Simao. A generalized model-based test generation method. In Antonio Cerone and Stefan Gruner, editors, *Sixth IEEE International Conference on Software Engineering and Formal Methods, SEFM*, pages 139–148, Cape Town, South Africa, 10–14, nov 2008. IEEE Computer Society.

[2] Adilson Luiz Bonifacio, Arnaldo Vieira Moura, and Adenilso da Silva Simão. Model partitions and compact test case suites. *Int. J. Found. Comput. Sci.*, 23(1):147–172, 2012.

[3] T. S. Chow. Testing software design modeled by finite-state machines. *IEEE Transactions on Software Engineering*, 4(3):178–187, 1978.

[4] Rita Dorofeeva, Khaled El-Fakih, and Nina Yevtushenko. An improved conformance testing method. In *FORTE*, pages 204–218, 2005.

[5] S. Fujiwara, G. V. Bochmann, F. Khendek, M. Amalou, and A. Ghedamsi. Test selection based on finite state models. *IEEE Transactions on Software Engineering*, 17(6):591–603, June 1991.

[6] A. Gill. *Introduction to the theory of finite-state machines.* McGraw-Hill, New York, 1962.

[7] F. C. Hennie. Fault detecting experiments for sequential circuits. In *Proceedings of the Fifth Annual Symposium on Switching Circuit Theory and Logical Design, 11-13 November 1964, Princeton, New Jersey, USA*, pages 95–110. IEEE, 1964.

[8] R. M. Hierons. Separating sequence overlap for automated test sequence generation. *Automated Software Engg.*, 13(2):283–301, 2006.

[9] Robert M. Hierons and Hasan Ural. Reduced length checking sequences. *IEEE Trans. Comput.*, 51(9):1111–1117, September 2002.

[10] Robert M. Hierons and Hasan Ural. Optimizing the length of checking sequences. *IEEE Trans. Comput.*, 55(5):618–629, May 2006.

[11] Adenilso da Silva Simao and Petrenko Petrenko. Checking completeness of tests for finite state machines. *IEEE Trans. Computers*, 59(8):1023–1032, 2010.

[12] Hasan Ural, Xiaolin Wu, and Fan Zhang. On minimizing the lengths of checking sequences. *IEEE Trans. Comput.*, 46(1):93–99, January 1997.