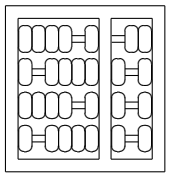

INSTITUTO DE COMPUTAÇÃO
UNIVERSIDADE ESTADUAL DE CAMPINAS

Test Suite Completeness and Black Box Testing

Adilson Luiz Bonifacio *Arnaldo Vieira Moura*

Technical Report - IC-16-04 - Relatório Técnico

August - 2016 - Agosto

Test Suite Completeness and Black Box Testing

Adilson Luiz Bonifacio* Arnaldo Vieira Moura†

Abstract

Model-based testing has been widely studied and successfully applied to generate and verify completeness of test suites. Roughly, completeness guarantees that a non-equivalent implementation under test will always be identified regarding complete deterministic Finite State Machines. Several approaches showed sufficient, and sometimes also necessary, conditions on specification models and test suites in order to guarantee completeness. In these studies, usually, test cases are required to be non-blocking — that is, they are required to run to completion — on both the specification and the implementation models. However, often it is desirable to have blocking test cases, and in some situations the presence of blocking test cases cannot be circumvented. In the present work we allow test cases to block, both in the specification as well as in the implementation models, and we study a natural variant of completeness, here called perfectness. Perfectness guarantees that non-compliance between a specification and an implementation will be detected, even in the presence of blocking test cases when an input action of the test case is not defined. We characterize perfectness in terms of isomorphisms, and establish a relationship between the classical notion of completeness and perfectness. We also give an upper bound on the number of states in implementations, beyond which no test suite can be complete, both in a conventional sense and in the presence of blocking test cases.

1 Introduction

Completeness of test suites has been largely studied for models based on Finite State Machines (FSMs) [4, 8, 6, 12, 2, 13, 11]. A test suite is said to be complete for a FSM specification when it provides complete fault coverage [4, 8] according to an appropriate fault model. Several works have proposed strategies for generating complete test suites [5], or for checking if a given test suite is complete for a given specification [2]. Some of them presented necessary conditions [9, 14] for test suite completeness, whereas other approaches gave sufficient, but not necessary, conditions for test suite completeness [6, 10, 12, 13]. Still other approaches described necessary *and* sufficient conditions for test suite completeness [2, 5]. All these works, however, imposed restrictions on the specification and implementation

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

models, or in some way restricted the fault domains [6, 10, 12, 13, 2]. Some of them considered specifications with n states and restricted implementations to have at most n states. Further, in some approaches specification and implementations are required to be reduced or completely specified machines.

Always, test cases have been required to be non-blocking on both the specifications and the implementations models, meaning that all test cases are assumed to run to the end in these models. In particular, test cases are assumed to run to completion on implementations even when implementations are treated as true black-boxes, which is not reasonable in practical applications since such implementations could be represented by partial machines. Hence test suites can not be assumed to run to completion in any black-box implementation.

In a recent work, Bonifacio and Moura [3] have proposed an alternative approach to deal with more general scenarios where test cases can block both in the specification or in implementation models. In that work test cases are not required to run to completion, even when implementations are *partial FSMs*, and furthermore, implementations are treated as *true black boxes*. A new notion of equivalence, called *alikeness*, was also introduced giving rise to the notion of *perfectness in lieu* of the classical notion of completeness.

A related issue that concerns test suite completeness is the maximum size, in terms of the maximum number of states, in implementations that can be put under test. Usually, earlier works constrained implementations to have at most the same number of states as the given specification. Some of them considered implementations with more states than the specification, but with an upper bound on the number of states, now imposed by the tester. We are not aware of any work that gives a precise relationship between the maximum number of states in implementations and the size of test suites in order to get positive verdicts when such implementations are put under test.

In this work we start by giving a new characterization of the notion of test suite perfectness in terms of isomorphisms. We establish a close relationship between the classical notion of completeness and the new notion of perfectness. We then give a precise upper bound on the number of states in implementations under test, beyond which no test suite can be guaranteed to be complete, both in the classical sense and in the more general scenario when blocking test cases can be present. The bound is based only on a measure of test suite size and the number of states in the given specification, and it does not depend on the implementation model.

We organize the paper as follows. Basic results, definitions and notations appear in Section 2. Section 3 characterizes perfectness in terms of isomorphisms. We investigate the relationship between completeness and perfectness in Section 4. In Section 5 we establish an upper bound on the number of states in candidate implementations beyond which a guarantee of completeness is lost. Section 6 states some conclusions.

2 Definitions and notation

In this section we introduce some basic concepts. We also present some preliminary results that will be important in the following sections.

Let $\mathcal{I}$ be an alphabet. The length of any finite sequence α of symbols over $\mathcal{I}$ is indicated

by $|\alpha|$. The empty sequence will be indicated by ε, with $|\varepsilon| = 0$. The set of all sequences of length k over $\mathcal{I}$ ($k \geq 0$) is denoted by $\mathcal{I}^k$, while $\mathcal{I}^\star$ names the set of all finite sequences over $\mathcal{I}$. When we write $x_1 x_2 \cdots x_n \in \mathcal{I}^\star$ ($n \geq 0$) we mean $x_i \in \mathcal{I}$ ($1 \leq i \leq n$), unless noted otherwise. Given any two sets of sequences $A, B \subseteq \mathcal{I}^\star$, their symmetric difference will be indicated by $A \ominus B$, that is $A \ominus B = (\overline{A} \cap B) \cup (A \cap \overline{B})$, where $\overline{A}$ indicates the complement of A with respect to $\mathcal{I}^\star$. The usual set difference is indicated by $A \setminus B$.

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

2.1 Finite state machines and test suites

Next, we write the definition of a Finite State Machine [2, 7].

Definition 1. *A deterministic FSM is a system $M = (S, s_0, \mathcal{I}, \mathcal{O}, D, \delta, \lambda)$ where*

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

In what follows 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 $\delta(r_{i-1}, x_i) = r_i$ and $\lambda(r_{i-1}, x_i) = a_i$ ($1 \leq i \leq n$), we may write $r_0 \xrightarrow{\sigma/\omega} r_n$. When the input sequence σ, or the output sequence ω, is not important we may write $r_0 \xrightarrow{\sigma/} r_n$, or $r_0 \xrightarrow{/\omega} r_n$, respectively, and when both sequences are not important we may write $r_0 \to r_n$. We can also drop the target state, and write $r_0 \xrightarrow{\sigma/\omega}$ 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 $\widehat{D} = \left\{(s,\sigma) \,\middle|\, s \xrightarrow{\sigma/}, \sigma \in \mathcal{I}^\star, s \in S\right\}$. Define the extensions $\widehat{\delta} : \widehat{D} \to S$ 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.

The function $U : S \to \mathcal{I}^\star$ will be useful, where $U(s) = \{\sigma \,|\, (s, \sigma) \in \widehat{D}\}$. Informally, $U(s)$ denotes all defined input action sequences that can be completely run starting at state s.

Now we are in a position to define test cases and test suites.

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

Definition 2. *A* test suite for M *is any finite nonempty subset of* $\mathcal{I}^\star$*. Any element of a test suite is a* test case*.*

Before we can define test completeness, we need the classical notions of distinguishability and equivalence.

Definition 3. *Let* M *and* N *be FSMs and let* $s \in S$*,* $q \in Q$*. Let* $C \subseteq \mathcal{I}^\star$*. We say that* s *and* q *are* C-distinguishable *iff* $\lambda(s,\sigma) \neq \tau(q,\sigma)$ *for some* $\sigma \in U(s) \cap U(q) \cap C$*, denoted* $s \not\approx_C q$*. Else,* s *and* q *are* C-equivalent*, denoted* $s \approx_C q$*. We say that* M *and* N *are* C-distinguishable *iff* $s_0 \not\approx_C q_0$*, and they are* C-equivalent *iff* $s_0 \approx_C q_0$*.*

When C is clear from the context we might drop the subscript. When there is no mention to C we understand that we are taking $C = \mathcal{I}^\star$. In this case, the condition $U(s_0) \cap U(q_0) \cap C$ reduces to $U(s_0) \cap U(q_0)$. For the ease of notation, we also write $M \approx_C N$ when M and N are C-equivalent, and write $M \not\approx_C N$ when they are C-distinguishable.

The conventional notion of m-completeness is as follows.

Definition 4. *Let* T *be a test suite for* M *and* $m \geq 1$*. Then* T *is* m-complete for M *iff for any FSM* N*, with* $U(s_0) \subseteq U(q_0)$ *and with at most* m *states, the following hold: Whenever* $M \not\approx N$ *then* $M \not\approx_T N$*, where there exists* $\sigma \in T$ *such that* $\sigma \notin U(s_0)$*.*

Note that if σ runs to completion from s_0, that is, $s_0 \xrightarrow{\sigma/}$, then σ must also run to completion from q_0, that is we must have $q_0 \xrightarrow{\sigma/}$. The definition says that any discrepancy between the behaviors of the specification M and any implementation N will be detected if we run the tests in T through M and N, provided that we consider implementations with at most m states. Note that the technical condition $U(s_0) \subseteq U(q_0)$ will always be satisfied if we were to test implementations that were complete FSM models. A FSM M is said to be complete when $D = S \times \mathcal{I}$, that is, for any state s and any input symbol x, we always have $s \xrightarrow{x/}$. We note that characterizations of m-completeness have appeared in earlier works [3, 2].

2.2 The notion of alikeness

A sequence of input symbols that does not run to completion in a FSM is called by a *blocking* test case.

Definition 5. *A* blocking *test case for* M *is a sequence* $\sigma \notin U(s_0)$*. When* σ *is not blocking we say that* σ runs to completion *in* M*.*

Given two FSM models M and N, if $\sigma \in U(s_0) \ominus U(q_0)$ then we must have that either σ blocks in M and runs to completion in N, or vice-versa. Given a test suite T and two FSM models M and N, we want to say when M and N are equivalent in some more general sense, that is, even considering that we may have blocking test cases, for M or N, in T. We want that all $\sigma \in T$ that is a blocking test case for M must also be a blocking test case for N, and vice-versa. Further, any test case that is non-blocking for both M and N must output identical behaviors when run through both models. Under these two conditions, M and N will be said to be T*-alike*.

Definition 6. *Let M and N be FSMs and let $s \in S$, $q \in Q$. Let $C \subseteq \mathcal{I}^\star$. We say that s and q are C-*alike*, denoted $s \sim_C q$, iff $\big(U(s) \ominus U(q)\big) \cap C = \emptyset$ and $\lambda(s,\sigma) = \tau(q,\sigma)$ for all $\sigma \in U(s) \cap U(q) \cap C$. Otherwise, s and q are C-*unlike*, denoted $s \not\sim_C q$. We say that M and N are C-*alike *iff $s_0 \sim_C q_0$, otherwise they are C-*unlike.*

We may also write $M \sim_C N$ when M and N are C-alike, or $M \not\sim_C N$ when they are C-unlike. Again, when C is not important, or when it is clear from the context, we might drop the subscript. When there is no other mention to C we understand that we are taking $C = \mathcal{I}^\star$.

Remark 2. *We note of the following simple observations.*

1. *Using Remark 1, we note that $s \sim q$ is equivalent to $U(s) = U(q)$ and $\lambda(s,\sigma) = \tau(q,\sigma)$ for all $\sigma \in U(s)$.*
2. *If $C_1 \subseteq C_2$, then $s \sim_{C_2} q$ implies $s \sim_{C_1} q$.*
3. *If $s \sim q$, then $s \sim_C q$, for all $C \subseteq \mathcal{I}^\star$.*

The alikeness relation, $\sim_C$, is an equivalence relation when M and N are the same machine, that is, when $\sim_C$ is defined over a single state set. We note that this is not the case, in general, with the distinguishability relation $\approx_C$.

Lemma 1. *Let M be a FSM and let $C \subseteq \mathcal{I}^\star$. Then $\sim_C$ is an equivalence relation on S.*

Proof. Let $s, r, p \in S$ be states of M. We clearly have $U(s) \ominus U(s) = \emptyset$ and $\lambda(s,\alpha) = \lambda(s,\alpha)$ for all $\alpha \in U(s) \cap C$. So, $\sim_C$ is reflexive. Also, set intersection, the symmetric set difference $\ominus$ and, of course, equality are commutative. Hence, $\sim_C$ is symmetric.

For transitivity, assume $s \sim_C r$ and $r \sim_C p$. Let $\alpha \in U(s) \cap C$. Thus $\alpha \in U(r)$ because $s \sim_C r$, and then $\alpha \in U(p)$ because $r \sim_C p$. So, $U(s) \subseteq U(p)$. Since we already have symmetry, we get $p \sim_C r$ and $r \sim_C s$, and a similar argument gives $U(p) \subseteq U(s)$, showing that $(U(s) \ominus U(p)) \cap C = \emptyset$. Now, let $\alpha \in U(s) \cap U(p) \cap C$. Since $s \sim_C r$, we get $\alpha \in U(r)$ and so $\lambda(s,\alpha) = \lambda(r,\alpha)$. But also $r \sim_C p$, and so $\lambda(r,\alpha) = \lambda(p,\alpha)$, thus establishing $\lambda(s,\alpha) = \lambda(p,\alpha)$. We may then conclude that $s \sim_C p$, and $\sim_C$ is transitive. □

Remark 3. *In Lemma 1, the transitivity of the alikness relation $\sim_C$ is still valid when it is defined as a relation among states of distinct machines.*

When reducing FSMs in the presence of blocking test cases, we will need the following technical result.

Lemma 2. *Let M be a FSM and let $s, r \in S$ with $s \sim r$.*

(1) *If $s \xrightarrow{x/a} p$ with $x \in \mathcal{I}$ and $a \in \mathcal{O}$, then $r \xrightarrow{x/a} q$ with $p \sim q$, for some $q \in S$.*

(2) *If $s \xrightarrow{\alpha/\omega} p$ with $\alpha \in \mathcal{I}^\star$ and $\omega \in \mathcal{O}^\star$, then $r \xrightarrow{\alpha/\omega} q$, with $p \sim q$ for some $q \in S$.*

Proof. We first treat item 1. We have $x \in U(s)$, and so $x \in U(r)$ because $s \sim r$, which leads to $r \xrightarrow{x/b} q$ for some $q \in S$, $b \in \mathcal{O}$. Now, $x \in U(s) \cap U(r)$ and, since $s \sim r$, we

get $a = \lambda(s, x) = \lambda(r, x) = b$. It remains to show that $p \sim q$. Let $\alpha \in U(p)$. Then $x\alpha \in U(s)$, and again $x\alpha \in U(r)$. Since M is deterministic, this gives $\alpha \in U(q)$, and so $U(p) \subseteq U(q)$. Using Remark 2(1) we have $r \sim s$, and a similar argument gives $U(q) \subseteq U(p)$. We conclude that $U(p) = U(q)$, and so $U(p) \ominus U(q) = \emptyset$. Now, let $\beta \in U(p) \cap U(q)$. Then, $x\beta \in U(s) \cap U(r)$, and since $s \sim r$ this gives $a\lambda(p, \alpha) = \lambda(s, x\beta) = \lambda(r, x\beta) = a\lambda(q, \alpha)$. We conclude that $\lambda(p, \alpha) = \lambda(q, \alpha)$, as desired.

Item (2) follows by a simple induction on $|\alpha| \geq 0$, and using the result of item 1. □

The notion of *perfectness* has been introduced by Bonifacio and Moura [3, 1], in order to cope with test cases that may not run to completion either in the specification or in the implementation models.

Definition 7 ([3]). *Let M be a FSM and T be a test suite for M. Then T is* perfect for *M iff for any FSM N, if $M \not\sim N$ then $M \not\sim_T N$.*

That is, when T is a perfect test suite for a specification M, then for any implementation under test N, if M and N are unlike, then they are also T-unlike.

The following result will be useful when we consider certain bi-similarities.

Lemma 3. *Let M and N be FSMs. Let $n \geq 0$, $r_i \in S$ $(1 \leq i \leq n+1)$, $x_i \in \mathcal{I}$, and $a_i \in \mathcal{O}$ be such that $r_i \stackrel{x_i/a_i}{\rightarrow} r_{i+1}$ $(1 \leq i \leq n)$. Assume that $r_0 \sim p_0$, for some $p_0 \in Q$. Then we have $p_i \in Q$ $(1 \leq i \leq n+1)$ such that $p_i \stackrel{x_i/a_i}{\rightarrow} p_{i+1}$ and $s_i \sim p_i$ $(1 \leq i \leq n)$.*

Proof. If $n = 0$ there is nothing to prove. Inductively, assume the result holds for some $0 \leq k < n$. Then we have $s_k \sim p_k$. Since $s_k \stackrel{x_k/a_k}{\rightarrow} s_{k+1}$, $x_k \in U(s_k)$ and $\lambda(s_k, x_k) = a_k$, Definition 6 immediately gives $p_k \stackrel{x_k/a_k}{\rightarrow} p_{k+1}$, for some $p_{k+1} \in Q$. For the sake of contradiction, assume that $s_{k+1} \not\sim p_{k+1}$. By Definition 6 we have two cases.

Case 1: $U(s_{k+1}) \ominus U(p_{k+1}) \neq \emptyset$.

Let $\beta \in U(s_{k+1})$ and $\beta \notin U(p_{k+1})$. This gives $\alpha\beta \in U(s_1)$ and $\alpha\beta \notin U(p_1)$. Hence $U(s_1) \ominus U(p_1) \neq \emptyset$, contradicting $s_1 \sim p_1$. The situation when $\beta \notin U(s_{k+1})$ and $\beta \in U(p_{k+1})$ is entirely analogous.

Case 2: $\beta \in U(s_{k+1}) \cap U(p_{k+1})$ and $\lambda(s_{k+1}, \beta) \neq \tau(p_{k+1}, \beta)$, for some $\beta \in \mathcal{I}^\star$.

This gives $\alpha\beta \in U(s_1) \cap U(p_1)$. Moreover,

$$\lambda(s_1, \alpha\beta) = \lambda(s_1, \alpha)\lambda(\delta(s_1, \alpha), \beta)) = \lambda(s_1, \alpha)\lambda(s_{k+1}, \beta), \text{ and}$$
$$\tau(p_1, \alpha\beta) = \tau(p_1, \alpha)\tau(\mu(p_1, \alpha), \beta)) = \tau(p_1, \alpha)\tau(p_{k+1}, \beta).$$

Because $|\lambda(s_1, \alpha)| = |\tau(p_1, \alpha)|$ and $\lambda(s_{k+1}, \beta) \neq \tau(p_{k+1}, \beta)$, we get $\lambda(s_1, \alpha\beta) \neq \tau(p_1, \alpha\beta)$. Since $\alpha\beta \in U(s_1) \cap U(p_1)$, this contradicts $s_1 \sim p_1$.

The proof is complete. □

The next result guarantees the existence of bi-simulations in the presence of blocking test cases.

Lemma 4. *Let T be a m-perfect test suite for a FSM M. Let N be a FSM with at most m states such that $M \sim_T N$. Then M and N are bi-similar.*

Proof. Define a relation $R_1 \subseteq S \times Q$ by letting $(s,q) \in R_1$ if and only if $\delta(s_0,\alpha) = s$ and $\mu(q_0,\alpha) = q$ for some $\alpha \in \mathcal{I}^\star$, $s \in S$ and $q \in Q$. Since $\delta(s_0,\varepsilon) = s_0$ and $\mu(q_0,\varepsilon) = q_0$ we get $(s_0,q_0) \in R_1$.

Now assume $(s,q) \in R_1$ and let $s \stackrel{x/a}{\rightarrow} r$ for some $r \in S$, $x \in \mathcal{I}$ and $a \in \mathcal{O}$. Since $(s,q) \in R_1$, the definition of R_1 gives some $\alpha \in \mathcal{I}^\star$ such that $\delta(s_0,\alpha) = s$ and $\mu(q_0,\alpha) = q$. Composing, we get $\delta(s_0,\alpha x) = \delta(s,x) = r$ and so $\alpha x \in U(s_0)$. Since T is m-perfect for M and $M \sim_T N$, Definition 7 gives $M \sim N$, that is $s_0 \sim q_0$. Further, Definition 6 and Remark 2 imply $U(s_0) = U(q_0)$, and so $\alpha x \in U(q_0)$. Then $\mu(q,x) = p$, for some $p \in Q$. Since $s_0 \sim q_0$, $\delta(s_0,\alpha) = s$ and $\mu(q_0,\alpha) = q$, Lemma 3 gives $s \sim q$. But $x \in U(s) \cap U(q)$, and so we must have $a = \lambda(s,x) = \tau(q,x)$. Thus, we have found $p \in Q$ with $q \stackrel{x/a}{\rightarrow} p$. Since $\delta(s_0,\alpha x) = r$ and $\mu(q_0,\alpha x) = p$, we also have $(r,p) \in R_1$. This shows that R_1 is a simulation relation.

A similar argument will show that $R_2 \subseteq Q \times S$, where $R_2 = R_1^{-1}$, is also a simulation relation. Thus M and N are bi-similar, as desired. □

2.3 Simulations and perfectness

In [3, 1] bi-simulation was used to *characterize* perfectness. Basically it is shown that a bi-simulation between two machines leads to the same observable behaviors produced by the machines when test cases are applied to them even in the presence of blocking test cases. In Section 3 we will characterize perfectness in terms of isomorphisms. The result presented in this subsection will be useful later to show the relationship between the notions of Perfectness and Isomorphism.

Definition 8. *Let M and N be FSMs. We say that a relation $R \subseteq S \times Q$ is a* simulation *(*of M by N*) iff $(s_0,q_0) \in R$, and whenever we have $(s,q) \in R$ and $s \stackrel{x/a}{\rightarrow} r$ in M, then there is a state $p \in Q$ such that $q \stackrel{x/a}{\rightarrow} p$ in N and with $(r,p) \in R$. We say that M and N are* bi-similar *iff there are simulation relations $R_1 \subseteq S \times Q$ and $R_2 \subseteq Q \times S$.*

The following simple facts will be used later.

Fact 1. *The simulation relation is transitive, that is, let $M_i = (S_i, s_i, \mathcal{I}, \mathcal{O}, D_i, \delta_i, \lambda_i)$ be FSMs, $i = 1,2,3$, and where M_2 simulates M_1 and M_3 simulates M_2. Then, M_3 simulates M_1.*

Proof. Let $R_1 \subseteq S_1 \times S_2$ and $R_2 \subseteq S_2 \times S_3$ be simulation relations. Define $R \subseteq S_1 \times S_3$ by $(s,p) \in R$ iff $(s,q) \in R_1$ and $(q,p) \in R_2$, for some $q \in S_2$. Firstly, since $(s_1,s_2) \in R_1$ and $(s_2,s_3) \in R_2$ we get $(s_1,s_3) \in R$, as needed. Moreover, let $(s,p) \in R$ and $s \stackrel{x/a}{\rightarrow} s_1$. We must have $(s,q) \in R_1$ and $(q,p) \in R_2$ for some $q \in S_2$. Since R_1 is a simulation, we get $q \stackrel{x/a}{\rightarrow} q_1$, with $(s_1,q_1) \in R_1$. Since R_2 is a simulation, we get $p \stackrel{x/a}{\rightarrow} p_1$ with $(q_1,p_1) \in R_2$. Then, $(s_1,p_1) \in R$, as desired. □

Fact 2. *Let M and N be FSMs, and let $R \subseteq S \times Q$ be a simulation of M by N. If $(s,q) \in R$ and $s \stackrel{\alpha/\omega}{\to} r$ in M for some $\alpha \in \mathcal{I}^\star$, $\omega \in \mathcal{O}^\star$, then $\stackrel{\alpha/\omega}{\to} t$ in N for a unique $t \in Q$, and $(r,t) \in R$.*

Proof. An easy induction on $|\alpha| \geq 0$. Since N is deterministic, $t \in Q$ is unique. □

Fact 3. *Let M and N be FSMs, let $R \subseteq S \times Q$ be a simulation of M by N, and let $L \subseteq Q \times S$ be a simulation of N by M. Let $(s,q) \in R$, $(q,s) \in L$, and $\alpha \in \mathcal{I}^\star$. If $\delta(s,\alpha) = r$, then $\mu(q,\alpha) = t$ with $(r,t) \in R$ and $(t,r) \in L$, for a unique $t \in Q$.*

Proof. From $\delta(s,\alpha) = r$ and $(s,q) \in R$ Fact 2 gives a unique $t \in Q$ with $\mu(q,\alpha) = t$ and $(r,t) \in R$. From $(q,s) \in L$ and $\mu(q,\alpha) = t$, Fact 2 again gives some $p \in S$ with $(t,p) \in L$ and $\delta(s,\alpha) = p$. Since M is deterministic and we already have $\delta(s,\alpha) = r$ we conclude that $p = r$. Hence, $(t,r) \in L$ as desired. □

The next lemma shows a useful relationship between bi-simulations and alikeness.

Lemma 5. *Let M and N be FSMs, let $R \subseteq S \times Q$ be a simulation of M by N, and let $L \subseteq Q \times S$ be a simulation of N by M. If $(s,q) \in R$ and $(q,s) \in L$ then $s \sim q$.*

Proof. If we have $\alpha \in U(s)$ then Fact 2 immediately implies that $\alpha \in U(q)$, so that $U(s) \subseteq U(q)$. Likewise, we have $U(q) \subseteq U(s)$, so that $U(s) \ominus U(q) = \emptyset$. Let $\alpha \in U(s) \cap U(q)$ with $s \stackrel{\alpha/\omega}{\to}$ in M, with $\alpha \in \mathcal{I}^\star$ and $\omega \in \mathcal{O}^\star$. Then, Fact 2 again gives $q \stackrel{\alpha/\omega}{\to}$ in N. Since N is deterministic we get $\lambda(\alpha) = \tau(\alpha)$. So, from Definition 6 we conclude that $s \sim q$. □

The next result reverses the direction in Lemma 5.

Lemma 6. *Let M and N be FSMs, and assume that $M \sim N$. Then M and N are bi-similar.*

Proof. Define a relation $R \subseteq S \times Q$ by letting $(s,q) \in R$ if and only if there is $\alpha \in \mathcal{I}^\star$ such that $s \stackrel{\alpha/}{\to}$ and $q_0 \stackrel{\alpha/}{\to}$. With $\alpha = \varepsilon$ we immediately get $(s_0,q_0) \in R$. Now let $(s,q) \in R$, and let $x \in \mathcal{I}$, $a \in \mathcal{O}$ be such that $s \stackrel{x/a}{\to} r$ in M. The definition of R gives $s_0 \stackrel{\alpha/}{\to} s$ and $q_0 \stackrel{\alpha/}{\to} q$, for some $\alpha \in \mathcal{I}^\star$. By Lemma 3 we have $s \sim q$. Since $x \in U(s)$ and $\lambda(s,x) = a$, the determinism of N and Definition 6 give $q \stackrel{x/a}{\to} t$, for some $t \in Q$. But then $s_0 \stackrel{\alpha x/}{\to} r$ and $q_0 \stackrel{\alpha x/}{\to} t$ give $(r,t) \in R$. This shows that R is a simulation of M by N.

Likewise, relation R^{-1} gives a simulation of N by M and we conclude that M and N are bi-similar. □

Now we have a characterization of alikness in terms of bi-similarity.

Theorem 1. *Let M and N be FSMs. Then M and N are alike iff they are bi-similar.*

Proof. If $M \sim N$, use Lemma 6 to conclude that M and N are bi-similar. Conversely, if M and N are bi-similar we get simulation relations $R \subseteq S \times Q$ and $L \subseteq Q \times R$ with $(s_0,q_0) \in R$ and $(q_0,s_0) \in L$. Then, Lemma 5 says that M and N are bi-similar. □

When we have a specific test suite at hand, we note the following result which also establishes a necessary and sufficient condition for it to be perfect.

Theorem 2 ([3])**.** *Let T be a test suite for M. Then T is perfect for M iff any T-alike FSM is bi-similar to M.*

In Definition 7, there is no limit in the size of the implementations. In the next definition, the key property of $M \not\sim N$ implying $M \not\sim_T N$ is required to hold only for implementations with up to a predefined number of states.

Definition 9. *Let M be a FSM, let T be a test suite for M, and let $m \geq 1$. Then T is* m-perfect for *M iff for any FSM N with at most m states, if $M \not\sim N$ then $M \not\sim_T N$.*

We can obtain a result very similar to Theorem 2, as stated in the next claim.

Theorem 3. *Let M be a FSM and T be a test suite for M. Then T is m-perfect for M iff any T-alike FSM with at most m states is bi-similar to M.*

Proof. Assume that T is m-perfect for M, and let N be a FSM with at most m states and such that $M \sim_T N$. Then, Definition 9 implies that $M \sim N$. From Lemma 6 we conclude that M and N are bi-similar. Now assume that any T-alike FSM with at most m states is bi-similar to M, and let N be a FSM with at most m states such that $M \not\sim N$. Then, M and N are bi-similar and so, using Theorem 2 we get $M \not\sim_T N$. Hence, T is m-perfect for M, by Definition 9. □

In the next section we show that the bi-similarity test, in Theorem 2, can be exchanged for an isomorphism test.

3 Perfectness and Isomorphism

In this section we characterize perfectness in terms of isomorphisms between FSMs.

3.1 Bi-simulation and isomorphism

Two FSMs are said to be isomorphic when they are identical, except for a state relabeling.

Definition 10. *Let M and N be FSMs with $\mathcal{O} = \mathcal{O}'$. An* isomorphism *(of M into N) is a bijection $f : S \to Q$ such that*

1. *$f(s_0) = q_0$; and*
2. *$s \xrightarrow{x/a} r$ in M if and only if $f(s) \xrightarrow{x/a} f(r)$ in N, for all $x \in \mathcal{I}$, $a \in \mathcal{O}$.*

Machines M and N are isomorphic *iff there is an isomorphism of M into N.*

Remark 4. *Let M and N be FSMs. The following are immediate consequences:*

1. *f is an isomorphism of M into N if and only if f^{-1} is an isomorphism of N into M.*
2. *Any isomorphism of M into N is also a simulation of M by N.*

The first half of the characterization is now easily obtained.

Lemma 7. *Let M and N be isomorphic FSMs. Then, M and N are bi-similar.*

Proof. Using Remark 4, we have a simulation of M by N, and vice-versa. □

Now let M and N be bi-similar. It is clear that if all states in M are unlike, but N has two distinct states that are alike, then it is possible for M and N not to be isomorphic, since these two distinct equivalent states in N would have to correspond to a single state in M. Machines illustrated in Figures 1 and 2 are a case in point. The problem, of

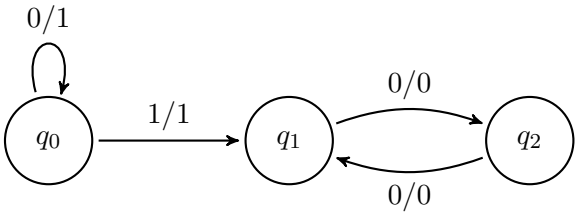

Figure 1: FSM N_1.

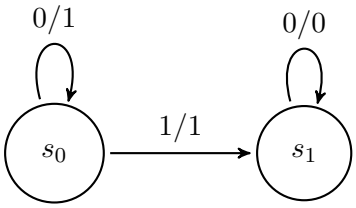

Figure 2: Specification FSM M.

course, is that states q_1 and q_2 in N_1 have exactly the same blocking input sequences and, moreover, the behaviors of q_1 and q_2 in N_1 are exactly the same under any input sequence that is non-blocking for both of them. We need to transform N_1 to an equivalent machine in which this behavior is avoided.

In the classical sense, a FSM M is reduced if every pair of distinct states in S are distinguishable. When treating partial FSM, however, we need also to take into consideration blocking input sequences. In order to differentiate from the classical notion of reduction in FSMs, we name reduction in the presence of blocking sequences as p-reduction. Both definitions are very similar.

Definition 11. *A FSM M is* reduced *iff every pair of distinct states of S are distinguishable, and for all state $s \in S$ there is a $\sigma \in \mathcal{I}^\star$ with $\delta(s_0, \sigma) = s$.*

Definition 12. *A FSM M is p-reduced iff any no two distinct states in M are alike and, moreover, for all $s \in S$ there is $\alpha \in \mathcal{I}^\star$ with $\delta(s_0, \alpha) = s$.*

Hence, for any two distinct states s and r in M there is an input sequence that blocking for one of them and is not blocking for the other, or there is an input sequence that is non-blocking for both s and r but yields different behaviors when starting at these two states. Returning to Figures 1 and 2, we see that the presence of q_1 and q_2 in N_1 shows that it is not a p-reduced FSM.

Remark 5. *If M is a reduced FSM with at least two reachable states, then there always exists a transition out of any reachable state s, that is $(s, x) \in D$ for some $x \in \mathcal{I}$. Otherwise, s could not be distinguished from any other reachable state in M.*

We proceed to show, by a series of simple facts, that if M and N are bi-similar and p-reduced, then they are isomorphic. We start by noting that the bi-similarity condition gives two simulation relations $R \subseteq S \times Q$ and $L \subseteq Q \times S$. Define a state relation $f \subseteq S \times Q$ as follows:

$$(s, q) \in f \quad \text{iff} \quad s_0 \stackrel{\alpha/}{\rightarrow} s \text{ and } q_0 \stackrel{\alpha/}{\rightarrow} q, \text{ for some } \alpha \in \mathcal{I}^\star.$$

Observe that $(s, q) \in f$ gives $s_0 \stackrel{\alpha/}{\rightarrow} s$ and $q_0 \stackrel{\alpha/}{\rightarrow} q$. Since $(s_0, q_0) \in R$, Fact 2 gives $q_0 \stackrel{\alpha/}{\rightarrow} p$ and $(s, p) \in R$, for some $p \in Q$. Since N is deterministic, we get $p = q$, and so $(s, q) \in R$. A symmetric argument gives $(q, s) \in L$. Using Lemma 5 we obtain $s \sim q$. Now let $(s, q_1) \in f$, and $(s, q_2) \in f$. Then, $s \sim q_1$ and $s \sim q_2$ and using Lemma 1 we get $q_1 \sim q_2$. Thus, Definition 12 gives $q_1 = q_2$, showing that f is a function. Similarly, if we have $(s_1, q) \in f$ and $(s_2, q) \in f$ then we must have $s_1 = s_2$ and we conclude that f is one-to-one. Further, for all $s \in S$, since M is p-reduced, Definition 12 gives $s_o \stackrel{\alpha/}{\rightarrow} s$, for some $\alpha \in \mathcal{I}^\star$. Since $(s_0, q_0) \in R$, Fact 2 implies $q_0 \stackrel{\alpha/}{\rightarrow} q$ for some $q \in Q$, and so $(s, q) \in f$, showing that f is a total function. Likewise, if $q \in Q$ we get some $s \in S$ such that $(s, q) \in f$, showing that f is onto. We have just argued showing that f is, in fact, a bijection. Finally, assume that $(s, q) \in f$, so that $s_0 \stackrel{\alpha/}{\rightarrow} s$ and $q_0 \stackrel{\alpha/}{\rightarrow} q$, for some $\alpha \in \mathcal{I}^\star$, and also $s \sim q$. If $s \stackrel{x/a}{\rightarrow} r$ in M, with $x \in \mathcal{I}$, $a \in \mathcal{O}$, then Definition 6 implies that we also have $q \stackrel{x/a}{\rightarrow} t$ in N, for some $t \in Q$. But now we have $s_0 \stackrel{\alpha x/}{\rightarrow} r$ and $q_0 \stackrel{\alpha x/}{\rightarrow} t$ and so we get $(r, t) \in f$. We conclude that the bijection f is, in fact, an isomorphism when M and N are p-reduced.

We can now state the main result of this section.

Theorem 4. *Let M and N be p-reduced FSMs. Then, M and N are bi-similar if and only if M and N are isomorphic.*

Proof. If M and N are isomorphic then they are bi-similar by Lemma 7. The argument just given establishes the converse. □

The next corollary exposes a strong relationship between perfectness of a test suite T for a FSM M and p-reduced FSMs that are T-alike to M.

Corollary 1. *Let M be a p-reduced FSM and T be a test suite for M. If T is perfect for M then any p-reduced T-alike FSM is isomorphic to M.*

Proof. Assume that T is perfect for M and let N be a p-reduced FSM that is T-alike M. By Theorem 2, we know that N is bi-similar to M. Then, M and N are isomorphic, using Theorem 4. □

3.2 p-reduced Finite State Machines

p-reduced Finite State Machines

The converse of Corollary 1 actually also holds. But, since Theorem 4 stipulates that *all* T-alike FSMs must simulate the specification M, we must first show that any FSM can be p-reduced without loosing the T-alikeness property.

Recall from Lemma 1 that $\sim$ is an equivalence relation on S. Let $[s]$ be the equivalence class of s under $\sim$. We now use the classical idea of taking quotients in order to construct a FSM $\overline{M}$ that is p-reduced and alike to M. Define

$$\overline{S} = \{[s] \,|\, s \in S, \text{and } s \stackrel{\alpha/\omega}{\to}, \text{ some } \alpha \in \mathcal{I}^\star,\, \omega \in \mathcal{O}^\star\},$$

and let $\overline{s_0} = [s_0]$. Next, if $s \sim r$ and $(s,x) \in D$, then Lemma 2(1) gives $(r,x) \in D$. Define $\overline{D} = \big\{\big([s],x\big) \,\big|\, (s,x) \in D\big\}$. Since $([s],x) \in \overline{D}$ implies $(s,x) \in D$, and Lemma 2(1) gives $\delta(s,x) \sim \delta(r,x)$ for all $r \in [s]$, we can define $\overline{\delta}\big([s],x\big) = \big[\delta(s,x)\big]$. If $s \sim r$ and $s \stackrel{x/a}{\to} p$, for some $p \in S$, $x \in \mathcal{I}$ and $a \in \mathcal{O}$, then Lemma 2(1) gives $r \stackrel{x/a}{\to} q$, for some $q \in S$, that is, $\lambda(s,x) = \lambda(r,x)$ whenever $s \sim r$ and $x \in U(s)$. Thus, we can define $\overline{\lambda}\big([s],x\big) = \lambda(s,x)$. The construction is complete.

Definition 13. *Let M be a FSM. Then $\overline{M} = (\overline{S}, \overline{s_0}, \mathcal{I}, \mathcal{O}, \overline{D}, \overline{\delta}, \overline{\lambda})$ is the FSM given by the preceding construction.*

The foregoing construction satisfy a number of properties that will be useful later.

Fact 4. *Let $s, r \in S$, and let $\alpha \in \mathcal{I}^\star$, $\omega \in \mathcal{O}^\star$. If $s \stackrel{\alpha/\omega}{\to} r$, then $[s] \stackrel{\alpha/\omega}{\to} [r]$.*

Proof. Assume that $s \stackrel{x/a}{\to} r$, with $x \in \mathcal{I}$ and $a \in \mathcal{O}$. Then $\delta(s,x) = r$ and $\lambda(s,x) = a$. From the construction of $\overline{M}$ we get $\delta([s],x) = [r]$ and $\overline{\lambda}([s],x) = a$. Hence, $[s] \stackrel{x/a}{\to} [r]$, and the result follows by an easy induction on $|\alpha| \geq 0$. □

Fact 5. *Let $r, q \in S$, and let $\alpha \in \mathcal{I}^\star$, $\omega \in \mathcal{O}^\star$. If $[r] \stackrel{\alpha/\omega}{\to} [q]$, then $r_1 \stackrel{\alpha/\omega}{\to} q_1$, for some $r_1, q_1 \in S$ with $r \sim r_1$ and $q \sim q_1$.*

Proof. Assume that $[r] \stackrel{x/a}{\to} [q]$, with $x \in \mathcal{I}$ and $a \in \mathcal{O}$. Then $\overline{\delta}([r],x) = [q]$ and $\overline{\lambda}([r],x) = a$. From $\overline{\delta}([r],x) = [q]$, the construction of $\overline{M}$ gives $r_1, q_1 \in S$ with $\delta(r_1,x) = q_1$, $r_1 \sim r$ and $q_1 \sim q$. From $\overline{\lambda}([r],x) = a$, we get $r_2 \in S$ with $\lambda(r_2,x) = a$ and $r_2 \sim r$. Hence, $r_1 \sim r_2$.

Since $r_1 \stackrel{x/b}{\to} q_1$, this gives $r_2 \stackrel{x/b}{\to} r_3$, for some $r_3 \in S$. But $\lambda(r_2,x) = a$, and so $a = b$ because machines are deterministic. Collecting, we have $r_1 \stackrel{x/a}{\to} q_1$, $r_1 \sim r$ and $q_1 \sim q$. The result now follows using a simple induction on $|\alpha|$. □

Lemma 8. *Let M be a FSM and $s, r \in S$. Let $\overline{M}$ be the FSM in Definition 13. If $[s] \neq [r]$, then $[s] \not\sim [r]$.*

Proof. Assume $[s] \sim [r]$ and show that $s \sim r$. First, we show that $U(s) \ominus U(r) = \emptyset$. Let $\alpha \in U(s)$. Then $s \stackrel{\alpha/\omega}{\rightarrow} p$, for some $p \in S$ and $\omega \in \mathcal{O}^\star$. Using Fact 4, we get $[s] \stackrel{\alpha/\omega}{\rightarrow} [p]$. Since $[s] \sim [r]$, Lemma 1 gives $[r] \stackrel{\alpha/\omega}{\rightarrow} [q]$, for some $[q] \in \overline{D}$. Using Fact 5 we obtain $r_1 \stackrel{\alpha/\omega}{\rightarrow} q_1$, for some $q_1 \in S$ with $r_1 \sim r$. Hence, Lemma 1 now gives $r \stackrel{\alpha/\omega}{\rightarrow} q_2$, for some $q_2 \in S$. We conclude that $\alpha \in U(r)$, thus establishing that $U(s) \subseteq U(r)$. A similar argument gives $U(r) \subseteq U(s)$, and so $U(s) = U(r)$, as needed. To finish, let now $\alpha \in U(s) \cap U(r)$. Then, $s \stackrel{\alpha/\omega}{\rightarrow} p$, for some $p \in S$. Repeating the preceding argument would give, again, $r \stackrel{\alpha/\omega}{\rightarrow} r_2$, for some $r_2 \in S$. Hence, $\lambda(s, \alpha) = \omega = \lambda(r, \omega)$. From Definition 6 we conclude that $s \sim r$. □

We can now establish that $\overline{M}$ is p-reduced.

Corollary 2. *Let $\overline{M}$ be the FSM in Definition 13. Then, $\overline{M}$ is p-reduced.*

Proof. Let $[s] \in \overline{S}$. By construction, $s_0 \stackrel{\alpha/\omega}{\rightarrow} s$, for some $\alpha \in \mathcal{I}^\star$, $\omega \in \mathcal{O}^\star$. Hence, Lemma 2(2) gives $\overline{s_0} \stackrel{\alpha/\omega}{\rightarrow} [s]$, because $\overline{s_0} = [s_0]$. Further, if $[s]$ and $[r]$ are distinct, Lemma 8 implies $[s] \not\sim [r]$. □

In the next result, we use the same symbol, $\sim$, to denote the alikeness relations between states of M, and also between states of M and of $\overline{M}$. The context will always make clear which relation we are referring to.

Lemma 9. *Let M be a FSM and $s, r \in S$. Let $\overline{M}$ be the FSM in Definition 13. If $s \sim r$, then $s \sim [r]$.*

Proof. We first show that $U(s) \ominus U([r]) = \emptyset$. Let $\alpha \in U(s)$. Since $s \sim r$, Lemma 2(2) gives $\alpha \in U(r)$. Hence, using Fact 4 we obtain $\alpha \in U([r])$, and so $U(s) \subseteq U([r])$. Conversely, let $\alpha \in U([r])$. Then, Fact 5 gives $\alpha \in U(r_1)$, where $r_1 \sim r$. Thus, $r_1 \sim s$, and so using Lemma 2(2) we get $\alpha \in U(s)$. This shows $U([r]) \subseteq U(s)$ and we may conclude that $U(s) = U([r])$. Hence, $U(s) \ominus U([r]) = \emptyset$ using Remark 1, as desired.

Now, let $\alpha \in U(s) \cap U([r])$. Then, $s \stackrel{\alpha/\omega}{\rightarrow} s_1$, for some $s_1 \in S$, $\omega \in \mathcal{O}^\star$, and also $[r] \stackrel{\alpha/\rho}{\rightarrow} [r_1]$, for some $[r_1] \in \overline{S}$, $\rho \in \mathcal{O}^\star$. In order to get $\lambda(s, \alpha) = \overline{\lambda}([r], \alpha)$ we just show that $\omega = \rho$. From $s \sim r$, and using Lemma 2(2), we have $r \stackrel{\alpha/\omega}{\rightarrow} r_2$, for some $r_2 \in S$ with $r_2 \sim s_1$. Hence, by Fact 4 we get $[r] \stackrel{\alpha/\omega}{\rightarrow} [r_2]$. The determinism of $\overline{M}$ now gives $\omega = \rho$. □

We can now say that the p-reduction construction preserves alikeness.

Corollary 3. *Let M be a FSM and let $\overline{M}$ be the FSM in Definition 13. Then, $M \sim \overline{M}$.*

Proof. Since $s_0 \sim s_0$, Lemma 9 gives $s_0 \sim [s_0]$, and we know that, by construction, $\overline{s_0} = [s_0]$. □

Besides preserving alikeness, the construction also yield bi-simulating machines.

Lemma 10. *Let M be a FSM and let $\overline{M}$ be the FSM in Definition 13. Then, M and $\overline{M}$ are bi-similar.*

Proof. Define the relation $R \subseteq S \times \overline{S}$ by letting $(s, [r]) \in R$ iff $s \sim r$. Clearly, $(s_0, [s_0]) \in R$. Now, let $(s, [r]) \in R$ with $s \stackrel{x/a}{\rightarrow} p$ for some $p \in S$, $x \in \mathcal{I}$, $a \in \mathcal{O}$. Since $s \sim r$, Lemma 2(1) gives $r \stackrel{x/a}{\rightarrow} q$ for some $q \in S$ with $q \sim p$. Then Fact 4 gives $[r] \stackrel{x/a}{\rightarrow} [q]$. But $(p, [q]) \in R$, and we conclude that R is a simulation relation. For the other direction, define the relation $L \subseteq \overline{S} \times S$ where $([r], s) \in L$ iff $r \sim s$. Again $([s_0], s_o) \in L$ clearly holds. Let $([s], r) \in L$ with $[s] \stackrel{x/a}{\rightarrow} [q]$ for some $[q] \in \overline{S}$, $a \in \mathcal{O}$, $x \in \mathcal{I}$. By Fact 5, we get $s_1 \stackrel{x/a}{\rightarrow} q_1$ for some $s_1, q_1 \in S$ with $s \sim s_1$ and $q \sim q_1$. Since $([r], s) \in L$, we have $s \sim r$, and so $r \sim s_1$. From $s_1 \stackrel{x/a}{\rightarrow} q_1$ we conclude that $r \stackrel{x/a}{\rightarrow} q_2$, for some $q_2 \in S$ with $q_2 \sim q_1$, using Lemma 2(1). Thus, $q_2 \sim q$, and so $([q], q_2) \in L$, and we conclude that L is also a simulation relation. □

The desired converse to Corollary 1 can now be established.

Corollary 4. *Let M be a p-reduced FSM and let T be a test suite for M. Assume that all p-reduced T-alike FSMs are isomorphic to M. Then T is perfect for M.*

Proof. In view of Theorem 2, it suffices to show that any FSM that is T-alike to M is also bi-similar to M. Let N be T-alike to M. Let $\overline{N}$ be as in Definition 13. By Corollary 2 N is p-reduced, and by Corollary 3 we have $N \sim \overline{N}$. Now, in view of Remark 2(2) we conclude that $N \sim_T \overline{N}$. Since we already have $M \sim N$, using Lemma 1 and Remark 3, we conclude that $M \sim \overline{N}$. So, $\overline{N}$ is p-reduced and T-alike M. By the hypothesis we know that M and $\overline{N}$ are isomorphic. Hence, using Theorem 4, we know that M and $\overline{N}$ are bi-similar. But $\overline{N}$ and N are also bi-similar, using Lemma 10. Using Fact 1, we conclude that M and N are bi-similar, as desired. □

Now we can combine the preceding results and those of the previous subsection to characterize perfectness in terms of isomorphisms.

Theorem 5. *Let M be a p-reduced FSM and let T be a test suite for M. Then T is perfect for M iff all p-reduced T-alike FSMs are isomorphic to M.*

Proof. Use Corollaries 1 and 4. □

4 Completeness and Perfectness

In this section we investigate the relationship between completeness and perfectness. We show that a test suite T that is not n-complete for a FSM M can not also be perfect for M, for any $n \geq 1$. In the other direction, we also show that there are test suites T which are perfect for M, but not n-complete for M, for $n \geq 2$.

We start by showing that perfectness only holds when n-completeness also holds. Let M be a FSM and let T be a test suite for M. We want to prove that if T is not n-complete for M, then T is not perfect for M, where $n \geq 1$. This will show that perfectness is at least as strong a condition as is completeness.

First, we need a measure on the length of blocking test cases in a test suite. Let $\alpha \in \mathcal{I}^\star$ be an input string for M. Define $F(M, \alpha)$ as:

$$F(M,\alpha) = \max\left\{|\beta| \,:\, \alpha = \beta x \gamma, \text{with } \beta \in U(s_0),\, \beta x \notin U(s_0),\, x \in \mathcal{I},\, \gamma \in \mathcal{I}^\star\right\}.$$

That is, $F(M, \alpha)$ is the maximum length of a prefix of α which does not block in M. For a test suite $T \subseteq \mathcal{I}^\star$ we overload the notation and define $F(M, T) = \sum_{\alpha \in T} F(M, \alpha)$.

Fact 6. *Given a FSM M and a test suite T for M, we have the upper bound $F(M,T) \leq \sum_{\alpha \in T} |\alpha|$.*

Proof. Immediate. □

Now, fix a FSM M, a test suite T, and assume that T is not n-complete for M, for some $n \geq 1$. Then, there is a FSM N such that $M \not\approx N$ and $M \approx_T N$. So, we have some $\sigma = x_1 x_2 \ldots x_{n+1}$, where $n \geq 0$ and $x_i \in \mathcal{I}$ $(1 \leq i \leq n+1)$, and such that

$$\sigma \notin T \qquad \text{and} \qquad \sigma \in U(s_0). \tag{1}$$

Let

$$s_0 \overset{x_1/a_1}{\rightarrow} s_1 \overset{x_2/a_2}{\rightarrow} s_2 \cdots s_{n-1} \overset{x_n/a_n}{\rightarrow} s_n \overset{x_{n+1}/a_{n+1}}{\rightarrow} s_{n+1}. \tag{2}$$

We show how to construct a sequence of FSMs N_i that satisfy, for all $i \geq 0$:

1. N_i is a labelled tree rooted at q_0.
2. $\sigma \in U_i(q_0)$.
3. for all $\alpha \in U_i(q_0) \cap T$ we have:
   (a) $\alpha \in U(s_0)$.
   (b) If $q_0 \overset{\alpha/\omega}{\underset{N_i}{\rightarrow}}$ and $s_0 \overset{\alpha/\eta}{\underset{M}{\rightarrow}}$, then $\omega = \eta$.

In order to ease the notation, we denote the states in each N_i as q_0, q_1, q_2, …, with q_0 the initial state. Moreover, by $U_i(q_0)$ we mean the set of all input strings α such that $q_0 \overset{\alpha/\omega}{\underset{N_i}{\rightarrow}}$, for some output string ω.

We start by defining N_0 as the FSM containing the transitions:

$$q_0 \overset{x_1/a_1}{\rightarrow} q_1 \overset{x_2/a_2}{\rightarrow} q_2 \cdots s_{n-1} \overset{x_n/a_n}{\rightarrow} q_n \overset{x_{n+1}/b}{\rightarrow} q_{n+1}, \tag{3}$$

where $b \neq a_{n+1}$. It is clear that N_0 is a labelled tree rooted at q_0, and that $\sigma \in U_0(q_0)$, and so properties (1) and (2) hold for N_0. Now, let $\alpha \in U_0(q_0) \cap T$. Since $\sigma \notin T$, we conclude that α is a prefix of $x_1 x_2 \cdots x_n$, and so property (3) also holds for N_0.

Now assume that N_i has been constructed satisfying properties (1) – (3), for some $i \geq 0$. If there is some input string $\alpha \in U(s_0) \cap T$ such that $\alpha \notin U_i(q_0)$ we show how to construct N_{i+1}. Since $\alpha \notin U_i(q_0)$, we can write $\alpha = y_1 y_2 \cdots y_k x \beta$, where $k \geq 0$, $y_j \in \mathcal{I}$ $(1 \leq j \leq k)$,

$x \in \mathcal{I}$, and where we also have $y_1 y_2 \cdots y_k \in U_i(q_0)$, $y_1 y_2 \cdots y_k x \notin U_i(q_0)$. So, in N_i we have the transitions

$$r_0 \overset{y_1/b_1}{\to} r_1 \overset{y_2/b_2}{\to} r_2 \cdots r_{k-1} \overset{y_k/b_k}{\to} r_k \tag{4}$$

with $r_0 = q_0$ and with no transition out of r_k on input x. Since $\alpha \in U(s_0)$, in M we get

$$p_0 \overset{y_1/b_1}{\to} p_1 \overset{y_2/b_2}{\to} p_2 \cdots p_{k-1} \overset{y_k/b_k}{\to} p_k \overset{x/c}{\to} p_{k+1}, \tag{5}$$

for some $c \in \mathcal{I}$ and with $p_0 = s_0$. We define N_{i+1} from N_i by adding to it a transition $r_k \overset{x/c}{\to} r$, and where r is a new state not present in N_i.

Since N_i is a labelled tree rooted at q_0, then so is N_{i+1} because r is a new state. Then property (1) holds for N_{i+1}. Also, since all transitions from N_i are present in N_{i+1}, then property (2), trivially, also holds for N_{i+1}.

Now, let $\gamma \in U_{i+1}(q_0) \cap T$. Since $\gamma \in U_{i+1}(q_0)$ we have two cases:

- Case 1: the new transition $r_k \overset{x/c}{\to} r$ does not occur in γ. Then, clearly, $\gamma \in U_i(q_0)$, and so (3a) and (3b) hold because N_i satisfies property (3).

- Case 2: the new transition $r_k \overset{x/c}{\to} r$ occurs in γ. Since r is a new state, we can write $\gamma = \theta x$, where $\theta \in U_i(q_0)$ and $q_0 \underset{N_{i+1}}{\overset{\theta/\eta}{\to}} r_k \underset{N_{i+1}}{\overset{x/c}{\to}} r$. Since N_i is a tree rooted at q_0, there is only one path from q_0 to r_k. Hence, from Eq. (4) we get $\theta = y_1 y_2 \cdots y_k$, and $\eta = b_1 b_2 \cdots b_k$. From Eq. (5) we get $s_0 \underset{M}{\overset{\theta/\eta}{\to}} p_k \underset{M}{\overset{x/c}{\to}} p_{k+1}$, and property (3) holds for N_{i+1}.

We conclude that properties (1) – (3) hold for N_{i+1}, as desired.

Because $\alpha = y_1 y_2 \cdots y_k x \beta$, $y_1 y_2 \cdots y_k x \notin U_i(q_0)$ and the construction of N_{i+1} gives $y_1 y_2 \cdots y_k x \in U_{i+1}(q_0)$ we conclude that $F(N_i, \alpha) < F(N_{i+1}, \alpha)$. Since we also have $\alpha \in T$, we then get $F(N_i, T) < F(N_{i+1}, T)$.

The preceding discussion makes it clear that we can construct the sequence of FSMs N_0, N_1, ... satisfying properties (1) – (3), and with $F(N_i, T) < F(N_{i+1}, T)$, as long as we have input strings $\alpha_i \in U(s_0) \cap T$ such that $\alpha_i \notin U_i(q_0)$, $i \geq 0$.

Fact 7. *There is some $\ell \geq 0$ such that there is no $\alpha \in U(s_0) \cap T$ and such that $\alpha \notin U_\ell(q_0)$.*

Proof. Fact 6 gives an upper limit to the sequence $F(N_0, T) < F(N_1, T) < \cdots$. □

From Eq (1), we take a test case $\sigma \notin T$, and use the fact that the construction gives $\sigma \in U(q_\ell)$ to show that T is not, in fact, perfect for M.

From Eqs. (1) and (2) we can write $s_0 \underset{M}{\overset{\sigma/\omega a_{n+1}}{\to}}$, where $\omega = a_1 a_2 \cdots a_n$. From Eq. (3) and property (2), we get $s_0 \underset{N_\ell}{\overset{\sigma/\omega b}{\to}}$. Since $a_{n+1} \neq b$ we conclude that $M \not\sim N_\ell$. If T was perfect for M we would have $M \not\sim_T N_\ell$. We now show that this leads to contradictions. There are two cases:

- Case A: there is some input string $\alpha \in U(s_0) \cap U_\ell(q_0) \cap T$ such that $s_0 \underset{M}{\overset{\alpha/\omega_1}{\to}}$, $q_0 \underset{N_\ell}{\overset{\alpha/\omega_2}{\to}}$, and $\omega_1 \neq \omega_2$. This contradicts property (3b).

- Case B: there is some input string $\alpha \in (U(s_0) \ominus U_\ell(q_0)) \cap T$. If $\alpha \in U_\ell(q_0) \cap T$ and $\alpha \notin U(s_0)$, we contradict property (3a). If $\alpha \in U(s_0) \cap T$ and $\alpha \notin U_\ell(q_0)$, we contradict Fact 7.

We conclude that T is not perfect for M.

Fact 8. *Let M be a FSM, and let T be a test suite that is not n-complete for M, for some $n \geq 1$. Then, T is not perfect for M.*

Proof. From the preceding discussion. □

Next, we also show that when T is n-complete for M, $n \geq 1$, it may be the case that T is not perfect for M. Let the input and output alphabets be $\mathcal{I} = \mathcal{O} = \{0, 1\}$, and let M be the specification with n states given by the transitions $s_i \stackrel{0/0}{\rightarrow} s_{i+1}$, $0 \leq i < n$. Let $T = \{0^n, 0^{n-1}\}$ be a test suite for M. We argue that T is n-complete for M. From Definitions 3 and 4, if that were not the case, we would have a FSM N with $U(s_0) \subseteq U(q_0)$, and such that $M \not\approx N$ and $M \approx_T N$. Since $U(s_0) \subseteq U(q_0)$ and $U(s_0) = \{0^{n-1}\}$, we get $U(s_0) \cap U(q_0) \cap T = \{0^{n-1}\}$. Hence $M \approx_T N$ gives $\lambda(s_0, 0^{n-1}) = 0^{n-1} = \mu(q_0, 0^{n-1})$. Since we also have $U(s_0) \cap U(q_0) \cap \mathcal{I}^\star = \{0^{n-1}\}$, Definition 3 and $M \not\approx N$ would require $\lambda(s_0, \alpha) \neq \mu(q_0, \alpha)$ for some $\alpha \in \{0^{n-1}\}$, and we reached a contradiction.

We now argue that $T = \{0^n, 0^{n-1}\}$ is not perfect for M. Let N be the FSM with the transitions $q_i \stackrel{0/0}{\rightarrow} q_{i+1}$ for $0 \leq i < n$, and also $q_{n-1} \stackrel{1/1}{\rightarrow} q_{n-1}$. It is clear that $0^{n-1}1 \in (U(s_0) \ominus U(q_0)) \cap \mathcal{I}^\star$. Hence, from Definition 6, we see that $M \not\sim N$. Since $T = \{0^n, 0^{n-1}\}$, we get $(U(s_0) \ominus U(q_0)) \cap T = \emptyset$. Also, $U(s_0) \cap U(q_0) \cap T = \{0^{n-1}\}$, and so $\lambda(s_0, \alpha) = \mu(q_0, \alpha)$ for all $\alpha \in U(s_0) \cap U(q_0) \cap T$. From Definition 6 we get $M \sim_T N$. Hence, Definition 7 says that T is not perfect for M.

Corollary 5. *Let M be a FSM. Then the following holds:*

1. *If T is a test suite which is perfect for M, then T is also n-complete for M, for all $n \geq 1$.*
2. *For all $n \geq 1$ there are test suites which are n-complete but not perfect for M.*

Proof. From Fact 8 and the preceding discussion. □

5 Test Suite Completeness and the Size of Implementations

In this section we show that if one allows for too large implementations, then test completeness, in the classical sense, is lost. More specifically, if T is a test suite for a FSM M, then T is not m-complete for M for every $m \geq g(T) + |S|$, where $g(T)$ is a constant depending only on T, and $|S|$ is the number of states in M. This means that T may not be able to detect all faults in implementations with m or more states. Moreover, we show that for all pairs (n, k), with $n \geq 2$ and $k \geq 1$, there is a reduced FSM M with $|S| = n$ states and a test suite T with $g(T) = k$ such that T is not $(n + k)$-complete for M, but T

is $(n+k-1)$-complete for M. This infinite family of FSMs and test suites shows that the bound $g(T)+|S|$ is sharp.

First, we establish some notation. Let M be a FSM, T a test suite for M and $\sigma \in T$. We say that σ is *extensible* in M and T if and only if for some α_1, $\alpha_2 \in \mathcal{I}^\star$ we have $\alpha_1\sigma\alpha_2 \in T$ where $s_0 \stackrel{\alpha_1/}{\rightarrow} s_0$ and $\alpha_2 \neq \varepsilon$. Otherwise, σ is *non-extensible* in T.

Remark 6. *If $T \cap U(s_0) = \emptyset$ then any FSM is trivially T-equivalent to M. Also, if $T = \{\varepsilon\}$ then, again, any FSM is trivially T-equivalent to M. Since M is reduced, one can easily construct a one-state FSM that is not equivalent to M. Hence, in both cases, T would not be 1-complete for M. We, therefore, can assume that there is a non-null $\sigma \in T \cap U(s_0)$. Clearly, we then get a non-extensible test case in $T \cap U(s_0)$.*

Throughout this section we fix a reduced FSM M with $|S| \geq 2$ states and a test suite T for M. Also, we fix $\sigma = x_0x_1\cdots x_k$, $k \geq 0$, as a smallest non-extensible test case in $T \cap U(s_0)$, and define $g(T) = |\sigma| - 1 = k$.

5.1 A tight upper bound for m-completeness

A tight upper bound for m-completeness The following construction, and the discussion in the sequel, will give us the desired upper bound on the size of implementations when testing for completeness.

Since $\sigma \in U(s_0)$, we get transitions in M:

$$\pi_i : s_i \stackrel{x_i/a_i}{\rightarrow} s_{i+1} \quad 0 \leq i \leq k. \tag{6}$$

Let $\omega = a_0a_1\ldots a_k$, so that $s_0 \stackrel{\sigma/\omega}{\rightarrow} s_{k+1}$.

We now construct a FSM N using the same input and output alphabets, respectively $\mathcal{I}$ and $\mathcal{O}$, of M. A simple example illustrating the construction is presented right after Theorem 6. Let $Q = S \cup R$, where $R = \{r_1, \ldots, r_k\}$ are new sates, that is, $S \cap R = \emptyset$ and $r_i \neq r_j$ $(1 \leq i < j \leq k)$. The initial state of N is the same as in M, that is, $q_0 = s_0$. For the ease of notation, we also define $r_0 = q_0$. Note that $|Q| = |S| + k = |S| + g(T)$.

We start the specification of N:

(a) Make all transitions of M also transitions of N, except that $\pi_0 : s_0 \stackrel{x_0/a_0}{\rightarrow} s_1$ in M is redirected to $\pi_0' : q_0 \stackrel{x_0/a_0}{\rightarrow} r_1$ in N.

(b) Replicate transitions from s_i: add the transitions $r_i \stackrel{x_i/a_i}{\rightarrow} r_{i+1}$ to N, for $1 \leq i < k$.

(c) Add return transitions from r_i: if $s_i \stackrel{z/b}{\rightarrow} s$ is in M with $z \neq x_i$, add $r_i \stackrel{z/b}{\rightarrow} s$ to N, $1 \leq i \leq k$.

(d) Transition from r_k: add $r_k \stackrel{x_k/a_k}{\rightarrow} r_{k+1}$ to N, where $r_{k+1} \neq s_{k+1}$.

We will indicate how to precisely choose r_{k+1} in the sequel. To ease the notation, define $\hat{s} = r_{k+1}$.

Next, we want to guarantee that $U(s_0) \subseteq U(q_0)$, but since $s_{k+1} \neq \hat{s}$ we could conceivably have some $\alpha \in U(s_{k+1})$ with $\alpha \notin U(\hat{s})$. So, we extend the specification of N in such a way

that all runs from s_{k+1} in M are also runs from $\hat{s}$ in N. More specifically, we extend N as follows:

(e) While we have $s_{k+1} \stackrel{\alpha/}{\to} p \stackrel{x/a}{\to} t$ in M and $\hat{s} \stackrel{\alpha/}{\to} q$ in N, with $\alpha \in \mathcal{I}^\star$, $x \in \mathcal{I}$, $a \in \mathcal{O}$, and there is no transition $q \stackrel{x/}{\to}$ in N, add $q \stackrel{x/a}{\to} t$ to N.

Since both M and N are finite, the construction clearly halts. Moreover, N is deterministic since M is deterministic.

Remark 7. *Item (e) of the construction clearly guarantees that if $s_{k+1} \stackrel{\alpha/}{\to}$ in M, then we also have $\hat{s} \stackrel{\alpha/}{\to}$ in N. Also, note that if a transition $s \stackrel{x/a}{\to} p$ is in M, then this transition is also in N, with the only exception that $s_0 \stackrel{x_0/a_0}{\to} s_1$ is also in N but redirected to r_1 when $k \geq 1$, or redirected to $\hat{s}$ when $k = 0$.*

In order to prove non-completeness, we need to relate runs in M to runs in N. The next lemma develops the main idea.

Lemma 11. *Let $p_0 \stackrel{\alpha/\beta}{\to}$ in M for some $\alpha \in \mathcal{I}^\star$, $\beta \in \mathcal{O}^\star$. Then, $p_0 \stackrel{\alpha/\gamma}{\to}$ in N, for some $\gamma \in \mathcal{O}^\star$. Further, if $\beta \neq \gamma$ then we must have $\alpha = \alpha_1\sigma\alpha_2$ with $p_0 \stackrel{\alpha_1/}{\to} s_0$ in M and $\alpha_2 \neq \varepsilon$.*

Proof. Let $\alpha = y_1 \cdots y_m$, $\beta = b_1 \cdots b_m$ with $m \geq 0$, $y_i \in \mathcal{I}$ and $b_i \in \mathcal{O}$ $(1 \leq i \leq m)$. In M we then have

$$p_0 \stackrel{y_1/b_1}{\to} p_1 \stackrel{y_2/b_2}{\to} \to \cdots \stackrel{y_m/b_m}{\to} p_m. \tag{7}$$

If the transition $\pi_0 : s_0 \stackrel{x_0/a_0}{\to} s_1$ does not occur in (7), then Remark 7 says that all those transitions are also in N, and we immediately have $p_0 \stackrel{\alpha/\beta}{\to}$ in N, and the result holds.

Otherwise assume that $p_j \stackrel{y_j/b_j}{\to} p_{j+1}$ is the first occurrence of π_0 in (7), $1 \leq j < m$, so that $p_j = s_0$, $p_{j+1} = s_1$, $y_j = x_0$ and $b_j = a_0$. By the minimality of j and Remark 7 we readily get

$$p_0 \stackrel{\alpha_1/\beta_1}{\to} p_j \qquad \text{in both } M \text{ and } N, \text{ with } \alpha_1 = y_1 \cdots y_{j-1},\ \beta_1 = b_1 \cdots b_{j-1}. \tag{8}$$

We have to examine that tail $y_j \cdots y_m$ of α and observe whether σ is a prefix of it or not. Recall that $\sigma = x_0x_1 \cdots x_k$ and that $y_j = x_0$.

Case A: σ is not a prefix. Then $k \geq 1$ and there is some $0 \leq \ell \leq k-1$ such that $y_{j+i} = x_i$ $(0 \leq i \leq \ell)$ and either (i) the tail is short, that is, $m = j + \ell$, or (ii) the tail is long enough, that is, $m > j + \ell$ and $y_{j+\ell+1} \neq x_{\ell+1}$.

Let $\alpha_2 = x_0 \cdots x_\ell = y_j \cdots y_{j+\ell}$ and let $\beta_2 = a_0 \cdots a_\ell = b_j \cdots b_{j+\ell}$, so that we may write $y_j \cdots y_m = \alpha_2 y_{j+\ell+1} \cdots y_m$ and $b_j \cdots b_m = \beta_2 b_{j+\ell+1} \cdots b_m$. Thus $\alpha = \alpha_1\alpha_2 y_{j+\ell+1} \cdots y_m$ and $\beta = \beta_1\beta_2 b_{j+\ell+1} \cdots b_m$.

Since $p_j = s_0$, in view of (6) we get $p_j \stackrel{\alpha_2/\beta_2}{\to} p_{j+\ell+1}$ in M, with $p_{j+\ell+1} = s_{\ell+1}$. Because of items (a) and (b) in the construction, we also have $s_0 \stackrel{\alpha_2/\beta_2}{\to} r_{\ell+1}$ in N. Combining

with (8) we now have

$$p_0 \overset{\alpha_1/\beta_1}{\to} p_j \overset{\alpha_2/\beta_2}{\to} s_{\ell+1} \text{ in } M \quad \text{and} \quad p_0 \overset{\alpha_1/\beta_1}{\to} p_j \overset{\alpha_2/\beta_2}{\to} r_{\ell+1} \text{ in } N, \tag{9}$$

So, if (i) holds and $m = j + \ell$ we get $\alpha = y_0 \cdots y_m = \alpha_1\alpha_2$ and $\beta = b_0 \cdots b_m = \beta_1\beta_2$, giving the desired result.

If (ii) holds with $m \geq j+\ell+1$ and $y_{j+\ell+1} \neq x_{\ell+1}$, then we may write $y_j \cdots y_m = \alpha_2 z \alpha_3$ with $z = y_{j+\ell+1}$ and $\alpha_3 = y_{j+\ell+2} \cdots y_m$. Likewise, $b_j \cdots b_m = \beta_2 c \beta_3$ with $c = b_{j+\ell+1}$ and $\beta_3 = b_{j+\ell+2} \cdots b_m$. Note that $\ell + 1 \leq k$ and $p_{j+\ell+1} = s_{\ell+1}$. So (6) gives $s_{\ell+1} \overset{z/c}{\to} t$ in M, with $t = p_{j+\ell+2}$. Also, since $y_{j+\ell+1} \neq x_{\ell+1}$ we get $z \neq x_{\ell+1}$ and item (c) of the construction gives $r_{\ell+1} \overset{z/c}{\to} t$ in N. Together with (9) we conclude that

$$p_0 \overset{\alpha_1\alpha_2/\beta_1\beta_2}{\to} s_{\ell+1} \overset{z/c}{\to} t \text{ in } M \quad \text{and} \quad p_0 \overset{\alpha_1\alpha_2/\beta_1\beta_2}{\to} r_{\ell+1} \overset{z/c}{\to} t \text{ in } N. \tag{10}$$

Since $t = p_{j+\ell+2}$, from (6) we get $t \overset{\alpha_3/\beta_3}{\to}$ in M. But, there are one less occurrences of π_0 in the transitions from t onwards in (6) so that, inductively, we conclude that $t \overset{\alpha_3/\gamma}{\to}$ in N. Combining with (10), we get $p_0 \overset{\alpha/\rho}{\to}$ in N with $\rho = \beta_1\beta_2 c\gamma$. If $\beta = \rho$ we are done. If not, recalling that $\beta = \beta_1\beta_2 c\beta_3$, we get $\beta_3 \neq \gamma$, and the induction now gives $\alpha_3 = \alpha_4\sigma\alpha_5$, for some α_4, $\alpha_5 \in \mathcal{I}^\star$ such that $t \overset{\alpha_4/}{\to} s_0$ in M and $\alpha_5 \neq \varepsilon$. Hence, $\alpha = \alpha_1\alpha_2 z\alpha_3 = \eta\sigma\alpha_5$ where $\eta = \alpha_1\alpha_2 z\alpha_4$, $\alpha_5 \neq \varepsilon$ and, using(10) again, we get $p_0 \overset{\alpha_1\alpha_2 z/}{\to} t \overset{\alpha_4/}{\to} s_0$ in M, that is $p_0 \overset{\eta/}{\to} s_0$ in M, as desired. This concludes Case A.

Case B: σ is a prefix. Then we have $j + k \leq m$ and $y_j \cdots y_m = x_0 \cdots x_k\alpha_2 = \sigma\alpha_2$ and $b_j \cdots b_m = a_0 \cdots a_k\beta_2 = \omega\alpha_2$, with $y_{j+i} = x_i$, $b_{j+i} = a_i$ $(0 \leq i \leq k)$, and $\alpha_2 = y_{j+k+1} \cdots y_m$, $\beta_2 = b_{j+k+1} \cdots b_m$.

Since $s_0 = p_j$, using (6) we obtain $p_j \overset{\sigma/\omega}{\to} p_{j+k+1}$ in M, with $p_{j+k+1} = s_{k+1}$. From items (b) and (d) of the construction, we get $p_j \overset{\sigma/\omega}{\to} \hat{s}$ in N. Combined with (8) we get

$$p_0 \overset{\alpha_1/\beta_1}{\to} p_j \overset{\sigma/\omega}{\to} s_{k+1} \text{ in } M \qquad \text{and} \qquad p_0 \overset{\alpha_1/\beta_1}{\to} p_j \overset{\sigma/\omega}{\to} \hat{s} \text{ in } N. \tag{11}$$

If $j + k = m$ we get $\alpha_2 = \beta_2 = \varepsilon$ and so $y_0 \cdots y_m = \alpha_1\sigma$, $b_0 \cdots b_m = \beta_1\omega$, thus establishing the desired result.

If $m > j + k$ we get $\alpha_2 = z\alpha_3$ with $z = y_{j+k+1}$, $\alpha_3 = y_{j+k+2} \cdots y_m$, and $\beta_2 = c\beta_3$ with $c = b_{j+k+1}$, $\beta_3 = b_{j+k+2} \cdots b_m$. Since $p_{j+k+1} = s_{k+1}$, from (6) we get $s_{k+1} \overset{z/c}{\to} t$ in M, with $t = p_{j+k+2}$. But now item (e) of the construction gives $\hat{s} \overset{z/d}{\to} q$ in N, for some $d \in \mathcal{O}$, $q \in S$. Combining with (11) we can now write

$$p_0 \overset{\alpha_1/\beta_1}{\to} p_j \overset{\sigma/\omega}{\to} s_{k+1} \overset{z/c}{\to} t \text{ in } M \qquad \text{and} \qquad p_0 \overset{\alpha_1/\beta_1}{\to} p_j \overset{\sigma/\omega}{\to} \hat{s} \overset{z/d}{\to} q \text{ in } N. \tag{12}$$

Recall that $\alpha = \alpha_1 \sigma z \alpha_3$ with $p_0 \stackrel{\alpha_1/}{\rightarrow} p_j$ in M, $p_j = s_0$, and $z\alpha_3 \neq \varepsilon$. Hence, to establish the result it is enough to show that $q \stackrel{\alpha_3/\eta}{\rightarrow}$ in N, for some $\eta \in \mathcal{O}^\star$ since, together with (12), this would give $p_0 \stackrel{\alpha/\gamma}{\rightarrow}$ in N where $\gamma = \beta_1 \omega d \eta$. To see that $q \stackrel{\alpha_3/\eta}{\rightarrow}$ in N, note that $t = p_{j+k+2}$, and so, using (6), we get $s_{k+1} \stackrel{z/c}{\rightarrow} t \stackrel{\alpha_3/}{\rightarrow}$ in M. Since we already have $\hat{s} \stackrel{z/d}{\rightarrow} q$ in N, the claim follows from an easy induction on $|\alpha_3| \geq 0$ using item (e) of the construction. This concludes Case B.

The the lemma is thus established. □

Now, let $\alpha \in U(s_0)$ in M, that is $s_0 \stackrel{\alpha/}{\rightarrow}$ in M. Since $q_0 = s_0$, Lemma 11 gives $q_0 \stackrel{\alpha/}{\rightarrow}$ in N, so that $\alpha \in U(q_0)$. We conclude that $U(s_0) \subseteq U(q_0)$.

Our next step is to show that $M \approx_T N$. Assume that we have $\alpha \in T$ such that $s_0 \stackrel{\alpha/\beta}{\rightarrow}$ in M and $q_0 \stackrel{\alpha/\gamma}{\rightarrow}$ in N with $\beta \neq \gamma$. Since $s_0 = q_0$ we may use Lemma 11 and obtain $\alpha_1, \alpha_2 \in \mathcal{I}^\star$ such that $\alpha = \alpha_1 \sigma \alpha_2$ with $s_0 \stackrel{\alpha_1/}{\rightarrow} s_0$ in M and $\alpha_2 \neq \varepsilon$. This shows that σ is extensible in M and T, contrary to our choice of σ. So, such an $\alpha \in T$ does not exist and we conclude that $M \approx_T N$.

Next, we want to argue that $M \not\approx N$. At this point we make precise our choice of $\hat{s} = r_{k+1}$ in item (d) of the construction of N. The choice of $\hat{s}$ will depend on the structure of M. To facilitate the notation we will write $\bar{x}$ to be 0 when $x = 1$, and $\bar{x}$ to be 1 when $x = 0$. Now, recall that $n \geq 2$ and that M is reduced, so that s_{k+1} must be distinguishable from any other state in M. So, we know that $s_{k+1} \stackrel{y/a}{\rightarrow}$ in M for some $y \in \mathcal{I}$, $a \in \mathcal{O}$. If there is a state t in M such that $t \stackrel{y/\bar{a}}{\rightarrow}$, then we choose $\hat{s} = t$. Clearly, from (6) we have $s_0 \stackrel{\sigma/\omega}{\rightarrow} s_{k+1} \stackrel{y/a}{\rightarrow}$ in M. From items (a), (b) and (d) of the construction, and from our choice of $\hat{s}$, we get $q_0 \stackrel{\sigma/\omega}{\rightarrow} \hat{s} \stackrel{y/\bar{a}}{\rightarrow}$ in N. Since $\omega a \neq \omega \bar{a}$ we conclude that $M \not\approx N$. Now assume that there are no state t such that $t \stackrel{y/\bar{a}}{\rightarrow}$ in M, so that if any state of M has a transition on input y, then the output must be a. If s_{k+1} has a transition on input $\bar{y}$, say $s_{k+1} \stackrel{\bar{y}/b}{\rightarrow}$, then either there is a state t in M with $t \stackrel{\bar{y}/\bar{b}}{\rightarrow}$, in which case we can choose $\hat{s} = t$ and proceed as before, or any other state in M that has a transition on input $\bar{y}$ must have b as output. But in this case, all states of M will have output a on input a, and will have output b on input $\bar{y}$, and no two states of M will be distinguishable, contrary to the fact that M is reduced. We then, conclude that $s_{k+1} \stackrel{y/a}{\rightarrow}$ is the only transition out of s_{k+1}, and that all other states of M that have transition on input y must also output a. Further, we cannot have $s_{k+1} \stackrel{y/a}{\rightarrow} s_{k+1}$ in M for, otherwise, if $s_{k+1} \stackrel{\alpha/\beta}{\rightarrow}$ then we would get $\alpha = y^m$ and $\beta = a^m$, for some $m \geq 0$. But for any other state t, if $t \stackrel{y^m/\gamma}{\rightarrow}$ then we we would get $\gamma = a^m$, so that s_{k+1} would not be distinguishable from any other state of M, a contradiction again. Hence, we must have $s_{k+1} \stackrel{y/a}{\rightarrow} t_1$ for some $t_1 \neq s_{k+1}$.

We now complete the specification of N by choosing $r_{k+1} = \hat{s} = t_1$.

Since t and s_{k+1} are distinguishable in M the distinguishing sequence must terminate

with $\bar{y}$, and we must have some $m \geq 1$ and states t_i in M $(1 \leq i \leq m)$ and some $b \in \mathcal{O}$ such that

$$s_{k+1} \xrightarrow{y/a} t_1 \xrightarrow{y/a} \cdots \xrightarrow{y/a} t_m \xrightarrow{\bar{y}/b} \quad \text{in } M \tag{13}$$

$$t_1 \xrightarrow{y/a} t_2 \xrightarrow{y/a} \cdots \xrightarrow{y/a} t_m \xrightarrow{y/a} t_{m+1} \xrightarrow{\bar{y}/\bar{b}} \quad \text{in } M. \tag{14}$$

Assume that m is minimal, and let $\alpha = y^m$, $\beta = a^m$. Suppose that we also have $t_1 \xrightarrow{\alpha/\beta} q \xrightarrow{\bar{y}/\bar{b}}$ in N. Recalling that $\hat{s} = t_1$, we can use (6) and items (a), (b) and (d) of the construction of N and write

$$s_0 \xrightarrow{\sigma/\omega} s_{k+1} \xrightarrow{\alpha/\beta} t_m \xrightarrow{\bar{y}/b} \quad \text{in } M$$

$$q_0 \xrightarrow{\sigma/\omega} \hat{s} \xrightarrow{\alpha/\beta} q \xrightarrow{\bar{y}/\bar{b}} \quad \text{in } N.$$

Since $\omega\beta b \neq \omega\beta\bar{b}$, we get $M \not\approx N$ again. Otherwise, from (14), Lemma 11 gives some α_1, $\alpha_2 \in \mathcal{I}^\star$ such that $t_1 \xrightarrow{\alpha_1/} s_0$, $\alpha\bar{y} = \alpha_1\sigma\alpha_2$ and $\alpha_2 \neq \varepsilon$. Since N is deterministic, we get

$$t_1 \xrightarrow{\alpha_1/\beta_1} t_j \xrightarrow{\sigma/\omega} t_\ell \xrightarrow{\alpha_2/\beta_2} \quad \text{in } N,$$

where $1 \leq j, \ell \leq m+1$, $\ell = j + |\sigma|$, $\alpha_1 = y^{j-1}$, $\beta_1 = a^{j-1}$, $\alpha_2 = y^{m+1-\ell}\bar{y}$, $\beta_2 = a^{m+1-\ell}\bar{b}$. Hence, $t_j = s_0 = q_0$, and items (a), (b) and (d) of the construction of N imply that $t_\ell = r_{k+1} = \hat{s} = t_1$. So, from (14), we may write $t_1 \xrightarrow{y^{m-\ell}/a^{m-\ell}} t_{m+1} \xrightarrow{\bar{y}/\bar{b}}$ in M. But $\ell \geq 1$ and so $m - \ell < m$ and we get a contradiction to the minimality of m. We, therefore, conclude that, in fact, $M \not\approx N$.

Combining, we have $U(s_0) \subseteq U(q_0)$, $M \approx_T N$ and $M \not\approx N$. Since N has $g(T) + |S|$ state, we conclude that T is not m-complete for M, for any $m \geq g(T) + |S|$.

We summarize this discussion in the next theorem.

Theorem 6. *Let M be a FSM and let T be a test suite for M. Let σ be a shortest test case in T that is non-extensible in $T \cap U(s_0)$. Then T is not m-complete for M, for any $m \geq |\sigma| + |S| - 1$.*

5.2 An upper bound example

We present a simple example to illustrate the construction in Subsection 5.1. Let M be as depicted in Figure 3. Note that M is a partial FSM since $(s_1, 1) \notin D$. Also let $T = \{0^5, 10^2\}$ be a test suite for M. So the shortest test case that is non-extensible in T is $\sigma = 10^2$. Notice that T could have included any other test cases, provided that σ remains as a shortest non-extensible test case in $T \cap U(s_0)$. Also, $|\sigma| + |S| - 1 = 3 + 3 - 1 = 5$ is the bound claimed in Subsection 5.1, and we know that T is not m-complete for M, for any $m \geq 5$.

Applying items (a) to (e), as proposed in Subsection 5, we obtain the FSM N depicted in Figure 4. Item (a) gives us the transitions $s_i \xrightarrow{0/0} s_{i+1}$, $i = 0, 1$, plus both transitions $s_2 \xrightarrow{0/1} s_2$, $s_2 \xrightarrow{1/1} s_2$, and redirects the transition $s_0 \xrightarrow{1/1} s_2$ of M to as the transition $s_0 \xrightarrow{1/1} r_1$

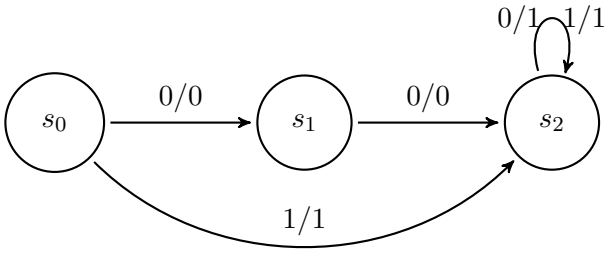

Figure 3: Specification FSM M.

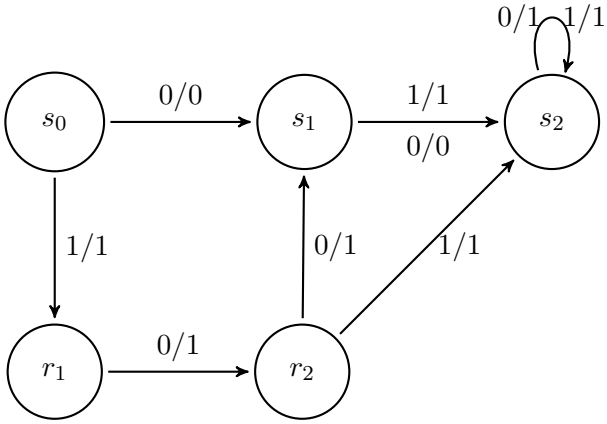

Figure 4: FSM N.

in N. This part of the construction of machine N shows that N and M yield the same behavior when we use test case 0^5.

From item (b) we obtain the transition $r_1 \stackrel{0/1}{\rightarrow} r_2$, and we also obtain $r_2 \stackrel{0/1}{\rightarrow} s_1$ from item (d). At this point of the construction we note that N and M yield the same output when the test case 10^2 is run on both machines.

Next, we get the return transition $r_2 \stackrel{1/1}{\rightarrow} s_2$ from item (c). For item (d) we can choose $\hat{s} = s_1$, and so we also add the transition $r_2 \stackrel{1/1}{\rightarrow} s_1$ to N. We complete the specification of machine N with transitions $s_1 \stackrel{1/1}{\rightarrow} s_2$, as required by item (e).

Since $\lambda(s_0, 00000) = 0011 = \tau(s_0, 00000)$ and $\lambda(s_0, 100) = 111 = \tau(q_0, 100)$ we get $M \approx_T N$. But $M \not\approx N$ because $\lambda(s_0, 1000) = 1111 \neq 1110 = \tau(q_0, 1000)$. We conclude that T is not m-complete for M for any $m \geq 5$, where 5 is the bound specified in Subsection 5.1.

5.3 A lower bound for m-completeness

A lower bound for m-completeness

Next we want to argue that the $|S| + g(T)$ bound is sharp in a strong sense. We show that for all pairs (n, k), with $n \geq 2$ and $k \geq 1$, there is a reduced FSM M with $|S| = n$

states and a test suite T with $g(T) = k$ such that T is not $(n+k)$-complete for M, but T is $(n+k-1)$-complete for M. This infinite family of FSMs and test cases will show that the bound cannot be improved over an infinite family of FSMs and test cases, whose sizes can be as large as desired.

Before giving the general argument, we illustrate the main ideas using a simple example with $n = 3$. Consider the specification depicted in Figure 5, and take the same test suite

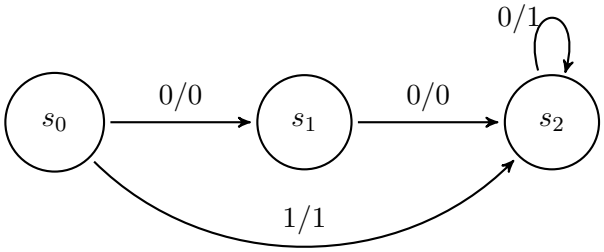

Figure 5: Specification FSM M.

$T = \{0^5, 10^2\}$.

We construct a machine N with at most four states such that $M \approx_T N$, but $M \not\approx N$. In order to maintain the equivalence $M \approx_T N$, preventing the test case 0^5 to distinguish N and M, an alternative for N would be as shown in Figure 6. We could then complete

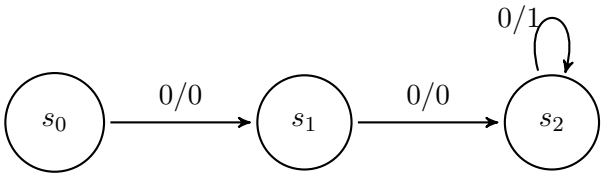

Figure 6: Implementation FSM N.

N with a transition $s_0 \xrightarrow{1/1} s_1$, obtaining a machine that is isomorphic to M, which clearly would imply $M_S \approx N$.

As an alternative to Figure 6, we can extend it with one more state s_3, and terminate by adding the $s_0 \xrightarrow{1/1} s_3$ transition, thus obtaining a 4-state machine as depicted in Figure 7. But then we would also have $M_S \approx_T N_4$ and $M_S \approx N_4$ as it is easily seen.

The last alternative would be to start as in Figure 6, but now use the fourth state s_3 as an intermediate state in a longer path $s_0 \xrightarrow{1/1} s_3 \xrightarrow{0/1} s_2$, as depicted at Figure 8. However, in this situation we still have $M \approx_T N$ and $M \approx N$, as it is easy to check.

A moments reflexion will show that there is no other alternative to construct a machine N with at most 4 states and such that $M \approx_T N$ but $M \not\approx N$. We are lead to conclude that T is indeed m-complete for M, for all $m \leq 4$.

We now proceed with a more formal and general reasoning. Let $n \geq 2$ and consider the FSM M with state set $S = \{s_0, s_1, \ldots, s_{n-1}\}$, and whose transitions are:

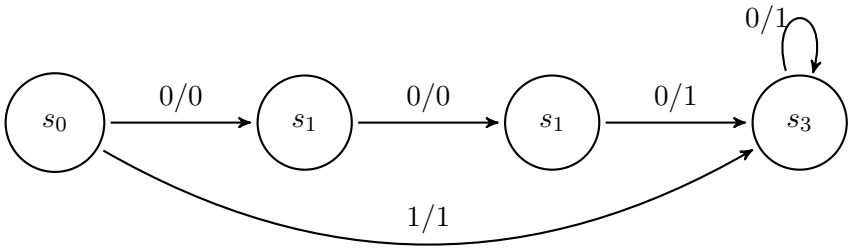

Figure 7: Implementation FSM N.

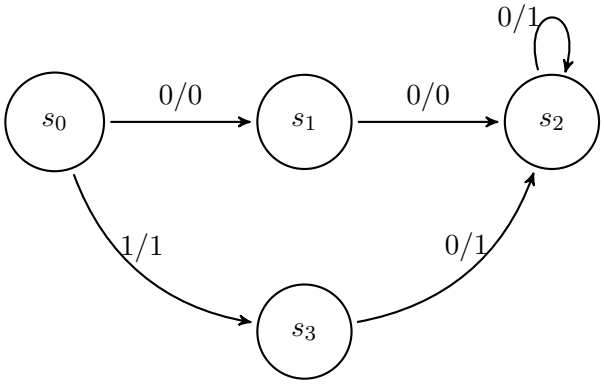

Figure 8: Implementation FSM N.

(i) $s_i \stackrel{0/0}{\rightarrow} s_{i+1}$, for $i = 0, 1, \ldots, n-2$

(ii) $s_{n-1} \stackrel{0/1}{\rightarrow} s_{n-1}$

(iii) $s_0 \stackrel{1/1}{\rightarrow} s_{n-1}$.

Let $k \geq 1$ and let

$$T = \{0^{n+k}, 10^k\}.$$

It is easily seen that M is reduced and deterministic, and that 10^k is the shortest non-extensible test case in T, so that $g(T) = |10^k| - 1 = k$.

For the sake of contradiction, assume that T is not $(n+k-1)$-complete for M. Then we must have an implementation N with at most $n+k-1$ states, $U(s_0) \subseteq U(q_0)$, and such that $M \not\approx N$ and $M \approx_T N$. Clearly, $T \subseteq U(s_0)$ and so we must have $T \subseteq U(q_0)$. Note that $0^{n+k} \in T$ and $s_0 \stackrel{0^{n-1}/0^{n-1}}{\rightarrow} s_{n-1}$ in M, and so in N we get

$$q_0 \stackrel{0/0}{\rightarrow} q_1 \stackrel{0/0}{\rightarrow} q_2 \stackrel{0/0}{\rightarrow} \cdots \stackrel{0/0}{\rightarrow} q_{n-1}.$$

We claim that $q_i \neq q_j$ for $0 \leq i < j \leq n-1$. If not, we would have a loop in the state sequence above and, in this case, N would clearly output 0^{n+k} when the test case $0^{n+k} \in T$ is run on N, but with this same input sequence, M would yield $0^{n-1}1^{k+1}$, contradicting

$M \approx_T N$. Continuing from q_{n-1} the run on N with the test case $0^{n+k} = 0^{n-1}0^{k+1}$ would then yield

$$q_0 \stackrel{0/0}{\to} q_1 \stackrel{0/0}{\to} q_2 \stackrel{0/0}{\to} \cdots \stackrel{0/0}{\to} q_{n-1} \stackrel{0/1}{\to} q_n \stackrel{0/1}{\to} q_{n+1} \stackrel{0/1}{\to} \cdots \stackrel{0/1}{\to} q_{n+k}.$$

Since N has at most $n+k-1$ states, we conclude that $q_u = q_v$ for some $0 \leq u < v < n+k$. Let v is the smallest such index. Since we already know that q_0, q_1, $\ldots$, q_{n-1} are distinct, we conclude that $v > n-1$. Suppose first that $u < n-1$. Note that $n+1 \leq v+1 \leq n+k$. So, on input sequence 0^v0 FSM M would output $0^{n-1}1^\ell 1$ where $\ell = v-(n-1) \geq 1$, and FSM N on the same input would output $0^{n-1}1^\ell 0$, showing that the input sequence 0^v0 distinguishes M and N. Since 0^v0 is a prefix of $0^{n+k} \in T$, this would contradict $M \approx_T N$. We conclude that $u \geq n-1$, and we have a loop in the state sequence q_{n-1}, q_n, $\ldots$, q_{n+k}. By the minimality of v, it follows that q_0, $\ldots$, q_{n-1}, q_n, $\ldots q_{v-1}$ are distinct states of N, and that $q_{v-1} \stackrel{0/1}{\to} q_u$ with $n-1 \leq u \leq v-1$ and $n \leq v \leq n+k-1$. That is, in N we now have

$$q_0 \stackrel{0/0}{\to} q_1 \stackrel{0/0}{\to} q_2 \stackrel{0/0}{\to} \cdots \stackrel{0/0}{\to} q_{n-1} \stackrel{0/1}{\to} q_n \stackrel{0/1}{\to} q_{n+1} \stackrel{0/1}{\to} \cdots \stackrel{0/1}{\to} q_u \stackrel{0/1}{\to} \cdots \stackrel{0/1}{\to} q_{v-1} \stackrel{0/1}{\to} q_u. \tag{15}$$

We also know that $M \not\approx N$, so that we must have some input $\alpha \in U(s_0) \cap U(q_0) \cap \mathcal{I}^\star = U(s_0)$ such that α distinguishes M and N. From the construction of M, there are two cases: (i) $\alpha = 0^m$ for some $m \geq 1$, or (ii) $\alpha = 10^m$, for some $m \geq 0$. Assume first that $\alpha = 0^m$ with $m \geq 1$. When $m \leq n-1$ both M and N would output 0^m, and when $m = (n-1)+\ell$, with $\ell > 0$, both M and N would output $0^{n-1}1^\ell$. So, in this case α does not distinguish between M and N.

Next, take $\alpha = 10^m$, with $m \geq 0$. We clearly need $m > k$ because $10^k \in T$ and we also have $M \approx_T N$. Assume that m is minimal. Since 10^m distinguishes M and N, m is minimal and the output of M over this sequence is $11^{m-1}1$, in N we must have

$$q_0 \stackrel{1/1}{\to} t_0 \stackrel{0/1}{\to} t_1 \stackrel{0/1}{\to} t_2 \stackrel{0/1}{\to} \cdots \stackrel{0/1}{\to} t_{m-1} \stackrel{0/0}{\to} t_m, \tag{16}$$

with corresponding output $11^{m-1}0$. The states t_0, $\ldots$, t_{m-1} must all be distinct for, otherwise, the output of N on 10^m would be 11^m, which is not the case. Consider a state t_i with $0 \leq i \leq m-2$ and assume that we have $t_i = q_j$, for some $0 \leq j \leq n-2$. Then, because $i \leq m-2$ and $j \leq n-2$, from the run (15) we see that the output of N on 10^m would be $11^i0\beta$, where $|\beta| = m-i-1 \geq 1$. But this contradicts the known output of N on 10^m as $11^i11^\ell 0$, with $\ell+1 = m-i-1 = |\beta| \geq 1$. If $t_i = q_j$ for some $n-1 \leq j \leq v-1$, then from run (15) again we see that the output of N on 10^m would now be $11^{m-1}1$ which also contradicts the established $11^{m-1}0$. We conclude, therefore, that t_0, $\ldots t_{m-2}$ are new states of N. Hence, from (15) and (16), we see that N has at least $v+m-1$ states. Recalling that $v \geq n$ and $m > k$ we conclude that N has at least $n+k$ states, contradicting the limit of $n+k-1$ states for N. We have, thus, established that T is $(n+k-1)$-complete for M. Since we $g(T) = k$, we now know that T is m-complete for M, for any $m \leq g(T) + |S|$, as desired.

We can now summarize the discussion in the following theorem.

Theorem 7. *There is an infinite family of specifications FSMs M_i and test suites T_i, with σ_i being a shortest non-extensible test case in T_i ($i = 1, 2, \ldots$), and such that: (1) T_i is not m-complete for M_i, for all $m \geq |\sigma_i| + |S_i| - 1$ ($i = 1, 2, \ldots$); and (2) T_i is m-complete for M_i, for all $m < |\sigma_i| + |S_i| - 1$ ($i = 1, 2, \ldots$).*

We note that the same questions, now related to m-perfectness, do not lead to interesting answers. When M is complete, that is, when $U(s_0) = \mathcal{I}^\star$, take a 1-state implementation with no transitions. For any nonempty test suite T we can trivially find an input sequence $\alpha \in U(s_0) \cap T$ such that $\alpha \notin U(q_0)$. This shows that T is not 1-perfect for M. When M is not complete, we have some input sequence $\alpha \notin U(s_0)$. Consider the 1-state complete implementation N wit transitions $q_0 \stackrel{0/0}{\rightarrow} q_0$ and $q_0 \stackrel{1/1}{\rightarrow} q_0$. Again, for any test suite T with $\alpha \in T$ we know that $\alpha \in \big[U(s_0) \ominus U(q_0)\big] \cap T$. Hence, again, T is not 1-perfect for M.

6 Conclusions

In this work we have studied test suite perfectness, a notion similar to the classical notion of test suite completeness, but that also allows for the presence of so called blocking test cases, that is, test cases that may not run to completion either in the specification or in the implementation models. An accompanying notion of p-reduction was also introduced, similar to the classical notion of reduction in FSMs.

We established that the notion of perfectness is equivalent to the notion of bi-similarity. This result then lead to a necessary and sufficient condition for testing for perfectness, even in the presence of partial models, either for the specifications or for the implementations.

We showed that any FSM can be p-reduced while maintaining the perfectness property, when it was already present in the original FSM. Using this result, we then proved that when the specification and implementation models are both p-reduced, then perfectness can be characterized in terms of an isomorphism between the specification and the implementation.

We then established a relationship between perfectness and the classical notion of completeness. We showed that perfectness is a strictly stronger relation, for specifications models of any sizes.

Further, we also proved that when testing for completeness one has to impose a limit on the number of states of the implementation models that are put under test. We established a sharp upper bound on the number of states of implementations, if the completeness property is required from the testing method that is being used.

For future studies, we mention developing and testing algorithms for testing perfectness, inspired by the theoretical results discussed here.

References

[1] Adilson Luiz Bonifacio and Arnaldo Vieira Moura. Partial fsm models and completeness with blocking test cases. Technical Report IC-13-33, Institute of Computing, University of Campinas, November 2013.

[2] Adilson Luiz Bonifacio and Arnaldo Vieira Moura. On the Completeness of Test Suites. In *Proceedings of the 29th ACM Symposium on Applied Computing (ACM SAC)*, volume 2, pages 1287–1293. ACM, march 2014.

[3] Adilson Luiz Bonifacio and Arnaldo Vieira Moura. Test suite completeness and partial models. In D. Giannakopoulou and G. SalaÃ$\frac{1}{4}$n, editors, *Proceedings of the 12th International Conference on Software Engineering and Formal Methods (SEFM)*, volume 8702 of *Lecture Notes in Computer Science*, pages 96–110, Grenoble, France, 01–05, sep 2014. Springer Verlag.

[4] 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.

[5] Adenilso da Silva Simao, Alexandre Petrenko, and Nina Yevtushenko. Generating reduced tests for fsms with extra states. In *TestCom/FATES*, pages 129–145, 2009.

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

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

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

[9] A. Petrenko and G. V. Bochmann. On fault coverage of tests for finite state specifications. *Computer Networks and ISDN Systems*, 29:81–106, 1996.

[10] Alex Petrenko and Nina Yevtushenko. On test derivation from partial specifications. In *In FORTE*, pages 85–102, 2000.

[11] Adenilso Simao, Alexandre Petrenko, and Nina Yevtushenko. On reducing test length for fsms with extra states. *Softw. Test. Verif. Reliab.*, 22(6):435–454, September 2012.

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

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

[14] Ming Yu Yao, Alexandre Petrenko, and Gregor von Bochmann. Fault coverage analysis in respect to an fsm specification. In *INFOCOM*, pages 768–775, 1994.