Contents lists available at ScienceDirect

Information and Computation

journal homepage: www.elsevier.com/locate/yinco

# Removing redundant refusals: Minimal complete test suites for failure trace semantics

Maciej Gazda*, Robert M Hierons

*Department of Computer Science, The University of Sheffield, Sheffield, S1 4DP, UK*

ARTICLE INFO

*Article history:*
Received 4 August 2022
Received in revised form 6 January 2023
Accepted 18 January 2023
Available online 25 January 2023

*Keywords:*
Process semantics
Refusal trace semantics
Complete test suite
Characteristic formulae

ABSTRACT

We explore the problem of finding a minimal complete test suite for refusal trace (or failure trace) semantics. Our approach is based on generating a minimal complete set of forbidden refusal traces and utilises several interesting insights into the semantics. In particular, we identify a key class of refusals called fundamental refusals which essentially determine the refusal trace semantics, and the associated equivalence relation. We then propose a small but not necessarily minimal test suite, which can be constructed with a simple algorithm. Subsequently, we provide an enumerative method to remove all redundant traces from our complete test suite, which comes in two variants, depending on whether we wish to retain the highly desirable uniform completeness.

We also address a related problem from modal logic, namely the construction of a characteristic formula of a given process with respect to refusal trace semantics, using a variant of Hennessy-Milner logic with recursion.

## 1. Introduction

Testing is arguably the most widely used form of software verification and validation, but it is typically largely manual and therefore expensive and error-prone. There has thus been significant interest in test automation, including *model-based testing (MBT)* approaches that base test generation on a formal specification or model [1]. There is evidence of the effectiveness of MBT in industrial projects [2] and, in addition, the use of MBT approaches can guarantee that well-defined classes of faults will be found (see, for example, [3]).

MBT work goes back to Moore's seminal 1956 paper [4] and initially considered finite state machine (FSM) specifications. Here, observations are traces: sequences of inputs and outputs. As a result, the notion of correctness used (the *implementation relation*) is typically either trace equivalence (the specification and system under test have the same sets of traces) or trace inclusion (all traces of the system under test are also traces of the specification). More generally, however, observations may be richer than this, leading to a wide range of implementation relations for labelled transition systems (LTSs) [5]. The interest in LTSs is motivated by their ability to capture the semantics of formal languages such as CSP (see, for example, [6]).

In testing, the consensus has been that observations are linear: they do not require the tester to force the system under test to backtrack. Observations are therefore types of traces and so implementation relations are based on decorated traces.

* Corresponding author.
*E-mail addresses:* m.gazda@sheffield.ac.uk (M. Gazda), r.hierons@sheffield.ac.uk (R.M. Hierons).

https://doi.org/10.1016/j.ic.2023.105009

Such traces can include, for example, refusals, where a refusal of a set $X$ of actions denotes the system being in a stable state in which no action in $X$ is enabled. Most work uses ioco [7] or one of its variants; see [8] for an overview. In ioco, an observation is a trace that can contain instances of quiescence: the situation in which the system under test (SUT) cannot produce output or change state without first receiving input. The ioco test theory assumes that the SUT cannot refuse inputs (is input-enabled) and the environment cannot block outputs, which is why quiescence is the only type of refusal.

A refusal of a set $X$ is observed through the tester offering the actions in $X$ and there then being a deadlock (detected through a timeout). A failure trace [5], also called a refusal trace [6], and initially introduced by Philips [9], is a sequence containing actions and refusals of sets of actions. Clearly, an observation made in ioco is a type of refusal trace. It has been noted that sometimes an SUT can block input. For example, a web-page might have fields that are greyed-out or an autonomous system might switch off sensors. As a result, it can make sense to allow the observation of the refusal of inputs during testing, and this is reflected in a variant of ioco called mioco [10].

This paper considers testing from an LTS specification where observations are refusal traces. The use of refusal traces was motivated by several factors. First, in testing it is not normally possible to backtrack or save the system state and so observations are linear; they are decorated traces. Second, it is often feasible to observe refusal traces in testing. Indeed, there is a standard approach to observing the refusal of a set $X$ of actions: one offers the actions in $X$ and concludes that the refusal has been observed if a timeout occurs.[1] Thus, from a testing perspective it makes sense to consider refusal traces as observations. Third, normally a tester will not have access to richer information than refusal traces since, for example, it is not possible to identify the set of actions that can be executed in a particular state. Thus, refusal traces normally provide the richest semantics that is consistent with testing under reasonably realistic assumptions [11]. Fourth, refusal traces are required if we wish to capture certain types of behaviour such as priorities [6]. Finally, the use of refusal traces is consistent with the widely used ioco implementation relation.

We initially investigated the recent approach of Peleska et al. [12], which showed how a finite test suite can be generated in order to check behaviour up to a given bound on length. This work introduced techniques for testing for both traces and failures but it did not consider refusal traces. Interestingly, however, it transpires that the approach of Peleska et al. [12] cannot be directly extended to test to refusal traces. This is due to the fundamental difference between testing failures and refusal traces refinement – the former only requires testing for *forbidden* refusals at the end of a trace and this can be performed by checking certain sets of allowed actions as in [12]. In contrast, for refusal traces we need – in addition – to establish the context of *allowed* refusals along the trace. As a result, we adopted the approach of Cavalcanti et al. [13] in which a test case is a refusal trace $\sigma$ that should *not* be a behaviour of the SUT; the observation of $\sigma$ in testing will denote a faulty behaviour. Test execution involves determining whether the SUT can exhibit $\sigma$. The problem then is to derive a *complete* test suite (one guaranteed to find all faults), by choosing a set of refusal traces to use when testing from LTS $P$.

This paper extends a previous conference version [14]. The extension includes a more detailed treatment, e.g. we have added explicit proofs of several statements such as the minimality proof for the case of general completeness; furthermore, we have largely updated proofs in Section 7 where the part leading up to the minimality results has now a clearer structure. We have included additional examples, in particular a more striking example motivating the uniform completeness assumption. The key difference, however, is that the previous conference version only considered concrete LTSs that have no internal (silent) actions: here, we drop this restriction. The inclusion of internal actions can lead to unstable states (those in which internal actions are available). Refusals cannot be observed in such unstable states since a refusal of a set $X$ of actions is the situation in which the system deadlocks if only the actions in $X$ are available; a system can always progress if internal actions are available. The inclusion of internal actions thus led to significant technical differences throughout: both new definitions and notation (since we can now have unstable states) and new proofs.

As far as we are aware, this is the first work to explore the problem of finding a *minimal* complete test suite when observations are refusal traces. Previous work has looked at testing using refusal traces. For example, Cavalcanti et al. [13] provide a new CSP semantics based on refusal traces, in the context of testing robotic systems, and showed how complete test suites can be defined. However, there was no attempt to minimise these test suites. Random test generation was adopted in the work on mioco [10]. Although the generation of an efficient test suite has been a major topic in FSM-based testing, along with the complexity of associated decision problems [15–17], even here, test suites need not be minimal.

Removing redundancies from a test suite was motivated by very practical issues: test execution takes time and has an associated cost. Importantly, test execution time will often exceed test generation time and, in addition, a test suite may be executed many times (e.g. in regression testing). It is thus desirable to use minimal test suites where possible and hence test suite reduction is the main topic of this article. Note that results in this paper are also applicable to scenarios when one cannot afford to work with a complete test suite – they identify tests that are redundant in *any* test suite.

In this work, we start off with an "obvious" complete test suite that may contain a lot of redundant behaviours. In the course of the paper we investigate different sources of redundancy, identifying tests that can be removed, or better, not even considered in the first place. The latter enhancement is possible due to interesting insights into the structure of local refusals, giving rise to the so-called fundamental equivalence. Equivalent refusals and refusal traces correspond to the same behaviour of a valid refinement, hence only one test per equivalence class is required. We progressively refine our test suites, obtaining first a substantially reduced suite that can be neatly characterised and constructed with reasonable

[1] This is exactly what is normally done when testing based on ioco.

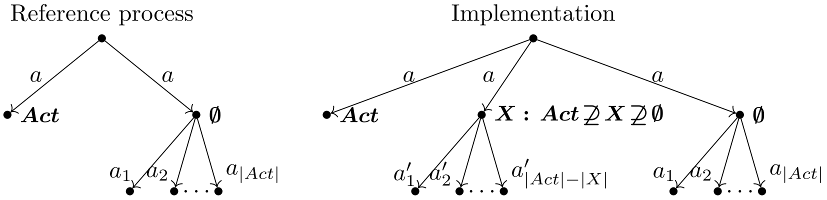


**Fig. 1.** The process on the right is a correct implementation of the reference process in failure semantics, but not in refusal trace semantics.

efficiency (we sketch the test generation algorithm). In order to remove the remaining redundant traces, we employ a more costly enumerative method, finally arriving at test suites that we prove to be minimal.

One discovery of particular interest are erroneous traces that entail the presence of other, *longer* disallowed traces in the implementation. The former traces are technically redundant according to the standard notion of completeness. However, we argue that the shortest counterexamples should be considered essential, and hence we require a stronger notion of *uniform completeness*. We also define two separate families of test suites which are provably minimal, depending on whether or not one prioritises the abovementioned usefulness criterion. We note that the phenomenon of non-uniform completeness and singular redundancies due to longer traces arises only in finer semantics where sufficiently rich observations are made along the trace (so not in e.g. traces or failures).

A line of research from the area of modal logic related to complete test suites are *characteristic formulae* [18,19] for process semantics. A characteristic formula is a logical encapsulation of process behaviour with respect to a given process semantics – a formula satisfied by exactly those processes which refine, or are equivalent to $P$. Characteristic formulae constructions have been offered by Aceto et al. [20,21] for a number of branching-time semantics. In addition, the so-called characterisation by primality principle has been shown for logics defining semantics in the linear time–branching time spectrum [22].

We show how our theory developed originally for complete test suites can be applied to construct characteristic formulae for refusal trace refinement. As the underlying logic, we use a variant of Hennessy-Milner logic with recursion. The construction produces formulae with a neat, succinct structure (though not necessarily minimal). Our contribution exemplifies close relationships between the different areas of testing and modal logic.

In general, our work offers an in-depth exploration of refusal/failure trace semantics and its interesting properties which distinguish it from other related semantics. As a simple example, consider the two processes in Fig. 1. In standard failure semantics, the process on the right is a correct implementation of the reference process for any refusal set $X$ strictly between the empty set and $Act$. However, this is not the case in refusal trace semantics – to see this, observe that the refusal trace $\emptyset.a.X.a'_1$ is not allowed by the reference process.

The example captures a subtle difference between failures and refusal trace semantics: in failure semantics, smaller refusals of individual states have no significance – only maximal refusals of the entire process after a trace determine the behaviour of a system. This is visible for instance in the labelling of the normalised graph in [12], where the only information stored are the maximal refusals after a trace. In refusal trace semantics, the *maximal refusals of individual states* play a much more prominent role – we call them *state refusals* (duals of the so-called ready sets from the literature). Ultimately, as we show throughout the paper, the refusal trace semantics is essentially determined by the *intersections of state refusals* – this crucial class we dub *fundamental refusals*.

### *Contributions*

The following are the main contributions of the paper.

- We identify a key class of refusals called *fundamental refusals*, whose traces essentially determine the refusal trace semantics. The associated *fundamental equivalence* captures a wide range of refusal traces which are always exhibited "as a whole" by a conforming system (a test suite needs only one representative per equivalence class).
- Using the aforementioned theory and further straightforward reduction methods, we provide a simple and substantially reduced test suite that can be generated reasonably efficiently.
- We investigate the more cumbersome sources of redundancy in test suites, such as redundancies due to continuation contexts and longer traces, and provide complete and minimal test suites for refusal trace refinement:
  - under uniform completeness assumption, which is natural and desirable, especially from an application perspective
  - for the general case, where additional refusal traces can be eliminated.
- We provide a characteristic formula construction for refusal trace refinement using a variant of Hennessy-Milner logic with recursion.

## 2. Preliminaries

For simplicity and generality, the presentation is based on labelled transition systems. While most of our definitions are standard, there are some exceptions:

- we introduce state refusals (maximal refusals of individual states); they provide the same observational power as initials / ready sets,
- our definition of refusal traces includes those that end in actions, as well as the empty trace – this assumption is convenient in our theory and makes no semantic difference.

A *labelled transition system* (LTS) is a tuple $\mathcal{L} = \langle S, \rightarrow, Act \cup \{\tau\}\rangle$ where $S$ is a finite set of states, $Act$ a finite set of visible/external actions, $\tau$ a special silent/internal action, and $\rightarrow \subseteq S \times Act \cup \{\tau\} \times S$ a transition relation. We use the shorthand notation $s \xrightarrow{a}$ [resp. $s \not\xrightarrow{a}$] whenever there exists [does not exist] a state $s'$ such that $s \xrightarrow{a} s'$. A state $s$ is *stable* if $s \not\xrightarrow{\tau}$.

We use the notation $\overset{\epsilon}{\Longrightarrow}$ for the reflexive transitive closure of $\xrightarrow{\tau}$, i.e. $s \overset{\epsilon}{\Longrightarrow} s'$ iff $s = s'$ or there is a chain of states and transitions $s = s_0 \xrightarrow{\tau} \cdots \xrightarrow{\tau} s_n = s'$. Moreover, $s \overset{a}{\Longrightarrow} s'$ denotes that $s \overset{\epsilon}{\Longrightarrow} t \xrightarrow{a} t' \overset{\epsilon}{\Longrightarrow} s'$ for some states $t, t'$.

A *process* can be defined by indicating a set of its possible initial states within a certain LTS. Formally, a process is a tuple $P = \langle S_I, \mathcal{L}\rangle$, where $\mathcal{L} = \langle S, \rightarrow, Act \cup \{\tau\}\rangle$ and $S_I \subseteq S$. Typically, the underlying LTS should be clear from the context and we shall identify a process $P$ with the set $S_I$. Moreover, the *internal closure* of $P$, notation $\mathrm{I}_\tau(P)$, is the set of states reachable from initial states of $P$ with internal actions, i.e. $\mathrm{I}_\tau(P) = \{s' \mid \exists s \in P : s \overset{\epsilon}{\Longrightarrow} s'\}$ [here, we identify the process $P$ with its set of initial states].

An important assumption that we make throughout the paper is divergence-freedom, that is, in all systems under consideration we assume that there are no infinite paths of $\tau$-labelled transitions. This is a reasonable assumption since: 1) divergence in a specification normally represents a fault; and 2) for a system under test, divergence looks the same[2] (to the tester) as deadlock and so is not needed.

For a stable state $s$, a set $X \subseteq Act$ is a *stable refusal*, or simply *refusal* of $s$, if for all $a \in X$, $s \not\xrightarrow{a}$. The set of all refusals of $s$ is denoted with $\mathbf{R}(s)$. Note that $\mathbf{R}(s) = \emptyset$ for unstable states. The *state refusal* of $s$ is the largest refusal of $s$, denoted with $\mathbf{SR}(s)$.

We shall use $\bullet$ to denote the null refusal observation which can be made in any state of an LTS (note: unlike a stable refusal, which can only be observed in stable states). A *refusal trace (failure trace)* $\sigma$ of a state $s$ is a sequence that is either an empty word $\epsilon$, or a word of the form $X_0 a_1 X_1 a_2 \ldots X_{n-1} a_n X_n$, or $X_0 a_1 X_1 a_2 \ldots X_{n-1} a_n$, where $a_i$ range over $Act$ and $X_i$ range over $\mathscr{P}(Act) \cup \{\bullet\}$, such that there is a chain of transitions $s \overset{\epsilon}{\Longrightarrow} s_0 \overset{a_1}{\Longrightarrow} s_1 \overset{a_2}{\Longrightarrow} \cdots \overset{a_{n-1}}{\Longrightarrow} s_{n-1} \overset{a_n}{\Longrightarrow} s_n \overset{\epsilon}{\Longrightarrow} q$ and for all $k \in \{0, \ldots, n-1, [n]\}$, $X_k \in \mathbf{R}(s_k)$ or $X_k = \bullet$. We denote the existence of such a chain of transitions by $s \overset{\sigma}{\Longrightarrow} q$. We define the *length* of a refusal trace in one of the above forms as *the number of refusals occurring in the trace*, that is

$$\begin{array}{lcl} |\epsilon| & \triangleq & 0 \\ |X_0 a_1 \ldots X_{n-1} a_n| & \triangleq & n \\ |X_0 a_1 \ldots X_{n-1} a_n X_n| & \triangleq & n+1 \end{array}$$

We stress that the length of a trace ending in refusal $X_n$ is $n+1$. We have chosen this definition, because a refusal $X_n$ provides a certain information about the process behaviour at $(n+1)$-th step (disallowed actions), and this notion of length proved most natural in definitions, proofs etc.

The set of refusal traces originating from a state $s$ [of length $\leq l$] is denoted with $\mathbf{RT}(s)$ [resp. $\mathbf{RT}^l(s)$]; the notation is lifted to processes. The language of *well-formed refusal traces* is defined as:

$$\begin{array}{l} \mathbf{RT} \triangleq \{\sigma \in (\mathscr{P}(Act) \cup \{\bullet\}) \times (Act \times (\mathscr{P}(Act) \cup \{\bullet\}))^* \cup ((\mathscr{P}(Act) \cup \{\bullet\}) \times Act)^* \\ \qquad \mid \sigma = \rho.X_i.a_{i+1}.\rho' \Longrightarrow X_i = \bullet \vee a_{i+1} \notin X_i\} \end{array}$$

Note that, since $\tau$ represents an unobservable event, all observations of a process $P$ are precisely those of its internal closure $\mathrm{I}_\tau(P)$ – the two processes are indistinguishable in this observational model. Moreover, we use internal closure to formally lift the notions of refusal, state refusal, and refusal trace from states to processes:

$$\begin{array}{ll} \mathbf{R}(P) & \triangleq \bigcup_{q \in \mathrm{I}_\tau(P)} \mathbf{R}(q) \\ \mathbf{SR}(P) & \triangleq \{\mathbf{SR}(q) \mid q \in \mathrm{I}_\tau(P)\} \\ \mathbf{RT}(P) & \triangleq \bigcup_{q \in \mathrm{I}_\tau(P)} \mathbf{RT}(q) \end{array}$$

[2] Note that although divergence and deadlock look the same to the tester, since there are no visible actions/observations, they would not look the same if the tester was able to determine whether a process is active (divergence involves progress, while deadlock does not).

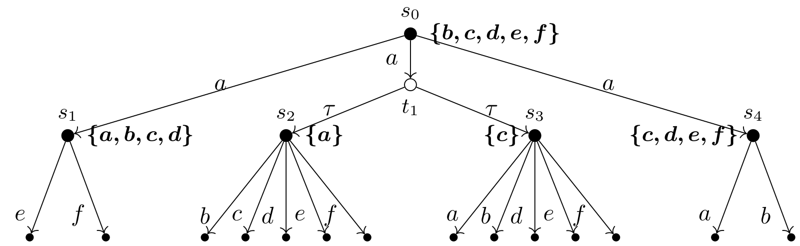


**Fig. 2.** A labelled transition system. Certain relevant states are labelled in bold with state refusal sets, a convention that we use throughout the paper.

In line with literature on testing, refusal trace refinement is defined as a restriction of behaviours: a process $Q$ is a *refusal trace refinement* [up to $\ell$ steps] of a process $P$, notation $P \sqsubseteq_{\mathbf{RT}} Q$ $[P \sqsubseteq^{\ell}_{\mathbf{RT}} Q]$, iff $\mathbf{RT}(Q) \subseteq \mathbf{RT}(P)$ $[\mathbf{RT}^{\ell}(Q) \subseteq \mathbf{RT}^{\ell}(P)]$. We also define the *refusal trace equivalence*: $P =_{\mathbf{RT}} Q$ $[P =^{\ell}_{\mathbf{RT}} Q]$ denotes the conjunction of $P \sqsubseteq_{\mathbf{RT}} Q$ and $Q \sqsubseteq_{\mathbf{RT}} P$ $[P \sqsubseteq^{\ell}_{\mathbf{RT}} Q$ and $Q \sqsubseteq^{\ell}_{\mathbf{RT}} P]$.

For a process $P$ and refusal trace $\sigma$, the process $P$ after $\sigma$, denoted with $P\|\sigma$, is the set $\{q \in S \mid \exists s \in P : s \overset{\sigma}{\Longrightarrow} q\}$. On a formal note, observe that for all $P$ we have $P\|\epsilon = \mathrm{I}_{\tau}(P)$.

**Example 1.** Consider the LTS from Fig. 2, representing a process $P$ with $S_I = s_0$. Process $P$ has one state refusal, i.e. $\mathbf{SR}(P) = \{\{b,c,d,e,f\}\}$, while its set of refusals $\mathbf{R}(P)$ consists of all subsets of $\{b,c,d,e,f\}$. After performing a visible action $a$, the process exhibits four state refusals, i.e. $\mathbf{SR}(P\|\{b,c,d,e,f\}.a) = \{\{a\},\{c\},\{a,b,c,d\},\{c,d,e,f\}\}$. All refusals of $P\|\{b,c,d,e,f\}.a$ are depicted in Fig. 3.

It is important to distinguish state refusals, which are the largest refusals of individual *states*, from *maximal refusals of a process* (after a given trace). Each maximal refusal of a process is a state refusal, but not the other way round – in our example, the maximal refusals of $P\|\{b,c,d,e,f\}.a$ are only $\{a,b,c,d\}$ and $\{c,d,e,f\}$. The refusals $\{a\}$ and $\{c\}$ are state refusals of $P\|\{b,c,d,e,f\}.a$, but not maximal refusals of that process.

The final notion we introduce here is a natural order on refusal traces which combines prefix and pointwise inclusion order. Intuitively, the larger refusal traces contain more detailed information about the system behaviour. We first define the following relations for individual refusals

$$\begin{aligned} X \preceq Y &\iff (X = \bullet) \vee (X \neq \bullet \wedge X \subseteq Y) \\ X \prec Y &\iff X \preceq Y \wedge X \neq Y \end{aligned}$$

Furthermore, for refusal traces we define

$$\begin{aligned} & X_0 a_1 \ldots a_n [X_n] \preceq Y_0 a_1 \ldots a_m [Y_m] \\ & \overset{def}{\iff} n \leq m \wedge \forall i \in \{0, \ldots, n-1, [n]\}\, X_i \preceq Y_i \\ & X_0 a_1 \ldots a_n [X_n] \prec Y_0 a_1 \ldots a_m [Y_m] \\ & \overset{def}{\iff} X_0 a_1 \ldots a_n [X_n] \preceq Y_0 a_1 \ldots a_m [Y_m] \wedge \big((n < m) \vee (\exists i : X_i \prec Y_i)\big) \end{aligned}$$

It is straightforward to observe that refusal traces of a process are always downward-closed w.r.t. $\preceq$, which we formally state below.

**Proposition 1.** *For any process P [and length bound $\ell$]*

$$\sigma \preceq \rho \wedge \rho \in \boldsymbol{RT}^{[\ell]}(P) \implies \sigma \in \boldsymbol{RT}^{[\ell]}(P)$$

We finish this section with a useful technical proposition, which also illustrates a key difference that working with processes, rather than individual states, brings. The following equivalence can be seen as a coinductive definition of refusal trace refinement – this is valid since we are reasoning on the normalised structure, comprising of subsets.

**Proposition 2.** *For all processes P, Q:*

$$(*) \qquad P \sqsubseteq^{[\ell]}_{\boldsymbol{RT}} Q \iff \begin{array}{l} [\forall k \in 0, \ldots, \ell]\ \forall \sigma \in \boldsymbol{RT}^{[k]}(P)\,. \\ P \parallel \sigma \sqsubseteq^{[\ell-k]}_{\boldsymbol{RT}} Q \parallel \sigma \end{array}$$

**Proof.** Fix $\ell$ and $k \leq \ell$. As $P =_{\mathbf{RT}} P\|\epsilon$, the nontrivial direction is from left to right. Observe that for any process $\widehat{P}$, and trace $\sigma$ of length $k$, the set $\mathbf{RT}^{[\ell-k]}(\widehat{P} \parallel \sigma)$ can be expressed as a function of $\mathbf{RT}^{[\ell]}(\widehat{P})$:

- for $\sigma = \rho.X$:

$$\mathbf{RT}^{[\ell-k]}(\widehat{P} \parallel \rho.X) \quad = \quad \{\epsilon\} \cup \{Z.\lambda \mid \exists Y \supseteq Z : \rho.Y.\lambda \in \mathbf{RT}^{[\ell]}(\widehat{P}) \;\wedge\; X \subseteq Y\}$$

- for $\sigma = \rho.X.a$ where $X \in \mathscr{P}(Act) \cup \{\bullet\}$:

$$\mathbf{RT}^{[\ell-k]}(\widehat{P} \parallel \rho.X.a) \quad = \quad \{\lambda \mid \rho.X.a.\lambda \in \mathbf{RT}^{[\ell]}(\widehat{P})\}$$

The above identities express $\mathbf{RT}^{[\ell-k]}(\widehat{P} \parallel \sigma)$ as a monotonically increasing function of $\mathbf{RT}^{[\ell]}(\widehat{P})$, that is:

$$\mathbf{RT}^{[\ell]}(Q) \subseteq \mathbf{RT}^{[\ell]}(P) \implies \mathbf{RT}^{[\ell-k]}(Q \parallel \sigma) \subseteq \mathbf{RT}^{[\ell-k]}(P \parallel \sigma)$$

which is equivalent to the statement we have been proving. □

## 3. Test suites, generic suite and forbidden refusals

In our framework, a *test suite* for a given process $P$ is a set of refusal traces forbidden by $P$. The research question that we explore in the paper is the generation, for a given reference process $P$, and the maximal number of steps $l$, a test suite $\mathsf{TS} \subseteq \mathbf{RT}$ such that for every process $Q$

$$P \sqsubseteq^{l}_{\mathbf{RT}} Q \iff \forall \sigma \in \mathsf{TS} : \sigma \notin \mathbf{RT}(Q)$$

Implication from left to right is called *soundness* while the implication from right to left is called *completeness*.

Our starting point is a generic suite consisting of all refusal traces within a given length bound that are not exhibited by the reference process. The only restriction that we make in this suite – an obvious one – is to consider only traces whose all proper prefixes are valid traces of $P$. This way only the first occurrence of forbidden behaviour in a trace is included.

$$\begin{aligned} \mathbf{TS}^{\ell}_{\mathbf{0}}(P) \quad \triangleq \quad & \{\sigma.X \in \mathbf{RT}^{\ell} \setminus \mathbf{RT}^{\ell}(P) \mid \sigma \in \mathbf{RT}^{\ell}(P)\} \\ & \cup \{\sigma.a \in \mathbf{RT}^{\ell} \setminus \mathbf{RT}^{\ell}(P) \mid \sigma \in \mathbf{RT}^{\ell}(P)\} \end{aligned}$$

Observe that for all $\sigma.X \in \mathbf{TS}^{\ell}_{\mathbf{0}}(P)$ we have $X \in \mathscr{P}(Act)$ – this is because $\sigma.\bullet \notin \mathbf{RT}^{\ell}(P)$ iff $\sigma \notin \mathbf{RT}^{\ell}(P)$.

Since in our case tests are the traces themselves, the soundness of the above test suite for refusal traces follows immediately from the definition.

**Proposition 3.** *Test suite $\mathbf{TS}^{\ell}_{\mathbf{0}}$ is sound for $\mathbf{RT}^{\ell}$.*

Moreover, as every test suite considered in the remainder of this work is a subset of $\mathbf{TS}_{\mathbf{0}}$, the above proposition immediately yields soundness of all those test suites.

Before stating the completeness property, we can apply our first optimisation. We restrict the first group of traces (ending with refusals) by including only those for which the forbidden refusals $X$ at the end of the trace are minimal. We note that this is a standard method in testing failure semantics [23,12].

$$\begin{aligned} \mathbf{TS}^{\ell}_{\mathbf{1}}(P) \quad \triangleq \quad & \{\sigma.X \in \mathbf{TS}^{\ell}_{\mathbf{0}} \mid X \in \mathsf{min}_{\not\subseteq}(\mathbf{R}(P \| \sigma))\} \\ & \cup \{\sigma.a \mid \sigma.a \in \mathbf{TS}^{\ell}_{\mathbf{0}}\} \end{aligned}$$

where for any family of refusals $\mathcal{X} \subseteq \mathscr{P}(Act)$

$$\mathsf{min}_{\not\subseteq}(\mathcal{X}) \triangleq \min\{Y \subseteq Act \mid \forall X \in \mathcal{X}.\, Y \not\subseteq X\}$$

**Lemma 1.** *Test suite $\mathbf{TS}^{\ell}_{\mathbf{1}}$ is complete for $\mathbf{RT}^{\ell}$.*

Our goal is to investigate which refusal traces can be removed safely from the test suite; ideally, we wish to arrive at a minimal suite that does not contain any redundant traces. In the remainder of the paper, we shall identify a number of different sources of redundancies in a test suite.

## 4. Fundamental equivalence

Redundancies of the first type are particularly interesting, have a neat characterisation and are potentially the most attractive from the algorithmic perspective. They can be identified due to a natural equivalence relation between local refusals (i.e. refusals of a process after a specific trace). The theory developed in this section gets well into the heart of refusal trace semantics, identifying a special class of refusals that determine the entire semantics.

### 4.1. Basic definitions and properties

The aforementioned class of refusals that will prove to be of paramount importance are intersections of state refusals. We shall call them *fundamental refusals*:

$$\mathbf{FR}(P) \quad \triangleq \quad \{X_1 \cap \cdots \cap X_k \mid X_i \in \mathbf{SR}(P)\}$$

The definition is extended to *fundamental refusal traces*:

$$\begin{aligned}\mathbf{FRT}^{[\ell]}(P) \quad \triangleq \quad & \{\epsilon\} \cup \{X_0 a_1 \dots X_n[a_{n+1}] \in \mathbf{RT}^{[\ell]}(P) \mid X_0 \in \mathbf{FR}(P) \\ & \wedge \forall i : X_i = \bullet \ \vee \ X_i \in \mathbf{FR}(P \| X_0 a_1 \dots a_i)\}\end{aligned}$$

Observe that in particular all state refusals are fundamental refusals. However, while refusal trace equivalent processes may differ on their state refusals, fundamental refusals are invariant under refusal trace semantics, which can be explained through the following neat syntactic characterisation.

**Proposition 4.** *A refusal $X \in \mathscr{P}(Act)$ of a process $P$ is fundamental if and only if every action outside $X$ is admissible directly after $X$ has been observed i.e.:*

$$\begin{aligned}X \in \boldsymbol{FR}(P) \iff \quad & X \in \boldsymbol{R}(P) \\ & \wedge \forall a \in Act \setminus X \ X.a \in \boldsymbol{RT}(P)\end{aligned}$$

*More generally, for any refusal trace $\sigma$ that does not end in a refusal (i.e. $\sigma = \epsilon$ or $\sigma = \sigma'.a$ for some $\sigma'$), we have:*

$$\begin{aligned}X \in \boldsymbol{FR}(P\|\sigma) \iff \quad & X \in \boldsymbol{R}(P\|\sigma) \\ & \wedge \forall a \in Act \setminus X \ \sigma.X.a \in \boldsymbol{RT}(P)\end{aligned}$$

**Proof.** "$\Longrightarrow$": Suppose that $X \in \mathbf{FR}(P\|\sigma)$. Let $X_1, \dots, X_n \in \mathbf{SR}(P\|\sigma)$ be such that $X = X_1 \cap \cdots \cap X_n$. Take any $a \in Act \setminus X$, and let $X_i$ be such that $a \notin X_i$. Since $X_i \in \mathbf{SR}(P\|\sigma)$, there must be some stable state $s \in P\|\sigma$ such that $\mathbf{SR}(s) = X_i \supseteq X$. We therefore have $s \xrightarrow{a}$ and $X \in \mathbf{R}(s)$, and hence obtain $\sigma.X.a \in \mathbf{RT}(P)$.

"$\Longleftarrow$": Suppose that $\forall a \in Act \setminus X \ \sigma.X.a \in \mathbf{RT}(P)$. Let $\mathcal{Y}_X = \{Y \in \mathbf{SR}(P\|\sigma) \mid X \subseteq Y\}$. $\mathcal{Y}_X$ is nonempty since $X \in \mathbf{R}(P\|\sigma)$; moreover, $X \subseteq \bigcap \mathcal{Y}_X$. For any $a \notin X$, we have $\sigma.X.a \in \mathbf{RT}(P)$, and from the definition of $\mathcal{Y}_X$ there must be some $\widehat{Y} \in \mathcal{Y}_X$ such that $a \notin \widehat{Y}$, from which we obtain $a \notin \bigcap \mathcal{Y}_X$. Hence $\bigcap \mathcal{Y}_X \subseteq X$ and thus $X = \bigcap \mathcal{Y}_X$, from which $X \in \mathbf{FR}(P\|\sigma)$ follows. □

As a corollary, we obtain the following preservation property for fundamental refusals and their traces. Note that the proof of the second statement (preservation of refusal traces up to $\ell$ steps) relies on our definition of the trace length (number of refusals rather than actions).

**Corollary 1.** *Suppose $P \sqsubseteq^{\ell}_{\boldsymbol{RT}} Q \left[P =^{\ell}_{\boldsymbol{RT}} Q\right]$. We have:*

1. *For any $\sigma \in \boldsymbol{RT}^{\ell-1}(P)$ that does not end with a refusal:*

$$X \in \boldsymbol{FR}(Q\|\sigma) \implies X \in \boldsymbol{FR}(P\|\sigma) \left[X \in \boldsymbol{FR}(P\|\sigma) \iff X \in \boldsymbol{FR}(Q\|\sigma)\right]$$

2. $\boldsymbol{FRT}^{\ell}(Q) \subseteq \boldsymbol{FRT}^{\ell}(P) \left[\boldsymbol{FRT}^{\ell}(P) = \boldsymbol{FRT}^{\ell}(Q)\right]$

**Proof.** Suppose $P \sqsubseteq^{\ell}_{\mathbf{RT}} Q$.

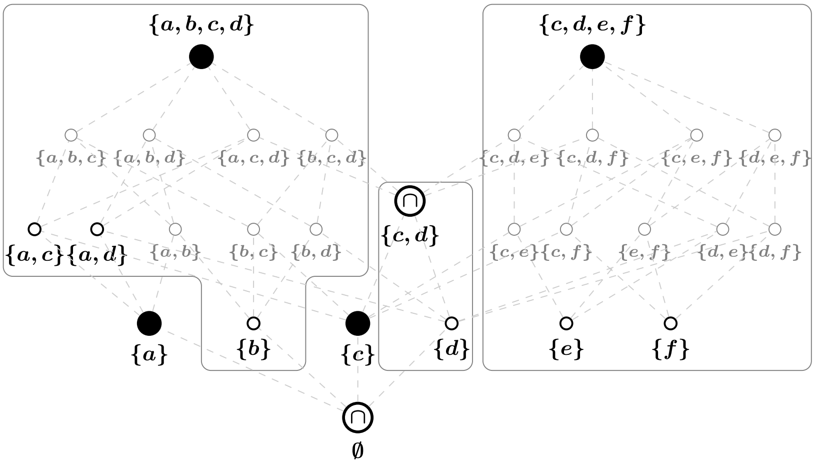
1. Take a $\sigma \in \mathbf{RT}^{\ell-1}(P)$ not ending in a refusal and let $X \in \mathbf{FR}(Q\|\sigma)$. From Proposition 4 we have $\forall a \in Act \setminus X \ \sigma.X.a \in \mathbf{RT}(Q)$. Since $P \sqsubseteq^{\ell}_{\mathbf{RT}} Q$, we have $\mathbf{RT}^{\ell}(Q) \subseteq \mathbf{RT}^{\ell}(P)$, hence $\forall a \in Act \setminus X \ \sigma.X.a \in \mathbf{RT}^{\ell}(P)$, from which and Proposition 4 follows $X \in \mathbf{FR}(Q\|\sigma)$.
2. Take any $\rho \in \mathbf{FRT}^{\ell}(Q)$. We proceed by structural induction on $\rho$. The base ($\rho = \epsilon$) is immediate. In the inductive step, we have two cases:
   - $\rho = \rho'.X$: this is the key case. From Proposition 4 we know that

     $$\forall a \in Act \setminus X \ \rho'.X.a \in \mathbf{RT}(Q)$$

     Observe that since $\rho'.X \in \mathbf{RT}^{\ell}(Q)$, formulae of the form $\rho'.X.a$ are also in $\mathbf{RT}^{\ell}(Q)$. Since $\mathbf{RT}^{\ell}(Q) \subseteq \mathbf{RT}^{\ell}(P)$, we thus have

     $$\forall a \in Act \setminus X \ \rho'.X.a \in \mathbf{RT}^{\ell}(P)$$

     Moreover, from the inductive hypothesis we have $\rho' \in \mathbf{FRT}^{\ell-1}(P)$. From the above, combined with Proposition 4 follows $\rho'.X \in \mathbf{FRT}^{\ell}(P)$.


{a, b, c, d}
{c, d, e, f}
{a, b, c} {a, b, d} {a, c, d} {b, c, d}
{c, d, e} {c, d, f} {c, e, f} {d, e, f}
∩
{c, d}
{a, c}{a, d} {a, b} {b, c} {b, d}
{c, e}{c, f} {e, f} {d, e}{d, f}
{a} {b} {c} {d} {e} {f}
∩
∅

Nontrivial clusters:

| top | min-base | other refusals in cluster |
|---|---|---|
| $\{\boldsymbol{a},\boldsymbol{b},\boldsymbol{c},\boldsymbol{d}\}$ | $\{a,c\}$, $\{a,d\}$, $\{b\}$ | $\{a,b\}$, $\{b,c\}$, $\{b,d\}$, $\{a,c,d\}$, $\{b,c,d\}$, $\{a,b,c\}$, $\{a,b,d\}$ |
| $\{\boldsymbol{c},\boldsymbol{d}\}$ | $\{d\}$ | |
| $\{\boldsymbol{c},\boldsymbol{d},\boldsymbol{e},\boldsymbol{f}\}$ | $\{e\}$, $\{f\}$ | $\{c,e\}$, $\{c,f\}$, $\{d,e\}$, $\{d,f\}$, $\{e,f\}$, $\{c,d,e\}$, $\{c,d,f\}$, $\{c,e,f\}$, $\{d,e,f\}$ |

**Fig. 3.** A detailed breakdown of all refusals of the process $P' = P \| \{b,c,d,e,f\}.a$, where $P$ is the process from Fig. 2, with the state refusals $\mathbf{SR}(P')$ denoted with large black circles. Additional fundamental refusals, that is, belonging to $\mathbf{FR}(P') \setminus \mathbf{SR}(P')$, are depicted with large circles labelled with "∩". The remaining refusals in $\mathbf{R}(P')$ are denoted with small circles. There are three nontrivial fundamental clusters, and in each such cluster elements of its minimum base are represented with small bold black circles. From the perspective of our test generation algorithm, the "grey" refusals will not appear in the generated traces (even though they are exhibited by the process).

- $\rho = \rho'.a$: Since $\rho'$ must be of the form $\rho' = \rho''.X$, from the proof in the previous case it follows that $\rho' \in \mathbf{FRT}^{\ell}(P)$, hence from the definition of **FRT** we immediately obtain $\rho'.a \in \mathbf{FRT}^{\ell}(P)$. □

For any refusal of a process, we define the so-called top refusal as the intersection of all state refusals that include $X$.

$$\mathsf{top}_P(X) \triangleq \bigcap_{Y \in \mathbf{SR}(P):\ X \subseteq Y} Y$$

Note that $\mathsf{top}_P(X)$ is the least element of $\{Y \in \mathbf{FR}(P) \mid X \subseteq Y\}$ – *the least fundamental refusal that subsumes* $X$. In addition, observe that $\mathbf{FR}(P) = \{\mathsf{top}_P(X) \mid X \in \mathbf{R}(P)\}$. Top refusals play a special role – as we shall prove, if $P$ refuses $X$, then it must be in a state[3] where it can refuse all actions in $\mathsf{top}_P(X)$.

For technical convenience, we extend the definition of $\mathsf{top}_P$ operator to null refusal $\bullet$ by simply defining

$$\mathsf{top}_P(\bullet) \triangleq \bullet$$

The related notions of *fundamental equivalence* and its equivalence classes called *fundamental clusters* (or simply *clusters*), are defined below:

$$X \sim_P Y \overset{def}{\iff} \mathsf{top}_P(X) = \mathsf{top}_P(Y)$$
$$[X]_P \triangleq \{Y \in \mathbf{R}(P) \mid X \sim_P Y\}$$

Also of relevance will be the set of minimal refusals in a fundamental cluster – we call it its *minimum base*. Note that while $\mathsf{top}_P(X)$ is a single refusal, $\mathsf{min\text{-}base}_P(X)$ may consist of several refusals.

$$\mathsf{min\text{-}base}_P(X) \triangleq \min\{Y \in \mathbf{R}(P) \mid Y \sim_P X\}$$

**Example 2.** An example illustrating the introduced notions is given in Fig. 3.

The crucial role of fundamental refusals and clusters is visible in the following proposition which characterises all forbidden actions occurring after a refusal.

[3] We essentially identify $P$ and its internal closure.

**Proposition 5.** *Suppose $X \in \mathscr{P}(Act)$, $\sigma.X \in \boldsymbol{RT}(P)$ and $a \in Act$. Then*

$$\sigma.X.a \notin \boldsymbol{RT}(P) \iff a \in \mathit{top}_{P\|\sigma}(X)$$

**Proof.** "$\Longrightarrow$": Follows immediately from Proposition 4.

"$\Longleftarrow$": Take any $a \in \mathsf{top}_{P\|\sigma}(X)$. Since

$$a \in \bigcap_{Y \in \mathbf{SR}(P\|\sigma):\ X \subseteq Y} Y,$$

for any stable $s \in P\|\sigma$ such that $X \subseteq \mathbf{SR}(s)$, we have $s \overset{a}{\not\longrightarrow}$. Hence $\sigma.X.a \notin \mathbf{RT}(P)$. □

As a corollary, we obtain the following property: for a conforming system, either all refusal sets in a fundamental cluster are refused, or none of them is. This is an interesting insight into refusals – while refusals are obviously downward-closed, the proposition describes a circumstance when a refusal entails the presence of a *larger* refusal in the conforming system. A more general property of refusal traces will be later provided by Proposition 6.

**Corollary 2.** *Let $P$ be a reference process, and $\sigma \in \boldsymbol{RT}(P)$. Suppose $X, Y \in \mathscr{P}(Act)$ are such that $X \sim_{P\|\sigma} Y$. Then for each $Q$ such that $P \sqsubseteq_{\boldsymbol{RT}} Q$:*

$$X \in \boldsymbol{R}(Q\|\sigma) \iff Y \in \boldsymbol{R}(Q\|\sigma)$$

**Proof.** We shall prove the statement by showing that

$$X \in \mathbf{R}(Q|\sigma) \iff \mathsf{top}_{P\|\sigma}(X) \in \mathbf{R}(Q|\sigma)$$

The implication from right to left is immediate as $X \subseteq \mathsf{top}_{P\|\sigma}(X)$ and refusals of a process are downward-closed. For the other direction, suppose that $X \in \mathbf{R}(Q|\sigma)$. Let $s \in Q\|\sigma$ be such that $X \in \mathbf{R}(s)$. From Proposition 5 and $\mathbf{RT}(Q) \subseteq \mathbf{RT}(P)$ it follows that for all $a \in \mathsf{top}_{P\|\sigma}(X)$, $s \overset{a}{\not\longrightarrow}$, from which in turn we obtain $\mathsf{top}_{P\|\sigma}(X) \in \mathbf{R}(s)$. Hence $\mathsf{top}_{P\|\sigma}(X) \in \mathbf{R}(Q|\sigma)$. □

### 4.2. Fundamental equivalence and test suites

Proposition 5 provides us with a full characterisation of locally forbidden actions, thanks to which we are able to update our test suite $\mathbf{TS_1}$ with a more accurate description.

$$\begin{aligned} \mathbf{TS}^\ell_\mathbf{1}(P) \quad = \quad & \{\sigma.X \in \mathbf{TS}^\ell_\mathbf{0} \mid X \in \min_{\not\subseteq}(\mathbf{R}(P\|\sigma))\} \\ & \cup \{\sigma.X.a \in \mathbf{TS}^\ell_\mathbf{0} \mid X \neq \bullet \wedge a \in \mathsf{top}_{P\|\sigma}(X) \setminus X\} \\ & \cup \{\sigma.\bullet.a \in \mathbf{TS}^\ell_\mathbf{0}\} \end{aligned}$$

Obviously, forbidden continuations for $a \in \mathsf{top}_{P\|\sigma}(X) \cap X$ are not included, as we only consider well-formed traces.

Furthermore, traces ending in forbidden actions can be restricted to only those where the last refusal belongs to the minimum base of a (nontrivial) cluster:

$$\begin{aligned} \mathbf{TS}^\ell_\mathbf{2}(P) \quad \triangleq \quad & \{\sigma.X \in \mathbf{TS}^\ell_\mathbf{1}\} \\ & \cup \{\sigma.X.a \in \mathbf{TS}^\ell_\mathbf{1} \mid X \neq \bullet \wedge a \in \mathsf{top}_{P\|\sigma}(X) \setminus X \wedge X \in \mathsf{min\text{-}base}_{P\|\sigma}(X)\} \\ & \cup \{\sigma.\bullet.a \in \mathbf{TS}^\ell_\mathbf{0}\} \end{aligned}$$

**Lemma 2.** *$\boldsymbol{TS}^\ell_\mathbf{2}$ is complete for $\boldsymbol{RT}^\ell$.*

**Proof.** Suppose $P \not\sqsubseteq^\ell_{\mathbf{RT}} Q$. From the completeness of $\mathbf{TS}^\ell_\mathbf{1}(P)$, there is some trace $\rho \in \mathbf{RT}^\ell(Q) \cap \mathbf{TS}^\ell_\mathbf{1}(P)$. In case $\rho$ is of the form $\sigma.X$ or $\sigma.\bullet.a$ in $\mathbf{RT}^\ell(Q) \setminus \mathbf{RT}^\ell(P)$, then from the definition of $\mathbf{TS_2}$ we have $\rho \in \mathbf{TS}^\ell_\mathbf{2}(P)$.

Suppose therefore that $\rho = \sigma.X.a$ for some $X \in \mathscr{P}(Act)$. From the definition of minimum base, there is a refusal $X_m \in \mathsf{min\text{-}base}_{P\|\sigma}(X)$ such that $X_m \subseteq X$. This, combined with $\mathsf{top}_{P\|\sigma}(X_m) = \mathsf{top}_{P\|\sigma}(X)$, yields $a \in \mathsf{top}_{P\|\sigma}(X) \setminus X \subseteq \mathsf{top}_{P\|\sigma}(X_m) \setminus X_m$, from which $\sigma.X_m.a \in \mathbf{TS}^\ell_\mathbf{2}(P)$ follows. Since refusals of a process are downward-closed, we have $\sigma.X_m.a \in \mathbf{RT}^\ell(Q)$, hence $\sigma.X_m.a \in \mathbf{RT}^\ell(Q) \cap \mathbf{TS}^\ell_\mathbf{2}(P)$. □

At this point, we have managed to restrict the test suite by analysing locally forbidden observations (single events and refusals) and their immediate context at the last step of a trace. There is still a potentially much larger scope for reduction, since the test suite $\mathbf{TS_2}$ may contain a lot of traces that are equivalent due to fundamentally equivalent intermediate refusals along the trace.

In what follows, we explain how to remove those redundancies by including only traces in which the intermediate refusals are fundamental refusals. We start with formally lifting the $\mathsf{top}_P$ operator to (valid) refusal traces of the reference process.

Give a process $P$, we define $\mathsf{top}_P(\epsilon) \triangleq \epsilon$, and for all nonempty refusal traces of $P$:

$$\mathsf{top}_P(X_0 a_1 \dots a_n[X_n]) \triangleq \mathsf{top}_{P_0}(X_0) a_1 \dots a_n[\mathsf{top}_{P_n}(X_n)]$$

where: $X_0 a_1 \dots X_{n-1} a_n [X_n] \in \mathbf{RT}(P)$, $P_0 = P$ and $P_i = P \| X_0 a_1 \dots a_i$ for $i \in \{1, \dots, n\}$.

The definition is extended to traces with a forbidden final observation, in which case *the final refusal observation is not affected by the operator*:

$$\begin{array}{lll} \mathsf{top}_P(\sigma.X) & \triangleq \mathsf{top}_P(\sigma).X & \text{if } \sigma \in \mathbf{RT}(P) \wedge \sigma.X \notin \mathbf{RT}(P) \\ \mathsf{top}_P(\sigma.X.a) & \triangleq \mathsf{top}_P(\sigma).X.a & \text{if } \sigma.X \in \mathbf{RT}(P) \wedge \sigma.X.a \notin \mathbf{RT}(P) \end{array}$$

We also lift fundamental equivalence to refusal traces of the specification. For a process $P$, we define:

$$X_0 a_1 X_1 \dots a_n[X_n] \sim_P Y_0 a_1 Y_1 \dots a_n[Y_n] \overset{def}{\iff} \forall i \in \{0, \dots, n-1, [n]\} \ \ X_i \sim_{P_i} Y_i$$

where $X_0 a_1 X_1 \dots a_n[X_n], Y_0 a_1 Y_1 \dots a_n[Y_n] \in \mathbf{RT}(P)$, $P_0 = P$ and $P_i = P \| X_0 a_1 \dots a_i$ for $i \in \{1, \dots, n-1\}$.

For traces with a forbidden final observation it is in addition required that the final refusals are identical (regardless of the type of the final observation):

$$\begin{array}{ll} \sigma.X.a \sim_P \sigma'.X.a & \text{if } \sigma \sim_P \sigma' \wedge \sigma.X, \sigma'.X \in \mathbf{RT}(P) \wedge \sigma.X.a, \sigma'.X.a \notin \mathbf{RT}(P) \\ \sigma.X \sim_P \sigma'.X & \text{if } \sigma \sim_P \sigma' \wedge \sigma, \sigma' \in \mathbf{RT}(P) \wedge \sigma.X, \sigma'.X \notin \mathbf{RT}(P) \end{array}$$

The following key proposition states that fundamentally equivalent refusal traces are indistinguishable from the perspective of a correct refinement.

**Proposition 6.** *Suppose $P \sqsubseteq^{\ell}_{\mathbf{RT}} Q$. For any $\sigma \in \mathbf{RT}^{\ell}(P)$:*

1. $\sigma \in \mathbf{RT}^{\ell}(Q) \iff \mathit{top}_P(\sigma) \in \mathbf{RT}^{\ell}(Q)$
2. $Q \| \sigma = Q \| \mathit{top}_P(\sigma)$

**Proof.** Fix $\ell \in \mathbb{N}$. The implication from right to left in statement 1, as well as inclusion from right to left in statement 2 above are both immediate due to $\sigma \preceq \mathsf{top}_P(\sigma)$ and downward closure of refusal traces. We therefore focus on showing that:

1. $\sigma \in \mathbf{RT}^{\ell}(Q) \implies \mathsf{top}_P(\sigma) \in \mathbf{RT}^{\ell}(Q)$
2. $Q \| \sigma \subseteq Q \| \mathsf{top}_P(\sigma)$

The above properties are proved simultaneously by induction on the length of $\sigma$. Proof of statement 2 makes an assumption that $\sigma \in \mathbf{RT}^{\ell}(Q)$ – the other case is trivial.

**Base.** Immediate since $top_P(\epsilon) = \epsilon$.

**Inductive step.** We assume that the conjunction of statements 1 and 2 holds for all $\sigma$ of length not exceeding $k$. Let $\sigma \in \mathbf{RT}^{k+1}(P)$, where $k+1 \leq \ell$. There are two cases:

- $\sigma = X_0 a_1 \dots a_k X_k$:
  The case when $X_k = \bullet$ follows immediately from the inductive hypothesis (this is because $X_0 a_1 \dots a_k \bullet$ and $X_0 a_1 \dots a_k$ are semantically equivalent). Hence we assume that $X_k \in \mathcal{P}(Act)$.
  Let $s \in Q \| X_0 a_1 \dots a_k X_k$; observe that $s$ is stable due to $X_k \in \mathcal{P}(Act)$. Since in particular $s \in Q \| X_0 a_1 \dots a_k$, from the inductive hypothesis we already know that $\mathsf{top}_P(X_0 a_1 \dots a_k) \in \mathbf{RT}^{\ell}(Q)$, and $s \in Q \| \mathsf{top}_P(X_0 a_1 \dots a_k)$. In order to prove statements 1 and 2, it suffices to show that $\mathsf{top}_{P_k}(X_k) \in \mathbf{R}(s)$, where $P_k = P \| X_0 a_1 \dots a_k$.
  Observe that $X_k \subseteq \mathbf{SR}(s)$ and since $\mathbf{SR}(s) \in \mathbf{FR}(Q \| \mathsf{top}_P(X_0 a_1 \dots a_k))$, from Proposition 2 and Corollary 1 we obtain $\mathbf{SR}(s) \in \mathbf{FR}(P \| \mathsf{top}_P(X_0 a_1 \dots a_k)) = \mathbf{FR}(P_k)$. Finally, $\mathsf{top}_{P_k}(X_k)$ is the smallest fundamental refusal in $P_k$ that includes $X_k$, yielding $\mathsf{top}_{P_k}(X_k) \subseteq \mathbf{SR}(s)$. Hence from downward closure of refusals we obtain $\mathsf{top}_{P_k}(X_k) \in \mathbf{R}(s)$.
- $\sigma = X_0 a_1 \dots X_k a_{k+1}$:
  Let $s \in Q \| X_0 a_1 \dots X_k a_{k+1}$. Then there must be some $q \in Q \| X_0 a_1 \dots X_k$ such that $q \overset{a_{k+1}}{\Longrightarrow} s$. From the proof of the previous case, we already know that $\mathsf{top}_P(X_0 a_1 \dots X_k) \in \mathbf{RT}^{k+1}(Q)$ and $q \in Q \| \mathsf{top}_P(X_0 a_1 \dots X_k)$. Hence

  $$Q \xRightarrow{\mathsf{top}_P(X_0 a_1 \dots X_k)} q \overset{a_{k+1}}{\Longrightarrow} s,$$

  which entails $\mathsf{top}_P(\sigma) = \mathsf{top}_P(X_0 a_1 \dots X_k a_{k+1}) = \mathsf{top}_P(X_0 a_1 \dots X_k) a_{k+1} \in \mathbf{RT}^{k+1}(Q)$, proving statement 1, and $s \in Q \| \mathsf{top}_P(X_0 a_1 X_1 \dots X_k a_{k+1})$ (statement 2). □

Proposition 6 allows us to considerably restrict our test suite by including only traces whose intermediate refusals (i.e. excluding the last one) are fundamental. Our new test suite based on fundamental equivalence is defined as:

$$\mathbf{TS}_{\mathbf{3}}^{\ell}(P) \quad \triangleq \quad \{\mathsf{top}_P(\sigma) \mid \sigma \in \mathbf{TS}_{\mathbf{2}}^{\ell}\}$$

**Lemma 3.** *For any process $P$ [and length bound $\ell$], $\boldsymbol{TS}_{\mathbf{3}}^{[\ell]}(P)$ is complete for $P$ w.r.t. $\sqsubseteq_{\boldsymbol{RT}}^{[\ell]}$.*

**Proof.** Fix $P$ and $\ell$, and let $Q$ be such that $P \not\sqsubseteq_{\mathbf{RT}}^{\ell} Q$. We assume in addition that $\ell$ is the lowest number with such a property, i.e. $P \sqsubseteq_{\mathbf{RT}}^{\ell-1} Q$.

From the completeness of $\mathbf{TS}_{\mathbf{2}}^{\ell}(P)$, there is some trace $\rho \in \mathbf{RT}^{\ell}(Q) \cap \mathbf{TS}_{\mathbf{2}}^{\ell}(P)$. Let $\rho = \sigma.\omega$, where $\sigma \in \mathbf{RT}^{\ell-1}(P)$ and $\omega \in \{X, X.a\}$. In particular, there must be a state $s \in Q \,\|\sigma$ such that $\omega \in \mathbf{RT}(s)$. Since $P \sqsubseteq_{\mathbf{RT}}^{\ell-1} Q$, from Proposition 6 we have $s \in P \| \mathsf{top}_P(\sigma)$, and hence $\mathsf{top}_P(\sigma).\omega \in \mathbf{RT}^{\ell}(Q)$. Since $\mathsf{top}_P(\sigma.\omega) = \mathsf{top}_P(\sigma).\omega$ and $\mathsf{top}_P(\sigma.\omega) \in \mathbf{TS}_{\mathbf{3}}^{\ell}$, we obtain $\mathsf{top}_P(\rho) \in \mathbf{RT}^{\ell}(Q) \cap \mathbf{TS}_{\mathbf{3}}^{\ell}$. □

We conclude this section with an equivalent, self-contained inductive definition of the test suite $\mathbf{TS}_{\mathbf{3}}^{\ell}$. We have

$$\mathbf{TS}_{\mathbf{3}}^{0}(P) = \emptyset$$

while for $\ell \geq 0$:

$$\begin{aligned}
\mathbf{TS}_{\mathbf{3}}^{\ell+1}(P) \; = & \bigcup_{X \in \min_{\not\subseteq} \mathbf{R}(P)} \{X\} \\
& \cup \bigcup_{X_\cap \in \mathbf{FR}(P)} \\
& \Big( \bigcup_{X_m \in \text{min-base}_P(X_\cap)} \bigcup_{a \in (X_\cap \setminus X_m)} \{X_m.a\} \\
& \cup \bigcup_{a \in Act \setminus X_\cap} \; \bigcup_{\sigma \in \mathbf{TS}_{\mathbf{3}}^{\ell}(P \| X_\cap.a)} \{X_\cap.a.\sigma\} \Big) \\
& \cup \bigcup_{a : \bullet.a \notin \mathbf{RT}(P)} \{\bullet.a\} \\
& \cup \bigcup_{a : \bullet.a \in \mathbf{RT}(P)} \bigcup_{\sigma \in \mathbf{TS}_{\mathbf{3}}^{\ell}(P \| \bullet.a)} \{\bullet.a.\sigma\}
\end{aligned}$$

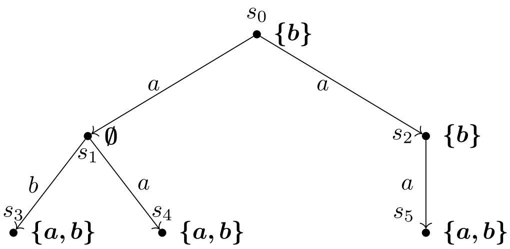
The first component of the sum (line 1 above) are refusals locally forbidden in $P$. Furthermore, iterating over all fundamental refusals $X_\cap$, the test suite includes all traces “contributed” by each fundamental cluster: locally forbidden actions (line 3), as well as longer traces prefixed with a fundamental refusal and relevant actions (line 4). The last two lines handle forbidden traces prefixed with a null refusal: locally forbidden events (line 5) and traces after valid events (line 6).

## 5. A small and simple test suite

In this section, we present a further refinement of our test suite which, while not necessarily minimal, has a reasonable balance between the size and ease/efficiency of construction.

### 5.1. Strictly smaller refusal traces

In the previous section, we have dealt with redundancies in test suites resulting from clusters of equivalent traces which represent the same behaviour. There is a further possible source of redundancy which is straightforward to remove: due to the downward-closure property of refusal traces (Proposition 1), a forbidden trace $\sigma$ is redundant in a test suite, if the suite contains a smaller trace w.r.t. $\preceq$.



Consider the process depicted above. Our test suite $\mathbf{TS}_{\mathbf{3}}$ for this process contains in particular traces $\{b\}.a.\emptyset.a.\emptyset.a$, and $\{b\}.a.\emptyset.a.\emptyset.a$. Clearly, if a faulty implementation fails the test $\{b\}.a.\{b\}.a.\emptyset.a$, then it must also fail the test $\{b\}.a.\emptyset.a.\emptyset.a$, hence the former trace (larger w.r.t. $\prec$) can be removed from the suite without affecting completeness. This observation gives rise to another important optimisation – our test suite can be restricted so that it contains only minimal traces w.r.t. $\prec$.

$$\mathbf{TS}_{\mathbf{4}}^{\ell}(P) \quad \triangleq \quad \min_{\prec} \mathbf{TS}_{\mathbf{3}}^{\ell}$$

**Lemma 4.** *For any process $P$ [and length bound $\ell$], $\boldsymbol{TS}_{\mathbf{4}}^{[\ell]}(P)$ is complete for $P$ w.r.t. $\sqsubseteq_{\boldsymbol{RT}}^{[\ell]}$.*

**Proof.** Suppose that $P \not\sqsubseteq_{\mathbf{RT}}^{\ell} Q$. From the completeness of $\mathbf{TS}_{\mathbf{3}}^{\ell}(P)$, we know that there is some trace $\rho \in \mathbf{RT}^{\ell}(Q) \cap \mathbf{TS}_{\mathbf{3}}^{\ell}(P)$. Since by definition $\mathbf{TS}_{\mathbf{4}}^{\ell}(P)$ contains all minimal elements of $\mathbf{TS}_{\mathbf{3}}^{\ell}(P)$ w.r.t. $\prec$, there must be in particular a trace $\rho_{min} \in \mathbf{TS}_{\mathbf{4}}^{\ell}(P)$ such that $\rho_{min} \preceq \rho$. Finally, from the downward closure of refusal traces w.r.t. $\preceq$ (Proposition 1), we obtain $\rho_{min} \in \mathbf{RT}^{\ell}(Q)$. □

### 5.2. Test generation algorithm

We shall now briefly describe an effective procedure to construct our test suite $\mathbf{TS_4}$ (Algorithm 1). It takes as input an LTS-based reference process $P$, from which information such as state refusals can be readily obtained. The test suite is stored in a global object $\mathbf{TS}$, whose insertion operation $\mathbf{TS}$.Add() must ensure that a refusal trace is added only if $\mathbf{TS}$ contains no larger trace w.r.t. $\prec$ and upon insertion of a trace, all larger traces in $\mathbf{TS}$ are removed.

**Algorithm 1:** Test generation algorithm. Invoking $GenerateTests(\ell, \epsilon, P)$ outputs a test suite for $\sqsubseteq_{\mathbf{RT}}^{\ell}$ w.r.t. specification $P$.

```
GenerateTests($\ell$,$\sigma$,$P$)
// generate part of the test suite for $\sqsubseteq_{RT}^{\ell}$ w.r.t. specification $P$
// consisting of $\sigma$-prefixed traces
  1. if $(|\sigma| \geq \ell)$ return;
     //I: generate forbidden refusals
  2. for all $(X \in \min_{\not\subseteq}(\mathbf{R}(P)))$ TS.Add($\sigma.X$)
     //II: add forbidden continuations and recursively generate extensions of the current trace
  3. $P_\sigma := P \| \sigma$;  compute $\mathbf{FR}(P_\sigma)$ and min-base$_{P_\sigma}$
  4. for each $X_\cap \in \mathbf{FR}(P_\sigma)$ in non-decreasing order
        //II.1 if the cluster is nontrivial then we need to add forbidden continuations
     (a) if $(|[X_\cap]|_{P_\sigma} > 1)$
            $\mathcal{Y} :=$ min-base$_{P_\sigma}(X_\cap)$
            for all $(Y \in \mathcal{Y})$
               for all $(a \in X_\cap \setminus Y)$ TS.Add($\sigma.Y.a$)
        // II.2 recursively generate extensions of the current trace
     (b) for all $(a \in Act \setminus X_\cap)$ $GenerateTests(k+1, \ell, \sigma.X_\cap.a, P)$
     //handling null refusals separately
  5. for each $a \in Act$
        // forbidden continuation – add to test suite
     (a) if $(\bullet.a \notin \mathbf{RT}(P_\sigma))$ TS.Add($\sigma.\bullet.a$)
        // valid continuation: recursively generate extensions of $\bullet.a$
     (b) if $(\bullet.a \in \mathbf{RT}(P_\sigma))$ $GenerateTests(k+1, \ell, \sigma.\bullet.a, P)$
```

The number of all refusal traces of length $\leq \ell$ is bounded by $\mathcal{O}\big((2^{|Act|} \cdot |Act|)^{\ell}\big)$. If we assume that a bound on the maximal number of fundamental refusals after every refusal trace is given by $\mathsf{Max}_{\cap}$, then the size of our test suite is $\mathcal{O}\big((|\mathsf{Max}_{\cap}| \cdot |Act|)^{\ell-1} \cdot (2^{|Act|} \cdot |Act|)\big) = \mathcal{O}(|Act|^{\ell} \cdot |\mathsf{Max}_{\cap}|^{\ell-1} \cdot 2^{|Act|})$. We note that $|\mathsf{Max}_{\cap}|$ can be polynomial w.r.t. $|Act|$ for instance if refusals have hierarchical structure (i.e. constitute a linear order), or are disjoint.

## 6. Additional sources of redundancies

In this section, we identify two additional types of redundancy in the test suite $\mathbf{TS_4}$. The first type concerns refusals directly preceding a forbidden action, the other a rather counter-intuitive possibility of detecting a violation through a different violation occurring at a later stage of computation.

### 6.1. Redundant event contexts

Suppose that the structure of refusals of some reference process $\widehat{P}$ is exactly as illustrated in Fig. 3. The test suite $\mathbf{TS_4}(\widehat{P})$ will contain in particular traces $\{a, c\}.b$, $\{a, c\}.d$, $\{a, d\}.b$. Observe that the first trace can be removed: any state $s$ that exhibits $\{a, c\}.b$ will either exhibit $\{a, c\}.d$ (in case $s \xrightarrow{d}$), or $\{a, d\}.b$ (in case $s \not\xrightarrow{d}$). Note that this pattern occurs for any $X.b \in \mathsf{TS}$ whenever there is an action $d$ such that $X.d \in \mathsf{TS}$, and some $Y \subseteq X$ such that $(Y \cup \{d\}).b \in \mathsf{TS}$.

We can characterise this type of redundancy with a rather broad definition: a test suite TS contains a redundant action context if there is a trace $\sigma.X.a \in \mathsf{TS}$ of length $k$ and a corresponding set $\mathsf{TS}_{\sigma.X.a} \subseteq \mathsf{TS}$ such that each trace in $\mathsf{TS}_{\sigma.X.a}$

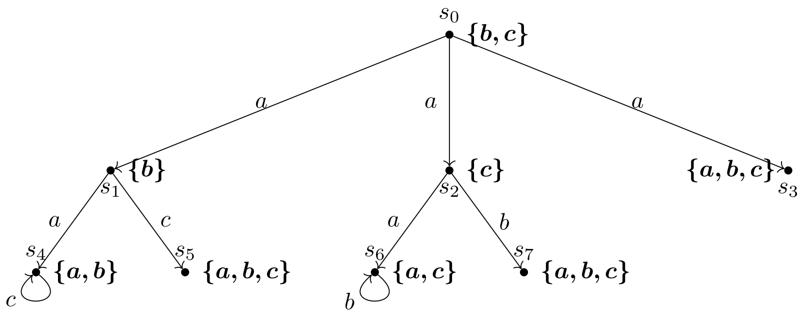


**Fig. 4.** A reference process $P$ for which the forbidden trace $\sigma = \{b,c\}.a.\{b,c\}.a$ may be removed from the test suite, since its presence will be detected indirectly through longer traces.

has length at most $k$ and for each process $Q$ such that $\sigma.X.a \in \mathbf{RT}(Q)$, there is some $\pi \in \mathsf{TS}_{\sigma.X.a} \cap \mathbf{RT}(Q)$. We stress that $\mathsf{TS}_{\sigma.X.a}$ may not contain traces longer that $k$ – redundancies due to longer traces will be described in the following section.

At the moment, apart from a somewhat brute force method described in Section 7, we have not found a more elegant/efficient technique to characterise and handle this type of redundancies – we leave it as a potential research question; we only briefly explain why we conjecture that the problem is likely of considerable computational complexity.

As we shall explain in greater detail in Section 7.1.1, to obtain completeness we need to ensure that a test suite is capable of detecting every incorrect *state refusal*, that is, a state refusal in the implementation that is different from a fundamental refusal. In particular, if we zoom into a local perspective of a cluster – say within the context of $P$ after a trace $\sigma$ we consider some $X_{top} \in \mathbf{FR}(P\|\sigma)$ and its corresponding cluster $[X_{top}]_{P\|\sigma}$ – then each state refusal in the cluster different from the top refusal (i.e. belonging to $[X_{top}]_{P\|\sigma} \setminus \{X_{top}\}$) needs to be detected by the test suite.

Observe that a refusal trace of the form $\sigma.X_m.a$ is capable of detecting all incorrect state refusals contained in the interval $[X_m, X_{top} \setminus \{a\}]$ (where $[X,Y] = \{Z \mid X \subseteq Z \subseteq Y\}$). Hence to ensure detection of all incorrect state refusals within a cluster $C = [X_{top}]_{P\|\sigma}$, we need to include refusal traces for which the corresponding intervals form an *interval covering* of the cluster $C$. As shown in [24], this problem is NP-complete for posets in general. While this does not immediately allow us to infer the complexity status of our problem, it indicates that finding a minimal set of refusal traces detecting all incorrect state refusals within a given cluster may be NP-hard. Importantly, our problem is much more complex than its local variant. This is because one may obtain additional material for inference from outside the cluster, in particular from a smaller trace context i.e. $\sigma'.Y.a \prec \sigma.X.a$, and a certain choice of locally complete traces from a cluster may lead to a more effective reduction modulo $\preceq$ (Section 5.1).

### 6.2. Longer traces make a shorter one redundant

We now arrive at the trickiest source of redundancy, although it can be debated if it is a true redundancy at all. Namely, it can happen that certain refusal traces can be safely removed from a test suite due to, somewhat surprising, distinguishing power of longer traces.

Consider the LTS representing a reference model $P$ (assuming $Act = \{a,b,c\}$), depicted in Fig. 4. A forbidden trace $\sigma = \{b,c\}.a.\{b,c\}.a$ appears in our test suite $\mathbf{TS_4}$ – this is because, in particular, $\{b,c\} \in \mathsf{min\text{-}base}_{P\|\{b,c\}.a}(\{a,b,c\})$, and $a \in \{a,b,c\} \setminus \{b,c\}$, and moreover, there are no smaller traces (w.r.t. $\prec$) than $\sigma$ in our test suite.

However, it turns out that removing $\sigma$ from our test suite does not affect completeness. This is because any system that exhibits $\sigma$ will also exhibit certain longer forbidden traces, which are present in the suite.

Indeed, suppose that a system $Q$ exhibits the trace $\sigma = \{b,c\}.a.\{b,c\}.a$. Let $s_\sigma$ be an arbitrary state in $Q\|\sigma$.

- if $s_\sigma \xrightarrow{b}$, then $Q$ exhibits in particular $\{b,c\}.a.\{b\}.a.\emptyset.b$, forbidden by $P$
- if $s_\sigma \overset{b}{\not\longrightarrow}$, then $Q$ exhibits in particular $\{b,c\}.a.\{c\}.a.\{b\}$, forbidden by $P$

Hence the lack of $\sigma$ in the test suite will always be compensated by longer traces, whose prefixes of length $|\sigma|$ coincide with $\sigma$ on every index except the last refusal in $\sigma$.

The key question is whether we really wish to remove $\sigma$ from our test suite in the scenario above. We believe that it should *not* be removed, as it provides the shortest and most accurate counterexample for a violation. From an application perspective, one strives to have the clearest, most succinct explanation of an error, and such a roundabout manner of detecting violations through longer traces clearly appears to be undesirable.

A more striking example illustrating our point is given in Fig. 5. For an arbitrary $L$, we can construct a specification $P_L$ which, similarly to the process $P$ in Fig. 4, has a forbidden trace $\{b,c\}.a.\{b,c\}.a$ that can be removed without affecting completeness. Indeed, for any specification $Q$ exhibiting $\{b,c\}.a.\{b,c\}.a$, $Q$ must also exhibit one of the two classes of forbidden traces:

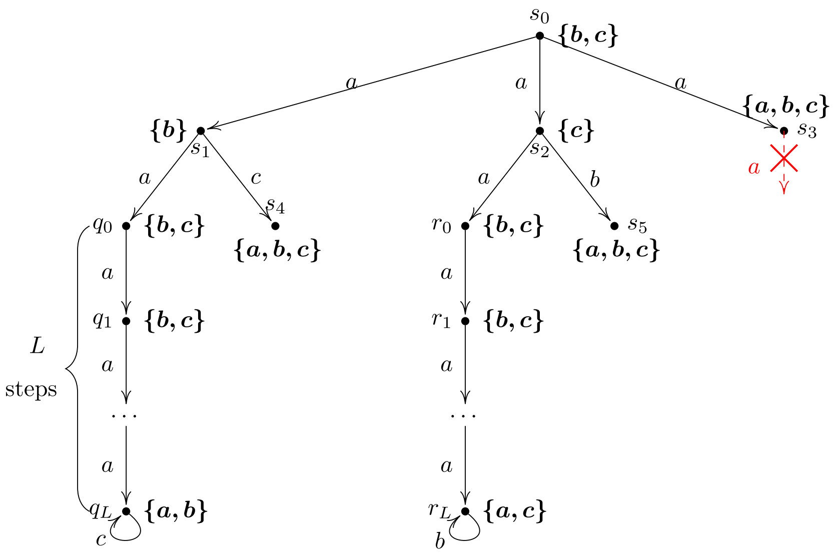


**Fig. 5.** A family of processes $P_L$, where $L \geq 1$ is a parameter specifying the length of two branches of computation through $a$-transitions, which terminate in states that give rise to irreconcilable requirements ($q_L$ and $r_L$).

- extensions of $\{b,c\}.a.\{b,c\}.a$ where $a$ is refused after less than $L$ steps i.e.

$$\{b,c\}.a.\{b\}.a.(\bullet.a.)^k\{a\}, \quad \{b,c\}.a.\{c\}.a.(\bullet.a.)^k\{a\}$$

- or, if $Q$ does *not* exhibit any of the above traces, then it must have a trace $\{b,c\}.a.\{b\}.a.(\{b,c\}.a.)^L$. In any stable state after that trace, action $b$ must be either enabled or refused, hence $Q$ must necessarily exhibit one of the two forbidden traces:

$$\{b,c\}.a.\{b\}.a.(\bullet.a.)^L \bullet .b, \ \ \{b,c\}.a.\{c\}.a.(\bullet.a.)^L\{b\}$$

In the latter scenario – if $Q$ exhibits $\{b,c\}.a.\{b,c\}.a$ and also has one of the two above traces, then a suite "optimised" by removing the trace $\{b,c\}.a.\{b,c\}.a$ will instead produce a counterexample of length $L+3$ – and our construction works for arbitrarily large $L$! Clearly, obtaining counterexample of length e.g. 5,000 where the fault occurs after 2 steps is undesirable in any reasonable testing scenario, and illustrates that one needs a stronger notion than mere completeness.

Observe that the key problem with the original definition of completeness is that it allows for potential gaps in conformance up to a smaller number of steps than the conformance relation under consideration – e.g. a test suite $TS$ may be complete for $\sqsubseteq^{\ell}_{\mathbf{RT}}$, but not complete for $\sqsubseteq^{k}_{\mathbf{RT}}$ where $k < \ell$. In view of that, we introduce a sanity condition on a test suite called uniform completeness, which for simplicity we define in the context of refusal trace refinement.[4] Formally, a test suite TS for a reference model $P$ is *uniformly complete (u.c.)* for $\sqsubseteq_{\mathbf{RT}}$ [$\sqsubseteq^{\ell}_{\mathbf{RT}}$] whenever for all $k$ [$k \leq \ell$] it is complete for $\sqsubseteq^{k}_{\mathbf{RT}}$.

## 7. Minimal test suites

Our goal in this section is to provide an effective method to obtain minimal complete test suites for finitary variants of refusal trace refinement. As argued in the previous section, we are primarily interested in minimal *uniformly complete* test suites, however, we shall also investigate the case of general completeness.

Let us first consider the most basic definition of the entities that we wish to construct. Mathematically speaking, we are interested in the following

$$\min\{\mathsf{TS} : \text{test suite for } P \ | \mathsf{TS} \text{ is [uniformly] complete for } P \text{ w.r.t. } \sqsubseteq^{\ell}_{\mathbf{RT}}\}$$

In order to obtain a more constructive characterisation of the above, we shall consider the following problems:

1. How can one restrict the universe of test suites to consider? In particular, can we consider only those contained in $\mathbf{TS_4}$?
2. How can we characterise completeness in a more efficient manner? In particular, we wish to decompose all possible faults into some class of atomic and independent instances of violations that can be then effectively used by some enumerative algorithm.

[4] One can define uniform completeness for an arbitrary conformance relation $\sqsubseteq$; the restrictions $\sqsubseteq^k$ should be then defined using projection mappings on processes, which mimick behaviour of a process up to $k$ number of steps – then $P \sqsubseteq^k Q$ iff $k$-th projections of $P$ and $Q$ are related by $\sqsubseteq$.

We shall start this section by exploring the second problem, and provide a characterisation of completeness using *violation traces* based on erroneous state refusals. The characterisation allows us to construct complete test suites which ensure that each occurrence of a violation trace will be detected. In the second part of the section, we show that the test suites we have defined are in fact minimal; as part of the proof, we answer in the affirmative the question whether we can restrict ourselves to test suites containing only traces from $\mathbf{TS_4}$.

### *7.1. Test suites based on detectors of violation traces*

#### *7.1.1. Violation traces*

Our goal is to characterise all causes of inadmissible behaviour w.r.t. a given specification through a certain set of *violations*, which represent *atomic*, and *independent* instances of faults. If, in addition, we can easily identify the sets of traces in a test suite that detect each violation, then devising the minimal test suite boils down to finding a set of traces of minimal size with the property that each violation is detected by some test trace.

As we have seen in the previous section, forbidden refusal traces themselves (such as those used in our test suites) cannot be used in such a characterisation. They are not sufficiently atomic: traces of a process always result from some model component, in particular a state with an erroneous transition or refusal, such a component typically manifests itself in multiple inadmissible traces.

In the setting without internal actions [14], where each state is stable, one can classify violations according to forbidden state refusals. Namely, for every non-conforming process $Q$, there must be a state $s$, reachable with some admissible trace $\sigma$, that has an incorrect state refusal – i.e. a state refusal which is either a disallowed refusal or does not contain a forbidden action. With silent actions present, a violation may also occur due to a transient state which admits a forbidden action. It transpires that the two abovementioned types of violations provide us with a comprehensive decomposition into independent violations that we require.

More specifically, for a reference process $P$, we define its *violation traces* as traces composed of a fundamental trace $\sigma$ of $P$, followed by either a state refusal $\widehat{X}$ forbidden in $P\|\sigma$, or a nullary refusal $\bullet$ followed by a forbidden action in $P\|\sigma$. Note that in the first case, the fact that a given refusal $\widehat{X}$ is forbidden in $P\|\sigma$ can be conveniently expressed as $\widehat{X} \notin \mathbf{FR}(P\|\sigma)$.

Our definition can be further optimised by excluding those state refusals whose complements (ready sets) contain an action $a$ such that $\bullet.a$ is forbidden in a given context. We first define the *strictly forbidden actions* of a process $P$ as:

$$\mathbf{SFA}(P) \triangleq \{a \in Act \mid \bullet.a \notin \mathbf{RT}(P)\}$$

Since strictly forbidden refusals are already handled by dedicated violation traces, for violation traces ending in state refusals we need to only consider state refusals that subsume all strictly forbidden actions. With this insight, the violation traces can now be formally defined as follows:

$$\begin{aligned}
\mathbf{VT}_P \quad &\triangleq \quad \{\sigma.\chi \in \mathbf{RT} \mid \sigma \in \mathbf{FRT}(P) \\
&\quad \wedge \big((\chi = \widehat{X} : \widehat{X} \neq \bullet \wedge \widehat{X} \notin \mathbf{FR}(P\|\sigma) \wedge \mathbf{SFA}(P\|\sigma) \subseteq \widehat{X}) \\
&\quad \vee (\chi = \bullet.a : \bullet.a \notin \mathbf{RT}(P\|\sigma))\big)\} \\
\mathbf{VT}_P^{\ell} \quad &\triangleq \quad \mathbf{VT}_P \cap \mathbf{RT}^{\ell}
\end{aligned}$$

The latter class $\mathbf{VT}_P^{\ell}$ are violation traces which give rise to forbidden refusal traces of length $\leq \ell$. The set of violations w.r.t. $P$ exhibited by an implementation $Q$ is defined as:

$$\begin{aligned}
\mathbf{VT}_P^{[\ell]}(Q) \quad \triangleq \quad &\{\sigma.\widehat{X} \in \mathbf{VT}_P^{[\ell]} \mid \sigma \in \mathbf{RT}(Q) \wedge \widehat{X} \in \mathbf{SR}(Q\|\sigma)\} \\
&\cup \{\sigma.\bullet.a \in \mathbf{VT}_P^{[\ell]} \cap \mathbf{RT}(Q)\}
\end{aligned}$$

In the above, $[l]$ is simply a shorthand, making it possible to simultaneously define $\mathbf{VT}_P(Q)$ and $\mathbf{VT}_p^{\ell}(Q)$. Note that we only consider violations that can be reached by fundamental traces; this is formally justified by the following lemma.

**Lemma 5.** $P \not\sqsubseteq_{\mathbf{RT}}^{\ell} Q \iff \mathbf{VT}_P^{\ell}(Q) \neq \emptyset$

**Proof.** "$\Longrightarrow$": Suppose $P \not\sqsubseteq_{\mathbf{RT}}^{\ell} Q$. From the completeness of $\mathbf{TS_4}$, there must be a trace $\rho \in \mathbf{TS}_4^{\ell}(P) \cap \mathbf{RT}^{\ell}(Q)$. From the definition of $\mathbf{TS_4}$, we have $\rho = \sigma.\omega$ where $\sigma \in \mathbf{FRT}^{\ell-1}(P)$ and either $\omega = X$ for $X \in \mathscr{P}(Act)$, or $\omega = X.a$ for $X \in \mathscr{P}(Act) \cup \{\bullet\}$. Let $s \in Q\|\sigma$ be such that $\omega \in \mathbf{RT}(s)$. The possible cases are:

- $\omega = X$: let $\widehat{X} = \mathbf{SR}(s)$; we have $X \subseteq \widehat{X}$. Since $X$ is a forbidden refusal in $P\|\sigma$, then so is $\widehat{X}$ (Proposition 1), thus in particular we have $\widehat{X} \notin \mathbf{FR}(P\|\sigma)$. Hence $\sigma.\widehat{X} \in \mathbf{VT}_P^{\ell}(Q)$
- $\omega = X.a$ for $X \in \mathscr{P}(Act)$: since $\sigma.X.a \notin \mathbf{RT}(P)$, from Proposition 5 we have $a \in \mathrm{top}_{P\|\sigma}(X)$. Since $\mathrm{top}_{P\|\sigma}(X)$ is the least fundamental refusal that subsumes $X$, every fundamental refusal subsuming $X$ must in particular contain $a$. Let $\widehat{X} = \mathbf{SR}(s)$; we have $X \subseteq \widehat{X}$ but $a \notin \widehat{X}$. Hence $\widehat{X} \notin \mathbf{FR}(P\|\sigma)$. Observe that if there is some $b \in \mathbf{SFA}(P\|\sigma) \setminus \widehat{X}$, then we simply have $\sigma.\bullet.b \in \mathbf{VT}_P^{\ell}(Q)$. Otherwise $\mathbf{SFA}(P\|\sigma) \subseteq \widehat{X}$, and therefore $\sigma.\widehat{X}$ meets all the conditions from the definition of $\mathbf{VT}_P^{\ell}$, hence $\sigma.\widehat{X} \in \mathbf{VT}_P^{\ell}(Q)$.

- $\omega = \bullet.a$: we have $\sigma.\bullet.a \in \mathbf{VT}^{\ell}_{P}$ immediately from the definition of $\mathbf{VT}^{\ell}_{P}$, hence $\sigma.\bullet.a \in \mathbf{VT}^{\ell}_{P}(Q)$.

"$\Longleftarrow$": Suppose $\mathbf{VT}^{\ell}_{P}(Q) \neq \emptyset$. Take any $\rho \in \mathbf{VT}^{\ell}_{P}(Q)$. From the definition of $\mathbf{VT}^{\ell}_{P}$, $\rho$ must be of the form $\rho = \sigma.\chi$, where $\sigma \in \mathbf{FRT}^{\ell-1}(P)$, and we have one of the following cases:

- $\chi = \widehat{X}$ such that $\sigma.\widehat{X} \notin \mathbf{RT}^{\ell}(P)$: since $\sigma.\widehat{X} \in \mathbf{VT}^{\ell}_{P}(Q)$, from the definition of $\mathbf{VT}^{\ell}_{P}(Q)$ we have in particular $\sigma.\widehat{X} \in \mathbf{RT}^{\ell}(Q)$, hence $P \not\sqsubseteq^{\ell}_{\mathbf{RT}} Q$.
- $\chi = \widehat{X}$ such that $\sigma.\widehat{X} \in \mathbf{RT}^{\ell}(P)$, and $\widehat{X} \notin \mathbf{FR}(P\|\sigma)$: since $\widehat{X} \notin \mathbf{FR}(P\|\sigma)$, there is some $a \in \mathsf{top}_{P\|\sigma}(\widehat{X}) \setminus \widehat{X}$, and from Proposition 5 we obtain $\sigma.\widehat{X}.a \notin \mathbf{RT}(P)$. On the other hand, since $\sigma.\widehat{X} \in \mathbf{VT}^{\ell}_{P}(Q)$, we have $\widehat{X} \in \mathbf{SR}(Q\|\sigma)$, and therefore $\widehat{X}.a \in \mathbf{RT}(Q\|\sigma)$, which yields $\sigma.\widehat{X}.a \in \mathbf{RT}(Q)$. Hence $P \not\sqsubseteq^{\ell}_{\mathbf{RT}} Q$.
- $\chi = \bullet.a$ such that $\sigma.\bullet.a \notin \mathbf{RT}(P)$: from the definition of $\mathbf{VT}^{\ell}_{P}(Q)$ it follows that $\sigma.\bullet.a \in \mathbf{RT}^{\ell}(Q)$, hence we immediately obtain $P \not\sqsubseteq^{\ell}_{\mathbf{RT}} Q$. □

### 7.1.2. Detectors

Refusal traces that witness a given violation trace are captured in the following notion of *detectors*. We start by defining, for each specification process $P$ and its violation trace $\rho = \sigma.\chi \in \mathbf{VT}_{P}$, the set of *maximal detectors* of $\rho$:

$$\mathsf{maxdetectors}_P(\sigma.\chi) \triangleq \begin{cases} \{\sigma.\widehat{Y}.a \mid a \in \mathsf{top}_{P\|\sigma}(\widehat{Y}) \setminus \widehat{Y}\} & \text{if } \chi = \widehat{Y} \in \mathscr{P}(Act) \text{ such that } \sigma.\widehat{Y} \in \mathbf{RT}^{\ell}(P) \\ \{\sigma.\widehat{Y}\} & \text{if } \chi = \widehat{Y} \in \mathscr{P}(Act) \text{ such that } \sigma.\widehat{Y} \notin \mathbf{RT}^{\ell}(P) \\ \{\sigma.\bullet.a\} & \text{if } \chi = \bullet.a \end{cases}$$

The set of detectors consists of all traces smaller w.r.t. $\preceq$ than some maximal detector and forbidden by $P$.

$$\mathsf{detectors}_P(\rho) \triangleq \{\delta \in \mathbf{RT} \setminus \mathbf{RT}(P) \mid \exists \delta_{max} \in \mathsf{maxdetectors}_P(\rho) \mid \delta \preceq \delta_{max}\}$$

In addition, for a test suite $\mathsf{TS} \subseteq \mathbf{RT} \setminus \mathbf{RT}(P)$, we define:

$$\mathsf{detectors}_P(\rho, \mathsf{TS}) \triangleq \mathsf{detectors}_P(\rho) \cap \mathsf{TS}$$

Our first observation is that detectors have the same length as the corresponding violation trace. For maximal detectors this holds immediately from their definition, whereas for arbitrary detectors it follows from the fact that for every trace $\delta$ such that $\delta \preceq \delta_{max} \in \mathsf{maxdetectors}_P(\sigma.\chi)$ and $|\delta| < |\delta_{max}|$ there is a trace $\sigma \in \mathbf{FRT}(P)$ such that $\delta \preceq \sigma$, and from the downward closure of refusal traces (Proposition 1) we have $\delta \in \mathbf{RT}(P)$.

The term *detectors* is justified as they indeed detect a given violation in every implementation that exhibits it:

**Lemma 6.** *Take any process $P$ and its violation trace $\rho \in \mathbf{VT}^{\ell}_{P}$. For any refusal trace $\delta \in \mathsf{detectors}_P(\rho)$ and process $Q$ we have:*

$$\rho \in \mathbf{VT}^{\ell}_{P}(Q) \implies \delta \in \mathbf{RT}^{\ell}(Q)$$

**Proof.** Let $\rho \in \mathbf{VT}^{\ell}_{P}$ be of the form $\rho = \sigma.\chi$ where $\sigma \in \mathbf{FRT}^{\ell-1}(P)$ and $\chi \in \mathbf{RT}^{1}$. From the definition of $\mathsf{detectors}_P$, $\delta$ is of the form $\delta = \sigma'.\omega$, where $\sigma' \preceq \sigma$. We have three possible cases:

- $\chi = \widehat{X}$ such that $\sigma.\widehat{X} \notin \mathbf{RT}^{\ell}(P)$: since $\sigma.\widehat{X} \in \mathbf{VT}^{\ell}_{P}(Q)$, we have in particular $\sigma.\widehat{X} \in \mathbf{RT}^{\ell}(Q)$. On the other hand, since $\delta \in \mathsf{detectors}_P(\sigma.\widehat{X})$, from the definition of $\mathsf{detectors}_P$, $\delta$ must be of the form $\delta = \sigma'.Y \preceq \sigma.\widehat{X}$, hence $\delta \in \mathbf{RT}^{\ell}(Q)$ from the downward closure of refusal traces (Proposition 1).
- $\chi = \widehat{X}$ such that $\sigma.\widehat{X} \in \mathbf{RT}^{\ell}(P)$, and $\widehat{X} \notin \mathbf{FR}(P\|\sigma)$: then there must be an action $a \in \mathsf{top}_{P\|\sigma}(\widehat{X}) \setminus \widehat{X}$ such that $\sigma.\widehat{X}.a \notin \mathbf{RT}^{\ell}(P)$. Since $\sigma.\widehat{X} \in \mathbf{VT}^{\ell}_{P}(Q)$, we have $\widehat{X} = \mathbf{SR}(s)$ for some $s \in Q\|\sigma$, and therefore $\sigma.\widehat{X}.a \in \mathbf{RT}^{\ell}(Q)$. Furthermore, since $\delta \in \mathsf{detectors}_P(\sigma.\chi)$, $\delta$ must be of the form $\delta = \sigma'.Y.a \preceq \sigma.\widehat{X}.a$, hence $\delta \in \mathbf{RT}^{\ell}(Q)$ from the downward closure of refusal traces.
- $\chi = \bullet.a$ such that $\sigma.\bullet.a \notin \mathbf{RT}(P)$: from the definition of $\mathbf{VT}^{\ell}_{P}(Q)$ it follows that $\sigma.\bullet.a \in \mathbf{RT}^{\ell}(Q)$. In this case, the detector trace $\delta$ must be of the form $\delta = \sigma'.\bullet.a \preceq \sigma.\bullet.a$, hence $\delta \in \mathbf{RT}^{\ell}(Q)$. □

For a test suite, we define its *detector cover* as the family of detector sets of all conceivable violation traces (with length bound $\ell$):

$$\mathsf{DetCov}_P(\mathsf{TS}, \ell) \quad \triangleq \quad \{\mathsf{detectors}_P(\sigma.\widehat{X}, \mathsf{TS}) \mid \sigma.\widehat{X} \in \mathbf{VT}^{\ell}_{P}\}$$

Using a detector cover, we can state a straightforward sufficient condition for uniform completeness.

**Lemma 7.** *Any test suite TS for P such that $DetCov_P(TS, \ell)$ does not contain an empty set, is uniformly complete for P w.r.t. $\sqsubseteq^{\ell}_{\boldsymbol{RT}}$.*

**Proof.** Suppose $P \not\sqsubseteq^k_{\mathbf{RT}} Q$ for some $k \le \ell$. From Lemma 5 there is some violation trace $\sigma.\chi \in \mathbf{VT}^k_P(Q)$. If $\mathsf{DetCov}_P(\mathsf{TS}, \ell)$ does not contain an empty set, then in particular there is some $\sigma'.\omega \in \mathsf{TS} \cap \mathsf{detectors}_P(\sigma.\chi)$. From the definition of detectors, we have $|\sigma'.\omega| \le k$ and from Lemma 6 we obtain $\sigma'.\omega \in \mathsf{TS} \cap \mathbf{RT}^k(Q)$. □

**Lemma 8.** *For any process P and length bound $\ell$, $DetCov_P(\boldsymbol{TS}^{\ell}_{\boldsymbol{4}}(P), \ell)$ does not contain an empty set.*

**Proof.** First, observe that the test suite $\mathbf{TS}^{\ell}_{\mathbf{3}}$ (where larger traces w.r.t. $\preceq$ have not been removed) meets sufficient condition for uniform completeness provided by Lemma 7. Indeed, for any violation $\sigma.\chi \in \mathbf{VT}^{\ell}_P$:

- if $\sigma.\chi = \sigma.\widehat{X} \notin \mathbf{RT}^{\ell}(P)$ where $\widehat{X} \in \mathscr{P}(Act)$, then we have $\sigma.Y \in \mathbf{TS}^{\ell}_{\mathbf{3}}$, where $Y$ is a minimal refusal such that $Y \subseteq \widehat{X}$ and $\sigma.Y \notin \mathbf{RT}^{\ell}(P)$
- if $\sigma.\chi = \sigma.\bullet.a \notin \mathbf{RT}^{\ell}(P)$, then we simply have $\sigma.\bullet.a \in \mathbf{TS}^{\ell}_{\mathbf{3}}$
- if $\sigma.\chi = \sigma.\widehat{X} \in \mathbf{RT}^{\ell}(P)$ where $\widehat{X} \in \mathscr{P}(Act)$ and $\widehat{X} \notin \mathbf{FR}(P\|\sigma)$, then we have

  $$\{\sigma.X_m.a \mid X_m \in \mathsf{min\text{-}base}_{P\|\sigma}(\widehat{X}) \wedge a \in \mathsf{top}_{P\|\sigma}(\widehat{X}) \setminus X_m\} \subseteq \mathbf{TS}^{\ell}_{\mathbf{3}}$$

  so from the definition of min-base in particular there is $X_m$ such that $X_m \subseteq \widehat{X}$ and moreover, since $\widehat{X} \neq \mathsf{top}_{P\|\sigma}(\widehat{X})$, for some $a \in \mathsf{top}_{P\|\sigma}(\widehat{X}) \setminus \widehat{X}$ we have $\sigma.X_m.a \in \mathbf{TS}^{\ell}_{\mathbf{3}}$

From their definition, detector sets of any violation are downward closed w.r.t. $\preceq$ for forbidden traces. Hence removing non-minimal traces (w.r.t. $\preceq$) from a test suite does not affect the nonemptiness of detector sets for any violation. Hence $\mathbf{TS}^{\ell}_{\mathbf{4}}(P)$ contains at least one detector for every violation in $\mathbf{VT}^{\ell}_P$. □

**Corollary 3.** *$\boldsymbol{TS}^{[\ell]}_{\boldsymbol{4}}(P)$ is uniformly complete for P w.r.t. $\sqsubseteq^{\ell}_{\boldsymbol{RT}}$.*

Lemmata 7 and 8 suggest an idea for obtaining minimal uniformly complete test suites using the so-called minimum hitting sets. For a family of sets $\mathcal{A}$, a *hitting set for* $\mathcal{A}$ is any set $H \subseteq \bigcup \mathcal{A}$ such that $\forall A \in \mathcal{A}.\, H \cap A \neq \emptyset$. The family $\mathsf{minHit}\,\mathcal{A}$ consists of all hitting sets of $\mathcal{A}$ with the smallest size.

We define the following family of test suites:

$$\boldsymbol{TS}^{\boldsymbol{uni}}_{\boldsymbol{min}}(P, \ell) \quad \triangleq \quad \mathsf{minHit}\ \mathsf{DetCov}_P(\mathbf{TS}^{\ell}_{\mathbf{4}}, \ell)$$

The key question is whether the above suites are really minimal: while detector sets suffice to prove a given violation, we have not shown yet if they exhaust all means of detecting the violation. In Section 7.2, we will show that $\boldsymbol{TS}^{\boldsymbol{uni}}_{\boldsymbol{min}}$ indeed contains minimal uniformly complete test suites.

### 7.1.3. *Beyond uniform completeness*

We have argued that redundancies due to longer traces are not actual redundancies: their omission would yield singularly behaved test suites, undesirable from a practical perspective. Still, determining the minimal suites that are complete without the uniformity requirement remains an interesting theoretical challenge. In this section, we consider this case of general (non-uniform) completeness.

The key insight is that when considering violations after a trace $\rho$, we only need to take care of those that are *realisably hidden*, that is, for which extensions consistent with further behaviour of $P$ exist. For such state refusals, it is possible to endow the original system with a fresh state exhibiting a disallowed behaviour $\widehat{X}$ in a way that does not violate any longer obligations. Such violations will always be "invisible" when inspecting longer traces.

The above considerations lead us to the formal definition of *realisable hidden violations* [with length bound $\ell$] of a process:

$$\begin{aligned} \mathsf{rhv}^{\ell}(P) \quad \triangleq \quad & \big\{\sigma.\widehat{X} \in \mathbf{VT}^{\ell}_P \mid \forall b \in Act \setminus \widehat{X}.\ \exists Q_b.\ \forall Y \subsetneq \widehat{X}. \\ & \quad P\|\sigma.Y.b \neq \emptyset \implies P\|\sigma.Y.b \sqsubseteq^{\ell-|\sigma|-1}_{\mathbf{RT}} Q_b \big\} \\ & \cup \{\sigma.\bullet.a \in \mathbf{VT}^{\ell}_P\} \end{aligned}$$

(note that $P \sqsubseteq^0_{\mathbf{RT}} Q$ is true for all processes $P, Q$).

The statement in line 1 implies that one can construct a process which has a state after $\sigma$ with state refusal $\widehat{X}$, but whose further behaviours are compatible with $P$ for every relevant continuation, hence the violation $\sigma.\widehat{X}$ is undetectable through longer traces (longer inadmissible traces that are minimal i.e. we do not take into account extensions of $\sigma.\widehat{X}$). Note also that strictly forbidden actions always give rise to realisable hidden violations (line 2).

**Example 3.** Consider the process $P$ in Fig. 4. We have:

- $\{b,c\}.a.\{a,b\} \in \mathsf{rhv}^\ell(P)$ (for any $\ell$): the 2nd line of definition of $\mathsf{rhv}^\ell$ requires one to check only one action $c$, hence we just need to show existence of a process that would refine both $P\|\{b,c\}.a.\{a\}.c$ and $P\|\{b,c\}.a.\{b\}.c$. This is straightforward since $P\|\{b,c\}.a.\{a\}.c$ yields an empty set, refined by any process, and therefore we can use $Q_c = P\|\{b,c\}.a.\{b\}.c$.
- $\{b,c\}.a.\{b,c\} \notin \mathsf{rhv}^\ell(P)$ for $\ell > 2$: as explained in Section 6.2, there is no process $Q_a$ that would refine both $P\|\{b,c\}.a.\{c\}.a$ and $P\|\{b,c\}.a.\{b\}.a$.

Intuitively speaking, every violation trace which is *not* a realisable hidden violation, entails the presence of a longer violation trace, and hence in every finitary variant of refusal trace refinement, a non-conforming implementation must exhibit some realisable hidden violation.

**Lemma 9.** *For every process $P$ and every incorrect implementation $Q$ such that $P \not\sqsubseteq^\ell_{\mathbf{RT}} Q$, $Q$ contains a realisable hidden violation, i.e. there exists some $\rho \in \mathbf{VT}^\ell_P(Q) \cap \mathsf{rhv}^\ell(P)$.*

**Proof.** Take any arbitrary $\rho \in \mathbf{VT}^\ell_P(Q)$ with maximal length (we can do this as $\mathbf{VT}^\ell_P(Q)$ is finite). Suppose, towards contradiction, that $\rho \notin \mathsf{rhv}^\ell(P)$. Then $\rho$ is of the form $\rho = \sigma.\widehat{X}$ for $\widehat{X} \in \mathscr{P}(Act)$, and there must be some $b \in Act \setminus \widehat{X}$ such that for all processes $Q_b$ there is some refusal $Y \subsetneq \widehat{X}$ such that $P\|\sigma.Y.b \neq \emptyset$ and $P\|\sigma.Y.b \not\sqsubseteq^{\ell-|\sigma|-1}_{\mathbf{RT}} Q_b$. Moreover, we know that there is a fundamental refusal with such a property: indeed, if we take $Y_{top} = \mathsf{top}_{P\|\sigma}(Y)$, we have $P\|\sigma.Y.b = P\|\sigma.Y_{top}.b$, from which $P\|\sigma.Y_{top}.b \not\sqsubseteq^{\ell-|\sigma|-1}_{\mathbf{RT}} Q_b$ follows.

Since $P\|\sigma.Y_{top}.b \not\sqsubseteq^{\ell-|\sigma|-1}_{\mathbf{RT}} Q_b$, from Lemma 5 we know that there must be some $\sigma'.\widehat{Z} \in \mathbf{VT}^{\ell-|\sigma|-1}_{P_b}(Q_b)$ where $P_b = P\|\sigma.Y_{top}.b$.

The combined violation trace $\sigma.Y_{top}.b.\sigma'.\widehat{Z}$ belongs to $\mathbf{VT}^\ell_P(Q)$ and is a longer violation than $\sigma.\widehat{X}$, which we assumed had maximal length – a contradiction. □

In our new test suite, among all local forbidden behaviours after $\sigma$, only realisable hidden violations need to be targeted directly by the test suite. The detection of the remaining violations will be ensured by longer traces in the suite.

Formally, we define the detector cover of a test suite restricted to realisable hidden violations as:

$$\mathsf{DetCov}^{\mathsf{rhv}}_P(\mathsf{TS},\ell) \quad \triangleq \quad \{\mathsf{detectors}_P(\sigma.\chi,\mathsf{TS}) \mid \sigma.\chi \in \mathsf{rhv}^\ell(P)\}$$

We can now define a family of test suites $\boldsymbol{TS_{min}}$ in a similar manner as $\boldsymbol{TS^{uni}_{min}}$.

$$\boldsymbol{TS_{min}}(P,\ell) \quad \triangleq \quad \mathsf{minHit}\ \mathsf{DetCov}^{\mathsf{rhv}}_P(\mathbf{TS}^\ell_\mathbf{4},\ell)$$

As a corollary of Lemma 9 we obtain completeness of $\boldsymbol{TS_{min}}$.

**Proposition 7.** *For each $P$, every $\boldsymbol{TS} \in \boldsymbol{TS_{min}}(P,\ell)$ is complete for $\boldsymbol{RT}^\ell$.*

**Proof.** Take any process $P$, test suite $\mathbf{TS} \in \boldsymbol{TS_{min}}(P,\ell)$, and implementation $Q$ such that $P \not\sqsubseteq^\ell_{\mathbf{RT}} Q$. From Lemma 9 we know that there is some $\rho \in \mathbf{VT}^\ell_P(Q) \cap \mathsf{rhv}^\ell(P)$. Since from the definition of $\boldsymbol{TS_{min}}(P,\ell)$ there is some $\delta \in \mathsf{detectors}_P(\rho) \cap \mathbf{TS}$, and from Lemma 6 we obtain $\delta \in \mathbf{RT}^\ell(Q) \cap \mathbf{TS}$. □

### 7.2. Minimality results

In this section, we prove that the families of test suites defined in the previous section are uniformly complete (in case of $\boldsymbol{TS^{uni}_{min}}$) and complete (in case of $\boldsymbol{TS_{min}}$) for bounded refusal trace refinement. The proofs have two stages: first, we show that the presence of detectors is a necessary condition for (uniform) completeness, hence our test suites are minimal within $\mathbf{TS_4}$. In the second stage, we show that an arbitrary test suite can be converted into a test suite contained in $\mathbf{TS_4}$ of equal cardinality, in a way that preserves (uniform) completeness.

#### 7.2.1. Presence of detectors essential for (uniform) completeness

In the first stage of the proof, we show that: each uniformly complete test suite must contain detector of each violation, and each complete test suite must contain detector of each realisable hidden violation.

As an intermediate step, we shall prove an important property of fundamental traces: each of them may occur as a state refusal in some context. More precisely, for any valid fundamental trace $\sigma \in \mathbf{FR}(P)$, we can construct a process $P \searrow \sigma$ that is refusal trace equivalent to the original process $P$, but in which the trace $\sigma$ is a *state refusal trace*. These are decorated traces labelled with state refusals of all visited stable states, and nullary refusals for each transient state from which a visible action originated. Formally, the set of state refusal traces of a state $s$, notation $\mathbf{SRT}(s)$, is defined inductively as follows:

$$
\begin{array}{rcll}
\mathbf{SRT}(s) & \triangleq & \{\epsilon\} & \\
& & \cup\{\mathbf{SR}(s) & \mid \mathbf{R}(s) \neq \emptyset\} \\
& & \cup\{\bullet & \mid \mathbf{R}(s) = \emptyset\} \\
& & \cup\{\mathbf{SR}(s).a.\rho & \mid \mathbf{R}(s) \neq \emptyset \wedge s \xrightarrow{a} q \wedge \rho \in \mathbf{SRT}(q)\} \\
& & \cup\{\bullet.a.\rho & \mid \mathbf{R}(s) = \emptyset \wedge s \xrightarrow{a} q \wedge \rho \in \mathbf{SRT}(q)\} \\
& & \cup\{\rho & \mid \mathbf{R}(s) = \emptyset \wedge s \xrightarrow{\tau} q \wedge \rho \in \mathbf{SRT}(q)\}
\end{array}
$$

**Proposition 8.** *Fix a process $P$. A trace $\sigma \in \boldsymbol{RT}$ is a fundamental refusal trace of $P$ ($\sigma \in \boldsymbol{FRT}(P)$) if and only if one can construct a process, denoted with $P \searrow \sigma$, such that*

1. *$P \searrow \sigma =_{\boldsymbol{RT}} P$, and*
2. *$\sigma \in \boldsymbol{SRT}(P \searrow \sigma)$.*

**Proof.** "$\Longleftarrow$": Observe that since $\sigma$ is a state refusal trace of $P \searrow \sigma$, it follows immediately that $\sigma \in \mathbf{FRT}(P \searrow \sigma)$, hence from $P \searrow \sigma =_{\mathbf{RT}} P$ and preservation of fundamental refusal traces (Corollary 1) we obtain $\sigma \in \mathbf{FRT}(P)$.

"$\Longrightarrow$": If $\sigma = \epsilon$, then we simply take $P \searrow \sigma = P$. Otherwise, let $\sigma = X_0 a_1 X_1 \ldots a_n X_n [a_{n+1}]$. We endow the LTS underlying $P$ with $n+1$ fresh states $s_0, s_1, \ldots, s_n$. For each $i \in \{0, \ldots, n-1, [n]\}$ such that $X_i \in \mathscr{P}(Act)$, we define, for each $a \in Act \setminus X_i$, a transition $s_i \xrightarrow{a} q$ iff $\exists s_Y \xrightarrow{a} q$, where $s_Y \in P \| X_0 a_1 \ldots a_i$ such that $X_i \subseteq \mathbf{SR}(s_Y)$. This ensures the state refusal of each such $s_i$ is equal to $X_i$. For each $i \in \{0, \ldots, n-1, [n]\}$ such that $X_i = \bullet$, we choose an arbitrary state $s \in P \| X_0 a_1 \ldots a_i$, and define a transition $s_i \xrightarrow{\tau} s$.

Observe that our construction ensures that for each state $s_i$ in the sequence, $s_i$ is unstable precisely when $X_i = \bullet$, and if $X_i \in \mathscr{P}(Act)$, then $\mathbf{SR}(s_i) = X_i$. Moreover, for each $i$ we have $\mathbf{RT}(s_i) = \mathbf{RT}(P \| X_0 \ldots a_{i-1} X_i)$.

What remains to be added is a chain of transitions between the states i.e. $s_i \xrightarrow{a_{i+1}} s_{i+1}$ for $i \in \{0, \ldots, n-1\}$. Finally, we set up the initial states as: $P \searrow \sigma = P \cup \{s_0\}$. □

As a side remark, we note that Proposition 8 offers an insight into the relationship between refusal trace semantics and the finer ready trace semantics induced by state refusals (state refusals are duals of ready sets). For instance, one may use it to describe different processes in ready trace semantics that give rise to the same process modulo $=_{\mathbf{RT}}$.

By $\mathbf{FRT}_{\mathsf{A}}(P)$, we shall denote the set of fundamental traces of $P$ that do not terminate with refusal, i.e.

$$\mathbf{FRT}_{\mathsf{A}}(P) = \{\epsilon\} \cup \{\pi.a \in \mathbf{FRT}(P)\}$$

The key corollary of Proposition 8 is that for any fundamental trace $\sigma \in \mathbf{FRT}_{\mathsf{A}}(P)$ and some partial process $Q$ (which in our context will contain some behaviour disallowed after $\sigma$), we can construct a process $P \searrow [\sigma, Q]$ that differs from $P$ only on traces prefixed with $\sigma$, or a trace smaller than $\sigma$ w.r.t. $\preceq$.

**Corollary 4.** *For any process $P$, its fundamental trace $\sigma \in \boldsymbol{FRT}_{\mathsf{A}}(P)$ and a process $Q$, there exists a process $P \searrow [\sigma, Q]$ such that*

$$\boldsymbol{RT}(P \searrow [\sigma, Q]) \setminus \boldsymbol{RT}(P) \;\subseteq\; \{\sigma'.\lambda \mid \sigma' \preceq \sigma \;\wedge\; |\sigma| = |\sigma'| \;\wedge\; \lambda \in \boldsymbol{RT}(Q)\}$$

**Proof.** If $\sigma = \epsilon$, then we simply take as the initial states of our new process the union of initial states of $P$ and $Q$.

If $\sigma \neq \epsilon$, then $\sigma$ is of the form $\sigma = \sigma'.a$. We start with $P \searrow \sigma$, constructed precisely as in the proof of Proposition 8. We add a fresh state $s_\sigma$ and set up the following transitions: $s_n \xrightarrow{a} s_\sigma$ and $\{s_\sigma \xrightarrow{\tau} s_Q \mid s_Q \in Q\}$. The initial states of $P \searrow [\sigma, Q]$ are the same as those of $P \searrow \sigma$.

Observe that if the newly constructed process has a trace $\rho$ forbidden in $P$, then the path that gives rise to $\rho$ must pass through an initial state of $Q$. As initial states of $Q$ can be reached only after passing through a path that gives rise to the trace $\sigma$ and smaller traces w.r.t. $\preceq$ (traces with larger refusals than those in $\sigma$ are not allowed as $\sigma$ is a state refusal trace), $\rho$ must have a prefix $\sigma'$ where $\sigma' \preceq \sigma$, hence the statement holds. □

We can now use Corollary 4 to show that detectors are essential for completeness. The general idea is outlined in Fig. 6: for a given violation $\sigma.\chi$ we need to find the suitable process $Q$ which exhibits the violation $\chi$, and then construct a process $P \searrow [\sigma, Q]$ that exhibits the violation $\sigma.\chi$ but differs from $P$ only on detectors of $\sigma.\chi$.

We first present the proof for uniform completeness, where the construction of $Q$ is more straightforward. In this case, we show that detectors must be present for any violation trace $\sigma.\chi$. Since completeness must hold for any $\sqsubseteq^k_{\mathbf{RT}}$ where $k \leq \ell$, in particular for $k = |\sigma.\chi|$, we just need to ensure that the initial state of $Q$ exhibits $\chi$; the behaviour after the first step is irrelevant.

In the proof of the following lemmata, when reasoning about forbidden traces exhibited by some implementation $Q$, we consider, rather than all forbidden traces i.e. $\mathbf{RT}(Q) \setminus \mathbf{RT}(P)$, the restricted set $\mathbf{RT}(Q) \cap \mathbf{TS_0}(P)$ of traces with admissible prefixes. This is justified as we only consider test suites contained in $\mathbf{TS_0}$ and allows us to make clearer, more

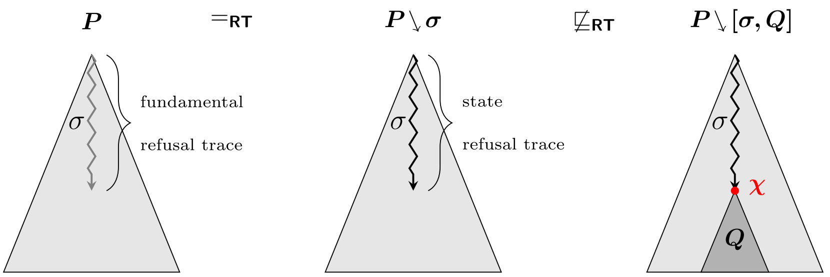


**Fig. 6.** Construction of an erroneous process that exhibits a violation $\sigma.\chi$ in a way which ensures that some detector $\delta \in \mathsf{detectors}_P(\sigma.\chi)$ must be present in any (uniformly) complete test suite. We use different sub-processes $Q$ depending on whether uniform completeness or general completeness is required; in each case $Q$ exhibits the violation $\chi$. The required process $P \searrow [\sigma, Q]$ is obtained by first constructing the equivalent process $P \searrow \sigma$ (Proposition 8) and then extending the trace $\sigma$ with $Q$ (Corollary 4).

succinct statements where the cases of redundant suffixes can be ignored. In particular, we use the observation that $\mathsf{detectors}_P(\sigma.\chi) \subseteq \mathbf{TS_0}(P)$

**Lemma 10.** *For every process P, length bound $\ell \in \mathbb{N}$, and a test suite $TS \subseteq \boldsymbol{TS}_{\boldsymbol{0}}^{\ell}$ uniformly complete for P w.r.t. $\sqsubseteq_{\boldsymbol{RT}}^{\ell}$, for every violation $\rho \in \boldsymbol{VT}^{\ell}(P)$, we have $\mathit{detectors}(\rho) \cap TS \neq \emptyset$.*

**Proof.** Let $\rho = \sigma.\chi \in \mathbf{VT}^{\ell}(P)$.

Our aim is to construct a process $P_{\sigma.\chi}$ that exhibits the violation $\rho = \sigma.\chi$ while its only forbidden traces (without redundant suffixes) up to $k = |\sigma.\chi|$ are the detectors of $\sigma.\chi$, i.e.

$$\mathbf{RT}^k(P_{\sigma.\chi}) \cap \mathbf{TS}_0^k(P) \subseteq \mathsf{detectors}_P(\sigma.\chi)$$

Since a uniformly complete test suite $\mathsf{TS} \subseteq \mathbf{TS}_{\mathbf{0}}^{\ell}(P)$ must be complete in particular for $k$, it shall then follow that TS must contain some detector trace $\delta \in \mathsf{detectors}_P(\sigma.\chi)$.

We first define a simple process $Q$ that exhibits a violation $\chi$ in the following way: $Q$ contains a state $s_Q$, endowed with the following transitions:

- if $\chi = \widehat{X} \in \mathscr{P}(Act)$, then for each $b \in (Act \setminus \widehat{X})$ we add a transition $s_Q \xrightarrow{b} s_Q$, so that $\mathbf{SR}(s_Q) = \widehat{X}$
- if $\chi = \bullet.a$, then we add a transition $s_Q \xrightarrow{a} s_Q$. Furthermore, for some $s \in P \| \sigma$ we add all states reachable from $s$ to the LTS underlying $Q$ and endow it with a transition $s_Q \xrightarrow{\tau} s$.

Observe that in both cases, the construction ensures that $Q$ does not exhibit any violation trace w.r.t. $P\|\sigma$ of length other than $\chi$, i.e. $\mathbf{VT}^1_{P\|\sigma}(Q) = \{\chi\}$. Hence we have

$$(*) \quad \mathbf{RT}^1(Q) \cap \mathbf{TS}_0^1(P\|\sigma) = \mathsf{detectors}_{P\|\sigma}(\chi)$$

Using Corollary 4, we can now construct the process $P \searrow [\sigma, Q]$ such that in particular for $k = |\sigma.\chi|$ we have

$$\mathbf{RT}^k(P \searrow [\sigma, Q]) \cap \mathbf{TS}_0^k(P) \subseteq \{\sigma'.\omega \mid \sigma' \preceq \sigma \wedge |\sigma| = |\sigma'| \wedge \omega \in \mathbf{RT}^1(Q)\}$$

We can further restrict the above set, as $\omega$ must range over traces forbidden in $P\|\sigma$:

$$\mathbf{RT}^k(P \searrow [\sigma, Q]) \cap \mathbf{TS}_0^k(P) \subseteq \{\sigma'.\omega \mid \sigma' \preceq \sigma \wedge |\sigma| = |\sigma'| \wedge \omega \in \mathbf{RT}^1(Q) \cap \mathbf{TS}_{\mathbf{0}}^1(P\|\sigma)\}$$

From (*) we have:

$$\mathbf{RT}^k(P \searrow [\sigma, Q]) \cap \mathbf{TS}_0^k(P) \subseteq \{\sigma'.\omega \mid \sigma' \preceq \sigma \wedge |\sigma| = |\sigma'| \wedge \omega \in \mathsf{detectors}_{P\|\sigma}(\chi)\}$$

From the definition of $\mathsf{detectors}_P$, we have

$$\mathsf{detectors}_P(\sigma.\chi) = \{\sigma'.\omega \in \mathbf{TS_0}(P) \mid \sigma' \preceq \sigma \wedge |\sigma| = |\sigma'| \wedge \omega \in \mathsf{detectors}_{P\|\sigma}(\chi)\}\big)$$

hence we obtain

$$\mathbf{RT}^k(P \searrow [\sigma, Q]) \cap \mathbf{TS}_0^k(P) \subseteq \mathsf{detectors}_P(\sigma.\chi) \quad \square$$

We now show a similar result for general completeness, where detectors are required only for realisable hidden violations.

**Lemma 11.** *For every process $P$, length bound $\ell \in \mathbb{N}$, and a test suite $TS$ complete for $P$ w.r.t. $\sqsubseteq^{\ell}_{\mathbf{RT}}$, for every realisable hidden violation $\rho \in \mathsf{rhv}^{\ell}(P)$, we have $\mathsf{detectors}^{\ell}(\rho) \cap TS \neq \emptyset$.*

**Proof.** Consider an arbitrary $\chi \in \mathsf{rhv}^{\ell}(P)$. Similarly to the proof of Lemma 10, we construct a process $P^{\ell}_{\sigma.\chi}$ whose forbidden traces – here, we consider all traces up to $\ell$ – are contained in $\mathsf{detectors}_P(\sigma.\chi)$:

$$\mathbf{RT}^{\ell}(P^{\ell}_{\sigma.\chi}) \cap \mathbf{TS}^{\ell}_0(P) \subseteq \mathsf{detectors}_P(\sigma.\chi)$$

Let $\rho = \sigma.\chi \in \mathsf{rhv}^{\ell}(P)$. We first construct a process $Q$ that exhibits a violation $\chi$ and does not contain any violation traces (w.r.t. the process $P\|\sigma$) longer than $\chi$.

$Q$ contains one initial state $s_Q$. We have the following cases:

- if $\chi = \widehat{X} \in \mathcal{P}(Act)$, then, since $\sigma.\chi \in \mathsf{rhv}^{\ell}(P)$, from the definition of realisable hidden violation we know that for every $b \in (Act \setminus \widehat{X})$ there is a process $Q_b$ such that
$$\forall Y \subsetneq \widehat{X}.\ P\|\sigma.Y.b \neq \emptyset \implies P\|\sigma.Y.b \sqsubseteq^{\ell-|\sigma|-1}_{\mathbf{RT}} Q_b$$
For each $b \in (Act \setminus \widehat{X})$, we endow our LTS with instances of such processes $Q_b$ and add transitions $\{s_Q \xrightarrow{b} s^I_b \mid s^I_b \in Q_b\}$.
- if $\chi = \bullet.a$, then construction is the same as in the analogous case of Lemma 10: we add a transition $s_Q \xrightarrow{a} s_Q$, choose any $s \in P\|\sigma$ and add a transition $s_Q \xrightarrow{\tau} s$

Similarly as in the proof of Lemma 10, the crucial property of the above construction is that $Q$ does not exhibit any violation trace – among traces of length $(\ell - |\sigma|)$ – other than $\chi$, i.e.

$$(\dagger) \quad \mathbf{VT}^{\ell-|\sigma|}_{P\|\sigma}(Q) = \{\chi\}$$

To prove (†), observe that a violation trace results from a path which originates in the sole initial state $s_Q$. We then have two cases:

In case $\chi = \widehat{X}$, $s_Q$ is stable and hence the only violation of length one is $\chi$. Any longer violation would entail a violation in some process $Q_b$ i.e. $\rho' \in \mathbf{VT}_{P\|\sigma.Y.b}(Q_b)$ for some $Y \in \mathbf{FR}(P\|\sigma)$, which would contradict $P\|\sigma.Y.b \sqsubseteq^{\ell-|\sigma|-1}_{\mathbf{RT}} Q_b$.

In case $\chi = \bullet.a$, any path from $s_Q$ must either start with a silent transition, in which case it is a path of the process $P\|\sigma$ and cannot give rise to a violation, or a transition $s_Q \xrightarrow{a} s_Q$ which can only give rise to one violation $\bullet.a$ (violations must have admissible prefixes).

We have thus shown that (†) holds and therefore we have:

$$(*) \quad \mathbf{RT}^{\ell-|\sigma|}(Q) \cap \mathbf{TS}^{\ell-|\sigma|}_0(P\|\sigma) = \mathsf{detectors}_{P\|\sigma}(\chi)$$

As in Lemma 10, we can now construct a process $P \searrow [\sigma, Q]$ such that

$$\mathbf{RT}^{\ell}(P \searrow [\sigma, Q]) \cap \mathbf{TS}^{\ell}_0(P) \subseteq \{\sigma'.\omega \mid \sigma' \preceq \sigma \ \wedge\ |\sigma| = |\sigma'| \ \wedge\ \omega \in \mathbf{RT}^{\ell-|\sigma|}(Q)\}$$

Since $\omega$ must range over traces forbidden in $P\|\sigma$, we have:

$$\begin{aligned}\mathbf{RT}^{\ell}(P \searrow [\sigma, Q]) \cap \mathbf{TS}^{\ell}_0(P) \subseteq \{\sigma'.\omega \mid \sigma' \preceq \sigma \ \wedge\ |\sigma| = |\sigma'| \\ \wedge\, \omega \in \mathbf{RT}^{\ell-|\sigma|}(Q) \cap \mathbf{TS}^{\ell-|\sigma|}_{\mathbf{0}}(P\|\sigma)\}\end{aligned}$$

From (*) we have:

$$\mathbf{RT}^{\ell}(P \searrow [\sigma, Q]) \cap \mathbf{TS}^{\ell}_0(P) \subseteq \{\sigma'.\omega \mid \sigma' \preceq \sigma \ \wedge\ |\sigma| = |\sigma'| \ \wedge\ \omega \in \mathsf{detectors}_{P\|\sigma}(\chi)\}$$

From the definition of $\mathsf{detectors}_P$, we have

$$\mathsf{detectors}_P(\sigma.\chi) = \{\sigma'.\omega \in \mathbf{TS_0}(P) \mid \sigma' \preceq \sigma \ \wedge\ |\sigma| = |\sigma'| \ \wedge\ \omega \in \mathsf{detectors}_{P\|\sigma}(\chi)\})$$

hence we obtain

$$\mathbf{RT}^{\ell}(P \searrow [\sigma, Q]) \cap \mathbf{TS}^{\ell}_0(P) \subseteq \mathsf{detectors}_P(\sigma.\chi) \quad \square$$

We have proved that detectors are essential for completeness, which in turn implies that $\boldsymbol{TS^{uni}_{min}}$ and $\boldsymbol{TS_{min}}$ contain, respectively, minimal uniformly complete and complete test suites among test suites contained in $\mathbf{TS_4}$.

### 7.2.2. $\mathbf{TS_4}$ contains minimal test suites

In the last part of our minimality proof, we show that there is always a minimal (uniformly) complete test suite contained in $\mathbf{TS_4}(P)$. To this end, we define the following mapping which, for any test suite contained in $\mathbf{TS_0}$, produces a test suite contained in $\mathbf{TS_4}$, and that we later show to preserve both uniform and general completeness.

The mapping $f_{\mathsf{top}} : \mathbf{TS_0} \rightarrow \mathbf{TS_4}$ is defined as follows[5]:

- $f_{\mathsf{top}}(\sigma.X) \triangleq \mathsf{top}_P(\sigma).X_{min}$
- $f_{\mathsf{top}}(\sigma.X.a) \triangleq \mathsf{top}_P(\sigma).X_m.a$
  where $X \in \mathcal{P}(Act)$ and $X_m \in \text{min-base}_{P\|\sigma}(X)$ and $X_m \subseteq X$
- $f_{\mathsf{top}}(\sigma.\bullet.a) \triangleq \mathsf{top}_P(\sigma).\bullet.a$

Observe that for every $\rho \in \mathbf{TS_0}$, $f_{\mathsf{top}}(\rho) \notin \mathbf{RT}^k(P)$, where $k$ is the length of $\rho$, hence an image of a sound test suite $\mathsf{TS} \subseteq \mathbf{TS_0}$ under $f$ is itself a sound test suite. Moreover, as an image under a function, $f_{\mathsf{top}}(\mathsf{TS})$ has cardinality not exceeding that of $\mathsf{TS}$.

We will now show that the image of a detector under the mapping $f_{\mathsf{top}}$ is also a detector.

**Lemma 12.** *For every violation $\rho \in \mathbf{VT}^{\ell}_P$, we have:*

$$\delta \in \mathit{detectors}_P(\rho) \implies f_{top}(\delta) \in \mathit{detectors}_P(\rho)$$

**Proof.** Observe that $f_{\mathsf{top}}$ is monotonic w.r.t. the prefix & pointwise inclusion preoder $\preceq$, that is:

$$(*) \quad \rho_1 \preceq \rho_2 \implies f_{\mathsf{top}}(\rho_1) \preceq f_{\mathsf{top}}(\rho_2)$$

From the definition of $f_{\mathsf{top}}$ and maximal detectors, for any maximal detector $\sigma.\omega \in \mathsf{maxdetectors}_P(\sigma.\chi)$, $f_{\mathsf{top}}(\sigma.\omega)$ is of the form $f_{\mathsf{top}}(\sigma.\omega) = \sigma.\omega'$, where $\omega' \preceq \omega$, hence $f_{\mathsf{top}}(\sigma.\omega) \preceq \sigma.\omega$, which, combined with (*), yields $f_{\mathsf{top}}(\sigma.\omega) \in \mathsf{detectors}_P(\sigma.\chi)$. That $f_{\mathsf{top}}(\sigma'.\omega') \in \mathsf{detectors}_P(\sigma.\chi)$ holds for an arbitrary detector follows immediately from the monotonicity of $f_{\mathsf{top}}$ and (*). □

**Lemma 13.** *The mapping $f_{top}$ preserves completeness and uniform completeness for finite test suites. That is, for every process $P$, and length bound $\ell$, if test suite $TS$ is [uniformly] complete for $P$ w.r.t. $\sqsubseteq^{\ell}_{\mathbf{RT}}$, then so is $f_{top}(TS)$.*

**Proof.** Take any specification $P$, length bound $\ell$, a complete test suite $\mathsf{TS}$, and an incorrect implementation $Q$ such that $P \not\sqsubseteq^{\ell}_{\mathbf{RT}} Q$. We will to show that there is some $\rho \in f_{\mathsf{top}}(\mathsf{TS})$ such that $\rho \in \mathbf{RT}^{\ell}(Q)$.

From Lemma 9, we know that there is some realisable hidden violation $\sigma.\widehat{X} \in \mathbf{VT}^{\ell}_P(Q) \cap \mathsf{rhv}^{\ell}(P)$. From Lemma 11, there must be some detector $\delta \in \mathsf{TS} \cap \mathsf{detectors}_P(\sigma.\widehat{X})$, and from Lemma 12 we obtain $f_{\mathsf{top}}(\delta) \in \mathsf{detectors}_P(\sigma.\widehat{X})$. From Lemma 6, every implementation that exhibits a given violation must exhibit all its detectors, hence $f_{\mathsf{top}}(\delta) \in \mathbf{RT}^{\ell}(Q)$.

Note that, since we have shown preservation of completeness for an arbitrary length bound $\ell \in \mathbb{N}$, it also follows that $f_{\mathsf{top}}$ preserves uniform completeness. □

We are finally in a position to state our main minimality result. From Lemma 10, we know that $\boldsymbol{TS^{uni}_{min}}(P,\ell)$ contains minimal uniformly complete test suites for $\sqsubseteq^{\ell}_{\mathbf{RT}}$ among test suites contained in $\mathbf{TS}^{\ell}_{\mathbf{4}}(P)$. On the other hand from Lemma 13 we know that at least one subset of $\mathbf{TS}^{\ell}_{\mathbf{4}}(P)$, as an image of a minimal test suite, must be a minimal uniformly complete test suite for $\sqsubseteq^{\ell}_{\mathbf{RT}}$. Hence we obtain the minimality theorem for the test suite family $\boldsymbol{TS^{uni}_{min}}(P,\ell)$.

**Theorem 1.** *For any process $P$, $\boldsymbol{TS^{uni}_{min}}(P,\ell)$ contains minimal test suites which are uniformly complete for $P$ w.r.t. refusal trace semantics $\sqsubseteq^{\ell}_{\boldsymbol{RT}}$.*

Using a similar reasoning, we can arrive at an analogous result for $\boldsymbol{TS_{min}}(P,\ell)$.

**Theorem 2.** *For any process $P$, $\boldsymbol{TS_{min}}(P,\ell)$ contains minimal test suites which are complete for $P$ w.r.t. refusal trace semantics $\sqsubseteq^{\ell}_{\boldsymbol{RT}}$.*

We remark that test suites in $\boldsymbol{TS_{min}}(P,\ell)$ are likely of purely theoretical significance. Apart from the clarity of feedback in debugging, also computational complexity of their generation appears to be immense, and we refrain from pursuing more exact estimates in this work.

[5] A fully formal definition of $f$ would require an unambiguous selection of refusals $X^f_{min}/X_m$ e.g. based on some ordering of actions.

## 8. Characteristic formulae for refusal trace refinement

In this section, we address a problem from the realm of modal logic that is intimately related to complete test suites: devising a formula through which refinement checking can be reduced to model checking. Formally, given some logical language $\mathcal{L}$, and a process $P$, a formula $\Phi_P \in \mathcal{L}$ is a *characteristic formula* for $P$ if for all processes $Q$:

$$P \sqsubseteq_{\mathbf{RT}} Q \iff Q \models \Phi_P$$

A similar idea as in our test case generation method can be used to construct characteristic formulae for refusal trace semantics. To this end, we use a recursive variant of Hennessy-Milner logic, a simple propositional modal logic on labelled transition systems. Recursion allows one to specify obligations for processes of an arbitrary length, while the universal syntax below facilitates expressing forbidden behaviours of systems (akin to test suites).

The recursive formulae $\Phi$ and modal formulae $\psi$ are defined as follows:

$$\begin{array}{lll} \Phi & ::= & (\nu\mathcal{Z} = \psi) \mid \epsilon \\ \psi & ::= & \mathsf{F} \mid \widetilde{X} \mid \psi \wedge \psi \mid [a]\,\psi \mid [\epsilon]\,\psi \mid [\widetilde{X}]\,\psi \mid \mathcal{Z} \end{array}$$

The language $\mathsf{HML}^{\mathbf{RT}}_{\nu+\Box}$ consists of all recursive formulae.

The syntax contains some standard constructs from propositional logic ($\mathsf{F}$, $\wedge$), as well as the standard modal box operator $[a]\,\phi$, signifying that a formula holds for all $a$-successors of a state. For brevity, and to keep the syntax closer to the observations that define refusal trace semantics, we use the construct $\widetilde{X}$, signifying the presence of refusal $X$, as well as $[\widetilde{X}]\,\psi$, equivalent to $\widetilde{X} \Longrightarrow \psi$ (note that monotonicity of the formulae is maintained). Recursion is expressed by the presence of recursive variables, ranged over with $\mathcal{Z}$, and recursive equations $\nu\mathcal{Z} = \psi_{\mathcal{Z}}$.

Since our logic contains only greatest fixpoints (no alternation), and moreover is interpreted over finite models, its semantics can be defined in a straightforward manner. Namely, we can annotate the satisfaction relation with an environment $\mathcal{E} \subseteq \mathcal{V}ar \times S$ to store the pairs of (fixpoint variable, state) already visited along the proof path – when a dependency is encountered again, the subformula is assumed to hold (greatest fixpoint coinductive principle).

Formally, semantics of a $\mathsf{HML}^{\mathbf{RT}}_{\nu+\Box}$ formula is defined in the context of an LTS $\mathcal{L} = \langle S, s_0, \rightarrow, Act\rangle$ and an environment $\mathcal{E} \subseteq \mathcal{V}ar \times S$ as follows:

$$\begin{array}{lll} s \not\models_{\Phi,\mathcal{E}} \mathsf{F} & & \\ s \models_{\Phi,\mathcal{E}} \widetilde{X} & \overset{def}{\iff} & X \in \mathbf{R}(s) \\ s \models_{\Phi,\mathcal{E}} \phi \wedge \psi & \overset{def}{\iff} & s \models_{\Phi,\mathcal{E}} \phi \wedge s \models_{\Phi,\mathcal{E}} \psi \\ s \models_{\Phi,\mathcal{E}} [a]\,\psi & \overset{def}{\iff} & \forall q \in S.\,(s \xrightarrow{a} q \implies q \models_{\Phi,\mathcal{E}} \psi) \\ s \models_{\Phi,\mathcal{E}} [\epsilon]\,\psi & \overset{def}{\iff} & \forall q \in S.\,(s \overset{\epsilon}{\Longrightarrow} q \implies q \models_{\Phi,\mathcal{E}} \psi) \\ s \models_{\Phi,\mathcal{E}} [\widetilde{X}]\,\psi & \overset{def}{\iff} & X \in \mathbf{R}(s) \implies s \models_{\Phi,\mathcal{E}} \psi \\ s \models_{\Phi,\mathcal{E}} \mathcal{Z} & \overset{def}{\iff} & \begin{cases} (\mathcal{Z}, s) \in \mathcal{E} \\ \text{or } (\mathcal{Z}, s) \notin \mathcal{E} \text{ and } s \models_{\mathcal{E}\cup\{(\mathcal{Z},s)\}} \psi_{\mathcal{Z}} \text{ where } (\mathcal{Z} = \psi_{\mathcal{Z}}) \in \Phi \end{cases} \end{array}$$

To complete the semantics, we also define:

$$\begin{array}{lll} s \models \Phi & \overset{def}{\iff} & s \models_{\Phi,\emptyset} \mathcal{Z} \\ & & \text{where } \Phi = (\mathcal{Z} = \psi_{\mathcal{Z}})\Phi' \text{ for some } \Phi' \\ P \models \Phi & \overset{def}{\iff} & \forall_{s \in P}\ s \models \Phi \end{array}$$

The characteristic formula $\Phi^P_{\mathbf{RT}}$ for a process $P$ consists of equations in the form presented below. Recursive variables are annotated with processes; the initial variable is $\mathcal{Z}_P$, and an equation for $\mathcal{Z}_Q$ occurs whenever $Q$ (that ranges over relevant processes reachable from $P$) appears on the right-hand side of some preceding equation:

$$\begin{array}{rl} \nu\mathcal{Z}_Q = [\epsilon] & \Big( \bigwedge_{X \in \min_{\not\subseteq} \mathbf{R}(Q)} [\widetilde{X}]\,\mathsf{F} \\ & \wedge \bigwedge_{X_\cap \in \mathbf{FR}(Q)} \\ & \quad \big( \bigwedge_{X_m \in \text{min-base}_Q(X_\cap)} [\widetilde{X_m}]\,\widetilde{X_\cap} \\ & \quad \wedge \bigwedge_{a \in Act \setminus X_\cap} [\widetilde{X_\cap}]\,[a]\,[\epsilon]\,\mathcal{Z}_{Q \,\|\, X_\cap . a} \big) \\ & \wedge \bigwedge_{a : \bullet . a \notin \mathbf{RT}(Q)} [a]\,\mathsf{F} \\ & \wedge \bigwedge_{a : \bullet . a \in \mathbf{RT}(Q)} [a]\,[\epsilon]\,\mathcal{Z}_{Q \,\|\, \bullet . a} \Big) \end{array}$$

Note that since processes are subsets of state spaces, given a finite LTS the construction always yields a finite formula.

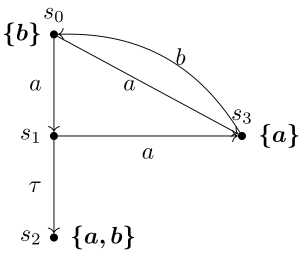


**Fig. 7.** A process $P$ for which a characteristic formula is constructed.

Observe that the structure of the formula $\Phi^P_{\mathbf{RT}}$ closely resembles the test generation algorithm (Section 5), a modification of which can be used to generate characteristic formulae.

**Theorem 3.** *$\Phi^P_{\mathbf{RT}}$ is a characteristic formula for $P$ w.r.t. refusal trace refinement, i.e.*

$$\forall Q \quad Q \models \Phi^P_{\mathbf{RT}} \iff P \sqsubseteq_{\mathbf{RT}} Q$$

**Proof.** We sketch the key parts of the proof.

I. Observe that $Q \not\models \Phi$ if and only if it can be proved with a finite *falsification path*, i.e. a finite path created by starting with $Q \not\models \Phi$ and successively applying semantic rules for $\mathsf{HML}^{\mathbf{RT}}_{\nu+\square}$, terminating in statements of the form either $s \not\models_{\Psi,\mathcal{E}} \mathsf{F}$, or $s \not\models_{\Psi,\mathcal{E}} \widetilde{X}$. If such a path exists, by its length we will denote the number of occurrences of statements of the form $s \not\models_{\Psi,\mathcal{E}} [a]\,\psi$, plus possibly the final $s \not\models_{\Psi,\mathcal{E}} \widetilde{X}$ in the path. The existence of a falsification path of length at most $k$ for a process $Q$ and formula $\Phi$ will be denoted with $Q \not\models^k \Phi$.

II. Since characteristic formulae have very similar structure to the complete test suites $\mathbf{TS_3}$, we can prove the main statement by showing that:

$$\forall P, Q \quad \left(Q \not\models^{\ell} \Phi^P_{\mathbf{RT}} \iff (\exists \sigma \in \mathbf{TS}^{\ell}_{\mathbf{3}}(P).\, \sigma \in \mathbf{RT}(Q))\right)$$

The proof proceeds by induction on $\ell$, assuming in IH that the above bi-implication holds for $\ell$, for all processes $P$ and $Q$. By juxtaposing the characteristic formula $\Phi^P_{\mathbf{RT}}$ and the inductive characterisation of the test suite $\mathbf{TS}^{\ell+1}_{\mathbf{3}}(P)$ (end of Section 4), we observe that the subformula in the first line of $\Phi^P_{\mathbf{RT}}$ does *not* hold precisely when $Q$ exhibits a trace of the form "$X$" given in the first line of the inductive definition of $\mathbf{TS}^{\ell+1}_{\mathbf{3}}$. Similar observation can be made for the correspondence of the lines 3 and 5 in both definitions (traces "$X_m.a$" and "$\bullet.a$", respectively). Finally, that falsifying line 4 [and 6] of the definition of $\Phi^P_{\mathbf{RT}}$ in $\leq \ell+1$ steps is equivalent to exhibiting a trace from line 4 [resp. 6] in $\mathbf{TS}^{\ell+1}_{\mathbf{3}}$, follows from Proposition 2 and the inductive hypothesis. □

**Example 4.** We apply our characteristic formula construction to the process $P$ depicted in Fig. 7.

$$\begin{aligned}
\nu\mathcal{Z}_{\{s_0\}} \quad &= \quad [\epsilon] \ \big([\widetilde{\{a\}}]\,\mathsf{F} \\
&\qquad\qquad \wedge [\emptyset]\,\widetilde{\{b\}} \\
&\qquad\qquad \wedge [\widetilde{\{b\}}]\,[a]\,[\epsilon]\,\mathcal{Z}_{\{s_1,s_2,s_3\}} \\
&\qquad\qquad \wedge [b]\,\mathsf{F} \\
&\qquad\qquad \wedge [a]\,[\epsilon]\,\mathcal{Z}_{\{s_1,s_2,s_3\}}\big) \\[1em]
\nu\mathcal{Z}_{\{s_1,s_2,s_3\}} \quad &= \quad [\epsilon] \ \big([\widetilde{\{a\}}]\,[b]\,\mathcal{Z}_{\{s_0\}} \\
&\qquad\qquad \wedge [\widetilde{\{b\}}]\,\widetilde{\{a\}} \\
&\qquad\qquad \wedge [a]\,[\epsilon]\,\mathcal{Z}_{\{s_3\}} \\
&\qquad\qquad \wedge [b]\,[\epsilon]\,\mathcal{Z}_{\{s_0\}}\big) \\[1em]
\nu\mathcal{Z}_{\{s_3\}} \quad &= \quad [\epsilon] \ \big([\widetilde{\{b\}}]\,\mathsf{F} \\
&\qquad\qquad \wedge [\widetilde{\emptyset}]\,\widetilde{\{a\}} \\
&\qquad\qquad \wedge [\widetilde{\{a\}}]\,[b]\,[\epsilon]\,\mathcal{Z}_{\{s_0\}} \\
&\qquad\qquad \wedge [a]\,\mathsf{F} \\
&\qquad\qquad \wedge [b]\,[\epsilon]\,\mathcal{Z}_{\{s_0\}}\big)
\end{aligned}$$

Observe that our construction contains certain redundancies, in particular weaker subformulae in conjuncts. For instance, in Example 4, the right hand side of the first equation defining $\mathcal{Z}_{\{s_0\}}$ the conjunction contains a subformula $[\emptyset]\,\widetilde{\{b\}}$ that can be removed due to the presence of the subformula $[b]\,\mathsf{F}$. As a simple optimisation, we can ensure that strictly forbidden actions are only handled once in the subformulae of the form $[a]\,\mathsf{F}$ and not include them in any refusal context.

We leave the topic of minimising characteristic formulae as an interesting idea for a future work. Note that while there is no equivalent technique to removing smaller traces in test suites, one can devise a special variant of simulation relation on the structure of the formulae (i.e. where recursive variables can be interpreted as states), and perform a minimisation modulo such simulation.

## 9. Conclusions and future work

This paper explored the problem of generating a complete and minimal test suite for testing from an LTS $P$. We considered the refusal (failure) trace semantics and used test sequences defined by refusal traces that the system under test should not have. In order to ensure that test suites are finite, we assumed a bound $\ell$ on the length of refusal traces used.

We set off by defining a simple but bulky complete test suite and then progressively kept refining it without reducing its effectiveness. In particular, the first important reduction step was a result of a few key insights into the distinct properties of refusal trace semantics, which in turn helped develop the fundamental equivalence on refusal traces – a test suite requires one trace per equivalence class. We described how to construct a simple and greatly reduced test suite which is not minimal in general, and moreover, provided an exhaustive method to remove any remaining redundant traces. We found that it is sometimes possible to remove certain refusal traces due to the presence of longer traces in the suite – such removal may not be desirable, so we have investigated minimality with either approach. Finally, we have also provided a characteristic formula construction for refusal trace refinement.

There are several potential lines of future work. Firstly, we may wish to adapt our work to distinguish between input and output actions. The basic difference is that, since the environment cannot block output, it is only possible to have a deadlock (and so observe a refusal) if the state is quiescent (i.e. not capable of producing output actions). As a result, we only need to consider refusals that include all outputs.

We would also like to explore the model-independent perspective where specification is given only in the form of a language of refusal traces; such an approach has been explored in [25] for trace semantics. This line of research may potentially lead to a language-based test generation algorithm, as well as equivalence class testing techniques akin to [25] and test suites with model-independent coverage properties.

Finally, we wish to extend our work on characteristic formulae to obtain similar constructions for other linear-time semantics and also devise more advanced minimisation methods along the lines suggested at the end of Section 8.

## Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Robert Hierons reports financial support was provided by Engineering and Physical Sciences Research Council.

## Data availability

No data was used for the research described in the article.

## Acknowledgments

We would like to thank Wan Fokkink and Tim Willemse for their useful comments and suggestions. We also thank the reviewers for their thorough feedback that helped improve the paper.

This work was supported by EPSRC grant EP/R025134/2, RoboTest: Systematic Model-Based Testing and Simulation of Mobile Autonomous Robots, and UKRI grant EP/V026801/2, Trustworthy Autonomous Systems Node in Verifiability.

## References

[1] R.M. Hierons, K. Bogdanov, J.P. Bowen, R. Cleaveland, J. Derrick, J. Dick, M. Gheorghe, M. Harman, K. Kapoor, P. Krause, G. Lüttgen, A.J.H. Simons, S.A. Vilkomir, M.R. Woodward, H. Zedan, Using formal specifications to support testing, ACM Comput. Surv. 41 (2) (2009) 9:1–9:76.

[2] W. Grieskamp, N. Kicillof, K. Stobie, V. Braberman, Model-based quality assurance of protocol documentation: tools and methodology, J. Software Test. Verif. Reliab. 21 (1) (2011) 55–71.

[3] M.-C. Gaudel, Testing can be formal too, in: 6th International Joint Conference CAAP/FASE Theory and Practice of Software Development (TAPSOFT'95), in: Lecture Notes in Computer Science, vol. 915, Springer, 1995, pp. 82–96.

[4] E.F. Moore, Gedanken-experiments, in: C. Shannon, J. McCarthy (Eds.), Automata Studies, Princeton University Press, 1956.

[5] R.v. Glabbeek, The linear time-branching time spectrum I. The semantics of concrete, sequential processes, in: J. Bergstra, A. Ponse, S. Smolka (Eds.), Handbook of Process Algebra, North Holland, 2001, Ch. 1.

[6] A.W. Roscoe, Understanding Concurrent Systems, Texts in Computer Science, Springer, 2011.

[7] J. Tretmans, A formal approach to conformance testing, Ph.D. thesis, University of Twente, Netherlands, 1992.

[8] J. Tretmans, Model based testing with labelled transition systems, in: Formal Methods and Testing, in: Lecture Notes in Computer Science, vol. 4949, Springer, 2008, pp. 1–38.

[9] I. Phillips, Refusal testing, Theor. Comput. Sci. 50 (3) (1987) 241–284.

[10] L. Heerink, J. Tretmans, Refusal testing for classes of transition systems with inputs and outputs, in: Formal Description Techniques and Protocol Specification, Testing and Verification (FORTE X/PSTV XVII), in: IFIP Conference Proceedings, vol. 107, Chapman & Hall, 1997, pp. 23–38.

[11] R.v. Glabbeek, Reactive bisimulation semantics for a process algebra with time-outs, in: 31st International Conference on Concurrency Theory (CONCUR 2020), in: LIPIcs, vol. 171, 2020, pp. 6:1–6:23.

[12] J. Peleska, W. Huang, A. Cavalcanti, Finite complete suites for CSP refinement testing, Sci. Comput. Program. 179 (2019) 1–23.

[13] A. Cavalcanti, R.M. Hierons, S.C. Nogueira, Inputs and outputs in CSP: a model and a testing theory, ACM Trans. Comput. Log. 21 (3) (2020) 24:1–24:53.

[14] M. Gazda, R.M. Hierons, Removing redundant refusals: minimal complete test suites for failure trace semantics, in: 36th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2021, Rome, Italy, June 29 - July 2, 2021, IEEE, pp. 1–13, 2021.

[15] K. El-Fakih, N. Yevtushenko, A. Saleh, Incremental and heuristic approaches for deriving adaptive distinguishing test cases for non-deterministic finite-state machines, Comput. J. 62 (5) (2019) 757–768.

[16] D. Lee, M. Yannakakis, Testing finite-state machines: state identification and verification, IEEE Trans. Comput. 43 (3) (1994) 306–320.

[17] H. Ural, X. Wu, F. Zhang, On minimizing the lengths of checking sequences, IEEE Trans. Comput. 46 (1) (1997) 93–99.

[18] S. Graf, J. Sifakis, A modal characterization of observational congruence on finite terms of CCS, Inf. Control 68 (1) (1986) 125–145.

[19] B. Steffen, A. Ingolfsdottir, Characteristic formulas for processes with divergence, Inf. Comput. 110 (1) (1994) 149–163.

[20] L. Aceto, A. Ingólfsdóttir, P.B. Levy, J. Sack, Characteristic formulae for fixed-point semantics: a general framework, Math. Struct. Comput. Sci. 22 (2) (2012) 125–173.

[21] L. Aceto, A. Ingólfsdóttir, J. Sack, Characteristic formulae for fixed-point semantics: a general framework, in: Proceedings EXPRESS 2009, in: EPTCS, vol. 8, 2009, pp. 1–15.

[22] L. Aceto, D.D. Monica, I. Fábregas, A. Ingólfsdóttir, When are prime formulae characteristic?, Theor. Comput. Sci. 777 (2019) 3–31.

[23] A. Cavalcanti, R.M. Hierons, Testing with inputs and outputs in CSP, in: 16th International Conference on Fundamental Approaches to Software Engineering (FASE 2013), in: Lecture Notes in Computer Science, vol. 7793, Springer, 2013, pp. 359–374.

[24] I. Bouchemakh, K. Engel, Interval stability and interval covering property in finite posets, Order 9 (2) (1992) 163–175.

[25] W. Huang, J. Peleska, Model-based testing strategies and their (in)dependence on syntactic model representations, Int. J. Softw. Tools Technol. Transf. 20 (4) (2018) 441–465.