
White Rose Research Online

eprints@whiterose.ac.uk
https://eprints.whiterose.ac.uk
Universities of Leeds, Sheffield and York

Deposited via The University of Sheffield.

White Rose Research Online URL for this paper:
https://eprints.whiterose.ac.uk/id/eprint/173724/

Version: Accepted Version

Proceedings Paper:
Gazda, M. and Hierons, R.M. (2021) Removing redundant refusals: minimal complete test suites for failure trace semantics. In: 2021 36th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS). Proceedings of the 2021 36th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS), 29 Jun - 02 Jul 2021, Rome, Italy. IEEE. ISBN: 9781665448963.

https://doi.org/10.1109/LICS52264.2021.9470737

© 2021 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other users, including reprinting/ republishing this material for advertising or promotional purposes, creating new collective works for resale or redistribution to servers or lists, or reuse of any copyrighted components of this work in other works. Reproduced in accordance with the publisher's self-archiving policy.

Reuse
Items deposited in White Rose Research Online are protected by copyright, with all rights reserved unless indicated otherwise. They may be downloaded and/or printed for private study, or other acts as permitted by national copyright laws. The publisher or other rights holders may allow further reproduction and re-use of the full text version. This is indicated by the licence information on the White Rose Research Online record for the item.

Takedown
If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing eprints@whiterose.ac.uk including the URL of the record and the reason for the withdrawal request.



Removing Redundant Refusals: Minimal Complete Test Suites for Failure Trace Semantics

Maciej Gazda
Department of Computer Science
University of Sheffield, UK

Robert M. Hierons
Department of Computer Science
University of Sheffield, UK

***Abstract*—We explore the problem of finding a minimal complete test suite for a refusal trace (or failure trace) semantics. Since complete test suites are typically infinite, we consider the setting with a bound ℓ on the length of refusal traces of interest. A test suite T is thus complete if it is failed by all processes that contain a disallowed refusal trace of length at most ℓ.**

The proposed approach is based on generating a minimal complete set of forbidden refusal traces. Our solution utilises several interesting insights into refusal trace semantics. In particular, we identify a key class of refusals called fundamental refusals which essentially determine the refusal trace semantics, and the associated fundamental equivalence relation. We then propose a small but not necessarily minimal test suite based on our theory, 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 (guarantee of shortest counterexamples).

A related problem is the construction of a characteristic formula of a process P, that is, a formula Φ_P such that every process which satisfies Φ_P refines P. Our test generation algorithm can be used to construct such a formula using a variant of Hennessy-Milner logic with recursion.

I. 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 [13]. There is evidence of the effectiveness of MBT in industrial projects [11] and, in addition, the use of MBT approaches can guarantee that well-defined classes of faults will be found (see, for example, [7]).

MBT work goes back to Moore's seminal paper [16] and initially considered finite state machine (FSM) specifications. Here, observations are traces: sequences of inputs and outputs. More generally, however, observations may be richer than this, leading to a wide range of implementation relations for labelled transition systems (LTSs) [8]. The interest in LTSs is motivated by their ability to capture the semantics of formal languages such as CSP (see, for example, [19]).

In testing, the consensus has been that observations are types of traces and so implementation relations are based on decorated traces. Traces can include, for example, refusals, where a refusal X denotes the system being in a stable state in which no action in X is enabled. Most work uses ioco [21] or one of its variants; see [22] 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 subsequent deadlock (detected through a timeout). A failure trace [8], also called a refusal trace [19], and initially introduced by Philips [18], 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 [12].

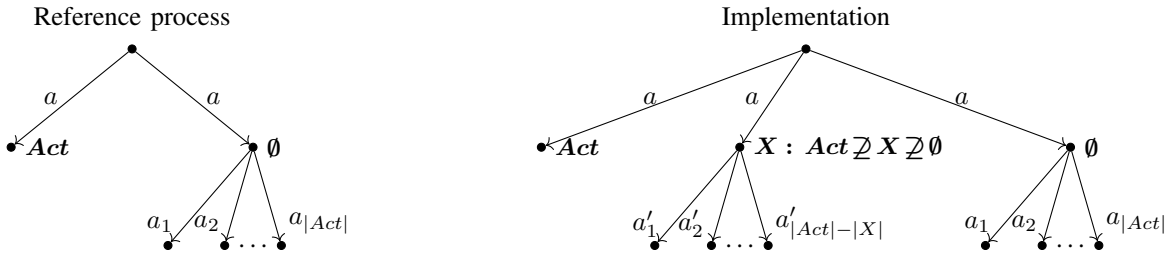
This paper considers testing from an LTS specification where observations are refusal traces. The use of refusal traces was motivated by several factors. First, in testing it is not normally possible to backtrack or save the system state and so observations are linear; they are decorated traces. Second, 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 [9]. Fourth, refusal traces are required if we wish to capture certain types of behaviour such as priorities [19]. 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. [17], which concerned generating a finite test suite to check behaviour up to a given bound on length. This work

978-1-6654-4895-6/21/$31.00 ©2021 IEEE

[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.



introduced techniques for testing for both traces and failures but not for refusal traces. Interestingly, however, it transpires that the approach of Peleska et al. [17] cannot be extended to test for refusal traces. This is due to the fundamental difference between testing failures and refusal traces refinement – the former semantics only requires testing for *forbidden* refusals (which can be efficiently performed by checking certain sets of allowed continuations as in [17]), whereas for the latter semantics 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. [5] in which a test case is a refusal trace σ that should *not* be a behaviour of the SUT; the observation of σ in testing will denote a faulty behaviour. Test execution involves determining whether the SUT can exhibit σ. 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.

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 and inputs may be refused. Previous work has looked at testing using refusal traces. For example, Cavalcanti et al. [5] 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 [12]. 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 [6], [15], [23], 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 which 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 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. In some cases we find that there may be a trade-off between minimality and usefulness of test suite in terms of clarity and succinctness of counterexamples – we address this by defining two different test suites which are provably minimal in both scenarios, depending on whether or not one prioritises the abovementioned usefulness criterion.

A line of research from the area of modal logic related to complete test suites are *characteristic formulae* [10], [20] 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. [1], [2] 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 [3].

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 [17], 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.

II. 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 $\mathcal{L} = \langle S, \rightarrow, Act \rangle$ where S is a finite set of states, Act a finite set of actions, and $\rightarrow \subseteq S \times Act \times S$ 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 *process* can be defined by indicating the set of possible initial states within a certain LTS. Formally, a process is a tuple $P = \langle S_I, \mathcal{L} \rangle$, where $\mathcal{L} = \langle S, \rightarrow, Act \rangle$ 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.

For a state s, a set $X \subseteq Act$ is a *refusal* of s, if for all $a \in X$, $s \not\xrightarrow{a}$. The set of all refusals of s is denoted with $\mathbf{R}(s)$. The *state refusal* of s is the largest refusal of s, denoted with $\mathbf{SR}(s)$.

A *refusal trace (failure trace)* σ of a state s is a sequence that is either an empty word ϵ, or a word of the form $X_0 a_1 X_1 a_2 \ldots X_{n-1} a_n X_n$, or $X_0 a_1 X_1 a_2 \ldots X_{n-1} a_n$ such that there is a chain of transitions $s = s_0 \xrightarrow{a_1} s_1 \xrightarrow{a_2} \ldots \xrightarrow{a_{n-1}} s_{n-1} \xrightarrow{a_n} s_n = q$ and for all $k \in \{0, \ldots, n-1, [n]\}$, $X_k \in \mathbf{R}(s_k)$. If in addition we have $X_k \in \mathbf{SR}(s_k)$, then the above trace is a *state refusal trace*. We denote the existence of such a chain of transitions by $s \xrightarrow{\sigma} q$. We define the *length* of a refusal trace in one of the above forms as n, that is, as the number of actions occurring in the trace. 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:

$$\mathbf{RT} \triangleq \{\sigma \in \mathcal{P}(Act) \times (Act \times \mathcal{P}(Act))^* \cup (\mathcal{P}(Act) \times Act)^* \mid \sigma = \rho.X_i.a_{i+1}.\rho' \Longrightarrow a_{i+1} \notin X_i\}$$

The notions of refusal, state refusal, and refusal trace are lifted from states to processes in a natural way, by taking the union over all (initial) states in a process. For instance, $\mathbf{R}(P) \triangleq \bigcup_{s \in P} \mathbf{R}(s)$ [recall that we identify the process P with its set of initial states].

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

For a process P and refusal trace σ, the process P after σ, denoted with $P|\sigma$, is the set $\{q \in S \mid \exists s \in P : \ s \xrightarrow{\sigma} q\}$.

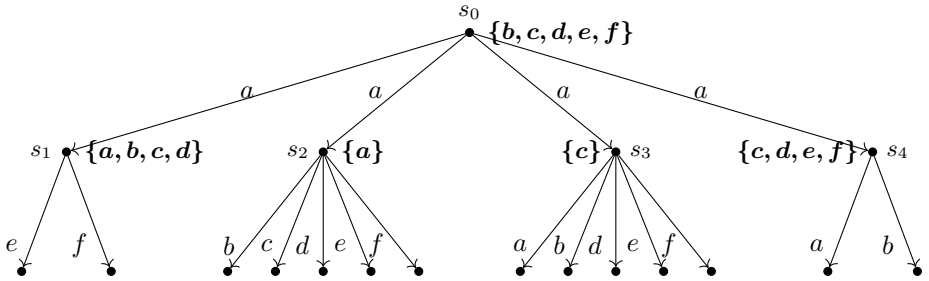
Example 1: Consider the LTS from Fig. 2, representing a process P with $s_I = s_0$. Process P has one state refusal, i.e. $\mathbf{SR}(P) = \{\{b, c, d, e, f\}\}$, while its set of refusals $\mathbf{R}(P)$ consists of all subsets of $\{b, c, d, e, f\}$. After performing action a, the process exhibits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

III. 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

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.



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.

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

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 1: 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 $\mathbf{TS_0}$, 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 [4], [17].

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

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

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

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 different sources of redundancies in a test suite.

IV. 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 / step). 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.

A. 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*:

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

The definition is extended to *fundamental refusal traces*:

$$\begin{aligned}\mathbf{FRT}(P) \triangleq\ & \{\epsilon\} \cup \{X_0 a_1 \ldots X_n \in \mathbf{RT}(P) \,|\, X_0 \in \mathbf{FR}(P) \\ & \wedge \forall i : X_i \in \mathbf{FR}(P|X_0 a_1 \ldots a_i)\}\end{aligned}$$

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.

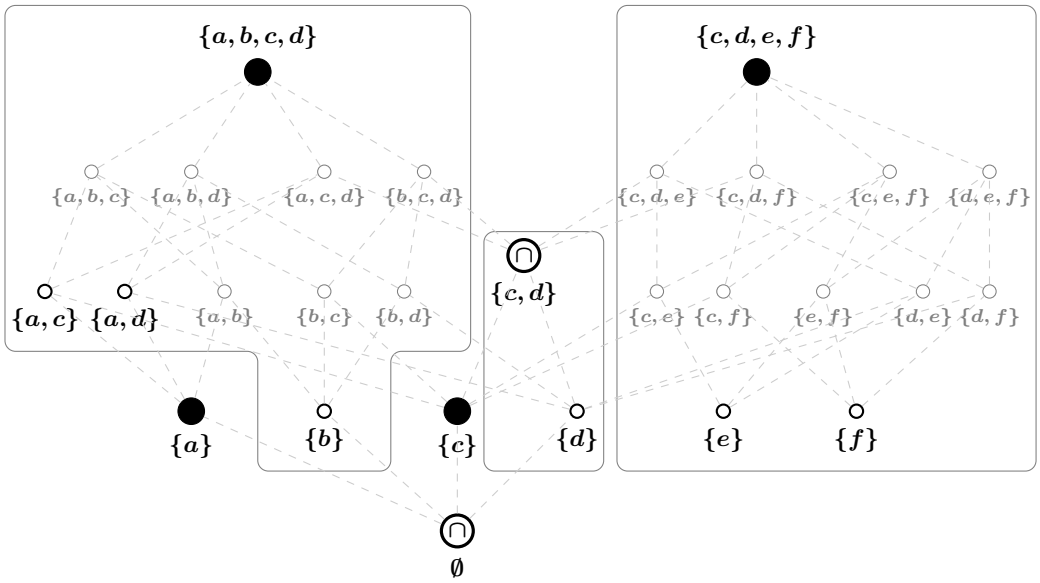
Proposition 2: A refusal X of a process P [after σ] is fundamental if and only if no action outside X is explicitly forbidden by the semantics directly after X:

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

Proof: "$\Longrightarrow$": Suppose that $X \in \mathbf{FR}(P|\sigma)$. Let $X_1, \ldots, X_n \in \mathbf{SR}(P|\sigma)$ 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 state $s \in P|\sigma$ such that $\mathbf{SR}(s) = X_i \supseteq X$. We have $s \xrightarrow{a}$ and hence $\sigma.X.a \in \mathbf{RT}(P)$.

"$\Longleftarrow$": Suppose that $\forall a \in Act \setminus X\ \sigma.X.a \in \mathbf{RT}(P)$. Let $\mathcal{Y}_X = \{Y \in \mathbf{SR}(P|\sigma) \,|\, X \subseteq Y\}$. $\mathcal{Y}_X$ 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 $a \notin \widehat{Y}$. Hence $\bigcap \mathcal{Y}_X \subseteq X$ and thus $X = \bigcap \mathcal{Y}_X$, from which $X \in \mathbf{FR}(P|\sigma)$ follows. ■

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 labeled with "$\cap$". 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).



Nontrivial clusters:

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

As a corollary, we obtain the following preservation property for fundamental refusals and their traces.

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

1) $\forall \sigma \in \mathbf{RT}^{\ell}(P)$ $X \in \mathbf{FR}^{\ell}(Q|\sigma) \implies X \in \mathbf{FR}^{\ell}(P|\sigma)$ $\left[X \in \mathbf{FR}^{\ell}(P|\sigma) \iff X \in \mathbf{FR}^{\ell}(Q|\sigma)\right]$
2) $\mathbf{FRT}^{\ell}(Q) \subseteq \mathbf{FRT}^{\ell}(P)$ $\left[\mathbf{FRT}^{\ell}(P) = \mathbf{FRT}^{\ell}(Q)\right]$

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 where it can refuse all elements of $\mathsf{top}_P(X)$.

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

$$X \sim_P Y \stackrel{def}{\iff} \mathsf{top}_P(X) = \mathsf{top}_P(Y)$$

$$[X]_P \triangleq \{Y \in \mathbf{R}(P) \mid X \sim_P Y\}$$

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 forbidden continuations.

Proposition 3: Suppose $\sigma.X \in \mathbf{RT}(P)$ and $a \in Act$. Then

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

Proof: "$\Longrightarrow$": Follows immediately from Proposition 2.
"$\Longleftarrow$": 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 $s \in P|\sigma$ such that $X \subseteq \mathbf{SR}(s)$, we have $s \stackrel{a}{\not\longrightarrow}$. 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 4.

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

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

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 3 and $\mathbf{RT}(Q) \subseteq \mathbf{RT}(P)$ it follows that for all $a \in \mathsf{top}_{P|\sigma}(X)$, $s \stackrel{a}{\not\longrightarrow}$, from which in turn we obtain $\mathsf{top}_{P|\sigma}(X) \in \mathbf{R}(s)$. Hence $\mathsf{top}_{P|\sigma}(X) \in \mathbf{R}(Q|\sigma)$. ∎

B. Fundamental equivalence and test suites

Proposition 3 provides us with a full characterisation of forbidden continuations, thanks to which we are able to update our test suite $\mathbf{TS_1}$ with a more accurate description.

$$\begin{aligned}\mathbf{TS}_\mathbf{1}^\ell(P) \quad = \quad & \{\sigma.X \in \mathbf{TS}_\mathbf{0}^\ell \mid X \in \mathsf{min}_{\mathbb{Q}}(\mathbf{R}(P|\sigma))\} \\ & \cup \{\sigma.X.a \in \mathbf{TS}_\mathbf{0}^\ell \mid a \in \mathsf{top}_{P|\sigma}(X) \setminus X\}\end{aligned}$$

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

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

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

Lemma 2: $\mathbf{TS}_\mathbf{2}^\ell$ is complete for $\mathbf{RT}^\ell$.

Proof: Suppose $P \not\sqsubseteq_{\mathbf{RT}}^\ell Q$. From the completeness of $\mathbf{TS}_\mathbf{1}^\ell(P)$, there is some trace $\sigma.X.a \in \mathbf{RT}^\ell(Q) \cap \mathbf{TS}_\mathbf{1}^\ell(P)$. From the definition of the 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 $\sigma.X_m.a \in \mathbf{TS}_\mathbf{2}^\ell(P)$ follows. Since refusals of a process are downward-closed, we have $\sigma.X_m.a \in \mathbf{RT}^\ell(Q)$, hence $\sigma.X_m.a \in \mathbf{RT}^\ell(Q) \cap \mathbf{TS}_\mathbf{2}^\ell(P)$ ∎

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 top_P operator to (valid) refusal traces of the reference process:

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

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 is not affected by the operator*:

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

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

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):

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

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

Proposition 4: Suppose $P \sqsubseteq_{\mathbf{RT}}^\ell Q$. For any $\sigma \in \mathbf{RT}^\ell$:

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

Proof: 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 smaller refusals occurring in σ as compared to $\mathsf{top}_P(\sigma)$. 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 σ. Proof of statement 2 makes the assumption that $\sigma \in \mathbf{RT}^\ell(Q)$ – the other case is trivial.

Base. In the base case, the trace σ is either empty (trivial) or has the form $\sigma = X$ for some refusal X:

1) If $X \in \mathbf{RT}^0(Q)$, then $X \in \mathbf{R}(Q)$; from Corollary 2 we have $\mathsf{top}_P(X) \in \mathbf{R}(Q)$, and hence $\mathsf{top}_P(X) \in \mathbf{RT}^0(Q)$.
2) Take any $s \in Q|X$, i.e. any $s \in Q$ such that $X \in \mathbf{R}(s)$. Since $\mathbf{SR}(s) \in \mathbf{FR}(Q)$, from Corollary 1 we have $\mathbf{SR}(s) \in \mathbf{FR}(P)$. Moreover, since $X \subseteq \mathbf{SR}(s)$, and $\mathsf{top}_P(X)$ is the smallest fundamental refusal in $\mathbf{FR}(P)$ that includes X, we have $\mathsf{top}_P(X) \subseteq \mathbf{SR}(s)$. Hence $\mathsf{top}_P(X) \in \mathbf{R}(s)$, which finally yields $s \in Q|\mathsf{top}_P(X)$.

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

- $\sigma = X_0 a_1 \dots X_k a_{k+1}$: Since $\sigma \in \mathbf{RT}^{k+1}(Q)$, there must be some $s \in Q|X_0 a_1 \dots X_k$ and $q \in Q|X_0 a_1 \dots X_k a_{k+1}$ such that $s \xrightarrow{a_{k+1}} q$. From the inductive hypothesis, we know that (1) $\mathsf{top}_P(X_0 a_1 \dots X_k) \in \mathbf{RT}^k(Q)$ and moreover (2) $s \in Q|\mathsf{top}_P(X_0 a_1 \dots X_k)$. Hence

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

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. Since the above reasoning holds for an arbitrary choice of $q \in Q|X_0 a_1 X_1 \dots X_k a_{k+1}$, we have $q \in Q|\mathsf{top}_P(X_0 a_1 X_1 \dots X_k a_{k+1})$, which proves statement 2.
- $\sigma = X_0 a_1 \dots a_{k+1} X_{k+1}$:
Let $s \in Q|X_0 a_1 \dots a_{k+1} X_{k+1}$. From the proof of the case above, we already know that $\mathsf{top}_P(X_0 a_1 \dots a_{k+1}) \in \mathbf{RT}^{k+1}(Q)$ and $s \in Q|\mathsf{top}_P(X_0 a_1 \dots a_{k+1})$. In order to prove statements 1 and 2, it suffices to show that $\mathsf{top}_{P_{k+1}}(X_{k+1}) \in \mathbf{R}(s)$, where $P_{k+1} = P|X_0 a_1 \dots a_{k+1}$. This can be done similarly as in base case (2): we have $X_{k+1} \subseteq \mathbf{SR}(s)$; from Corollary 1 we also have $\mathbf{SR}(s) \in \mathbf{FR}(P_{k+1})$, and finally $\mathsf{top}_{P_{k+1}}(X_{k+1})$ is the smallest fundamental refusal in P_{k+1} that includes X_{k+1}, yielding $\mathsf{top}_{P_{k+1}}(X_{k+1}) \subseteq \mathbf{SR}(s)$. Hence $\mathsf{top}_{P_{k+1}}(X_{k+1}) \in \mathbf{R}(s)$.

■

Proposition 4 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:

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

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

$$\mathbf{TS}_{\mathbf{3}}^{0}(P) = \bigcup_{X \in \min_{\subseteq} \mathbf{R}(P)} \{X\}$$

while for $\ell \geq 0$:

$$\begin{aligned}\mathbf{TS}_{\mathbf{3}}^{\ell+1}(P) = {} & \textstyle\bigcup_{X \in \min_{\subseteq} \mathbf{R}(P)} \{X\} \\ & \cup \textstyle\bigcup_{X_\cap \in \mathbf{FR}(P)} \\ & \big(\textstyle\bigcup_{X_m \in \mathsf{min\text{-}base}_P(X_\cap)} \bigcup_{a \in (X_\cap \setminus X_m)} \{X_m.a\} \\ & \cup \textstyle\bigcup_{a \in Act \setminus X_\cap} \bigcup_{\sigma \in \mathbf{TS}_{\mathbf{3}}^{\ell}(P|X_\cap.a)} \{X_\cap.a.\sigma\} \big)\end{aligned}$$

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 continuations (line 3), as well as longer traces prefixed with a fundamental refusal and relevant actions (line 4).

V. 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.

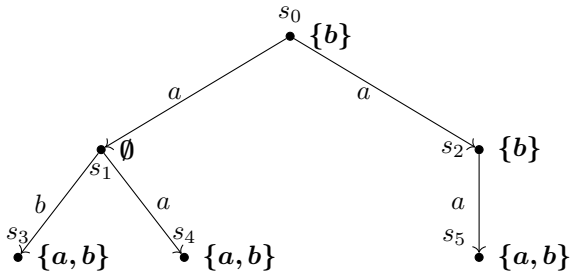
A. Strictly smaller refusal traces

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 reasonably easy to remove – we do not need to include a forbidden trace in a test suite, if the suite also contains a certain "smaller" trace. Here, by smaller we mean the natural order on refusal traces which combines pointwise prefix and inclusion order.

$$\begin{aligned} & X_0 a_1 \dots a_n[X_n] \preceq Y_0 a_1 \dots a_m[Y_m] \\ & \overset{def}{\iff} \; n \leq m \wedge \forall i \in \{0, \dots, n-1, [n]\}\, X_i \subseteq Y_i \end{aligned}$$

$$\begin{aligned} & X_0 a_1 \dots a_n[X_n] \prec Y_0 a_1 \dots a_m[Y_m] \\ & \overset{def}{\iff} \; X_0 a_1 \dots a_n[X_n] \preceq Y_0 a_1 \dots a_m[Y_m] \wedge \exists i : X_i \subsetneq Y_i \end{aligned}$$

It is immediate to see that if a trace is exhibited by a process, then so are all traces below it w.r.t. $\preceq$. Hence a forbidden trace σ is redundant in a test suite, if the test suite contains a trace σ' such that $\sigma' \prec \sigma$. As an example, consider the following LTS representing a reference model (assuming $Act = \{a, b\}$):



Our test suite $\mathbf{TS_3}$ contains in particular traces $\{b\}.a.\{b\}.a.\emptyset.a$, 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. $\prec$.

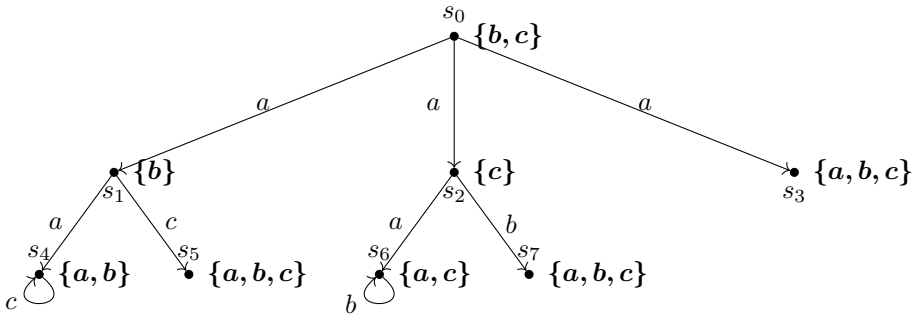
$$\mathbf{TS}_{\mathbf{4}}^{\ell}(P) \quad \triangleq \quad \min_{\prec} \mathbf{TS}_{\mathbf{3}}^{\ell}$$

B. Test generation algorithm

We shall now briefly describe an effective procedure to construct our test suite $\mathbf{TS_4}$. 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 $\mathbf{TS}.\mathsf{Add}()$ must ensure that a refusal trace is added only if $\mathbf{TS}$ contains no larger trace w.r.t. $\prec$.

The number of all refusal traces of length $\leq \ell$ is bounded by $\mathcal{O}(2^{|Act|} \cdot (|Act| \cdot 2^{|Act|})^{\ell}) = \mathcal{O}(2^{|Act|^{\ell+1}})$. 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 $\mathcal{O}\big((|\mathsf{Max}_\cap| \cdot |Act|)^{\ell} \cdot 2^{|Act|}\big) = \mathcal{O}(|Act|^{\ell} \cdot |\mathsf{Max}_\cap|^{\ell} \cdot 2^{|Act|})$. We note that $|\mathsf{Max}_\cap|$ can be polynomial w.r.t. $|Act|$ for instance if refusals have hierarchical structure (linear order), or are disjoint.

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.



***GenerateTests*(k, ℓ, σ, P)**

//I: generate forbidden refusals

1) for all $(X \in \mathsf{min}_{\trianglelefteq}(\mathbf{R}(P)))$ $\mathbf{TS}.\mathsf{Add}(\sigma.X)$
2) if $(k > \ell)$ return;

//II: add forbidden continuations and recursively generate longer tests with extensions of current trace

3) compute $\mathbf{FR}(P)$ and $\mathsf{min\text{-}base}_P$
4) for each $X_\cap \in \mathbf{FR}(P)$ in non-decreasing order

//II.1 if the cluster is nontrivial then we need to add forbidden continuations

a) if $(|[X_\cap]|_P > 1)$
$\mathcal{Y} := \mathsf{min\text{-}base}_P(X_\cap)$
for all $(Y \in \mathcal{Y})$
for all $(a \in X_\cap \setminus Y)$ $\mathbf{TS}.\mathsf{Add}(\sigma.Y.a)$

// II.2 recursively generate longer tests with extensions of current trace

b) for all $(a \in Act \setminus X_\cap)$
$GenerateTests(k+1, \ell, \sigma.X_\cap.a, P)$

Algorithm 1: Test case generation algorithm

VI. 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 continuation, the other a rather counter-intuitive possibility of detecting a violation through a different violation occurring at a later stage of computation.

A. Redundant continuation 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 $X.b$ 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 continuation 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 $\mathsf{TS}_{\sigma.X.a}$ 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 VII, 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 note that the problem is more complex than finding a minimal number of "locally complete" tests which would cover every violation within a fixed cluster (i.e. caused by state refusals $\widehat{X}$ that belong to the same cluster). This is because one may obtain additional material for inference from outside the cluster, in particular from a smaller trace context i.e. $\sigma'.Y.a \prec \sigma.X.a$.

B. 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 σ in our test suite.

However, it turns out that removing σ from our test suite does not affect completeness. This is because any system that exhibits σ 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_σ 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 $s_\sigma \not\xrightarrow{b}$, then Q exhibits in particular $\{b,c\}.a.\{c\}.a.\{b\}$, forbidden by P

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

The key question is whether we really wish to remove σ from our test suite in the scenario above. It gives 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 it appears undesirable to detect violations in such a roundabout way (through longer traces). In view of that, we introduce a sanity condition on a test suite called uniform completeness – the lack of "gaps" in completeness for all lengths smaller or equal to the length of interest ℓ.

Formally, a test suite TS for a reference model P [and length bound ℓ] is *uniformly complete (u.c.)* for $\sqsubseteq^{\ell}_{\mathbf{RT}}$ whenever for all $k \leq \ell$ it is complete for $\sqsubseteq^{k}_{\mathbf{RT}}$.

VII. Minimal test suites

We now proceed to present an enumerative method that allows to remove the redundancies described in the previous section. We establish two families of test suites: the first one consists of minimal uniformly complete suites, whereas the other contains potentially smaller suites which are minimal and complete, but may provide suboptimal (i.e. long) counterexamples.

A. Minimal uniformly complete test suites

To accomplish minimality, we employ a method that utilises a slightly different perspective on incorrect behaviour – rather than looking solely at traces, we focus on causes of forbidden behaviours in terms of "wrong" *state refusals*.

The set of *state refusal violatons* w.r.t. a reference process P contains traces composed of a fundamental trace σ of P followed by a refusal that cannot occur as a state refusal after σ in any refinement of P (or, equivalently, refusal that is not fundamental in $P|\sigma$).

$$\begin{array}{rcl} \mathbf{VSR}_P & \triangleq & \{\sigma.\widehat{X} \in \mathbf{RT} \mid \sigma \in \mathbf{FRT}(P) \wedge \widehat{X} \notin \mathbf{FR}(P|\sigma)\} \\ \mathbf{VSR}^{\ell}_P & \triangleq & \mathbf{VSR}_P \cap (\mathbf{RT}^{\ell-1} \cup \{\sigma.\widehat{X} \notin \mathbf{RT}^{\ell}(P)\}) \end{array}$$

The latter class $\mathbf{VSR}^{\ell}_P$ are state refusal violations which result in forbidden traces[2] of length $\leq \ell$. The set of violations w.r.t. P exhibited by an implementation Q is defined as:

$$\begin{array}{rcl} \mathbf{VSR}^{[\ell]}_P(Q) & \triangleq & \{\sigma.\widehat{X} \in \mathbf{VSR}^{[\ell]}_P \\ & & \mid \sigma \in \mathbf{RT}(Q) \wedge \widehat{X} \in \mathbf{SR}(Q|\sigma)\} \end{array}$$

In the above, $[l]$ is simply a shorthand to allow both $\mathbf{VSR}_P(Q)$ and $\mathbf{VSR}^{\ell}_p(Q)$ to be defined. Note that we only consider violations that can be reached by fundamental traces. This is sufficient due to Proposition 4, and witnessed by the following lemma – a lack of refinement is always caused by some state refusal violation:

Lemma 3: $P \not\sqsubseteq^{\ell}_{\mathbf{RT}} Q \iff \mathbf{VSR}^{\ell}_P(Q) \neq \emptyset$

[2]Observe that if $\sigma.\widehat{X} \in \mathbf{VSR}^{\ell}_P \cap \mathbf{RT}^{\ell}(P)$ has length ℓ, then the resulting action-terminated forbidden trace has length $\ell + 1$, hence we do not include these violations in $\mathbf{VSR}^{\ell}_P$.

For each process P, its state refusal violation $\sigma.\widehat{X} \in \mathbf{VSR}_P$, and a test suite $\mathsf{TS} \subseteq \mathbf{RT} \setminus \mathbf{RT}(P)$, the set of *detectors* of $\sigma.\widehat{X}$ in TS contains traces in TS, smaller or equivalent[3] to $\sigma.\widehat{X}$ w.r.t. preorder $\preceq$, which can always "detect" $\sigma.\widehat{X}$ – that is, traces exhibited by any process Q such that $\widehat{X} \in \mathbf{SR}(Q|\sigma)$.

$$\begin{array}{l} \mathsf{detectors}_P(\sigma.\widehat{X}, \mathsf{TS}) \\ \triangleq \begin{cases} \{\sigma'.Y.a \in \mathsf{TS} \mid a \notin \widehat{X} \wedge \sigma'.Y.a \preceq \sigma.\widehat{X}.a\} & \text{if } \sigma.\widehat{X} \in \mathbf{RT}(P) \\ \{\sigma'.Y \in \mathsf{TS} \mid \sigma'.Y \preceq \sigma.\widehat{X}\} & \text{if } \sigma.\widehat{X} \notin \mathbf{RT}(P) \end{cases} \end{array}$$

Lemma 4: Take any test suite TS for P, violation $\sigma.\widehat{X} \in \mathbf{VSR}^{\ell}_P$, and any of its detectors $\rho \in \mathsf{detectors}_P(\sigma.\widehat{X}, \mathsf{TS})$. For any Q such that $\sigma.\widehat{X} \in \mathbf{VSR}^{\ell}_P(Q)$, we have $\rho \in \mathbf{RT}^{\ell}(Q)$.

The *detector cover* of a test suite is the family of detector sets of all conceivable violations (with legnth bound ℓ):

$$\begin{array}{rcl} \mathsf{DetCov}_P(\mathsf{TS}, \ell) & \triangleq & \{\mathsf{detectors}_P(\sigma.\widehat{X}, \mathsf{TS}) \\ & & \mid \sigma.\widehat{X} \in \mathbf{VSR}^{\ell}_P\} \end{array}$$

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

Lemma 5: Any test suite TS for P such that $\mathsf{DetCov}_P(\mathsf{TS}, \ell)$ does not contain an empty set, is uniformly complete for P w.r.t. $\sqsubseteq^{\ell}_{\mathbf{RT}}$.

Observe that $\mathbf{TS}^{\ell}_{\mathbf{4}}$ meets the above sufficient condition. Indeed, for any violation $\sigma.\widehat{X} \in \mathbf{VSR}^{\ell}_P$:

- if $\sigma.\widehat{X} \notin \mathbf{RT}^{\ell}(P)$, then $\sigma.Y \in \mathbf{TS}^{\ell}_{\mathbf{4}}$, where Y is a minimal refusal such that $Y \subseteq \widehat{X}$ and $\sigma.Y \notin \mathbf{RT}^{\ell}(P)$
- otherwise we have $\sigma.X_m.a \in \mathbf{TS}^{\ell}_{\mathbf{4}}$, where $X_m \in \mathsf{min\text{-}base}_{P|\sigma}(\widehat{X})$, $X_m \subseteq \widehat{X}$ and $a \notin \widehat{X}$

This suggests an idea of obtaining minimal 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:

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

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. We will show that $\mathcal{TS}^{\boldsymbol{uni}}_{\boldsymbol{min}}$ indeed contains minimal uniformly complete test suites.

As an intermediate step, we shall prove an important property of fundamental traces. Namely, for any valid fundamental trace $\sigma \in \mathbf{FR}(P)$, we can construct a process $P\backslash\sigma$ that is refusal trace equivalent to the original process P, but in which the trace σ is a *state refusal trace*

Proposition 5: Fix a process P. A trace $\sigma \in \mathbf{RT}$ is a fundamental refusal trace of P ($\sigma \in \mathbf{FRT}(P)$) if and only if one can construct a process, denoted with $P\backslash\sigma$, such that

1) $P\backslash\sigma =_{\mathbf{RT}} P$, and
2) σ is a *state refusal trace* of $P\backslash\sigma$.

[3]Formally speaking, they are smaller or equal w.r.t. $\preceq$ than the maximal detectors of $\sigma.\widehat{X}$, defined in the proof of Theorem 1.

Proof: "$\Longleftarrow$": If σ is a state refusal trace of $P\backslash\sigma$, then in particular $\sigma \in \mathbf{FRT}(P\backslash\sigma)$, from $P\backslash\sigma =_{\mathbf{RT}} P$ and preservation of fundamental refusals we obtain $\sigma \in \mathbf{FRT}(P)$.

"$\Longrightarrow$": 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 $i \in \{0, \ldots, n-1, [n]\}$, and each $a \in Act \setminus X_i$, we define transitions: $s_i \xrightarrow{a} q$ 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 that the state refusals of each s_i are equal to X_i. We also create 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: $P\backslash\sigma = P \cup \{s_0\}$. ■

As a side remark, we note that Proposition 5 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 $=_{\mathbf{RT}}$.

Theorem 1: For any process P, $\mathcal{TS}^{\boldsymbol{uni}}_{\boldsymbol{min}}(P, \ell)$ is a family of minimal test suites (with the smallest number of traces) which are uniformly complete for P w.r.t. $\sqsubseteq^{\ell}_{\mathbf{RT}}$.

Proof: We first establish that:

(*) Any test suite $\mathsf{TS}^{uc}_4 \subseteq \mathbf{TS}^{\ell}_{\mathbf{4}}$ uniformly complete for P and $\sqsubseteq^{\ell}_{\mathbf{RT}}$, is a hitting set of $\mathsf{DetCov}_P(\mathbf{TS}^{\ell}_{\mathbf{4}}, \ell)$.

Fix a process P, length $\ell \in \mathbb{N}$, and assume that a test suite $\mathsf{TS}^{uc}_4 \subseteq \mathbf{TS}^{\ell}_{\mathbf{4}}$ is uniformly complete for $\sqsubseteq^{\ell}_{\mathbf{RT}}$. Suppose, towards contradiction, that there is some violation $\sigma.\widehat{X} \in \mathbf{VSR}^{\ell}_P$ such that $\mathsf{detectors}^{\ell}_P(\sigma.\widehat{X}, \mathsf{TS}^{uc}_4) = \emptyset$, and let $k \leq \ell$ be the minimal integer for which $\sigma.\widehat{X} \in \mathbf{VSR}^{k}_P$. We will show that TS^{uc}_4 is not complete for $\sqsubseteq^{k}_{\mathbf{RT}}$.

We shall need the auxiliary definitions of maximal detectors of a violation and their downward closure:

$$\mathsf{maxdets}_P(\rho.\widehat{Y}) \triangleq \begin{cases} \{\rho.\widehat{Y}.a \mid a \in \mathsf{top}_{P|\rho}(\widehat{Y}) \setminus \widehat{Y}\} & \text{if } \rho.\widehat{Y} \in \mathbf{RT}(P) \\ \{\rho.\widehat{Y}\} & \text{if } \rho.\widehat{Y} \notin \mathbf{RT}(P) \end{cases}$$

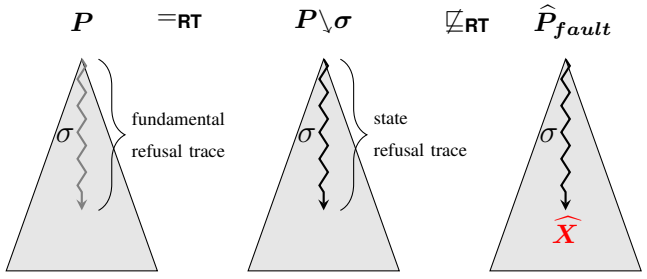
$$\downarrow_{\preceq} \mathsf{maxdets}_P(\rho.\widehat{Y}) \triangleq \{\lambda \mid \exists \pi \in \mathsf{maxdets}_P(\rho.\widehat{Y}) \mid \lambda \preceq \pi\}$$

Observe that for any $\rho.\widehat{Y} \in \mathbf{VSR}_P$ and test suite TS', we have $\mathsf{detectors}_P(\rho.\widehat{Y}, \mathsf{TS}') = \mathsf{TS}' \cap \downarrow_{\preceq} \mathsf{maxdets}_P(\rho.\widehat{Y})$.

We will construct a process $\widehat{P}_{fault}$ that exhibits $\sigma.\widehat{X}$, but differs from P (w.r.t. $\sqsubseteq^{k}_{\mathbf{RT}}$) only on traces contained in $\downarrow_{\preceq} \mathsf{maxdets}_P(\sigma.\widehat{X})$, i.e. $\widehat{P}_{fault}$ is such that $\left(\mathbf{RT}^k(\widehat{P}_{fault}) \setminus \mathbf{RT}^k(p)\right) \subseteq \downarrow_{\preceq} \mathsf{maxdets}_P(\sigma.\widehat{X})$. This suffices to show that TS^{uc}_4 is not complete for $\sqsubseteq^{k}_{\mathbf{RT}}$, since for such $\widehat{P}_{fault}$ we have: $\mathsf{TS}^{uc}_4 \cap \left(\mathbf{RT}^k(\widehat{P}_{fault}) \setminus \mathbf{RT}^k(p)\right) \subseteq \mathsf{TS}^{uc}_4 \cap \downarrow_{\preceq} \mathsf{maxdets}_P(\sigma.\widehat{X}) = \mathsf{detectors}^{\ell}_P(\sigma.\widehat{X}, \mathsf{TS}^{uc}_4) = \emptyset$.

Since $\sigma \in \mathbf{FRT}(P)$, according to Proposition 5 there is a process $P\backslash\sigma$ refusal trace equivalent to P, and containing σ as a *state refusal trace*. Our new process $\widehat{P}_{fault}$ is constructed by modifying the LTS underlying the process $P\backslash\sigma$.

We add a fresh state s_{new} to the LTS ensuring that it appears in the new process after *state refusal trace* σ – this way s_{new} will be contained in $\widehat{P}_{fault}|\sigma$, *but not in* $\widehat{P}_{fault}|\sigma'$ *for any* $\sigma \prec \sigma'$. This can be done as follows: if $\sigma = \epsilon$, then s_{new} is simply included as an initial state of $\widehat{P}_{fault}$. Otherwise σ is



of the form $\sigma'.b$, and we add a b-transition into s_{new} from the final state of the chain which realises the state refusal trace σ, as described in the proof of Proposition 5. Finally, we endow s_{new} with outgoing transitions: for each $b \in (Act \setminus \widehat{X})$ we add a transition $s_{new} \xrightarrow{b} s_{new}$, so that $\mathbf{SR}(s_{new}) = \widehat{X}$.

We have thus constructed a process $\widehat{P}_{fault}$ such that:

- $\sigma.\widehat{X} \in \mathbf{VSR}^k_P(\widehat{P}_{fault})$, and
- $\mathbf{RT}^k(\widehat{P}_{fault}) \setminus \mathbf{RT}^k(P) \subseteq \downarrow_{\preceq} \mathsf{maxdets}_P(\sigma.\widehat{X})$.

This shows that TS^{uc}_4 is not complete w.r.t. $\sqsubseteq^{k}_{\mathbf{RT}}$, contradicting uniform completeness of TS^{uc}_4; we have thus proved (*).

Statement (*) entails that test suites in $\mathcal{TS}^{\boldsymbol{uni}}_{\boldsymbol{min}}(P, \ell)$ are uniformly complete and minimal w.r.t. size *among the subsets of* $\mathbf{TS}_{\mathbf{4}}$. We will now show that no uniformly complete test suite has smaller size than those in $\mathcal{TS}^{\boldsymbol{uni}}_{\boldsymbol{min}}(P, \ell)$.

Take any test suite TS^{uc} uniformly complete for P w.r.t. $\sqsubseteq^{\ell}_{\mathbf{RT}}$. Let us define the following function[4] $f : \mathsf{TS}^{uc} \to \mathbf{TS}_{\mathbf{4}}$:

- $f(\sigma.X) \triangleq \mathsf{top}_P(\sigma).X^f_{min}$ where X^f_{min} is a minimal refusal such that $X^f_{min} \subseteq X$ and $\sigma.X^f_{min} \notin \mathbf{RT}(P)$
- $f(\sigma.X.a) \triangleq \mathsf{top}_P(\sigma).X_m.a$ where $X_m \in \mathsf{min\text{-}base}_{P|\sigma}(X)$ and $X_m \subseteq X$

Observe that for every $\sigma \in \mathsf{TS}^{uc}$, $f(\sigma) \notin \mathbf{RT}^k(P)$, where k is the length of σ. Moreover, for any implementation Q and $\sigma \in \mathsf{TS}^{uc}$, we have $\sigma \in \mathbf{RT}(Q) \implies f(\sigma) \in \mathbf{RT}(Q)$, and thus uniform completeness of the test suite $f(\mathsf{TS}^{uc}) \subseteq \mathbf{TS}_{\mathbf{4}}$ follows from the uniform completeness of TS^{uc}. Hence for an arbitrary uniformly complete test suite one can construct a uniformly complete test suite of no larger size contained in $\mathbf{TS}_{\mathbf{4}}$. Combined with (*) and the definition of $\mathcal{TS}^{\boldsymbol{uni}}_{\boldsymbol{min}}$, this proves the main statement. ■

B. Minimality 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 "absolutely" minimal test suites remains an interesting theoretical challenge. Here, we consider this general case (non-uniform completeness).

The key insight is that when considering violations after a trace ρ, we only need to take care of those that can be caused by state refusals $\widehat{X}$ that are *realisably hidden*, that is, for which extensions consistent with further behaviour of P exist. For

[4] 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.

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 "hidden" when inspecting longer traces.

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

$$\begin{array}{rcl}\mathsf{rhv}^{\ell}(P) & \triangleq & \{\sigma.\widehat{X} \in \mathbf{VSR}^{\ell}_{P} \mid \\ & & \Big((|\sigma| < \ell) \implies (\forall b \in Act \setminus \widehat{X} \\ & & \exists Q_b.\ \forall Y \subsetneq \widehat{X}\ P|\sigma.Y.b \sqsubseteq^{\ell-|\sigma|-1}_{\mathbf{RT}} Q_b)\Big)\}\end{array}$$

where we define $P \sqsubseteq^{-1}_{\mathbf{RT}} Q$ as true for all processes P, Q.

The statement in lines 2-3 implies that one can construct a process which has a state after σ 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.

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

- $\{b,c\}.a.\{a,b\} \in \mathsf{rhv}^{\ell}(P)$ (for any ℓ): the 2nd line of definition of rhv^{ℓ} requires 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 VI-B, there is no process Q_a that would refine both $P|\{b,c\}.a.\{a\}.a$ and $P|\{b,c\}.a.\{b\}.a$.

In our new test suite, among all local forbidden behaviours after σ, 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:

$$\begin{array}{rcl}\mathsf{DetCov}^{\mathsf{rhv}}_{P}(\mathsf{TS},\ell) & \triangleq & \{\mathsf{detectors}_{P}(\sigma.\widehat{X},\mathsf{TS}) \mid \\ & & \sigma.\widehat{X} \in \mathsf{rhv}^{\ell}(P)\}\end{array}$$

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

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

It transpires that $\mathcal{TS}_{\boldsymbol{min}}$ consists of minimal complete test suites (though not necessarily uniformly complete).

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

We remark that test suites in $\mathcal{TS}_{\boldsymbol{min}}$ 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.

VIII. 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 Φ and modal formulae ψ are defined as follows:

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

The language $\mathsf{HML}^{\mathbf{RT}+\nu}_{\square}$ 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 $\widetilde{X} \implies \psi$ (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 $\mathsf{HML}^{\mathbf{RT}+\nu}_{\square}$ 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:

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

To complete the semantics, we also define:

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

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:

$$\begin{array}{rcl} \nu\mathcal{Z}_Q & = & \bigwedge_{X \in \min_{\sqsubseteq}\, \mathbf{R}(Q)} [\widetilde{X}]\,\mathsf{F} \\ & & \wedge \bigwedge_{X_\cap \in \mathbf{FR}(Q)} \\ & & \big(\bigwedge_{X_m \in \text{min-base}_Q(X_\cap)} [\widetilde{X}_m]\, \widetilde{X_\cap} \\ & & \wedge \bigwedge_{a \in Act \setminus X_\cap} [X_\cap]\,[a]\mathcal{Z}_{Q|X_\cap.a} \big) \end{array}$$

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

Observe that the structure of the formula $\Phi^P_{\mathbf{RT}}$ closely resembles the test generation algorithm (Section V), 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. $\forall Q \;\; Q \models \Phi^P_{\mathbf{RT}} \iff P \sqsubseteq_{\mathbf{RT}} Q$

Proof: We sketch the key parts of the proof.

I. We first show that for all processes P, Q:

$$(*) \qquad P \sqsubseteq^{[\ell]}_{\mathbf{RT}} Q \iff [\forall k \in 0,\ldots,\ell]\; \forall \sigma \in \mathbf{RT}^{[k]}(P)\,.\; P \,|\, \sigma \sqsubseteq^{[\ell-k]}_{\mathbf{RT}} Q \,|\, \sigma$$

The nontrivial direction is from left to right. Observe that for any process $\widehat{P}$, and trace $\sigma = \tau.X$ of length k, the set $\mathbf{RT}(\widehat{P} \,|\, \tau.X)$ can be expressed as a function of $\mathbf{RT}(\widehat{P})$:

$$\begin{array}{rcl} \mathbf{RT}^{[\ell-k]}(\widehat{P} \,|\, \tau.X) & = & \{Z.\lambda \,|\, \exists Y \supseteq Z : \tau.Y.\lambda \in \mathbf{RT}^{[\ell]}(\widehat{P}) \\ & & \wedge\; X \subseteq Y\} \end{array}$$

Using the above identity we can establish that if $\mathbf{RT}^{[\ell]}(Q) \subseteq \mathbf{RT}^{[\ell]}(P)$, then $\mathbf{RT}^{[\ell-k]}(Q \,|\, \tau.X) \subseteq \mathbf{RT}^{[\ell-k]}(P \,|\, \tau.X)$.

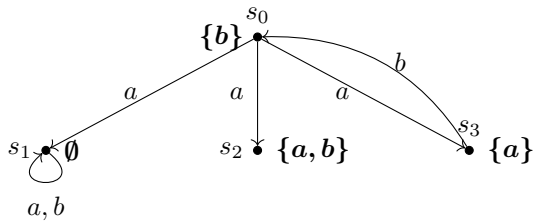
II. Observe that $Q \not\models \Phi$ if and only if it can be proved with a finite *falsification path*, i.e. a finite path created by starting with $Q \not\models \Phi$ and successively applying semantic rules for $\mathsf{HML}^{\mathbf{RT}+\nu}_{\Box}$, terminating in statements of the form either $s \not\models_{\Psi,\mathcal{E}} \mathsf{F}$, 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 Φ will be denoted with $Q \not\models^k \Phi$.

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

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

The proof proceeds by induction on ℓ, assuming in IH that the above bi-implication holds for ℓ, 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 IV), 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 third lines in both definitions (traces "$X_m.a$"). Finally, that falsifying line 4 of the definition of $\Phi^P_{\mathbf{RT}}$ in $\leq \ell+1$ steps is equivalent to exhibiting a trace from line 4 in $\mathbf{TS}^{\ell+1}_{\mathbf{3}}$, follows from (*) and the inductive hypothesis. ∎

Fig. 5.



Example 4: We apply our characteristic formula construction to the process P depicted in Fig. 5.

$$\begin{array}{rcl} \nu\mathcal{Z}_{\{s_0\}} & = & [\widetilde{\{a\}}]\,\mathsf{F} \\ & & \wedge\, [\emptyset]\,\widetilde{\{b\}} \\ & & \wedge [\widetilde{\{b\}}]\,[a]\,\mathcal{Z}_{\{s_1,s_2,s_3\}} \\ \\ \nu\mathcal{Z}_{\{s_1,s_2,s_3\}} & = & [\emptyset]\,[a]\,\mathcal{Z}_{\{s_1\}} \\ & & \wedge [\emptyset]\,[b]\,\mathcal{Z}_{\{s_0,s_1\}} \\ & & \wedge [\widetilde{\{a\}}]\,[b]\,\mathcal{Z}_{\{s_0\}} \\ & & \wedge [\widetilde{\{b\}}]\,\widetilde{\{a\}} \\ \\ \nu\mathcal{Z}_{\{s_1\}} & = & [\widetilde{\{a\}}]\,\mathsf{F} \wedge [\widetilde{\{b\}}]\,\mathsf{F} \\ & & \wedge [\widetilde{\emptyset}]\,[a]\,\mathcal{Z}_{\{s_1\}} \\ & & \wedge [\widetilde{\emptyset}]\,[b]\,\mathcal{Z}_{\{s_1\}} \\ \\ \nu\mathcal{Z}_{\{s_0,s_1\}} & = & [\widetilde{\{a\}}]\,\mathsf{F} \\ & & \wedge [\widetilde{\emptyset}]\,[a]\,\mathcal{Z}_{\{s_1,s_2,s_3\}} \\ & & \wedge [\widetilde{\emptyset}]\,[b]\,\mathcal{Z}_{\{s_1\}} \\ & & \wedge [\widetilde{\{b\}}]\,[a]\,\mathcal{Z}_{\{s_1,s_2,s_3\}} \end{array}$$

IX. Future work

There are several potential lines of future work. Firstly, we did not consider internal actions. We believe that while including silent steps requires a few technical adjustments, the essential part of our results should carry over to the extended setting. The key difference in the context of internal actions is that refusals can only be observed in stable states. To cater for traces passing through transient states (having an outgoing internal action), the refusals can be endowed with an additional element $\bullet$, corresponding to the minimal observation, which can be recorded in any state. We also use an extended order on refusals $\sqsubset$, assuming that $\bullet \sqsubset X$ for all $X \subseteq Act$.

We can then adapt our framework by treating $\bullet$ as a (potential) state refusal – whenever a specification process P has a transient state after a trace σ, $\bullet$ is included in $\mathbf{SR}(P|\sigma)$. Our definitions regarding clusters from Section IV can then be extended in a natural way, using $\bullet$ as a special case of state/fundamental refusal, for instance:

$$\mathsf{top}_P(\bullet) \triangleq \begin{cases} \bullet & \text{if } \bullet \in \mathbf{SR}(P) \\ \bigcap_{Y \in \mathbf{SR}(P)} Y & \text{if } \bullet \notin \mathbf{SR}(P) \end{cases}$$

An interesting open question is whether redundancies in continuation contexts described in Section VI-A can be characterised in a more productive manner. In particular, it would be desirable to reduce the test suite $\mathbf{TS_4}$ with a more efficient method than those described in Section VII.

Another possible direction of research is 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.

Finally, we would like to explore the model-independent perspective where specification is given only in the form a language of refusal traces; such approach has been explored in [14] 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 [14] and test suites with model-independent coverage properties.

X. Conclusions

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 ℓ 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.

Acknowledgements

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.

References

[1] Luca Aceto, Anna Ingólfsdóttir, Paul Blain Levy, and Joshua Sack. Characteristic formulae for fixed-point semantics: a general framework. *Math. Struct. Comput. Sci.*, 22(2):125–173, 2012.

[2] Luca Aceto, Anna Ingólfsdóttir, and Joshua Sack. Characteristic formulae for fixed-point semantics: A general framework. In *Proceedings EXPRESS 2009*, volume 8 of *EPTCS*, pages 1–15, 2009.

[3] Luca Aceto, Dario Della Monica, Ignacio Fábregas, and Anna Ingólfsdóttir. When are prime formulae characteristic? *Theoretical Computer Science*, 777:3 – 31, 2019.

[4] Ana Cavalcanti and Robert M. Hierons. Testing with inputs and outputs in CSP. In *16th International Conference on Fundamental Approaches to Software Engineering (FASE 2013)*, volume 7793 of *Lecture Notes in Computer Science*, pages 359–374. Springer, 2013.

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

[6] Khaled El-Fakih, Nina Yevtushenko, and Ayat Saleh. Incremental and heuristic approaches for deriving adaptive distinguishing test cases for non-deterministic finite-state machines. *Comput. J.*, 62(5):757–768, 2019.

[7] Marie-Claude Gaudel. Testing can be formal too. In *6th International Joint Conference CAAP/FASE Theory and Practice of Software Development (TAPSOFT'95)*, volume 915 of *Lecture Notes in Computer Science*, pages 82–96. Springer, 1995.

[8] Rob van Glabbeek. The linear time-branching time spectrum I. The semantics of concrete, sequential processes. In J.A. Bergstra, A. Ponse, and S.A. Smolka, editors, *Handbook of process algebra*, chapter 1. North Holland, 2001.

[9] Rob van Glabbeek. Reactive Bisimulation Semantics for a Process Algebra with Time-Outs. In *31st International Conference on Concurrency Theory (CONCUR 2020)*, volume 171 of *LIPIcs*, pages 6:1–6:23, 2020.

[10] S. Graf and J. Sifakis. A modal characterization of observational congruence on finite terms of CCS. *Information and Control*, 68(1):125 – 145, 1986.

[11] Wolfgang Grieskamp, Nicolas Kicillof, Keith Stobie, and Victor Braberman. Model-based quality assurance of protocol documentation: tools and methodology. *The Journal of Software Testing, Verification and Reliability*, 21(1):55–71, 2011.

[12] Lex Heerink and Jan 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)*, volume 107 of *IFIP Conference Proceedings*, pages 23–38. Chapman & Hall, 1997.

[13] Robert M. Hierons, Kirill Bogdanov, Jonathan P. Bowen, Rance Cleaveland, John Derrick, Jeremy Dick, Marian Gheorghe, Mark Harman, Kalpesh Kapoor, Paul Krause, Gerald Lüttgen, Anthony J. H. Simons, Sergiy A. Vilkomir, Martin R. Woodward, and Hussein Zedan. Using formal specifications to support testing. *ACM Computing Surveys*, 41(2):9:1–9:76, 2009.

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

[15] David Lee and Mihalis Yannakakis. Testing finite-state machines: State identification and verification. *IEEE Transactions on Computers*, 43(3):306–320, 1994.

[16] Edward F. Moore. Gedanken-experiments. In C. Shannon and J. McCarthy, editors, *Automata Studies*. Princeton University Press, 1956.

[17] Jan Peleska, Wen-ling Huang, and Ana Cavalcanti. Finite complete suites for CSP refinement testing. *Science of Computer Programming*, 179:1–23, 2019.

[18] Iain Phillips. Refusal testing. *Theoretical Computer Science*, 50(3):241–284, 1987.

[19] Andrew William Roscoe. *Understanding Concurrent Systems*. Texts in Computer Science. Springer, 2011.

[20] Bernhard Steffen and Anna Ingolfsdottir. Characteristic formulas for processes with divergence. *Information and Computation*, 110(1):149 – 163, 1994.

[21] Jan Tretmans. *A formal approach to conformance testing*. PhD thesis, University of Twente, Netherlands, 1992.

[22] Jan Tretmans. Model based testing with labelled transition systems. In *Formal Methods and Testing*, volume 4949 of *Lecture Notes in Computer Science*, pages 1–38. Springer, 2008.

[23] Hasan Ural, Xiaolin Wu, and Fan Zhang. On minimizing the lengths of checking sequences. *IEEE Transactions on Computers*, 46(1):93–99, 1997.