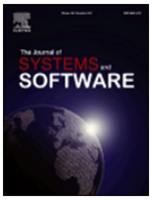
Contents lists available at ScienceDirect

The Journal of Systems and Software

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



Model-based mutant equivalence detection using automata language equivalence and simulations[☆]

Xavier Devroey[a,*], Gilles Perrouin[b,1], Mike Papadakis[c], Axel Legay[d], Pierre-Yves Schobbens[b], Patrick Heymans[b]

[a] SERG, Delft University of Technology,Postbus 5, 2600 AA Delft, The Netherlands
[b] Faculty of Computer Science, Namur Digital Institute, PReCISE, University of Namur, rue Grandgagnage 21, B-5000 Namur, Belgium
[c] Interdisciplinary Centre for Security, Reliability and Trust (SnT), SERVAL Team, University of Luxembourg, 29, Avenue J.F Kennedy, L-1855 Luxembourg, Luxembourg
[d] INRIA Rennes, Campus universitaire de Beaulieu, 35042 Rennes Cedex, France

ARTICLE INFO

Article history:
Received 20 November 2017
Revised 8 March 2018
Accepted 13 March 2018
Available online 15 March 2018

Keywords:
Model-based mutation analysis
Automata language equivalence
Random simulations

ABSTRACT

Mutation analysis is a popular technique for assessing the strength of test suites. It relies on the mutation score, which indicates their fault-revealing potential. Yet, there are mutants whose behaviour is equivalent to the original system, wasting analysis resources and preventing the satisfaction of a 100% mutation score. For finite behavioural models, the Equivalent Mutant Problem (EMP) can be transformed to the language equivalence problem of non-deterministic finite automata for which many solutions exist. However, these solutions are quite expensive, making computation unbearable when used for tackling the EMP. In this paper, we report on our assessment of a state-of-the-art exact language equivalence tool and two heuristics we proposed. We used 12 models, composed of (up to) 15,000 states, and 4710 mutants. We introduce a random and a mutation-biased simulation heuristics, used as baselines for comparison. Our results show that the exact approach is often more than ten times faster in the weak mutation scenario. For strong mutation, our biased simulations can be up to 1000 times faster for models larger than 300 states, while limiting the error of misclassifying non-equivalent mutants as equivalent to 8% on average. We therefore conclude that the approaches can be combined for improved efficiency.

1. Introduction

Mutation analysis is a technique that injects artificial defects, called *mutations*, into the code under test, yielding *mutants*. Mutants are typically used to evaluate the effectiveness of test suites (Andrews et al., 2006; Offutt, 2011; Papadakis et al., 2018) and to support test generation (Papadakis and Malevris, 2010; Fraser and Arcuri, 2014; Offutt, 2011). The technique is quite popular in research due to the ability of mutants to simulate the behaviour of real faults (Andrews et al., 2006; Just et al., 2014). There is also evidence showing that tests designed to detect mutants reveal significantly more faults than other test criteria (Chekam et al., 2017; Offutt, 2011; Baker and Habli, 2013).

These characteristics of mutation inspired researchers to apply the method on artefacts other than code and particularly models (Offutt, 2011; Papadakis et al., 2014b; 2018). The usual advantages of model-based testing technique is the ability to identify defects related to missing functionality or misinterpreted specifications (Budd and Gopal, 1985) where code-based testing fails (Howden, 1976; Voas and McGraw, 1997). The method has been shown to be practical and can complement existing approaches. For instance, Aichernig et al. (2014) report that model mutants lead to tests that are able to reveal implementation faults that were found neither by manual tests, nor by the actual operation, of an industrial system.

Despite its potential, mutation analysis faces a number of challenges that currently prevent wider adoption (Papadakis et al., 2015; 2018). One of them is the *Equivalent Mutants Problem* (EMP). This problem concerns the identification of the mutants whose behaviour is identical to the original artefact (code or model). Such mutants cannot be distinguished by any test, a situation that raises

[☆] This article is an extension of our previous work Devroey et al. (2017), published as a short paper in the proceedings of the 10th IEEE International Conference on Software Testing, Verification and Validation (ICST 2017)

[*] Corresponding author.
E-mail addresses: x.d.m.devroey@tudelft.nl (X. Devroey), gilles.perrouin@unamur.be (G. Perrouin), michail.papadakis@uni.lu (M. Papadakis), axel.legay@inria.fr (A. Legay), pierre-yves.schobbens@unamur.be (P.-Y. Schobbens), patrick.heymans@unamur.be (P. Heymans).

[1] FNRS research associate

https://doi.org/10.1016/j.jss.2018.03.010
0164-1212/© 2018 Elsevier Inc. All rights reserved.

two issues: (i) they hamper the use of the criterion as a stopping rule by skewing the mutation score measurement (the number of detected mutants divided by the total number of mutants), and (ii) they do not bring any new value to the test generation techniques as they attempt to kill mutants that have no chance to be killed.

In this paper, we focus on the model-based formulation of the EMP, which can be expressed in terms of language equivalence. Language equivalence has been studied by the formal verification community who determined its PSPACE complexity (Kupferman and Vardi, 1996) and derived exact equivalence checking algorithms (Bonchi and Pous, 2013a; Doyen and Raskin, 2010). While potentially helpful, such tools have, to our knowledge, never been used to tackle the EMP. This is the main contribution and novelty of this paper.

In summary, the contributions of this paper are:

- The design of two simulation algorithms relying on random simulations (RS) and biased simulations (BS) that aim at covering infected states (Ammann and Offutt, 2008) (i.e., exploiting syntactical differences between original and mutant models) to improve the chances to distinguish non-equivalent mutants;
- A configurable implementation of our simulations (available at https://projects.info.unamur.be/vibes/) that benefits from the fact that simulation can be easily distributed among processor cores;
- The definition of an experimental setup to apply an automata language equivalence tool (ALE) (Bonchi and Pous, 2013a) to the EMP. We employed twelve models of varying origins and sizes, from 9 to 15,000 states. We produced 4710 mutants using seven operators, and considered four mutation orders (one, two, five, ten), according to strong and weak mutation scenarios.
- The assessment of the ALE tool with respect to our baseline algorithms. We measured the speed and accuracy of equivalence detection. The ALE tool is particularly efficient for weak mutation by being, on average, ten times faster than simulations. However, biased simulations perform well for strong mutation on models larger than 300 states: they can be 1000 times faster. The ratio of tagging non-equivalent mutants as equivalent is 8% for biased simulations and 15% for random ones. To ease reproducibility, all our models and experimental results are available at: https://projects.info.unamur.be/vibes/mutants-equiv.html.

This paper extends our previous work (Devroey et al., 2017) on the major following points: the empirical analysis is now performed on 12 models of size up to 15,000 states and 4710 mutants (instead of 3 models and 1170 mutants); it adds a new research question to analyse the impact of strong and weak mutation on automata language equivalence performance; finally, we provide statistical significance evidence.

The remainder of the paper is organised as follows. Section 2 presents background information on the models used and language equivalence, while Section 3 details the design of our simulation heuristics and the ALE approach we used. Section 4 describes our empirical assessment and provides some lessons learned. Section 5 covers relevant literature. Finally Section 6, wraps up the paper.

2. Background

In this section we introduce the main formalism, namely, finite transition systems, and the relevance of language equivalence for equivalent mutant detection, that we use throughout the paper.

2.1. Transition systems & finite automata

We consider transition systems as a powerful abstract formalism to model system behaviour. We adapt and follow the definition of Baier and Katoen's (2008), where atomic propositions have been omitted (we do not consider state internals). Thus, we consider:

Definition 1 (Transition System (TS))**.** A TS is a tuple $(S, Act, trans, i)$ where S is a set of states, Act is a set of actions, $trans \subseteq S \times Act \times S$ is a non-deterministic transition relation (with $(s_1, \alpha, s_2) \in trans$, denoted $s_1 \xrightarrow{\alpha} s_2$), and $i \in S$ is the initial state.

To deal with test generation activities, where finite behaviours are sought, we first require the sets S and Act to be finite. To mimic weak and strong mutation scenarios (see Section 3.1), we impose the requirement of stoping the test execution at specific states. These requirements make the non-deterministic finite automata (NFA) sematics be equivalent to our executions. This key observation enables the comparison of our simulations with the ALE tools. In the remainder of this paper, unless otherwise stated, we refer to TSs with such restrictions so that the term can be used interchangeably with NFAs[2].

Definition 2 (Trace)**.** Let $ts = (S, Act, trans, i)$ be a TS, let $t = (\alpha_1, \ldots, \alpha_n)$ where $\alpha_1, \ldots, \alpha_n \in Act$ be a finite sequence of actions. The trace t is valid iff:

$$ts \overset{(\alpha_1,\ldots,\alpha_n)}{\Longrightarrow}$$

where $ts \overset{(\alpha_1,\ldots,\alpha_n)}{\Longrightarrow}$ is equivalent to $\exists s \in S : i \overset{(\alpha_1,\ldots,\alpha_n)}{\Longrightarrow} s$, meaning that there exists a non-empty sequence of transitions labelled $(\alpha_1, \ldots, \alpha_n)$ from i to a state s of the TS.

2.2. Equivalent mutant problem

In this paper, we focus on the model-based instance of the Equivalent Mutant Problem (EMP). The equivalent mutant problem is a well-known issue in mutation analysis (Papadakis et al., 2018; Ammann and Offutt, 2008; Papadakis et al., 2015). It stems from the fact that two program variants may exhibit the same behaviour and therefore cannot be distinguished by test cases. This is particularly problematic with respect to both generation and assessment of test suites, since in the former case resources are spent on trying to kill non-killable mutants and in the later case skewing the assessment score (100% of killed mutants is impossible to reach in case of equivalence). Mutant equivalence can take two forms (Papadakis et al., 2015): (a) equivalence between mutants and the original system; (b) equivalence between two mutants (not with the original system). Mutants of case (a) are called *equivalent* while mutants of case (b) are called *duplicate*. In the context of this paper, we focus on mutants that are behaviourally *equivalent* to the original system, i.e., mutants of case (a).

2.3. Automata language equivalence & EMP

In our context, the EMP corresponds to a classic problem in automata theory: *Automata Language Equivalence* (ALE). The accepted language of an automaton is formed by all the sequences of actions (words) that can be accepted *i.e.*, starting in the initial state and ending in a final state. Therefore, if a mutant m accepts the same language as the original o (language-equivalent), then there is no trace t that can distinguish the mutant from the original:

$$\forall t, t \in \mathcal{L}(o) \Leftrightarrow t \in \mathcal{L}(m)$$

[2] Our MBT framework, VIBeS, uses TSs as its underlying formalism so we stick to the term "TS" for consistency.