Applying valued booleans in testing of cyber-physical systems

Applying Valued Booleans in Testing of Cyber-Physical Systems*

Koen Claessen*, Nicholas Smallbone*, Johan Lidén Eddeland†‡, Zahra Ramezani‡, Knut Åkesson‡ and Sajed Miremadi†
*Department of Computer Science and Engineering
Chalmers University of Technology, Gothenburg, Sweden
Email: {koen, nicsma}@chalmers.se
†Volvo Car Corporation
Gothenburg, Sweden
Email: {firstname.lastname}@volvocars.com
‡Department of Electrical Engineering
Chalmers University of Technology, Gothenburg, Sweden
Email: {johedd,rzahra,knut}@chalmers.se

In software testing, as in cyber-physical systems testing, test suites are traditionally developed by hand. In this work we consider one framework for putting the computer in charge of the testing instead: *constrained random test case generation* as supported by the tool QuickCheck [2]. This is implemented by the use of *Valued Booleans* (VBools). VBools naturally allow for an extension of QuickCheck into cyber-physical systems, which is useful particularly since QuickCheck can perform *shrinking* of test cases. Shrinking is a technique to make test cases simpler while preserving failure.

This work has three main contributions. First, we adapt random testing with QuickCheck to hybrid systems. Second, we define VBools as a way to simplify counterexamples (*shrinking*) when testing hybrid systems. Finally, we compare random testing and shrinking to existing falsification tools, to illustrate strengths and weaknesses of random testing and the importance of simplifying counterexamples.

Formally, a VBool is a pair of a Boolean value and a robustness value, which is a non-negative number:

$$\mathbb{V} = \mathbb{B} \times \mathbb{R}_{\geq 0}$$

The robustness of a false VBool represents the severity of the failure. The robustness of a true VBool represents the severity with which its *negation* would have failed, which roughly coincides with how convincingly the test passed. The most common VBool operator is $\leq_{\mathrm{v}}$, which takes the difference between its arguments as its robustness:

$$\leq_{\mathrm{v}} : \mathbb{R} \times \mathbb{R} \to \mathbb{V}$$
$$x \leq_{\mathrm{v}} y = \begin{cases} (\top, x - y) & \text{if } x \leq y \\ (\bot, y - x) & \text{otherwise} \end{cases}$$

The other comparison operators are defined in terms of $\leq_{\mathrm{v}}$. $\top$ and $\bot$ denote true and false, respectively.

We define conjunction $\wedge_+$ as follows. The Boolean part of $x \wedge_+ y$ is computed as $x \wedge y$. For robustness, suppose that we have a property which is a conjunction of two sub-properties, both of which fail; we would like the total severity to be the *sum* of the two severities. Therefore, when we take the conjunction of two false VBools, their robustnesses are added. We do the same if both VBools are true. If one VBool is true and the other is false, there is no obvious way to combine the two robustnesses, so we take the robustness of the false VBool (as shown in (1) - (4)). The other boolean operators are defined as in (5) - (8).

$$(\bot, x) \wedge_+ (\bot, y) = (\bot, x + y) \tag{1}$$
$$(\bot, x) \wedge_+ (\top, y) = (\bot, x) \tag{2}$$
$$(\top, x) \wedge_+ (\bot, y) = (\bot, y) \tag{3}$$
$$(\top, x) \wedge_+ (\top, y) = \left(\top, \frac{1}{\frac{1}{x} + \frac{1}{y}}\right) \tag{4}$$
$$\top_{\mathrm{v}} = (\top, \infty) \tag{5}$$
$$\bot_{\mathrm{v}} = (\bot, \infty) \tag{6}$$
$$\neg_{\mathrm{v}}(b, x) = (\neg b, x) \tag{7}$$
$$x \vee_+ y = \neg_{\mathrm{v}}(\neg_{\mathrm{v}} x \wedge_+ \neg_{\mathrm{v}} y) \tag{8}$$

We perform some experiments on a model of a heater, where we test against the following specification: *If the setpoint temperature has been constant (steady) for*

* This abstract summarizes the work made in a conference paper which is accepted to WODES 2018 [3].

978-1-5386-6748-4/18/$31.00 ©2018 IEEE
DOI 10.1109/MT-CPS.2018.00011

50 minutes, then the difference between the setpoint temperature and the actual room temperature should be at most $1°C$.

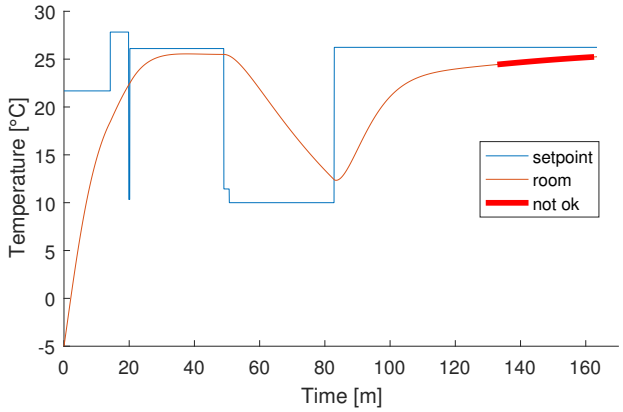

Figure 1: Random testing performed with QuickCheck on the heater example.

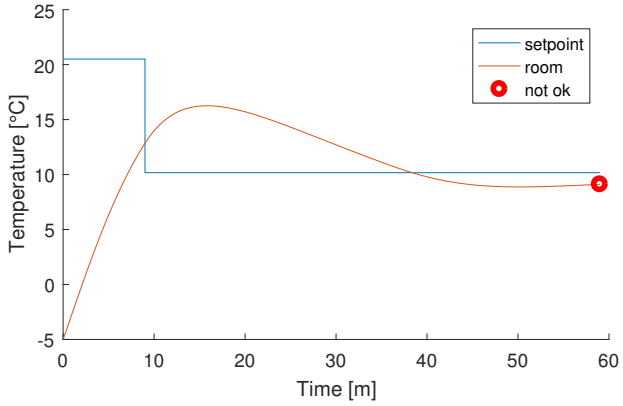

Figure 2: Standard shrinking turns the counterexample into a glitch.

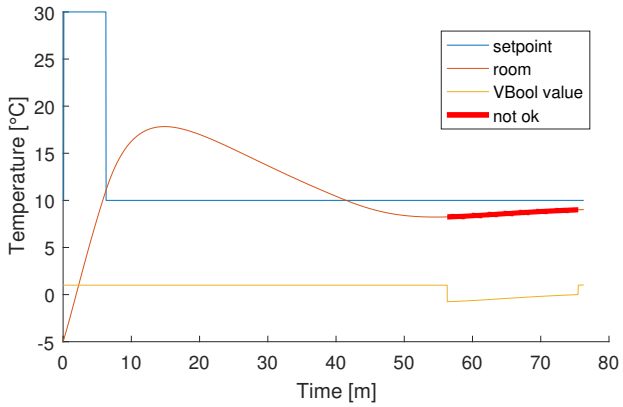

Figure 3: Shrinking with valued Booleans. We have included the VBool value in the graph for clarity, even though it is not a temperature.

Random testing without shrinking gives complicated counterexamples, as shown in Figure 1. Standard QuickCheck shrinking only yields short glitches, which is exemplified in Figure 2. To get the desired output of shrinking, we apply shrinking with VBools instead. As seen in Figure 3, the test case is much simpler but the failure is preserved.

A similar framework is falsification of temporal logic properties, which is a black-box approach to testing of hybrid systems. The tools S-TaLiRo [1] and Breach [4] use Metric Temporal Logic (MTL) and Signal Temporal Logic (STL) respectively to calculate robustness of trajectories, which are minimized by gradient-free optimizers. One key difference between VBools and the similar robust satisfaction of temporal logic is that VBools are meant to be a logic for expressing truth *and* robustness at the same time, rather than robustness being defined after the fact. Of course, the "+" semantics of VBools can also be used to alter the robustness values of MTL or STL formulas, to be used in falsification. Further details on VBools are found in the conference paper by the authors [3].

References

[1] Yashwanth Annpureddy et al. "S-TaLiRo: A tool for temporal logic falsification for hybrid systems". In: *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*. Springer. 2011, pp. 254–257.

[2] Koen Claessen and John Hughes. "QuickCheck: A Lightweight Tool for Random Testing of Haskell Programs". In: *Proceedings of the Fifth ACM SIGPLAN International Conference on Functional Programming*. ICFP '00. New York, NY, USA: ACM, 2000, pp. 268–279. ISBN: 1-58113-202-6. DOI: 10.1145/351240.351266. URL: http://doi.acm.org/10.1145/351240.351266.

[3] Koen Claessen et al. *Using Valued Booleans to Find Simpler Counterexamples in Random Testing of Cyber-Physical Systems*. Accepted to WODES. 2018.

[4] Alexandre Donzé. "Breach, a toolbox for verification and parameter synthesis of hybrid systems." In: *CAV*. Vol. 10. Springer. 2010, pp. 167–170.