Test for Posix Utilities

Mihaela Sighireanu and Ralf Treinen

mihaela.sighireanu@lsv.fr,ralf.treinen@irif.fr

Location: Laboratoire Méthodes Formelles (LMF), ENS Paris-Saclay 4 avenue des Sciences, Gif-sur-Yvette, France
or IRIF, Université de Paris, 8 place Aurelie Némours, 75013 Paris, France.

Context: The Feature Tree Logic (FTL) has been proposed in the research project Colis for the specification of POSIX primitives that transform the file system (e.g., `mv` or `mkdir`) [3]. The specifications obtained in FTL have been used for the verification of installation scripts in Debian [2]. A symbolic execution engine [1] constructs an encoding of the behavior of the script as an FTL formula. This is achieved by unrolling the constructions of the scripting language (for instance POSIX shell) and combining them with the known behavior of the UNIX primitives.

The open question with this approach is in which degree the existing POSIX primitives are conform with our FTL specification. An approach to answer to this question is to employ property-based testing. It consists in extracting test cases (input configurations and expected output) from the specification which are used to test the specified implementation. To apply this approach, one has to be able to extract models from a specification (to obtain the starting configuration for a test), and to check that a model satisfies a specification or model-checking (to test that the result obtained conforms to the specification).

The solver impplemented for the logic FTL, called COLIS-DP, decides satisfiability of quantifier-free FTL formulas, but it does not include a model extraction or a model-checking procedure.

Objective: This internship aims at improving COLIS-DP [4, colis-language/tree/master/src/constraint by adding a model extraction and a model-checking procedures. These procedures shall be useful to generate test cases for the POSIX implementations of primitives dealing with the file system.

Workplan: The intern will start by getting familiar with the logic, the decision procedure proposed and the present version of the solver COLIS-DP. Then, s.he will design and implement the two procedures (model extraction and model-checking). The intern will test her/his implementation on the formulas used for the specification of POSIX primitives.

Requirement: Basic knowledge of first order logic and mastering of OCaml programming.

References

[1] Becker, B., Marché, C.: Ghost Code in Action: Automated Verification of a Symbolic Interpreter. In: Chakraborty, S., A.Navas, J. (eds.) Verified Software: Tools, Techniques and Experiments. Lecture Notes in Computer Science (2019), https://hal.inria.fr/hal-02276257

[2] Becker, B.F.H., Jeannerod, N., Marché, C., Régis-Gianas, Y., Sighireanu, M., Treinen, R.: Analysing installation scenarios of debian packages. In: Biere, A., Parker, D. (eds.) Tools and Algorithms for the Construction and Analysis of Systems - 26th International Conference, TACAS 2020, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2020, Dublin, Ireland, April 25-30, 2020, Proceedings, Part II. Lecture Notes in Computer Science, vol. 12079, pp. 235–253. Springer (2020). https://doi.org/10.1007/978-3-030-45237-7_14, https://doi.org/10.1007/978-3-030-45237-7_14

[3] Jeannerod, N., Régis-Gianas, Y., Marché, C., Sighireanu, M., Treinen, R.: Specification of UNIX utilities. Technical report, HAL Archives Ouvertes (Oct 2019), https://hal.inria.fr/hal-02321691

[4] The CoLiS project: The CoLiS toolchain. https://github.com/colis-anr