Presentation Notes

Ana Nora Evans

January 30, 2017

1 Paper

These notes are prepared for research group meeting discussing the ASE 2016 paper, *Inferring annotations for device drivers from verification histories* [1].

2 Program Verifiers

Model checking is the exhaustive exploration of the state space of a system, typically to see if an error state is reachable. It produces concrete counterexamples.

A program verifier checks if a program satisfies certain properties.

SLAM does verification by model checking. It creates a boolean model of the program and then checks for satisfiability of the formula.

Input:

- standard C program
- Specification (written in SLIC) given as a finite state machine.

Output:

- Verified which means ”program does not violate the given specification” (with a proof).
- counterexample

The Static Driver Verifier Static Driver Verifier [2] is a verification tool included in the Windows Driver Kit (WDK). It uses SLAM. Among other things, SDV provides a number of class-specific components (for example, API rules and an environment model). API rules are expressed in the SLIC language and describe the proper way to use the driver APIs.

Note, SDV is available for academic use.

Corral

Houdini

Annotations are candidate predicates.

Invariants are logical formulas over program variables that are always true.

References

[1] Zvonimir Pavlinovic, Akash Lal, and Rahul Sharma. Inferring annotations for device drivers from verification histories. In *Proceedings of the 31st IEEE/ACM International Conference on Automated Software Engineering*, ASE 2016, pages 450–460, New York, NY, USA, 2016. ACM.

[2] Thomas Ball, Ella Bounimova, Vladimir Levin, Rahul Kumar, and Jakob Lichtenberg. The static driver verifier research platform. In *Proceedings of the 22Nd International Conference on Computer Aided Verification*, CAV'10, pages 119–122, Berlin, Heidelberg, 2010. Springer-Verlag.