

**Title:** Generating Data Race Witnesses by an SMT-based Analysis

**Authors:** Mahmoud Said, Chao Wang, Zijiang Yang, and Karem Sakallah<sup>3</sup>

**Engineering Research Paper Published in:**

**• What is your take-away message from this paper?**

This paper is wanting to improve upon previously existing techniques for detecting data races by giving all possibilities of witnesses/traces a data race could happen in the execution of a program. This improvement is because currently existing tools can't reliably reproduce a data race error and aren't very useful for programmers due to this.

**• What is the motivation for this work (both people problem and technical problem), and its distillation into a research question? Why doesn't the people problem have a trivial solution? What are the previous solutions and why are they inadequate?**

Data races are often very bad to programs and crucial to ensure that none of these errors happen. This is because data races aren't errors in logic and a single problem can take weeks for programmers to identify. The techniques used previously some don't scale and none of them can deterministically help programmers reproduce the data race which makes them unhelpful to debugging.

**• What is the proposed solution (hypothesis, idea, design)? Why is it believed it will work? How does it represent an improvement? How is the solution achieved?**

The writers propose that they use a trace and existing data race detection algorithms to give a set of potential data races. They believe it will work because the set of potential data races is claimed to be a **maximal set** which means this set will always include the possible data races a program can have. This solution is achieved by setting constraints on events such that they are recursive-lock-free, synchronization consistency, and read-write consistency are enforced. If there are recursive locks, the events are encoded differently. Another improvement is using SMT solvers versus previous implementations of enumerating on the feasible interleavings.

**• What is the author's evaluation of the solution? What logic, argument, evidence, artifacts (e.g., a proof-of-concept system), or experiments are presented in support of the idea?**

The authors use proofs and first order logic to prove why their solution is correct in what they want,

and as well as the complexity of the solution. The constraints set on the first order logic equations they use and proof for how the constraints on the equation ensure that all possible witnesses are created yet still optimizes it to some extent. The experiments are also given and they claim that it is scalable to medium length traces. A problem here that is given is that they evaluate with the potential data races don't have concrete witnesses due to the algorithms that detect them. This makes the manual effort for programmers to check very tedious.

- **What is your analysis of the identified problem, idea and evaluation? Is this a good idea? What flaws do you perceive in the work? What are the most interesting or controversial ideas? For work that has practical implications, ask whether this will work, who would want it, what it will take to give it to them, and when might it become a reality?**

I think the idea is good and the evaluation is also solid to some extent. While the tool is scalable to medium size traces, it isn't for large traces. The added problem of witnesses not actually being concrete for many possible data races is only due to the previous tools that are used in conjunction with this one.

- **What are the paper's contributions (author's and your opinion)? Ideas, methods, software, experimental results, experimental techniques...?**

The paper adds a way to improve existing data race detection algorithms to give all possible witnesses for data races and also changes to using SMT solvers and FOL which are faster for time complexity and won't bottleneck the runtime of the algorithms.

- **What are future directions for this research (author's and yours, perhaps driven by shortcomings or other critiques)?**

The authors didn't actually have any future research in the paper, but obviously the evolution aware version of this is a future direction. This evolution-aware will further increase the scalability of the tool and also the programmer effort required to only check the possible data races in new commits.

- **What questions are you left with? What questions would you like to raise in an open discussion of the work (review interesting and controversial points, above)? What do you**

**find difficult to understand? List as many as you can, at least three, not including questions that can be answered quickly by searching the internet.**

How come a large project wasn't used, it doesn't really seem that scalable in their claims since I'm sure people would want to use this for larger traces. Although I don't know the exact ranges for medium vs large traces

Is it possible to pinpoint the more likely witnesses for data races, even if we give developers all possible witnesses and ignore the warnings, it would still be a lot for them to go in by themselves to check up on the output.

What exactly do recursive locks do to a program? Is it infinite depth of an execution trace?