# To be implemented…

* I would have liked to see a table summarizing, which properties the other tools support.
* please explain your methodology w.r.t. "fair failures"
* The paper is unclear about the fact that the authors developed their own benchmark suite. This is only clearly mentioned in the appendix.
* Claiming user assertions and then verifying only two threads is slightly misleading. Assertions about the local state of a thread should still be verifiable, but anything that depends on a global invariant will result into inconclusive, right?
* The statistics are somewhat hard to interpret.
* The missing info about why an incorrect verification is incorrect (incapable or false positive) is missing from the online data summary (spreadsheet) as well.
* The experiments were performed on a CPU with 8 (hyper-) cores. The text does not state whether time is measured as wall time or CPU time, and nothing is said about how many cores were assigned to each tool.
  + Given that the tool 'time' is used, the reader can assume that wall time is meant, because time cannot measure the CPU time of multi-process programs, and given that no other tool is used for controlling the resources, I should assume that the tools were run completely uncontrolled.
  + Furthermore, since no resource limits are given, the runtime and success of a tool also depends on how much memory the operating system currently uses, i.e., the possible max. memory usage is being different for each run, obviously. **Resources (mem, time, cores) must always be specified and enforced**, in order to assign equal resources to all tools.
  + The connection between SMT-based bounded model checking and explicit state-space exploration (bottom of page 1) remains unclear to me.
  + On page 3, "State Hashing": It would be interesting to learn more about this. How is this done? Explain briefly with a hint to the reader or a reference. I know state hashing only from explicit domains, not from symbolic bounded model checking.
  + page 2: approach ... address them -> addresses
  + Section 5: overview ... are shown -> is shown
  + CIVL uses symbolic model checking and maybe you should investigate and clarify the difference with your tool.