

Zoo: A framework for the verification of concurrent OCAML 5 programs using separation logic

ANONYMOUS AUTHOR(S)

The release of OCAML 5, which introduced parallelism into the language, drove the need for safe and efficient concurrent data structures. New libraries like [Saturn](#) aim at addressing this need. From the perspective of formal verification, this is an opportunity to apply and further state-of-the-art techniques to provide stronger guarantees.

We present Zoo, a framework for verifying fine-grained concurrent OCAML 5 algorithms. Following a pragmatic approach, we defined a limited but sufficient fragment of the language to faithfully express these algorithms: ZOO_{LANG}. We formalized its semantics carefully via a deep embedding in the [Rocq](#) proof assistant, uncovering subtle aspects of physical equality. We provide a tool to translate source OCAML programs into ZOO_{LANG} syntax inside [Rocq](#), where they can be specified and verified using the [Iris](#) concurrent separation logic. To illustrate the applicability of Zoo, we verified a subset of the standard library and a collection of fine-grained concurrent data structures from the [Saturn](#) and [Eio](#) libraries.

In the process, we also extended OCAML to more efficiently express certain concurrent programs.

References