Improving Detection of Data Races and Misuses of Lock-Free Queues via Semantics

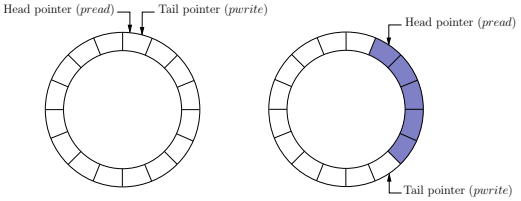
Manuel F. Dolz,[1] David del Rio Astorga,[1] Javier Fernández,[1] J. Daniel García,[1]
Félix García-Carballeira,[1] Marco Danelutto,[2] and Massimo Torquati[2]

[1]*Department of Computer Science, University Carlos III of Madrid, 28911–Leganés, Spain.*
[2]*Department of Computer Science, University of Pisa, 56127–Pisa, Italy.*

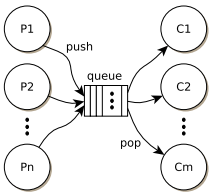
ABSTRACT

Introduction. In the advent of concurrent programming, frameworks offering structured patterns have alleviated developers' burden adapting such applications to multithreaded architectures. While some of these patterns are implemented using synchronization mechanisms, others avoid them using lock-free data structures. However, lock-free programming is not straightforward: ensuring an appropriate use of their interfaces can be challenging, since data races can occur. The benefits of race detectors are formidable in this sense, nevertheless they may emit false positives and are not able to detect misuses of lock-free structure semantics. To face this problem, we formulate the semantics of two lock-free data structures: the Single Producer/Single-Consumer (SPSC) and the Multiple-Producer/Multiple-Consumer (MPMC) queues. Next, we implement them into ThreadSanitizer (TSan) in order to drop false positives and detect semantic violations.

Semantics. We describe formally the bounded and unbounded SPSC and the MPMC queues along with their semantics for the concurrent lock-free version. These definitions allows us to proceed further with our rationale in order to develop rules that guarantee the proper use among entities of these lock-free parallel structures.



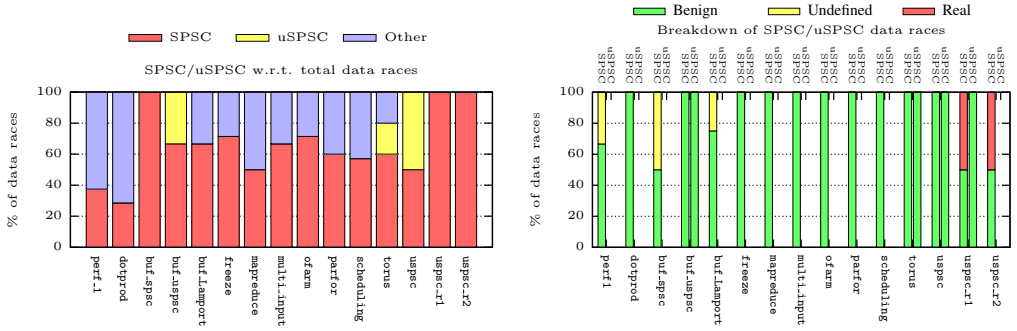
(a)SPSC queue circular buffer.



(b)MPMC queue schema.

Motivation. We use semantics to improve the detection of data races within TSan of lock-free queue structures: we found that TSan has some shortcomings when dealing with this kind of data types, since it is not capable of determining whether a data race is real or not, nor determining if there has been a misuse in a lock-free queue.

Evaluation. We perform an experimental evaluation of the semantics implemented for the SPSC and MPMC lock-free queues into TSan. To evaluate the SPSC queue semantics, we employ a series of μ-benchmarks and real applications of FastFlow using two versions of the SPSC lock-free queue implementation: the bounded and unbounded variants. For the case of the MPMC lock-free queue semantics, we use four different implementations from the state-of-the-art and used within different synthetic μ-benchmarks.



Conclusions. We provide two novel features: *i)* filtering data race warnings classified as false positives, and *ii)* detecting misuses via semantics of such lock-free data structures. The ability of filtering out false positives at runtime is a very helpful feature to prevent overwhelming due to warning reports. In a similar way, detecting violations of the semantics, even when data races are not detected, can greatly aid developers to debug applications. Through these extensions we demonstrate that we are able to discard, on average 60 % of data races classified as false positives. We also observe that some wrong uses of lock-free data structures cannot be detected with a race detector, but via semantics.