Internship proposal for Tristan Charrier

Jade Alglave

University College London

Most multiprocessors exhibit subtle weak memory behaviour, with writes from one thread not immediately visible to all others; they do not provide sequentially consistent (SC) memory [8]. For some, such as Intel x86 [2] and IBM Power [1], the vendor documentation is in inevitably ambiguous informal prose, leading to confusion.

Recent progress has been made towards the formal specification of the behaviour of multiprocessors, e.g. [9, 5]. These formal foundations allow us to step towards verification of concurrent systems code running on top of deployed multiprocessors e.g. [3, 4].

But much remains to be done. A classical path to verify software is to use a program logic, such as Hoare logic [7]. Recent work by Richard Bornat shows that such a logic is possible for weak memory [6].

Yet, this logic takes an unusual form, with the proofs being underpinned by a graph structure which describes the execution paths of the program. This unusual form asks all the more for tool support: we crucially need to develop algorithms and tools to automate the discovery of proofs in this new logic. This is what I would like Tristan to work on during his internship.

References

1. *Power ISA Version 2.06*. January 2009.
2. Intel 64 and IA-32 Architectures Developer's Manual, rev. 30, March 2009.
3. J. Alglave, D. Kroening, V. Nimal, and M. Tautschnig. Software Verification for Weak Memory via Program Transformation. In *ESOP 2013*.
4. J. Alglave, D. Kroening, and M. Tautschnig. Partial Orders for Efficient BMC of Concurrent Software. In *CAV*, 2013.
5. J. Alglave, L. Maranget, S. Sarkar, and P. Sewell. Fences in Weak Memory Models (Extended Version). In *FMSD*, 2012.
6. R. Bornat. Abducing barriers. 2012.
7. T. Hoare. An axiomatic basis for computer programming. In *CACM 12(10): 576-580 (1969)*.
8. L. Lamport. How to Make a Correct Multiprocess Program Execute Correctly on a Multiprocessor. *IEEE Trans. Comput.*, 1979.
9. S. Owens, S. Sarkar, and P. Sewell. A better x86 model: x86-TSO. In *TPHOL 09*.