Abstract

I present a potential example of the Hobbit tool (Higher Order Bounded BIsimulation Tool) by compiling a subset of the Rust programming language into the ML-like language used by Hobbit. When realised, this compilation can serve as a method of comparing practical code segments for program equivalence. This was done by taking a subset of Rust and proceeding to create a mapping and compilation tool, named rust2bils into the verification language. Examples and benchmarks within the compiler tool subset that can be applied to the tool were tested and verified with the Hobbit tool, showing that it succeeds in compiling semi-practical code segments. I also sketched out how the compiler tool subset can be extended with future work, even to a state where it can potentially be applied to more practical systems.