Abstract: SMT-Friendly Formalization of the Solidity Memory Model

Ákos Hajdu and Dejan Jovanović

Abstract

Solidity is the dominant programming language for Ethereum smart contracts. This paper presents a high-level formalization of the Solidity language with a focus on the memory model. The presented formalization covers all features of the language related to managing state and memory. In addition, the formalization we provide is effective: all but few features can be encoded in the quantifier-free fragment of standard SMT theories. This enables precise and efficient reasoning about the state of smart contracts written in Solidity. The formalization is implemented in the solc-verify verifier and we provide an extensive set of tests that covers the breadth of the required semantics. We also provide an evaluation on the test set that validates the semantics and shows the novelty of the approach compared to other Solidity-level contract analysis tools.

François Bobot, Tjark Weber (eds.); SMT 2020, pp. 59–59