Bartoletti, Massimo; Lande, Stefano; Murgia, Maurizio; Zunino, Roberto

Verifying liquidity of recursive Bitcoin contracts. (English) Zbl 07471711


Summary: Smart contracts – computer protocols that regulate the exchange of crypto-assets in trustless environments – have become popular with the spread of blockchain technologies. A landmark security property of smart contracts is liquidity: in a non-liquid contract, it may happen that some assets remain frozen, i.e. not redeemable by anyone. The relevance of this issue is witnessed by recent liquidity attacks to Ethereum, which have frozen hundreds of USD millions. We address the problem of verifying liquidity on BitML, a DSL for smart contracts with a secure compiler to Bitcoin, featuring primitives for currency transfers, contract renegotiation and consensual recursion. Our main result is a verification technique for liquidity. We first transform the infinite-state semantics of BitML into a finite-state one, which focusses on the behaviour of a chosen set of contracts, abstracting from the moves of the context. With respect to the chosen contracts, this abstraction is sound, i.e. if the abstracted contract is liquid, then also the concrete one is such. We then verify liquidity by model-checking the finite-state abstraction. We implement a toolchain that automatically verifies liquidity of BitML contracts and compiles them to Bitcoin, and we assess it through a benchmark of representative contracts.

MSC:
03B70 Logic in computer science
68-XX Computer science

Keywords:
Bitcoin; BitML; blockchain; smart contracts; liquidity; verification

Full Text: arXiv Link

References:


