Institut für Informationssicherheit und Verlässlichkeit (KASTEL)
Anwendungsorientierte Formale Verifikation
Prof. Dr. Bernhard Beckert

Relational Verification of Smart Contracts

Project Group "Praxis der Forschung" – Winter Term 2021/22

1 Motivation

Ethereum is a platform for *smart contracts*: Programs which automatically manage resources in a decentralised network. Changing an Ethereum smart contract directly (e.g., to fix a bug or improve performance) is not possible: The contract has to be deleted, and a new contract can be deployed instead. But how to convince the stakeholders that the new version does the right thing? Does it really only fix a bug? Does it really behave identically except for improved performance? In a smart contract network, different agents do not necessarily trust each other - so we better had proof!

2 Project Description

Formal verification for smart contracts is a very active research area. However, there is no established methodology for comparing the behavior of two different smart contracts.

Ethereum smart contracts are usually written in the Solidity programming language. In this project, the goal is to develop an approach for *Relational Verification* of smart contracts written in Solidity.

3 Requirements

Understanding of first order and temporal logic as taught, e.g., in the *Formale Systeme* lecture.

4 Contact / Supervision

Jonas Schiffl, jonas.schiffl@kit.edu, Room 226 (Geb. 50.34)