Simplicity: A New Language for Blockchains

Russell O'Connor
roconnor@blockstream.com

2017-12-13

Abstract

Simplicity is a typed, combinator-based, functional language without loops and recursion, designed to be used for crypto-currencies and blockchain applications. It aims to improve upon existing crypto-currency languages, such as Bitcoin Script and Ethereum's EVM, while avoiding some of the problems they face. Simplicity comes with formal denotational semantics defined in Coq, a popular, general purpose software proof assistant. Simplicity also includes operational semantics that are defined with an abstract machine that we call the Bit Machine. The Bit Machine is used as a tool for measuring the computational space and time resources needed to evaluate Simplicity programs. Owing to its Turing incompleteness, Simplicity is amenable to static analysis that can be used to derive upper bounds on the computational resources needed, prior to execution. While Turing incomplete, Simplicity can express any finitary function, which we believe is enough to build useful "smart contracts" for blockchain applications.

0 License

1 Introduction

Blockchain and distributed ledger technologies define protocols that allow large, ad-hoc, groups of users to transfer value between themselves without needing to trust each other or any central authority. Using public-key cryptography, users can sign transactions that transfer ownership of funds to other users. To prevent transactions from conflicting with each other, for example when one user attempts to transfer the same funds to multiple different users at the same time, a consistent sequence of blocks of transactions is committed using a proof of work scheme. This proof of work is created by participants called miners. Each user verifies every block of transactions; among multiple sequences of valid blocks, the sequence with the most proof of work is declared to be authoritative.