Mi-Cho-Coq, a framework for certifying Tezos Smart Contracts

Bruno Bernardo, Raphaël Cauderlier, Zhenlei Hu, Basile Pesin, and Julien Tesson

Nomadic Labs, Paris, France
{first.last}@nomadic-labs.com

Abstract. We present Mi-Cho-Coq, a Coq library for verifying the functional correctness of smart contracts running in the Tezos blockchain.

Keywords: Certified programming · Programming languages · Blockchains · Smart contracts.

Introduction

Blockchains are distributed databases that can be updated in a decentralised way. Each node of the network stores a copy of the database and updates it by validating new blocks. A consensus algorithm helps ensuring that nodes agree on the choice of new blocks without any central entity. Blockchains are typically used as public ledgers for crypto-currencies such as Bitcoin [10], Ethereum [5,14], Tezos [8,9,3], etc.

Smart contracts, popularised by Ethereum which pushed further the concept introduced by Bitcoin's scripts [1], are programs associated to an account in a blockchain. Transactions made to an account containing a smart contract will trigger the execution of this program. Accounts of a public ledger stored in a blockchain thus become programmable, making possible many use cases such as crowdfunding, votes, auctions, etc.

Blockchains manage assets with a significant value, making them a target of choice for attackers willing to exploit vulnerabilities, particularly those that occur in smart contracts. One famous example is the attack of the DAO contract [13] where a reentrancy attack moved about a third of the contract's funds (worth about $15M out of $50M, estimations at the time of the attack). This attack was *canceled* by a vast majority of the block validators who agreed on rewriting the history of the chain. This *hardfork* of the Ethereum blockchain was very controversial and divided the Ethereum community.

1 Tezos and Michelson

Tezos is a blockchain launched in June 2018. It is written in OCaml and supports smart contracts. It also has an explicit on-chain governance process: token holders can vote for changes to the code.

The smart contract language of Tezos, called Michelson, has been designed with formal verification in mind. It is stack based, statically typed with a rich type system (options, products, sums, lists, sets and maps) and a mixture of low level (stack manipulation, arithmetic, etc.) and high level (cryptographic primitives, data manipulation, etc.) instructions. Its syntax, type system, and semantics are clearly documented [2]. Furthermore, the Michelson interpreter, run by every node of the Tezos network, is implemented as an OCaml GADT, which gives subject-reduction for free.

2 Mi-Cho-Coq

Formally verifying smart contracts is extremely relevant given the financial impact that bugs in them can have and because they are usually small programs.

Mi-Cho-Coq [11] is a Coq framework for verifying the functional correctness of smart contracts of the Tezos blockchain. It consists of an implementation of a Michelson interpreter as well as a weakest precondition calculus à la Dijkstra [7].

The abstract syntax tree of a Michelson program is represented as inductive type indexed by the types of the input and output stacks. [1] Similarly to the OCaml's implementation with GADTs, this gives us subject reduction for free. Notations are used so that contracts in Mi-Cho-Coq look very much like proper Michelson contracts.

Michelson semantics is formalised by an evaluator that takes as an input an instruction, a fuel value, an input stack and returns an output stack inside an error monad so that explicit failure can be represented. Some domain specific instructions that are hard to define (cryptography, serialisation, query to the execution environment) are axiomatised.

In order to make it easier to write correctness proofs in Mi-Cho-Coq, a weakest precondition calculus is defined through a Coq function that takes as input an instruction and a predicate over the possible output stacks (postcondition) and returns a predicate about the possible input stacks (precondition). This function is proved correct in regards with the evaluator.

A portfolio of verified contracts is under development. One interesting example is the verification of a *multisig* contract [4,6], that is a contract whose ownership is split between several members: a subset of the owners need to agree on some action for it to take place. We have also verified a smart contract for voting [12], where users can make transactions to choose among a predefined set of choices.

Mi-Cho-Coq is a work in progress and we are planning to add the Michelson cost model to it. We would also like to replace the current OCaml implementation of the Michelson interpreter in the Tezos codebase with an extracted version of the interpreter implemented in Mi-Cho-Coq. Furthermore, the formalised Michelson semantics in Mi-Cho-Coq could be used as a certified compilation target for higher level languages.

References

1. Bitcoin script wiki. `https://en.bitcoin.it/wiki/Script`
2. Michelson: the language of Smart Contracts in Tezos. `http://tezos.gitlab.io/mainnet/whitedoc/michelson.html`
3. Tezos developer documentation. `https://tezos.gitlab.io`
4. Breitman, A.: Multisig contract in Michelson. `https://github.com/murbard/smart-contracts/blob/master/multisig/michelson/generic_multisig.tz`
5. Buterin, V.: Ethereum: A next-generation smart contract and decentralized application platform. `https://github.com/ethereum/wiki/wiki/White-Paper` (2013)
6. Cauderlier, R.: Certification of Breitman's Multisig implementation in Mi-Cho-Coq. `https://gitlab.com/nomadic-labs/mi-cho-coq/blob/master/src/contracts_coq/multisig.v`
7. Dijkstra, E.W.: Guarded commands, nondeterminacy and formal derivation of programs. Commun. ACM **18**(8), 453–457 (Aug 1975). https://doi.org/10.1145/360933.360975
8. Goodman, L.M.: Tezos: A self-amending crypto-ledger. position paper. `https://tinyurl.com/tezospp` (2014)
9. Goodman, L.M.: Tezos: A self-amending crypto-ledger. white paper. `https://tinyurl.com/tezoswp` (2014)
10. Nakamoto, S.: Bitcoin: A peer-to-peer electronic cash system. `https://bitcoin.org/bitcoin.pdf` (2008)
11. Nomadic Labs: Mi-Cho-Coq public repository. `https://gitlab.com/nomadic-labs/mi-cho-coq`
12. Pesin, B.: Certification of a voting smart contract in Mi-Cho-Coq. `https://gitlab.com/nomadic-labs/mi-cho-coq/blob/master/src/contracts_coq/vote.v`
13. Vessenes, P.: `https://vessenes.com/deconstructing-thedao-attack-a-brief-code-tour/` (2016)
14. Wood, G.: Ethereum: a secure decentralised generalised transaction ledger. `http://gavwood.com/paper.pdf` (2014)

[1] Stacks are implemented by lists and their types are lists of the types of their elements.