Checking the Unchecked: Demonstrating Correct Behaviour of the Cardano Blockchain node using Lightweight Formal Methods

Kevin HAMMOND[1], Ulf NORELL[2], Maximilian ALGEHED[2], James CHAPMAN[1], and Thomas ARTS[2]

[1] Input Output Global, UK
kevin.hammond@iohk.io, james.chapman@iohk.io

[2] QuviQ AB, Göteborg, Sweden
ulf.norell@quviq.com, maximilian.algehed@quviq.com, thomas.arts@quviq.com

Abstract. This talk introduces Cardano, a top-10 blockchain that has been constructed using Haskell and lightweight formal methods. A key part of the blockchain's functionality is based around a set of around 100 formal ledger rules, which have been formulated using a structured transition system (yielding a small-step operational semantics).

The talk will describe the formal specification of the Cardano ledger in Agda, highlighting key properties (specifically preservation of value), showing how the corresponding Haskell implementation can be shown to conform to the formal specification, and discussing how new QuickCheck Dynamic capabilities can be used to demonstrate correct behaviour under both normal and adversarial conditions.

Keywords: Distributed Systems · Cardano Blockchain · Formal Specification · Property Testing · QuickCheck Dynamic · Formal Methods · Conformance Testing · Functional Programming · Haskell · Agda

1 The Cardano Blockchain and its Implementation

Cardano [6] is a top-10 blockchain, with a total market capitalisation in the tens of billions of dollars, that is designed to work in a highly decentralised way, using novel proof-of-stake consensus protocols. The Cardano network comprises around 3,000 active block producing nodes, which are globally distributed and which cooperate to produce new blocks to extend the chain. These core nodes are supported by other nodes that act as relays. Wallets, explorers and other blockchain applications (dApps) connect to these relay nodes to follow the chain and extract information from it.

The Cardano node [5] is implemented in Haskell and comprises around 500K lines of code in several modules. The ledger code forms the heart of this codebase [4], comprising about 200K lines of code that define a structured transition system, transforming blockchain inputs into unpsent transaction outputs (UTxO) under the guidance of specific transactions. Each new block then comprises a group of (generally unrelated) transactions that build on the previous blocks in the chain in a cryptographically secure way following the rules of the Ouroboros Praos proof-of-stake protocol [1]. This talk will focus on these ledger rules and the properties of the associated transition system.

1.1 Demonstrating Correctness

Our approach is based around a new formalisation of the ledger rules in Agda [2] that is associated in a systematic way with the underlying Haskell implementation. There are three main areas of concern for the ledger rules:

1. That major semantic properties hold, the most important of which is that value is preserved, *i.e.* that tokens are not created or destroyed by the ledger operations.
2. That the implementation conforms to the formal specification.
3. That the design performs correctly in the presence of adversarial as well as honest actors.

Each of these is dealt with in a different way:

1. We construct proofs at the Agda level to demonstrate the correctness of the most important semantic properties.
2. We generate Haskell code from the Agda sources to give a reference implementation of the ledger rules, and generate property tests (*conformance tests*) that show that the behaviour of the actual implementation conforms to this reference implementation.
3. We use QuickCheck Dynamic [3] to run the reference implementation against simulated adversarial behaviour.

2 Talk Outline

The talk will:

1. introduce Cardano and the architecture of the Cardano node;
2. describe the formal specification of (part of) the ledger code in Agda;
3. show how key properties can be proved in Agda;
4. give examples to show how the reference implementation in Haskell is generated from the Agda formal specification;
5. describe the conformance testing process, with examples;
6. show how QuickCheck Dynamic can be used to determine that the system behaves correct under both normal and adversarial behaviour;
7. draw some general conclusions relating to the approach that has been used and improvements that could be made in future.

Acknowledgments. The work described in this paper was funded by Input Output Global.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. David, B., Gaži, P., Kiayias, A., Russell, A.: Ouroboros Praos: An Adaptively-Secure, Semi-synchronous Proof-of-Stake Blockchain. In: Nielsen, J.B., Rijmen, V. (eds.) Advances in Cryptology – EUROCRYPT 2018. pp. 66–98. Springer International Publishing, Cham (2018)
2. Knispel, A., Melkonian, O., Chapman, J., Hill, A., Jääger, J., DeMeo, W., Norell, U.: Formal Specification of the Cardano Blockchain Ledger, Mechanized in Agda. In: Bernardo, B., Marmsoler, D. (eds.) 5th International Workshop on Formal Methods for Blockchains (FMBC 2024). Open Access Series in Informatics (OASIcs), vol. 118, pp. 2:1–2:18. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany (2024). https://doi.org/10.4230/OASIcs.FMBC.2024.2, https://drops.dagstuhl.de/entities/document/10.4230/OASIcs.FMBC.2024.2
3. QuViQ AB: QuickCheck Dynamic (2024), https://hackage.haskell.org/package/quickcheck-dynamic,
4. The Cardano Development Team: The cardano ledger (2024), https://github.com/IntersectMBO/cardano-ledger, the Intersect Members Based Organisation
5. The Cardano Development Team: The Cardano Node (2024), https://github.com/IntersectMBO/cardano-node, the Intersect Members Based Organisation
6. The Cardano Foundation: The Cardano Blockchain (2024), https://cardano.org,