Thesis proposal for the degree of Engineering in Informatics

Formal verification of financial smart contracts in Marlowe for the Cardano blockchain

Advisor: Phd. Mariano G. Beiró

Co-Advisor: Phd. Simon Thompson (Kent University, IOHK)

Alumno: Julián Ferres, *(Padrón #101.483)*
`jferres@fi.uba.ar`

Facultad de Ingeniería, Universidad de Buenos Aires

May 16, 2022

Contents

0.1 Introduction

Blockchains are data structures in which the information is divided into sets (the blocks) that contain additional information relative to previous blocks in the chain. With this organization and some cryptographic techniques, the information inside a block can only be altered by modifying all preceding blocks. This property enables their deployment as a distributed system, in such a way that the chain of blocks can model a non-relational public database, which contains an irrefutable historical record of information. In practice, blockchains enable the implementation of a distributed ledger that supports and guarantees the security of transactions and digital assets. The concept of blockchain was first applied in 2009 as a core part of Bitcoin.

Another interesting feature of blockchains is their support of smart contracts. A contract is a legally binding agreement (loans, sales, tenancy, etc). A smart contract forces the achievement of what is agreed in it through a software-ensured guarantee, in which none of the parties involved can sabotage or modify the contract. This is achieved by immortalizing both the program and its results in the blockchain ledger underlying the contract, thus ensuring that all the history (including the current status of the contract) is recorded with a high degree of reliability. From the perspective of the author of the smart contract, the blockchain is a distributed ledger system, which keeps track of who owns the amount of a virtual asset (Bitcoin, Ada, etc.) and which assets are transferred from one entity to another. The owners of digital assets are identified by their public keys, and they may be people or machines.

Given the relevance of smart contracts supporting a wide range of activities in every sector of industry, including supply-chains, finance, legal and medical services, there is a strong demand for verification and validation techniques over them. However, the vast majority of smart contracts lack of any kind of formal specification, which is essential to establish their correctness.

In this paper we propose the study of low-level verification of a specific group of financial contracts defined in the ACTUS standard[1], in particular for the Cardano blockchain[2]. We will use the Plutus platform, which is a software development kit (SDK) built into Haskell, so that developers can write smart contracts, including the logic that will eventually run on Cardano. The advantage behind the Plutus approach is the guarantee of security in general, and for smart contracts in particular, this is due to the usage of functional programming with advanced type systems.

This work will be co-supervised by Prof. Simon Thompson (Kent University) who is a Senior Research Fellow at IOHK and specializes in verification techniques. The properties that we will verify are defined, for each type of contract, within the ACTUS standard[3]. The proofs will be written in the Isabelle/HOL proof assistant. The contracts developed in Marlowe will also be integrated into the Marlowe Playground developed by IOHK, allowing the automatic generation of Marlowe code from the definition of the contract conditions.

0.2 State of the art

Currently, smart contracts are supported by several blockchains, such as Ethereum, Steller, EOS and Cardano. Ethereum was the first platform enabled for smart contracts, and it still remains the most popular option for developers. It was publicly released in 2015 and supports the deployment of a wide range of contracts, from “Initial Coin Offer” to smart-contract-based insurance.

As noted in [Permenev et al., 2020], smart contracts –as well as other systems where security is critical, suchs as controllers in cars and airplanes–, must be formally verified before deployment. While this observation is not recent, only a handful of smarts contracts projects (e.g, MakerDAO[4]) have been formally verified so far. Current verification efforts are carried out using theorem proff assintants, such as Isabelle/HOL. This requires manual effort and experience. As an example, [Bhargavan et al., 2016], verified some simple contracts written in Solidity and compiled into EVM bytecode. In order to achieve this, the bytecode was translated into F* (a functional programming language, inspired by ML and whose main purpose is program verification).

Cardano is currently integrating smart contracts functionalities into the blockchain, being at the beginning of what is called its “Goguen Era”[5], the third of five stages in the project roadmap. The stages are:

1. Byron Era: Foundation and implementation of the ledger.
2. Shelley Era: Multi-Asset Capability and Decentralization.

[1] https://www.actusfrf.org/
[2] https://cardano.org/
[3] https://www.actusfrf.org/techspecs
[4] Formal verification of multicollateral dai, 2019, [online] Available at: https://github.com/dapphub/k-dss/.
[5] https://roadmap.cardano.org/en/goguen/

3. Goguen Era: Smart contracts.
4. Basho Era: Optimization and scalability.
5. Voltaire Era: Governance.

The Goguen era also includes work to make Cardano accessible to a wider audience via *Marlowe*[6], a new domain specific language (DSL) designed for writing financial smart contracts. Marlowe allows contracts to be written in a financial language, rather than using general-purpose blockchain languages, although the code it generates is Haskell code. It is currently still under development. When fully operational, it will allow organizations to write their own contracts or download the contracts already made from the repositories, and transfer assets according to agreed conditions.

Currently Marlowe contracts are written directly in Haskell, Javascript, or through the syntax introduced by Marlowe, and can be visualized using the Marlowe Playground, where it is also possible to simulate and analyze them. In the next few months the implementation of Marlowe in Cardano will conclude, and in this way the contracts will be executed directly on the Blockchain.

The work up to date regarding contract verification for Marlowe is summarized in the papers [Lamela Seijas and Thompson, 2018], [Lamela Seijas et al., 2020a] and [Lamela Seijas et al., 2020b].

In [Lamela Seijas et al., 2020a], the Isabelle/HOL verifier was used to test some basic properties of the generated contracts[7]:

- Balance of the assets involved in the contract.
- Return of unspent resources to their original owners, at the end of the contract.
- Termination of the contract.

In [Lamela Seijas et al., 2020b] some optimizations are proposed –based on the Haskell-specific syntax of the contracts generated by Marlowe– to make static analysis using SBV library more efficient. SBV is a tool used to express properties intended to be verified as functions in Haskell, and it also performs the conversion to symbolic language so they can be tested with an SMT solver (in this paper, Z3). To analyze the optimizations performance, the *Auction*, *Crowdfunding*, *Rent* and *Coupon Bond* financial contracts were benchmarked and a significant improvement was found in proof times.

With regard to validation and verification techniques, in the survey by [Tolmach et al., 2021] different formal models and specifications of smart contracts of the literature are analysed and classified and an extensive and systematic description is presented. Current approaches to verify specifications are also discussed. In particular, Isabelle/HOL is mentioned as one

[6] https: //alpha.marlowe.iohkdev.io

[7] Marlowe github (2018). https://github.com/input-output-hk/marlowe. Accessed 6 Aug 2021

of the common theorem proving's tools for the formal semantics of low, medium and high levels smart contract languages, including domain specific languages (DSL) as is the case in [Lamela Seijas et al., 2020a]. This is the proof assistant that we will use for verification.

0.3 Goals

The main goal of this work is to study the incorporation of ACTUS smart contracts to the Cardano blockchain, verifying that the implementations fulfill the properties specified in the ACTUS standard for those contracts.

The specific goals are:

1. To design a translation mechanism for the selected contracts in the ACTUS formal specification into Marlowe code.
2. To verify the produced Marlowe contracts using the proof assistant Isabelle/HOL.
3. To incorporate the designed contracts into the Actus Labs website, so that the contracts can be created interactively from a simple block model.

0.4 Work plan

We propose the following schedule for the development of this thesis:

Tasks	Months									
	1	2	3	4	5	6	7	8	9	10
Bibliography exploration	■	■								
Plutus Pioneers Program		■	■							
Development			■	■	■	■	■	■		
Manuscript								■	■	■

This thesis has an estimated load of 750 working hours, plus 120 study hours to follow the Plutus Pioneers Program course.

- Bibliography exploration [120 hours]: Reading articles, scientific publications and literature on which this study is based.
- Plutus Pioneers Program [120 hours]: Course offered by Cardano Developers, 2 months long, whose aim is to train developers for the Cardano ecosystem.

- Development [450 hours]:
  - Translating the formal specification of the selected Actus contracts into Marlowe. [100 hours]
  - Proofs writing in Isabelle/HOL, to validate the properties of each contract specified in the Actus standard. [200 hours]
  - Integration of the new contracts to the interactive Actus Labs platform. [150 hours]
- Manuscript writing [180 hours].

Bibliography

[Bhargavan et al., 2016] Bhargavan, K., Delignat-Lavaud, A., Fournet, C., Gollamudi, A., Gonthier, G., Kobeissi, N., Kulatova, N., Rastogi, A., Sibut-Pinote, T., Swamy, N., and Zanella-Béguelin, S. (2016). Formal Verification of Smart Contracts: Short Paper. In *ACM Workshop on Programming Languages and Analysis for Security*, Vienna, Austria.

[Lamela Seijas et al., 2020a] Lamela Seijas, P., Nemish, A., Smith, D., and Thompson, S. (2020a). Marlowe: Implementing and analysing financial contracts on blockchain. In Bernhard, M., Bracciali, A., Camp, L. J., Matsuo, S., Maurushat, A., Rønne, P. B., and Sala, M., editors, *Financial Cryptography and Data Security*, pages 496–511, Cham. Springer International Publishing.

[Lamela Seijas et al., 2020b] Lamela Seijas, P., Smith, D., and Thompson, S. (2020b). Efficient static analysis of marlowe contracts. In Margaria, T. and Steffen, B., editors, *Leveraging Applications of Formal Methods, Verification and Validation: Applications*, pages 161–177, Cham. Springer International Publishing.

[Lamela Seijas and Thompson, 2018] Lamela Seijas, P. and Thompson, S. (2018). Marlowe: Financial contracts on blockchain. In Margaria, T. and Steffen, B., editors, *Leveraging Applications of Formal Methods, Verification and Validation. Industrial Practice*, pages 356–375, Cham. Springer International Publishing.

[Permenev et al., 2020] Permenev, A., Dimitrov, D., Tsankov, P., Drachsler-Cohen, D., and Vechev, M. (2020). Verx: Safety verification of smart contracts. In *2020 IEEE Symposium on Security and Privacy (SP)*, pages 1661–1677.

[Tolmach et al., 2021] Tolmach, P., Li, Y., Lin, S.-W., Liu, Y., and Li, Z. (2021). A survey of smart contract formal specification and verification. *ACM Comput. Surv.*, 54(7).