Propuesta de Tesis de Grado de Ingeniería en Informática

Verificación de smart contracts en Marlowe para la blockchain Cardano

Director: Dr. Ing. Mariano G. Beiró
mbeiro@fi.uba.ar

Co-director: Phd. Simon Thompson (Kent University, IOHK)
S.J.Thompson@kent.ac.uk

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

Facultad de Ingeniería, Universidad de Buenos Aires

2 de mayo de 2022

Índice general

0.1. Introducción

Las cadenas de bloques, conocidas en inglés como *blockchains*, son estructuras de datos en las cuales la información se divide en conjuntos (bloques) que cuentan con información adicional relativa a bloques previos de la cadena. Con esta organización relativa, y con ayuda de técnicas criptográficas, la información de un bloque solo puede ser alterada modificando todos los bloques posteriores. Esta propiedad facilita su aplicación en un entorno distribuido, de manera tal que la cadena de bloques puede modelar una base de datos pública no relacional, que contenga un registro histórico irrefutable de información. En la práctica esta técnica ha permitido la implementación de un registro contable o *ledger* distribuido que soporta y garantiza la seguridad de transacciones y dinero digital. El concepto de cadena de bloque fue aplicado por primera vez en 2009 como parte central de Bitcoin.

Otra característica interesante de las cadenas de bloques es que soportan la definición de contratos inteligentes (*"smart contracts"*). Un contrato es un acuerdo legalmente vinculante (por ejemplo, sobre un préstamo, venta, arrendamiento, etc). Un contrato inteligente permite forzar el cumplimiento de lo pactado en el mismo a través de una garantía asegurada por software, en la cual ninguna de las partes involucradas puede sabotear o alterar el contrato. De esta forma se puede renunciar a la toma de acciones legales por parte de un individuo, empresa o gobierno, y en su lugar optar por la ejecución de un programa, o contrato inteligente, para controlar la transferencia de fondos entre los participantes. Este objetivo se logra inmortalizando tanto el programa como su resultado en el ledger de la cadena de bloques subyacente al contrato, y garantizando así que todo el historial (incluido el estado actual del contrato) se registre de forma inmutable con un alto grado de fiabilidad. Desde la perspectiva del autor del contrato inteligente, la blockchain es un sistema de contabilidad distribuido, que realiza un seguimiento de quién posee una cantidad de un recurso virtual (Bitcoin, Ada, etc.) y cuándo los activos se transfieren de una entidad a otra. Los propietarios de activos digitales se identifican por sus

claves públicas, y pueden ser personas o máquinas.

Dada la importancia de los smart contracts para respaldar actividades en todos los sectores de la industria, incluyendo cadenas de abastecimiento, finanzas, servicios legales y médicos, existe una fuerte demanda de verificación y técnicas de validación sobre los mismos. Sin embargo, la gran mayoría de los contratos inteligentes carecen de cualquier tipo de especificación formal, que es esencial para establecer que el mismo es correcto.

En este trabajo nos proponemos estudiar la verificación a bajo nivel de un grupo específico de contratos financieros definidos en el estándar ACTUS[1], en particular para la cadena de bloques *Cardano*[2]. Utilizaremos la plataforma *Plutus*, que es un kit de desarrollo de software (SDK) integrado en Haskell, para que los desarrolladores puedan escribir contratos inteligentes para Cardano, incluida la lógica que eventualmente se ejecutará en la cadena de bloques. La ventaja detrás del enfoque de Plutus es la garantía de seguridad en general, y en particular en contratos inteligentes, debido al uso de programación funcional con sistemas de tipado avanzado.

El trabajo contará con la co-dirección del profesor Simon Thompson (Kent University, (`https://scholar.google.com/citations?user=twtKi84AAAAJ&hl=es`)) quien se desempeña como Senior Research Fellow en IOHK, a cargo del desarrollo de Cardano, y se especializa en técnicas de verificación. Como parte del trabajo, se buscaran las propiedades a verificar sobre los contratos, algunas de ellas derivan directamente de la especificación formal definida por el estándar ACTUS[3], mientras que otras resultan mas abstractas o generales. Las pruebas se escribirán a través del asistente de pruebas Isabelle/HOL. Los contratos serán desarrollados en Marlowe (un lenguaje construido sobre Plutus) y también se integrarán al Marlowe Playground en la web, para permitir la generación automática del código Marlowe a partir de la definición de las condiciones de contrato.

0.2. Estado del arte

Actualmente, varias de las blockchains existentes soportan contratos inteligentes, tales como Ethereum, Steller, EOS y Cardano. Ethereum es la primer plataforma para contratos inteligentes, y permanece como la opción más popular para los desarrolladores. La plataforma se lanzó al público en 2015 y ahora soporta y facilita el desarrollo de aplicaciones desde *"Initial Coin Offering"* hasta seguros basados en smart contracts.

Como se señala en [Permenev et al., 2020], los smart contracts, así como otros sistemas en que la seguridad es crítica (por ejemplo, controladores en autos y aviones), deben ser formalmente verificados antes de su despliegue. Si bien esta observación no es reciente, solo un puñado de proyectos de smarts contracts (por ejemplo, MakerDAO[4]) se han verificado formalmente hasta ahora. Los esfuerzos de verificación actuales se llevan a cabo utilizando verificadores de

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

teoremas interactivos, como Isabelle/HOL. Estorequiere de un esfuerzo manual y experiencia. Un antecedente que encontramos es [Bhargavan et al., 2016], en donde se analizan y verifican algunos contratos simples redactados en Solidity y compilados en bytecode de EVM. Paraello, se traduce dicho bytecode a F* (un lenguaje de programación funcional, inspirado por ML y que tiene como objetivo la verificación de programas)

Cardano se encuentra actualmente integrando las funcionalidades de smart contracts a la cadena de bloques, en el inicio de lo que se llama su *Era Goguen*[5], la tercera de las cinco etapas en la hoja de ruta del proyecto. Las mismas son:

1. Era Byron: Fundación e implementación del ledger.
2. Era Shelley: Capacidad de múltiples activos y descentralización.
3. Era Goguen: Contratos inteligentes.
4. Era Basho: Optimización y escalabilidad.
5. Era Voltaire: Gobernanza.

La era Goguen también abarca el trabajo para hacer que Cardano sea accesible a un público más amplio a través de Marlowe [6], un nuevo lenguaje de dominio específico para escribir Smart Contracts financieros en Cardano. Marlowe permite que los contratos sean escritos en un lenguaje financiero, en lugar de usar lenguajes de propósito general para blockchain, si bien el código que genera es código Haskell. Actualmente se encuentra aún en desarrollo. Cuando esté en funcionamiento, las organizaciones podrán escribir sus propios contratos o descargar los contratos ya realizados de los repositorios y transferir activos conforme las condiciones pactadas.

Actualmente los contratos Marlowe pueden ser escritos directamente en Haskell, en Javascript, o a través de la semántica introducida por Marlowe, y pueden ser visualizados usando el Marlowe Playground, donde también es posible simularlos y analizarlos. En los próximos meses finalizará la implementación de Marlowe en Cardano, y de esa forma los contratos se ejecutarán directamente en la Blockchain.

El trabajo realizado hasta la fecha relativo a verificación de contratos en Marlowe se resume en los trabajos [Lamela Seijas and Thompson, 2018], [Lamela Seijas et al., 2020a] y [Lamela Seijas et al., 2020b].

En [Lamela Seijas et al., 2020a] se utilizó el verificador Isabelle/HOL para probar algunas propiedades básicas de los contratos que genera[7]:

- Balance de los recursos involucrados en el contrato.
- Devolución de los recursos no gastados a sus dueños originales, al finalizar el contrato.
- Finalización del contrato.

[5] https://roadmap.cardano.org/en/goguen/
[6] https://alpha.marlowe.iohkdev.io/
[7] Marlowe github (2018). https://github.com/input-output-hk/marlowe. Accessed 6 Ago 2021

En [Lamela Seijas et al., 2020b] se propusieron algunas optimizaciones (basadas en la sintaxis específica en Haskell de los contratos generados por Marlowe) para hacer más eficiente el análisis estático con la biblioteca SBV.SBVes una herramienta que permite expresar propiedades a verificar sobre funciones en Haskell, y realiza su conversión a lenguaje simbólico para que sean probadas por un SMT solver (en el caso de este trabajo, Z3). Para analizar la performance de las optimizaciones, se armó un benchmark con contratos financieros de tipo *Auction*, *Crowdfunding*, *Rent* y *Coupon Bond* y se encontró que las optimizaciones mejoraban significativamente los tiempos de prueba de los contratos.

Por otra parte, en [Kondratiuk et al., 2021] se presentan algunos lineamientos sobre la verificación de un contrato en especifico definido por el estándar ACTUS. Se explora también un framework para la construcción de criptomonedas estandarizadas utilizando Marlowe y con el estándar ACTUS como base. También se profundiza en las formas en que los contratos descritos en ACTUS pueden definirse en Marlowe, Plutus y Haskell. Por supuesto, Plutus o Haskell pueden expresar estos contratos, pero representarlos en Marlowe trae ventajas adicionales. Marlowe se define para proporcionar garantías por diseño: un contrato de Marlowe sólo hará un número finito de interacciones con su entorno, y su vida útil puede leerse en el código de un contrato; además, cuando finalice el contrato, todos los activos mantenidos por el contrato se devolverán automáticamente a los participantes. Ninguna de estas garantías puede ser proporcionada por un lenguaje de propósito general.

Con respecto a las técnicas de validación y verificación de smart contracts, en el estudio exploratorio de [Tolmach et al., 2021] se investigan modelos formales y especificaciones de smart contracts presentes en la literatura y se presenta una extensiva y sistemática descripción de los mismos. También se discuten los enfoques actuales, usados para la verificación de dichas especificaciones. En particular, se menciona como una de las herramientas de Theorem Proving a Isabelle/HOL, utilizada para desarrollar semántica formal en lenguajes de smart contracts de bajo, medio y alto nivel, incluidos lenguajes de dominio específico (DSL) para contratos financieros, como en [Lamela Seijas et al., 2020a]. Este es el asistente que utilizaremos para la verificación.

0.3. Objetivos

El objetivo general del trabajo consiste en estudiar la incorporación de contratos inteligentes del estándar ACTUS a la blockchain Cardano, mediante la búsqueda y verificación de propiedades para las implementaciones de contratos especificados en el estándar.

Los objetivos particulares son:

1. Diseñar un mecanismo de traducción de la especificación formal de ACTUS a código en Marlowe.

2. Verificar los contratos escritos en Marlowe utilizando el asistente de pruebas Isabelle/HOL.

3. Incorporar los contratos diseñados a la plataforma ACTUS Labs, de manera que los contratos puedan generarse en forma interactiva desde el sitio web, obteniendo directamente el código a partir de un diseño por bloques.

0.4. Cronograma de trabajo

Se establece el siguiente cronograma estimativo para el desarrollo de la tesis:

Tareas	Meses									
	1	2	3	4	5	6	7	8	9	10
Lectura de bibliografía	■	■								
Plutus Pioneers Program		■	■							
Desarrollo			■	■	■	■	■	■		
Redacción de Manuscrito								■	■	■

La tesis tiene una carga de trabajo estimada de 750 horas, a las que se suman 120 horas de formación siguiendo el curso de Plutus Pioneers Program.

- Lectura de bibliografía [120 horas]: Lectura de artículos, publicaciones científicas y literatura en los cuales se basará el estudio de esta tesis.
- Plutus Pioneers Program [120 horas]: Se tomará este curso brindado por Cardano Developers, con duración de 2 meses, para entrenar desarrolladores en Plutus para el ecosistema Cardano.
- Desarrollo [450 horas]:
    - Escritura y ejecución de las pruebas de validación para plantillas (contratos con algunas variables cuyos valores quedan a definir por el usuario) de cada tipo de contrato con Isabelle/HOL.[300 horas]. Esta tarea se divide en las siguientes subtareas:
        - Estudiar la representación de los contratos en el estÁndar ACTUS.
        - Diseñar una metodología general para traducir los contratos ACTUS a Marlowe.
        - Implementar los contratos seleccionados en Marlowe.
        - Definir las propiedades a validar.

    - Escribir pruebas de validación en Isabelle/HOL y verificar que los contratos generales (es decir, sin asignar a las variables valores específicos) pasen todas las pruebas.
  - Incorporación de los contratos a la plataforma interactiva ACTUS Labs. [150 horas]
- Redacción de manuscrito de tesis [180 horas].

Bibliografía

[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.

[Kondratiuk et al., 2021] Kondratiuk, D., Seijas, P. L., Nemish, A., and Thompson, S. (2021). Standardized crypto-loans on the cardano blockchain. In *5th Workshop on Trusted Smart Contracts, Financial Cryptography and Data Security 2021*.

[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).