Digital Currencies as Types

Timothy A. K. Zakian
Novi
https://tzakian.github.io/
tzakian@fb.com

Abstract

Linear types have been well studied since their inception by Girard; a linear value can be moved from one place to another, but can never be copied or forgotten. From its inception Move – a new programming language developed to implement custom transactions and smart contracts on the Libra Blockchain – has had values–or resources–that behave in this linear manner as a central part of its semantics. On the Libra Blockchain, Move enables significant parts of the Libra protocol, including the Libra Coins, transaction processing, and validator management. In this talk, we will look at how different digital assets are represented with Move on the Libra Blockchain.

In the process of exploring the representation of digital assets on-chain in Move, we will revisit one of the first examples used in the paper that introduced linear logic; that of payments, and encounter other ideas from programming languages along the way, such as type-indexed data types and code modularity. We will see how we can leverage these ideas to provide strong guarantees of key asset properties such as losslessness, value conservation, and explicit representation of an asset, its currency, and its value.

As we explore the implementation of a digital asset in Move, we will see how, in Move, code is organized into a number of different modules, with each module consisting of resources and functions that can be used with the resources defined in that module. This gives rise to a type of strong encapsulation around the resources defined within a Move module: only functions within the module that define the resource can create, destroy, or access the fields of that resource.

We will see how representing a digital asset as a resource, coupled with this strong encapsulation, and privileging the creation and destruction operations within the module means that we can build a digital asset representation on-chain that is lossless by design: wherever it may go on-chain, such a digital asset cannot ever be "lost" or accidentally forgotten, and, no new digital assets can be created on-chain without the correct privileges.

We can then index this digital asset resource that we've built in Move by a type-level representation of each currency in the system to arrive at an explicit static representation of the currency of a digital asset. This representation statically disallows entire classes of possible issues, such as trying to combine two assets in different currencies, while still preserving all of the properties that we previously had, such as losslessness.

With this representation of a digital asset that we have built in Move, we can also test and verify that the value of the digital assets on-chain are preserved outside of creation and destruction operations; since the only functions that can change the value of an asset must be defined within the same module we can heavily test, and in fact verify, that these functions preserve the value of any digital assets that they may interact with. At the end of this process we arrive at a testable, verifiable, and explicit representation of a digital asset in Move that is lossless, conserves value, and represents its currency and value explicitly.

2012 ACM Subject Classification Theory of computation → Semantics and reasoning; Theory of computation → Linear logic; Theory of computation → Type theory

Keywords and phrases Digital Currencies, Linear Types, Move, Blockchains

Digital Object Identifier 10.4230/OASIcs.Tokenomics.2020.3

Category Invited Talk

2nd International Conference on Blockchain Economics, Security and Protocols (Tokenomics 2020).
Editors: Emmanuelle Anceaume, Christophe Bisière, Matthieu Bouvard, Quentin Bramas, and Catherine Casamatta; Article No. 3; pp. 3:1–3:1

OpenAccess Series in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany