Lean Course

Guilherme Silva

2024-03-17 Tue

Outline

1. Solidity
2. My research
3. Lean Course

Motivation

- Ethereum is the second-biggest cryptocurrency and the biggest smart contract platform
- Solidity is the most used programming language to create contracts in this platform
- Code in the smart contracts domain can not be changed after deployment
- It is totally open
- Used for financial assets
- There are adversary interests in the smart contracts ecosystem
- Identity behind addresses is anonymous

Motivation

- Ethereum is the second-biggest cryptocurrency and the biggest smart contract platform
- Solidity is the most used programming language to create contracts in this platform
- Code in the smart contracts domain can not be changed after deployment
- It is totally open
- Used for financial assets
- There are adversary interests in the smart contracts ecosystem
- Identity behind addresses is anonymous

Motivation

- Ethereum is the second-biggest cryptocurrency and the biggest smart contract platform
- Solidity is the most used programming language to create contracts in this platform
- Code in the smart contracts domain can not be changed after deployment
- It is totally open
- Used for financial assets
- There are adversary interests in the smart contracts ecosystem
- Identity behind addresses is anonymous

Motivation

- Ethereum is the second-biggest cryptocurrency and the biggest smart contract platform
- Solidity is the most used programming language to create contracts in this platform
- Code in the smart contracts domain can not be changed after deployment
- It is totally open
- Used for financial assets
- There are adversary interests in the smart contracts ecosystem
- Identity behind addresses is anonymous

Motivation

- Ethereum is the second-biggest cryptocurrency and the biggest smart contract platform
- Solidity is the most used programming language to create contracts in this platform
- Code in the smart contracts domain can not be changed after deployment
- It is totally open
- Used for financial assets
- There are adversary interests in the smart contracts ecosystem
- Identity behind addresses is anonymous

Motivation

- Ethereum is the second-biggest cryptocurrency and the biggest smart contract platform
- Solidity is the most used programming language to create contracts in this platform
- Code in the smart contracts domain can not be changed after deployment
- It is totally open
- Used for financial assets
- There are adversary interests in the smart contracts ecosystem
- Identity behind addresses is anonymous

Motivation

- Ethereum is the second-biggest cryptocurrency and the biggest smart contract platform
- Solidity is the most used programming language to create contracts in this platform
- Code in the smart contracts domain can not be changed after deployment
- It is totally open
- Used for financial assets
- There are adversary interests in the smart contracts ecosystem
- Identity behind addresses is anonymous

Motivation

- Ethereum is the second-biggest cryptocurrency and the biggest smart contract platform
- Solidity is the most used programming language to create contracts in this platform
- Code in the smart contracts domain can not be changed after deployment
- It is totally open
- Used for financial assets
- There are adversary interests in the smart contracts ecosystem
- Identity behind addresses is anonymous

Solidity details

- Solidity has two kinds of memory, one is called “storage” and another one is called “memory”
- Storage is persistent in the blockchain while memory is erased after a transaction

Solidity details

- Solidity has two kinds of memory, one is called “storage” and another one is called “memory”
- Storage is persistent in the blockchain while memory is erased after a transaction

Solidity details

- Solidity has two kinds of memory, one is called “storage” and another one is called “memory”
- Storage is persistent in the blockchain while memory is erased after a transaction

Building for Solidity

- Program logic
- Calculus
- Verification system

Building for Solidity

- Program logic
- Calculus
- Verification system

Building for Solidity

- Program logic
- Calculus
- Verification system

Building for Solidity

- Program logic
- Calculus
- Verification system

Axiomatization of Solidity data types

- In storage and memory
- Structs
- Maps
- Array
- I developed an algebra for writing and reading in memory and storage

Axiomatization of Solidity data types

- In storage and memory
- Structs
- Maps
- Array
- I developed an algebra for writing and reading in memory and storage

Axiomatization of Solidity data types

- In storage and memory
- Structs
- Maps
- Array
- I developed an algebra for writing and reading in memory and storage

Axiomatization of Solidity data types

- In storage and memory
- Structs
- Maps
- Array
- I developed an algebra for writing and reading in memory and storage

Axiomatization of Solidity data types

- In storage and memory
- Structs
- Maps
- Array
- I developed an algebra for writing and reading in memory and storage

Axiomatization of Solidity data types

- In storage and memory
- Structs
- Maps
- Array
- I developed an algebra for writing and reading in memory and storage

Main issues

- Shallow and deep copy
- Structured defaults
- “Eager” and “lazy” formalizations
- Eager is similar to low-level language (such as Ethereum Virtual Machine)
- Lazy is necessary for automatic theorem proving

Main issues

- Shallow and deep copy
- Structured defaults
- "Eager" and "lazy" formalizations
- Eager is similar to low-level language (such as Ethereum Virtual Machine)
- Lazy is necessary for automatic theorem proving

Main issues

- Shallow and deep copy
- Structured defaults
- "Eager" and "lazy" formalizations
- Eager is similar to low-level language (such as Ethereum Virtual Machine)
- Lazy is necessary for automatic theorem proving

Main issues

- Shallow and deep copy
- Structured defaults
- “Eager” and “lazy” formalizations
- Eager is similar to low-level language (such as Ethereum Virtual Machine)
- Lazy is necessary for automatic theorem proving

Main issues

- Shallow and deep copy
- Structured defaults
- “Eager” and “lazy” formalizations
- Eager is similar to low-level language (such as Ethereum Virtual Machine)
- Lazy is necessary for automatic theorem proving

Main issues

- Shallow and deep copy
- Structured defaults
- “Eager” and “lazy” formalizations
- Eager is similar to low-level language (such as Ethereum Virtual Machine)
- Lazy is necessary for automatic theorem proving

Lean

- Formalization of datatypes in Lean
- Theorems of lazy solution by the eagle implementation

Lean

- Formalization of datatypes in Lean
- Theorems of lazy solution by the eagle implementation

Lean

- Formalization of datatypes in Lean
- Theorems of lazy solution by the eagle implementation