Fundação Getúlio Vargas

Modelagem Matemática

Programação de criptomoeda em linguagem Agda

Mestrando:
Guilherme Horta Alvares da Silva
Orientador:
Professor Doutor Flávio Codeço Coelho

Conteúdo

1 Introdução

The bitcoin Satoshi Nakamoto created it in 2008 using many ideas from cypherpunk community.

Before bitcoin, there were a lot of cryptograph cash ideas based on ecash protocol. Adam Back developed a proof-of-work scheme for spam control. Wei Dai propose b-money for the first proposal for distributed digital scarcity. And Hal Finney created a reusable proof of work for hashcash for its algorithm of proof of work.

Since 2008, the total market of Bitcoin came to 330 billions dollars in 17 of December of 2018 and his value has a historic record of 20 thousands dollars. Other criptocurrencies like Ethereum and Monero were created and they both have a good market value. Cripto currencies will be a great finance instrument for a near future.

Cripto currencies are used for smart contracts too. For example, it is possible to reserve some part of money to the seller in the blockchain. To unlock the money, it is necessary to get a signature of the buyer. If the buyer receive the product, he will sign the contract. If the product does not come in time, the buyer receive his money back. Smart contracts are widely adopted today the big funds and these contracts are fully governed by algorithms.

1.1 O bitcoin

Já que cripto-moedas são governadas integralmente por algoritmos. Precisamos garantir que esses algoritmos estejam corretos, uma vez que em caso de falha, não existe nada que se possa fazer. Por exemplo, o bitcoin pode ficar preso em algum contrato e não seria possível usá-lo novamente. O único jeito de resolver problemas é criando um consenso entre usuários e mineradores, a partir de um soft fork (atualização retro compatível) ou um hard fork (atualização não retrocompatível). A possibilidade de erro é um dos maiores riscos às cripto-moedas. Por exemplo, no protocolo inicial do bitcoin, a unicidade dos IDs das transações não era garantida. Com isso, era possível que houvesse ataques de gasto duplo. Isso pôde ser evitado incluindo o número do bloco na transação (foi feito em um soft fork). Outro problema do bitcoin é que o tamanho do bloco do bitcoin não é suficiente para cobrir todas as transações. Para resolver isso, foi criada a Lightning Network, que é um protocolo de segunda camada do bitcoin.

Por isso, é importante a verificação completa dos protocolos criptográficos. Um alto nível de verificação pode ser encontrado criando um método formal para cripto-moedas e provando que ele está correto. Métodos formais são análises matemáticas

para verificar se o software comporta conforme o esperado. Ainda não foi criado nenhum método completamente formal para a formalização de alguma cripto-moeda.

O crescimento de contratos inteligentes criou alguns problemas em relação a segurança de instrumentos financeiros. O maior incidente foi a falha do cripto-contrato DAO. Quando o DAO chegou a um valor de 150 milhões de dólares, um usuário usou uma vulnerabilidade para hackear o contrato. A perda do dinheiro dos investidores foi somente evitada por causa de um hard fork na rede ethereum, que apagou a maioria das transações investidas no DAO. Esse hard fork violou o principio de que o dinheiro dos usuários em cripto-moedas devem ser apenas governados por algoritmos, sem nenhuma interferência humana. Por isso, existe a necessidade de provas formais, que garantem sua corretude do software.

1.2 Agda Introduction

Agda is a dependently typed functional language developed by Norell at Chalmers University of Technology as his PhD Thesis. The current version of Agda is Agda 2.

Agda is also a proof assistance based on intensional Martin-Löf type theory. It looks like CoQ, but does not have tatics. Agda is a total language, so it is garanteed that the code always terminal and coverage all inputs. Agda needs it to be a consistent language.

Agda has inductive data types that are similar to algebric data types in non-depently typed programming language. The definition of Peano numbers in Agda:

```
data ℕ : Set where
  zero : ℕ
  suc : ℕ → ℕ
```

1.3 UTXO Bitcoin

Uma das características que diferencia o Bitcoin de outras criptomoedas (como o Ethereum) e outros bancos é a sua estrutura de dados de transação. Em um banco, o saldo dessa conta é armazenado em uma linha do banco de dados. Para o bitcoin, isso não existe. O que está armazenado na blockchain são apenas as transações. No bitcoin, existe a transação não gasta, cujo termo é UTXO (unspent transaction output).

No bitcoin, cada transação possui várias entradas e saídas. As entradas são todas as transações não gastas do usuário e a saída são todas as contas receberão a transação mais uma transação não gasta. Como pode ser visto nessa figura:

Existem várias vantagens e desvantagens desse tipo de estrutura de dados. Uma das desvantagens é que é bastante custoso verificar o saldo da conta, pois é necessário calcular todas as transações vinculadas a conta. Outra desvantagem é que esse modelo é mais complexo que o modelo bancário.

Já suas vantagens são outras. Esse tipo de estrutura de dados permite escalabilidade, pois o paralelismo é permitido. Já que uma mesma conta pode por exemplo gastar várias transações não gastas ao mesmo tempo. Outra vantagem seria a privacidade, pois um mesmo usuário pode criar uma conta nova por transação.

No paper de Anton Setzer, a modelagem da estrutura de dados do UTXO está quase terminada.

1.4 Blockchain em Agda

A estrutura de dados blockchain significa cadeia de blocos. Existem dois tipos de blocos, o primeiro bloco (genesis block) e os outros. Cada bloco, possui uma ligação para o bloco anterior (exceto o primeiro). Essa ligação é o hash (identificador único de algum elemento) do bloco anterior. Dessa forma, é possível sempre garantir a ordem dos blocos. Cada bloco armazena transações, o hash do bloco anterior, o nonce e o seu próprio hash. Cada bloco precisa ser minerado, ou seja, o minerador precisa calcular o valor do nonce (que necessita muito poder computacional).

Dessa forma, no código de Agda, foi adicionada a especificação de que o bloco deve conter o hash do bloco anterior e a especificação do hash do bloco, usando tipos dependentes.

2 Metodologia

O trabalho está sendo programado na linguagem Agda, provador de teoremas.

Para controle de versão, é utilizado o Git com Github. Com o Github, é possível fazer integração contínua com Travis CI ou Hydra. Ou seja, a cada novo commit, é instanciada uma máquina virtual para compilar o projeto todo.

O gerenciador de pacotes utilizado é o Nix, que é um gerenciador de pacotes funcional. Ou seja, é feito de forma descritiva como será compilado o projeto.

Para compilação dos slides e da dissertação é utilizado Latex e latexmk e todo o resto acima. Todo código colocado nos slides e na dissertação são compilados antes. Dessa forma, não existe perigo de haver algum código que não funcione.

O trabalho será feito usando a metodologia do livro Type Driven Development. Ou seja, enquanto a criptomoeda for programada, o seu tipo será refinado conforme às necessidades. Caso algo ainda não seja provado com os tipos, uma prova formal será adicionada posteriormente.

3 Agradecimento

Agradeço o professor Flávio por me auxiliar em todas as minhas dúvidas sobre bitcoin e me ajudar na escrita da minha dissertação.

Referências