Fundação Getúlio Vargas

Modelagem Matemática

Programming a cryptocurrency in Agda

Student:
Guilherme Horta Alvares da Silva
Professor:
Doctor Flávio Codeço Coelho

Contents

1 Introduction **2**
1.1 Cryptocurrencies 2
1.2 Agda Introduction 3
1.3 Martin-Löf type theory 3
1.4 Types 3
1.5 Types Constructors 3
1.6 UTXO Bitcoin 5
1.7 TXTree in Agda 7

2 Methods **7**

3 Conclusion **7**

References **8**

1 Introduction

1.1 Cryptocurrencies

In 1983, David Chaum created ecash (Panurach, 1996) , an anonymous cryptographic eletronic money. This cryptocurrency use RSA blind signatures (Chaum, 1983) to spend transactions. Later, in 1989, David Chaum found an eletronic money corporation called DigiCash Inc. It was declared bankruptcy in 1998.

Adam Back developed a proof-of-work (PoW) scheme for spam control, Hashcash (Back et al., 2002). To send an email, the hash of the content of this email plus a nounce has to have a numerically value smaller than a defined target. So, to create a valide email, the sender (miner) has to spend a considerable CPU resource on it. Because, hash functions produces pratically random values, so the miner has to guess a lot of nounce values before find some nounce that make the hash of the email less than the target value. This idea is the same that is used in Bitcoin proof of work, because each block has a nounce guessed by the miner and the hash of the block has to be less than the target value.

Wei Dai propose b-money (Dai, 1998) for the first proposal for distributed digital scarcity. And Hal Finney created Bit Gold, a reusable proof of work for hashcash for its algorithm of proof of work.

In 31 October 2008, Satoshi Nakamoto registered the website "bitcoin.org" and put a link for his paper (Nakamoto et al., 2008) in a cryptography mailing list. In January 2009, Nakamoto released the bitcoin software as open-source code. The identity of Satoshi Nakamoto is still unknown. Since that time, 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 cryptocurrencies like Ethereum (Wood et al., 2014), Monero (Noether, 2015) and ZCash (Hopwood, Bowe, Hornby, & Wilcox, 2016) were created after Bitcoin, but Bitcoin is still the cryptocurrency with the biggest market value.

Ethereum is a cryptocurrency that uses account model instead of UTXO used in bitcoin for its transaction data struct. It uses Solidity as its programming language for smart contracts, it looks like Java Script, so it is easier to programming in it than in stack machine programming language of Bitcoin. Ethereum is now changing from proof of work (used in Bitcoin) to proof of stake.

Monero and ZCash are both cryptocurrency that focus on fungibility, privacy and descentralization. Monero uses obfuscated public ledger, so anyone can send transcations, but nobody can tell the source, amount or destination. Zcash uses the

concept of zero-knowledge proof called zk-SNARKs, which garantee privacy for its users.

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.

1.3 Martin-Löf type theory

Agda is also a proof assistance based on intensional Martin-Löf type theory.

1.4 Types

In Martin-Löf type theory, there are 3 finites types and 5 types constructors. The 0 type contain 0 terms, it is called empty type and it is written bot.

The 1 type is the type with just 1 canonical term and it represents existence. It is called unit type and it is written top .

The 2 type contains 2 canonical terms. It represents a choice between two values.

The Boolean Type is defined using the Trivial type and the Either type

If statement is defined using booleans

1.5 Types Constructors

The sum-types contain an ordered pair. The second type can depend on the first type. It has the same meaning of exist.

```
data ∑ (A : Set) (B : A → Set) : Set where
  ⟨_,_⟩ : (x : A) → B x → ∑ A B

∑-elim : ∀ {A : Set} {B : A → Set} {C : Set}
  → (∀ x → B x → C)
  → ∑ A B
  ---------------
```

```
  → C
∑-elim f ⟨ x , y ⟩ = f x y
```

The pi-types contain functions. So given an input type, it will return an output type. It has the same meaning of a function

In Inductive types, it is a self-referential type. Naturals numbers are examples of that

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

Other data structs like linked list of natural numbers, trees, graphs are too. Proofs in inductive types are made by induction.

```
ℕ-elim : (target : ℕ) (motive : (ℕ → Set)) (base : motive zero)
  (step : (n : ℕ) → motive n → motive (suc n) ) → motive target
ℕ-elim zero motive base step = base
ℕ-elim (suc target) motive base step = step target (ℕ-elim target motive base step)
```

Universe types are created to allow proofs written in all types. For example, the type of Nat is U0.

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
```

Definitions in Agda are done using induction. For example, the sum of two numbers in Agda:

```
_+'_ : ℕ → ℕ → ℕ
zero +' m = m
suc n +' m = suc (n + m)
```

In Agda, because of dependent types, it is possible to make some restrictions in types that is not possible in other language. For example, get the first element of a

vector. For it, it is necessary to specify in the type that the vector should have at size greater or equal tha than one.

```
head : {A : Set} {n : ℕ} (vec : Vector A (suc n)) → A
head (x :: vec) = x
```

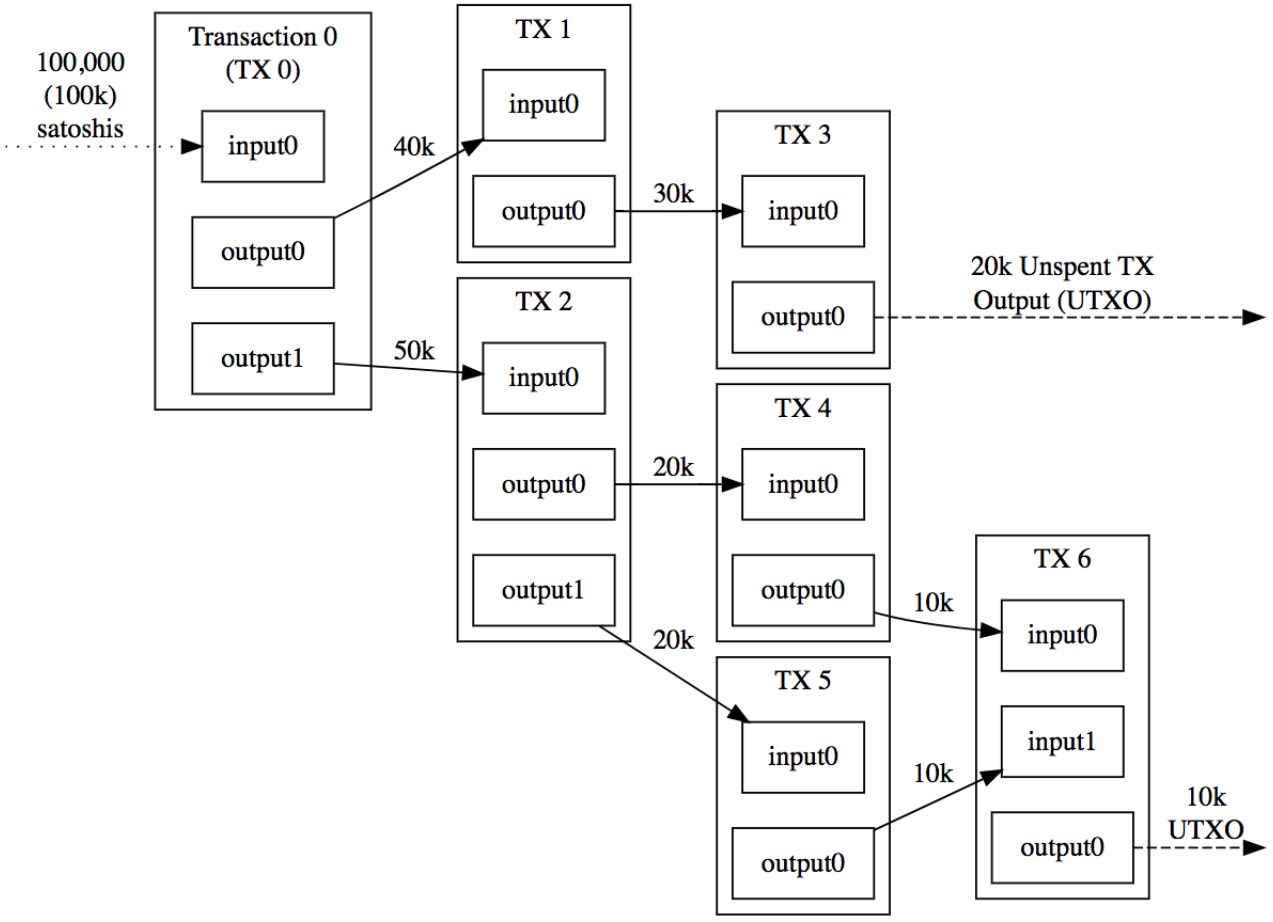
Another good example is that in sum of two matrices, they should have the same dimentions.

```
_+m_ : {m n : ℕ} (P Q : Matrix ℕ m n) → Matrix ℕ m n
[] +m [] = []
(vx :: P) +m (vy :: Q) = (vx +v vy) :: (P +m Q)
```

1.6 UTXO Bitcoin

There are two kinds of data structures to modeling accounts records and savings states. The UTXO model used in Bitcoin and the account model used in Ethereum.

In account model, it is saved the address and the balance of each address. For example, the data struct will look like this [(0xabc01, 1.01), (0xabc02, 2.02)]. So the address 0xabc01 has 1.01a of balance and the address 0xabc02 has 2.02 of balance. In this way, it is possible to easily know how much of balance each address has, but it is not possible to know how they got in this state.



Triple-Entry Bookkeeping (Transaction-To-Transaction Payments) As Used By Bitcoin

In UTXO model, each transaction is saved in the transaction tree. Every transaction is composed of multiples inputs and multiples outputs. But all inputs have to never been spent before.

Because of that, in UTXO model, it is easy to make a new transaction from previous one, but it is harder to know how much each one has. The wallet that calculate how much balance each address has.

In account model, there could be one kind of vulnerability that is less probabable to happen in UTXO model. Because there is an undesirable intermediary state that there is some address without balance while another has not already received his money.

For example:
bobBalance -= 1
Intermediary State
aliceBalance += 1

In account model, it is straight foward to know how much balance each address has. In UTXO model, this calculation is made offchain. It can be a good thing, because each user has more privacy.

1.7 TXTree in Agda

2 Methods

3 Conclusion

References

Back, A., et al. (2002). Hashcash-a denial of service counter-measure.

Chaum, D. (1983). Blind signatures for untraceable payments. In *Advances in cryptology* (pp. 199–203).

Dai, W. (1998). B-money. *Consulted*, *1*, 2012.

Hopwood, D., Bowe, S., Hornby, T., & Wilcox, N. (2016). Zcash protocol specification. *Tech. rep. 2016–1.10. Zerocoin Electric Coin Company, Tech. Rep.*.

Nakamoto, S., et al. (2008). Bitcoin: A peer-to-peer electronic cash system.

Noether, S. (2015). Ring signature confidential transactions for monero. *IACR Cryptology ePrint Archive*, *2015*, 1098.

Panurach, P. (1996). Money in electronic commerce: Digital cash, electronic fund transfer, and ecash. *Communications of the ACM*, *39*(6), 45–51.

Wood, G., et al. (2014). Ethereum: A secure decentralised generalised transaction ledger. *Ethereum project yellow paper*, *151*, 1–32.