Bitcoin transactions

Digital Assets - Week 2 (Pre-record)

Rhys Bidder
rhys.m.bidder@kcl.ac.uk

KBS, QCGBF

Autumn 2024

Disclaimer 1: Any opinions expressed, errors or omissions should be regarded as those of the author and not necessarily those of KCL, KBS, QCGBF, QCB, CBI, or BoE.

Disclaimer 2: These notes on cryptography are heavily simplified and leave out many important details. For any real-world security applications these notes should not be relied upon. Instead you should consult appropriate security resources and reliable security professionals.

Disclaimer 3: Cryptoasset transactions are illegal in some jurisdictions. Do not violate these or any other laws. I am not promoting the use of crypto in countries where it is illegal in any form and these slides are not a promotion of crypto or an invitation to participate in crypto-related activities in such countries. They are purely for educational purposes.

Bitcoin transactions

Bitcoin adopts the Unspent Transaction Output (UTXO) model

- UTXOs are unspent chunks of value, denominated in bitcoin, leftover from transactions, which can then be used in later transactions.
- **Note:** Other important blockchains use UTXO, including some private chains (e.g Corda)

Very readable discussions can be found here and here

- The intuition of UTXOs is *a bit like* using different coins/notes in our pockets to execute transactions
- Those coins/notes were 'unspent outputs' from previous transactions and we combine them to make future transactions
- **Warning:** The parallels are not complete between UTXOs and coins/notes

Bitcoin transactions

The state in Bitcoin is represented by many chains of transactions, each traceable back to a 'coinbase transaction(s)'

- As we will discuss later, bitcoin is created through **'mining'**
- New bitcoin emerges from rewards paid to 'miners' in the **coinbase transaction** (the 'money supply' in Bitcoin)
- **Chains arise from outputs (UTXOs) from earlier transactions being used up as inputs to new transactions**

Terminology warning: These chains are distinct from the overall *block*chain

- They capture the conceptual book-keeping of *payments*
- The blockchain and its protocols provide the technology implementing and recording this

Bitcoin transactions

- Every transaction must spend **all** of each UTXO associated with a transaction
  - The UTXOs from sending address(es) become **inputs**
  - Once they are 'consumed', they no longer exist
- Every transaction must have at least one output
  - **Outputs** are new UTXOs associated with receiving address(es)
- Reiterate: **All** of a UTXO used as an input must be spent
  - We need to allow for **'change'** when paying someone else
  - Typically a transaction *also* sends residual to *sender's* address!
  - So, unlike change in the real world, this isn't *'given back'*

Bitcoin transactions

- A typical (non-coinbase) Bitcoin transaction is a message containing various 'fields' of information
- We will highlight the most important (see here and here for richer detail)
  - Transaction ID (TXID)
  - SignatureScript
  - Input information
  - Output information

Bitcoin transactions

- Transaction ID (TXID)
  - Hash of all of the other information in the transaction
- SignatureScript
  - Reflects the signing of the transaction by authorized parties
  - **Relies on careful protection of private keys**
  - In simplest case, only sender signs, but additional signers can be used for extra security

Bitcoin transactions

Input information

- TXIDs of previous transactions whose UTXOs are being used as inputs for payment in *this* transaction
- The indices of the UTXOs in the previous transactions
  - **Q:** Why do we need 'OutIndices'?
  - A transaction may have many UTXOs as outputs
- PubKeyScript
  - Checks if signatures match authorizers' public keys
- Total input value (in BTC)
  - Adds up value of all UTXOs being sent

Each transaction is linked to a previous transaction(s)

$\Rightarrow$ **Chains of TXID-OutIndices**

Bitcoin transactions

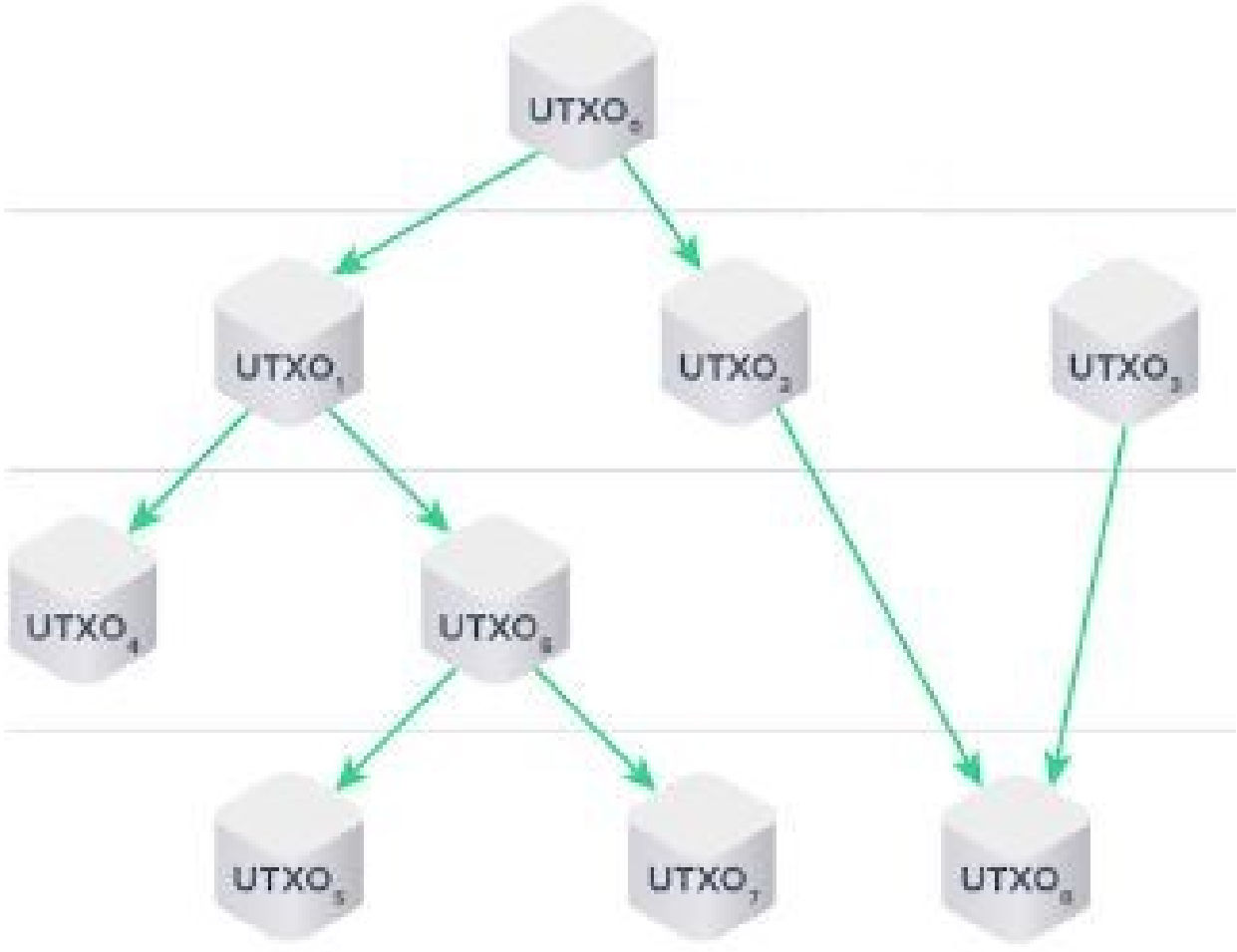
Output information

- Total output value (in BTC)
  - This must be $\leq$ total input value
  - Residual is optional, but advisable, TX fee payment to miner
  - Fee used to incentivise inclusion in a block
- Output addresses, values and indices
  - To whom the outputs are sent (can be one or multiple)
  - How much in BTC in each UTXO
  - Indices referencing particular UTXOs (previously mentioned in inputs discussion)
- PubKeyScript
  - Specifies **requirements** for UTXOs to be spent (when used as inputs in future transaction)
  - **Note:** Distinct from the PubKeyScript mentioned in the *inputs*

Bitcoin transactions

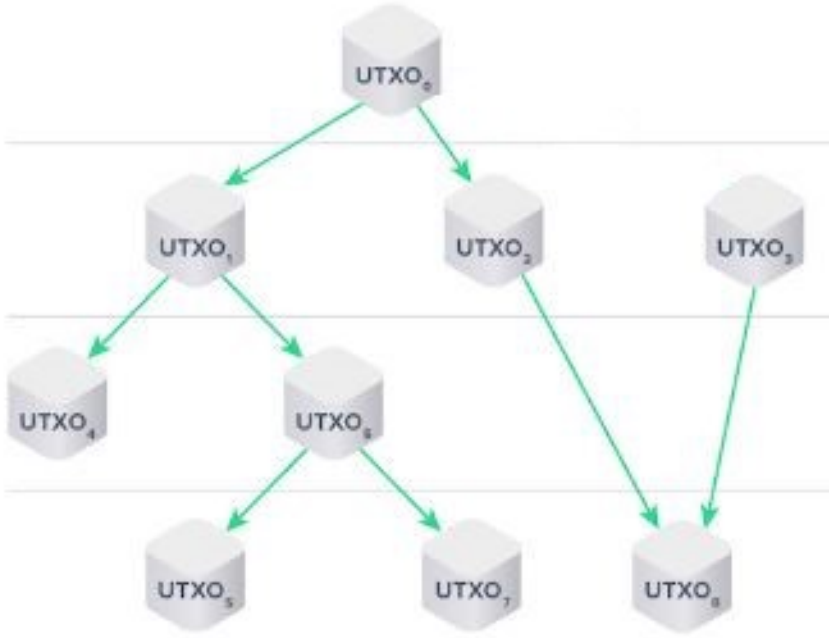
- Each transaction is linked to a previous transaction(s)
  - By iterating on that link, the transaction is connected to all transactions back to the coinbase transaction(s)...
  - ...**and** to any transactions that succeed it, using inputs derived from the outputs of this transaction
- So transactions are chained via the creation/use of UTXOs
  - In particular, *via TXID-OutIndices* pairs
- As we will discuss later this has implications for privacy
  - Esp. when combined with tools to de-pseudonymize addresses

Bitcoin transactions



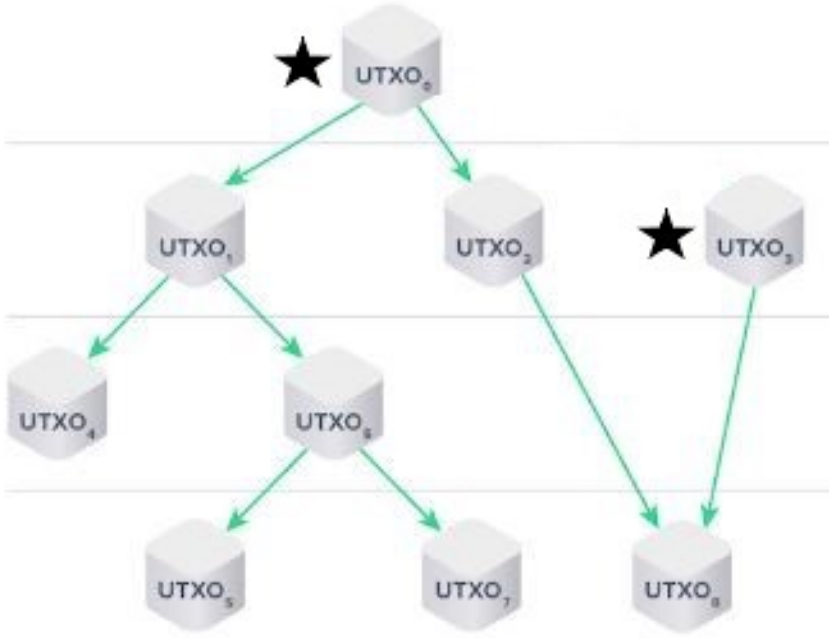
Connecting UTXOs - Directed Acyclic Graph (DAG). Source: Horizen Academy

Bitcoin transactions



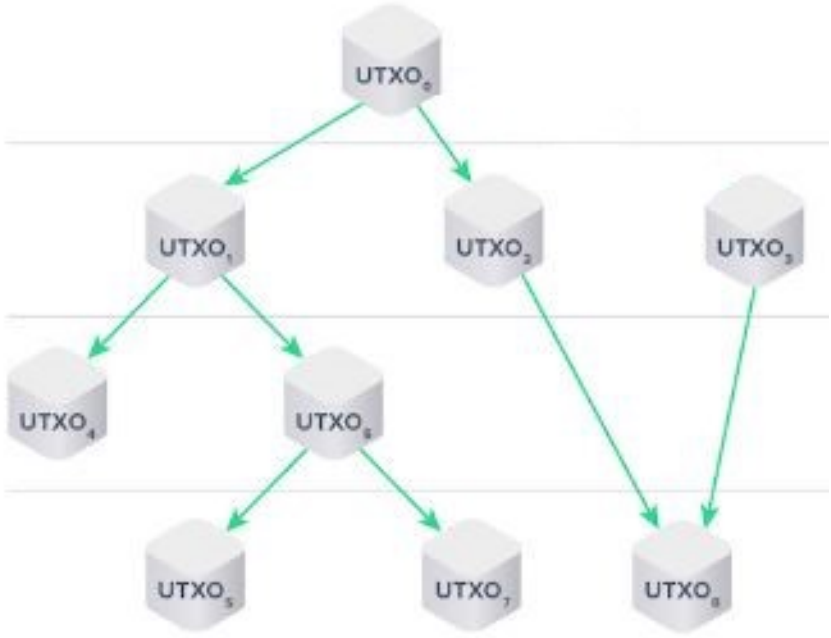
- Each transaction represented by arrows coming out of a node

Bitcoin transactions



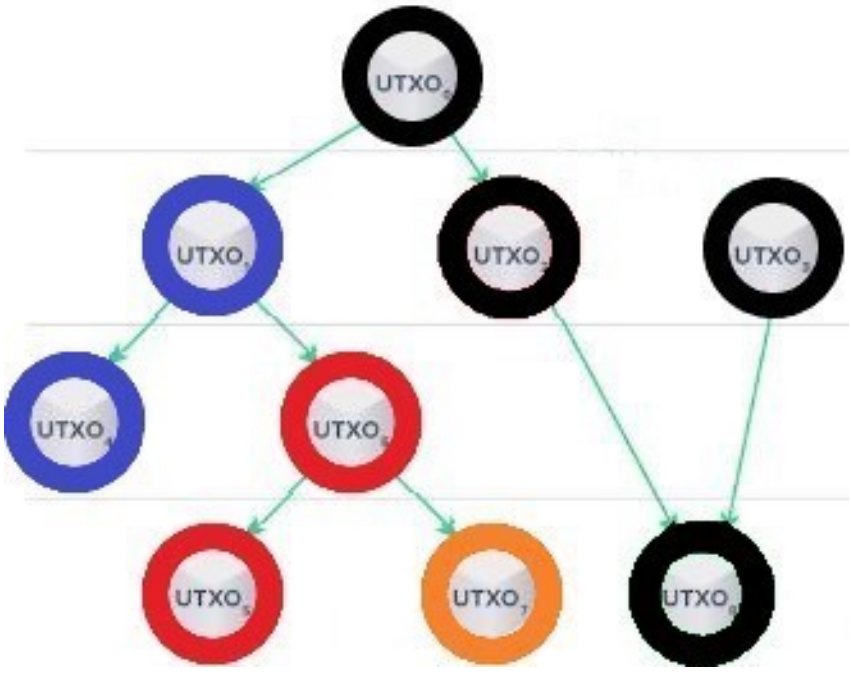
- Two UTXOs assumed to arise from mining

Bitcoin transactions



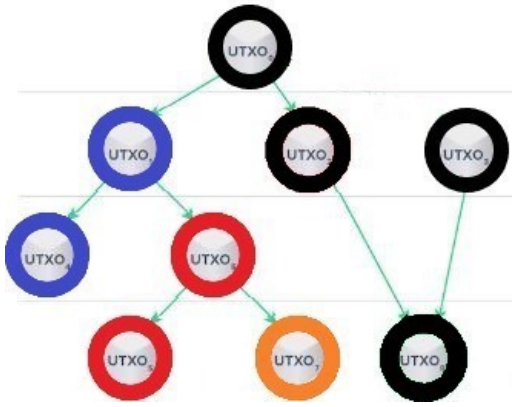
- UTXOs destroyed and created in every transaction

Bitcoin transactions



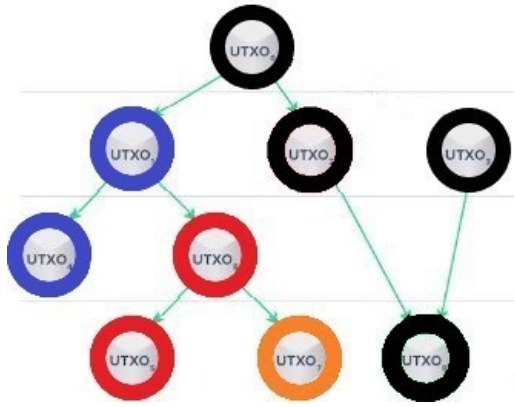
- Let each color correspond to an address

Bitcoin transactions



- **Black** address transacts with blue
  - Sends some of its UTXO balance to them
  - Sends 'change' to itself
- Blue transacts with red
  - Uses UTXO from **black** as input
  - But not all of it - so sends some 'change' to itself
- Red transacts with orange
  - Uses UTXO from blue as input
  - But not all of it - so sends some 'change' to itself

Bitcoin transactions



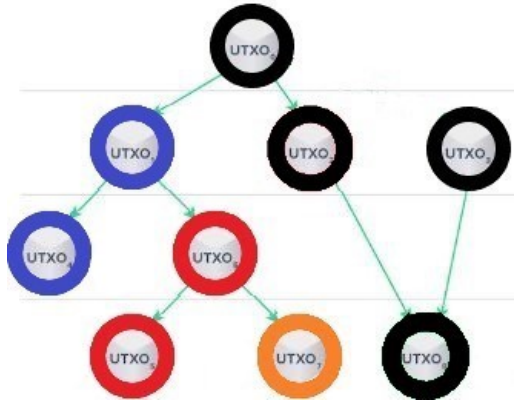
Consider the right hand side - why is the miner using two UTXOs to send a single UTXO to herself?

- Common practice called 'consolidation'
- Using multiple small UTXOs for a large payment makes transaction computationally expensive for miners

$\Rightarrow$ Less likely transaction will be added to block quickly

Note that transactions to 'yourself' and the use of multiple addresses makes it very hard to assess (naively) the true amount of activity in such blockchains

Bitcoin transactions



Note: More than one color may correspond to a user

- For example, several of these addresses might be the miner's
- Blue could be an address stored in a cold wallet
- Red might be a transactional address managed by a hot wallet

Bitcoin transactions

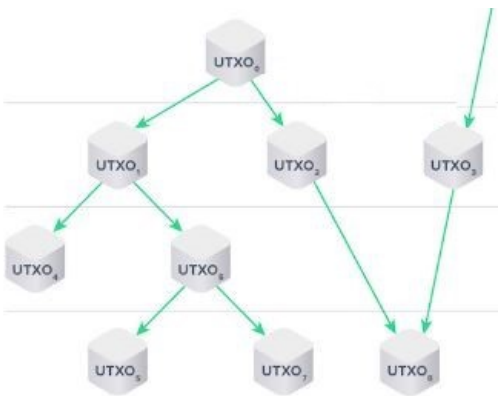
How does (this part) of the blockchain state evolve in our example

- I say 'this part' because there will be many such DAGs recorded in the blockchain - and they generally will interact
- Indeed, I *could* have drawn an arrow coming from off-screen
  - **Q:** What would that represent?

Bitcoin transactions

How does (this part) of the blockchain state evolve in our example

- I say 'this part' because there will be many such DAGs recorded in the blockchain - and they generally will interact
- Indeed, I *could* have drawn an arrow coming from off-screen
  - **Q:** What would that represent?
  - **A:** Transaction using as input a UTXO from a DAG emerging from a *different* coinbase transaction

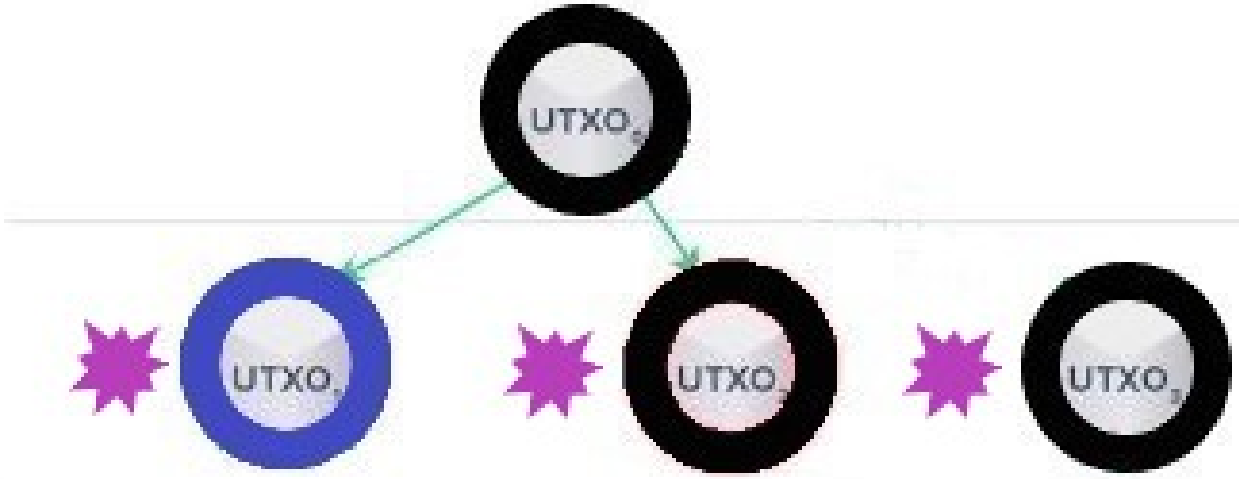


But for the next few slides, set that aside and assume our original DAG is the full set of UTXOs. . .

Bitcoin transactions

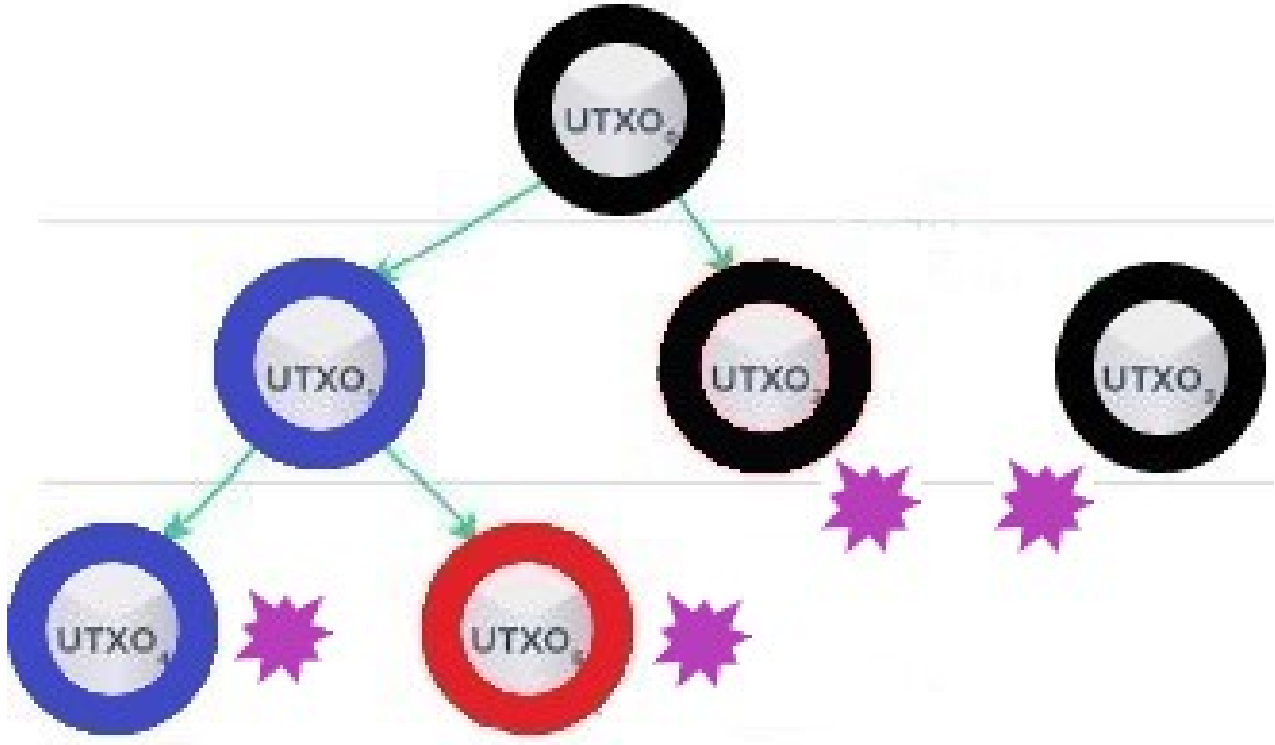
Evolution of state (purple splodge). Source: Horizen Academy

Bitcoin transactions



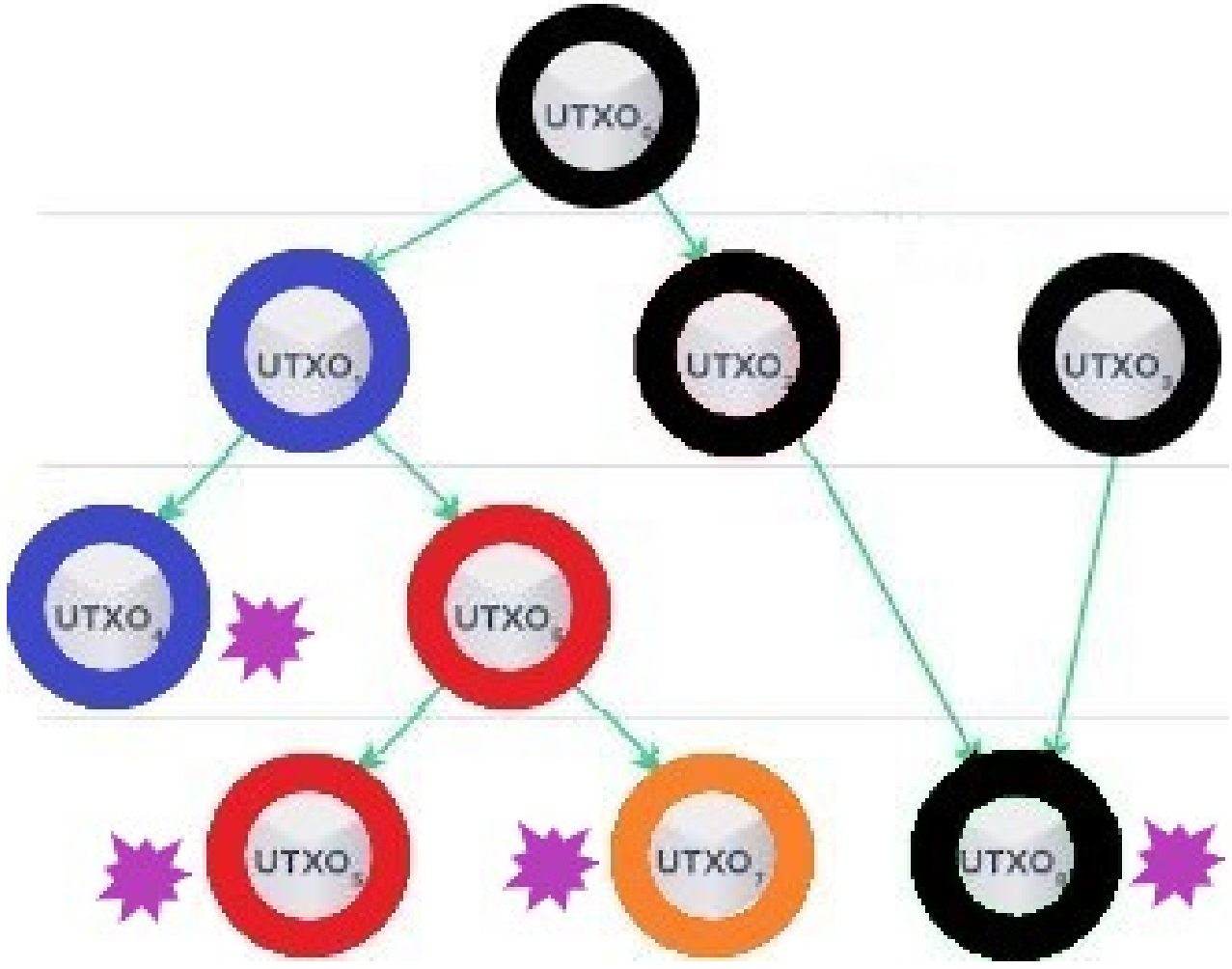
Evolution of state (purple splodge). Source: Horizen Academy

Bitcoin transactions



Evolution of state (purple splodge). Source: Horizen Academy

Bitcoin transactions



Evolution of state (purple splodge). Source: Horizen Academy

Bitcoin transactions

The purple splodges were highlighting the currently unspent outputs (the **UTXOs**) and the **addresses** that have ownership of them

- These are the state of the Bitcoin system
- Nevertheless, the blockchain records the entire history of transactions, implicit in the entire history of blocks

Bitcoin transactions

The resources an *address* can draw upon comprise the aggregation of all the unspent outputs sent to that address

- The resources that a *user* can draw on is an aggregation over the resources of her addresses
- Let's proceed by assuming each user has just one address

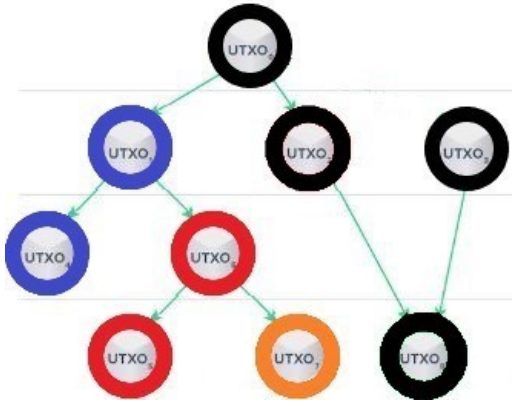
Why not represent state as the aggregation - an **account** balance?

- Why track a load of transaction chains when all I care about is whether someone has enough resources to pay?
- Ethereum adopts the account representation

Both representations can work (pros and cons)

- See here and Ch. 5-6 of Lipton and Treccani's *Blockchain and Distributed Ledgers*

Bitcoin transactions



Evolution of state. Source: Horizen Academy

Suppose we want to think of this not in UTXO, but more intuitively in terms of what the different addresses are sending/receiving:

- **Q:** What would that look like?
- **Q:** How are their BTC balances evolving?

Bitcoin transactions

Period	Transfers	# Transactions	Balances				Total BTC
			Bk	B	R	O	
1	(Coinbase; {Bk100})	1	100	0	0	0	
2	(Bk; {B70, Bk30}), (Coinbase; {Bk100})	2					
3	(B; {R5, B65})	1					
4	(R; {O4, R1}),(Bk30,Bk100; {Bk130})	2					

Assume:

- Addresses receiving coinbase previously controlled no UTXO
- Other addresses also initially controlled no UTXO
- No other TX associated with these addresses in the blockchain
- We are also ignoring (optional but advisable) transaction fees

Note: Numbers are arbitrary/illustrative

Bitcoin transactions

Period	Transfers	# Transactions	Balances				Total BTC
			Bk	B	R	O	
1	(Coinbase; {Bk100})	1	100	0	0	0	
2	(Bk; {B70, Bk30}), (Coinbase; {Bk100})	2	130	?	?	0	
3	(B; {R5, B65})	1	130	65	?	0	
4	(R; {O4, R1}),(Bk30,Bk100; {Bk130})	2	130	?	1	4	

Assume:

- Addresses receiving coinbase previously controlled no UTXO
- Other addresses also initially controlled no UTXO
- No other TX associated with these addresses in the blockchain
- We are also ignoring (optional but advisable) transaction fees

Note: Numbers are arbitrary/illustrative

Bitcoin transactions

Period	Transfers	# Transactions	Balances				Total BTC
			Bk	B	R	O	
1	(Coinbase; {Bk100})	1	100	0	0	0	100
2	(Bk; {B70, Bk30}), (Coinbase; {Bk100})	2	130	70	0	0	200
3	(B; {R5, B65})	1	130	65	5	0	200
4	(R; {O4, R1}),(Bk30,Bk100; {Bk130})	2	130	65	1	4	200

Assume:

- Addresses receiving coinbase previously controlled no UTXO
- Other addresses also initially controlled no UTXO
- No other TX associated with these addresses in the blockchain
- We are also ignoring (optional but advisable) transaction fees

Note: Numbers are arbitrary/illustrative

Bitcoin transactions

Period	Transfers	# Transactions	Balances				Total BTC
			Bk	B	R	O	
1	(Coinbase; {Bk100})	1	100	0	0	0	100
2	(Bk; {B70, Bk30}), (Coinbase; {Bk100})	2	130	70	0	0	200
3	(B; {R5, B65})	1	130	65	5	0	200
4	(R; {O4, R1}),(Bk30,Bk100; {Bk130})	2	130	65	1	4	200

Comments:

- **Note:** Total 'money supply' only $\Uparrow$ with coinbase tx
  - Other transactions simply divide up value
- Had we allowed for payment of transaction fees (to miner) we would have had to adjust the numbers
- Second coinbase *could* have been for mining block containing transaction implementing (Bk; {B70, Bk30})