

# Utreexo - What is it?

What it is and how it works

# Utreexo

## Agenda for today

# What is it?

# How does it work?

**What is it?**

**Turn the UTXO set into a  
Merkle tree**

# The UTXO set

## Quick overview

- TXO - An Output of a TX (transaction)

# Typical TX

8ad63210d059908229f00ad177469d9f16a5d37a7ea64cede9c92684ff7cf06e

DETAILS +

#0 f758ad929fbefac99ef0d1f8b13990fa75efb517ef6c919178c8af7308f145 0.04852489 BTC  
07:1

>

#0 17dSwJytZBszAafyWrK8Pue7rsRh9s5xeJ 0.04033963 BTC

#1 bc1qwqdg6squsna38e46795at95yu9atm8azzmyvckulcc7kytlcckxswvvej 0.00758526 BTC

1 CONFIRMATION 0.04792489 BTC

source: <https://blockstream.info/tx/8ad63210d059908229f00ad177469d9f16a5d37a7ea64cede9c92684ff7cf06e>

# Outputs of a TX

8ad63210d059908229f00ad177469d9f16a5d37a7ea64cede9c92684ff7cf06e

DETAILS +

#0 f758ad929fbefac99ef0d1f8b13990fa75efb517ef6c919178c8af7308f145 0.04852489 BTC  
07:1

>

#0 17dSwJytZBszAafyWrK8Pue7rsRh9s5xeJ 0.04033963 BTC

#1 bc1qwqdg6squsna38e46795at95yu9atm8azzmyvckulcc7kytlcckxswvvzej 0.00758526 BTC

1 CONFIRMATION 0.04792489 BTC

# The UTXO set

## Quick overview

- TXO - An Output of a TX (transaction)
- UTXO - Unspent TXO



# The UTXO set

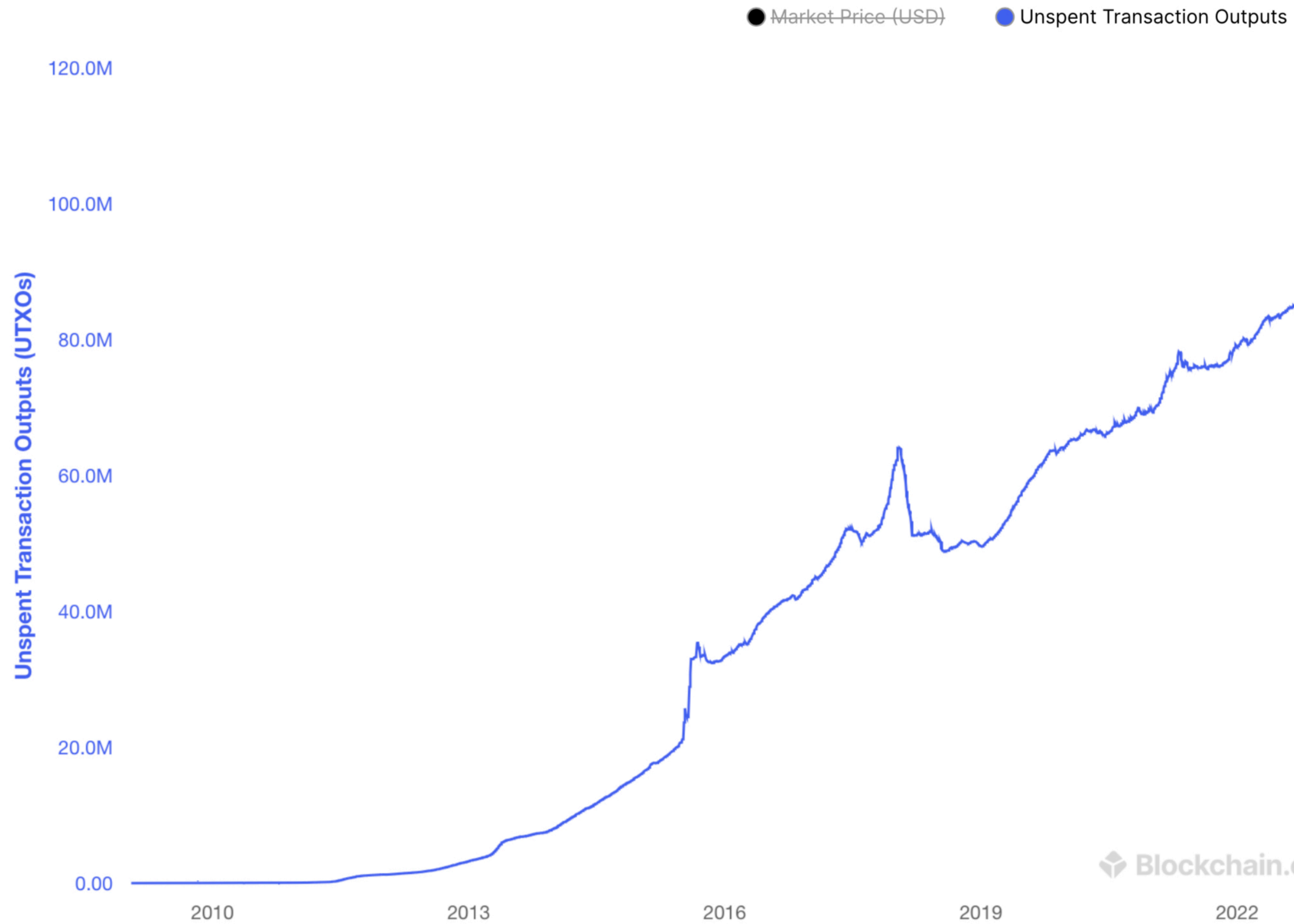
## Quick overview

- TXO - An Output of a TX (transaction)
- UTXO - Unspent TXO
- UTXO set - Set of all UTXOs

# The UTXO set

Why merklize it?

- Puts a bound to the UTXO set growth



$O(N) \rightarrow O(\log N)$

**N=8 billion UTXOs, at 59B per entry**

**472GB -> 608B**

# The UTXO set

## Why merklize it?

- Puts a bound to the UTXO set growth
- Allows for tiny nodes

**5.8G chainstate/**

**384B**

**chainstate/**



# Immediate node bootstrap

# AssumeUTXO

Quick way to sync up

- Receive a UTXO set hash commitment with the binary

# AssumeUTXO

Quick way to sync up

- Receive a UTXO set hash commitment with the binary
- Download the ~5GB of UTXO set from torrents

# AssumeUTXO

Quick way to sync up

- Receive a UTXO set hash commitment with the binary
- Download the ~5GB of UTXO set from torrents
- Download blocks from peers

# AssumeUtreexo

Quicker way to sync up

- Receive the UTXO set with the binary

# AssumeUtreexo

Quicker way to sync up

- Receive the UTXO set with the binary
- Download blocks/tx from peers



Download Bitcoin Core

# The UTXO set

## Why merklize it?

- Puts a bound to the UTXO set growth
- Allows for tiny nodes
- **Faster block validation**



# Current block validation

## Sequential validation

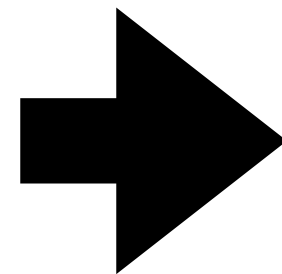


Genesis

# Current block validation

## Sequential validation

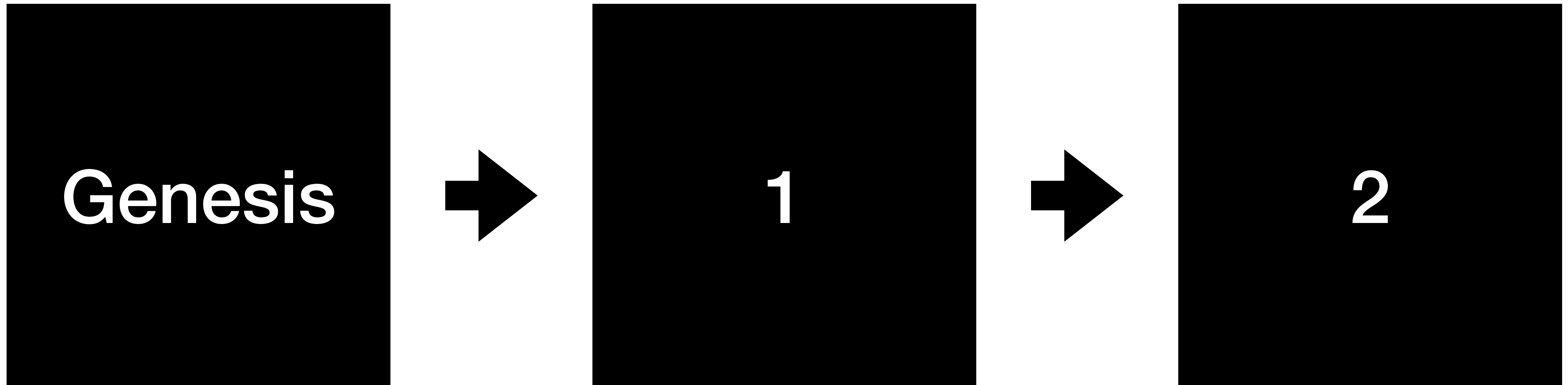
Genesis



1

# Current block validation

## Sequential validation



# Parallel block validation

Replay blocks in any order

- Utreexo roots committed into the binary
- Process blocks starting from any of the roots that are committed
- <https://github.com/mit-dci/utcd/blob/master/chaincfg/mainnetroots.go>

# With Utreexo

Efficient parallel validation

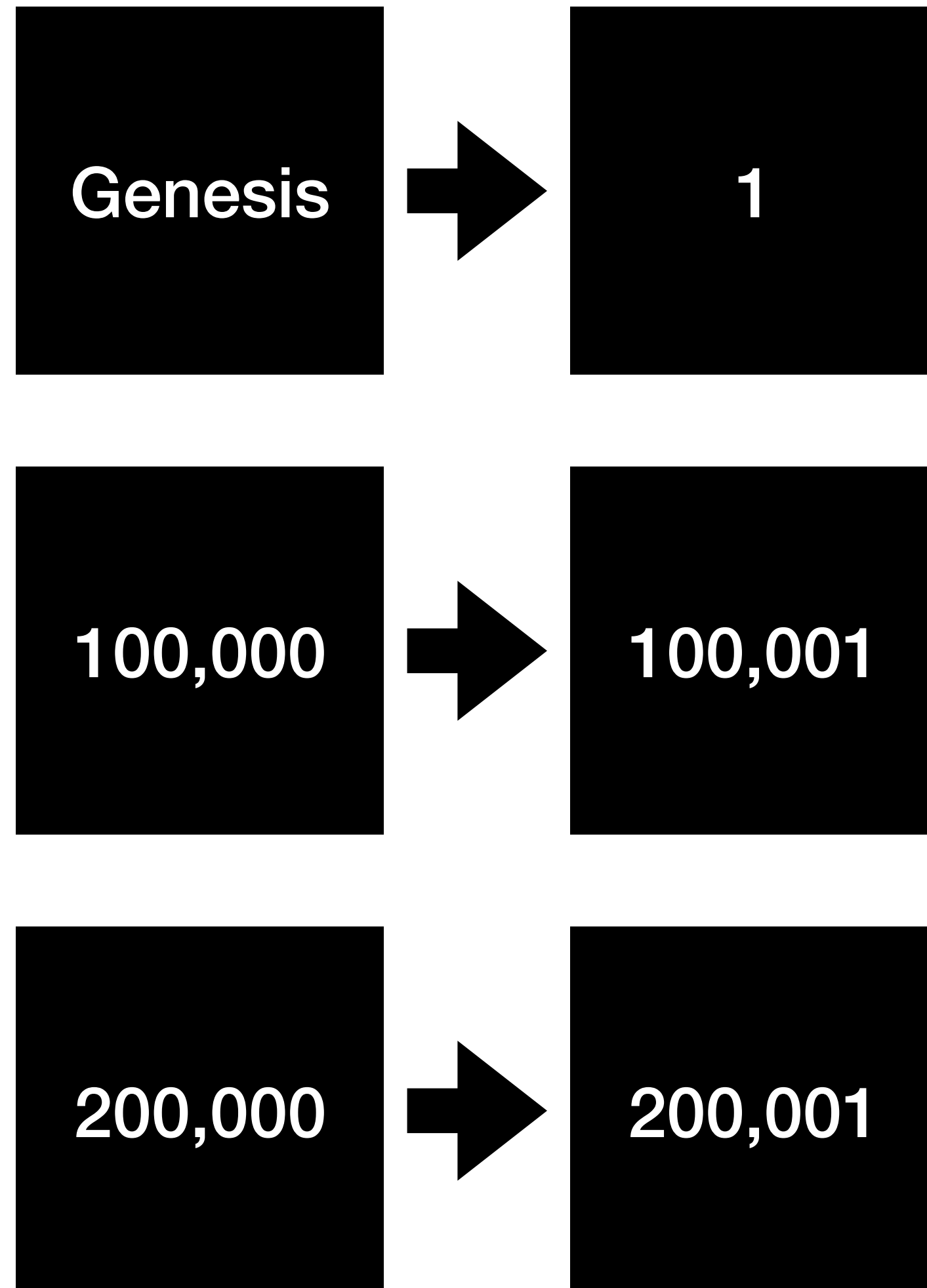
Genesis

100,000

200,000

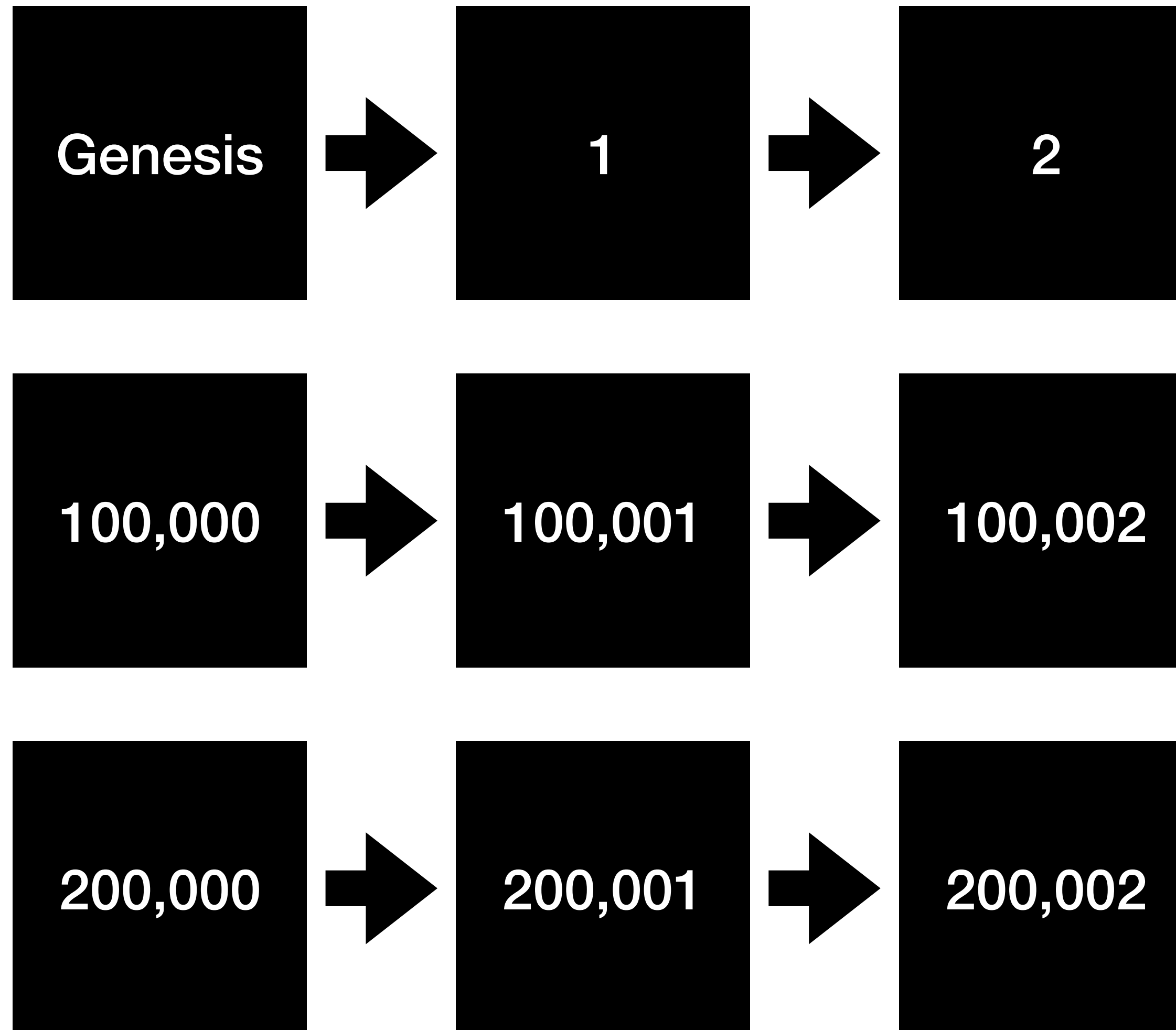
# With Utreexo

Efficient parallel validation



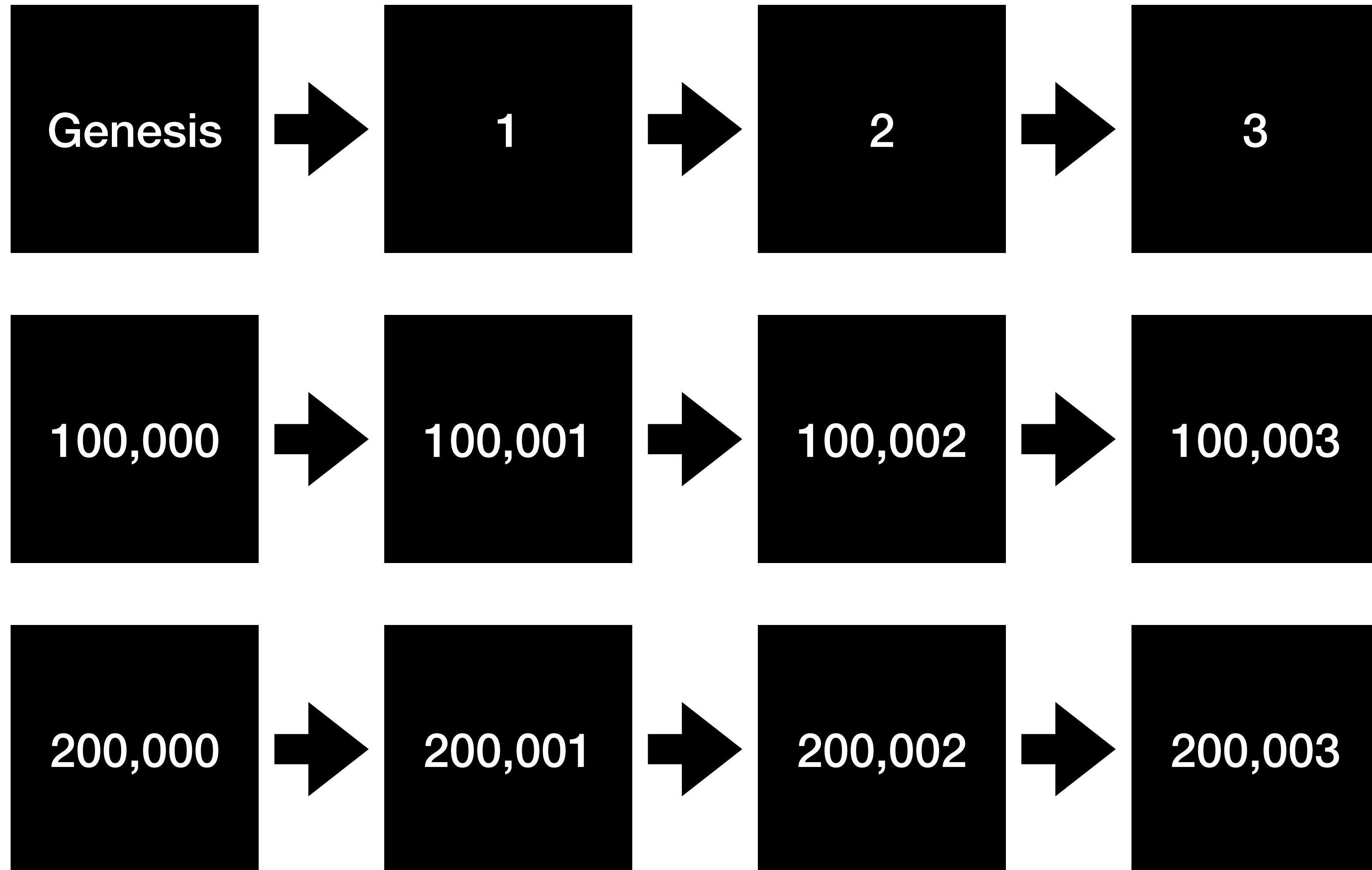
# With Utreexo

Efficient parallel validation



# With Utreexo

Efficient parallel validation

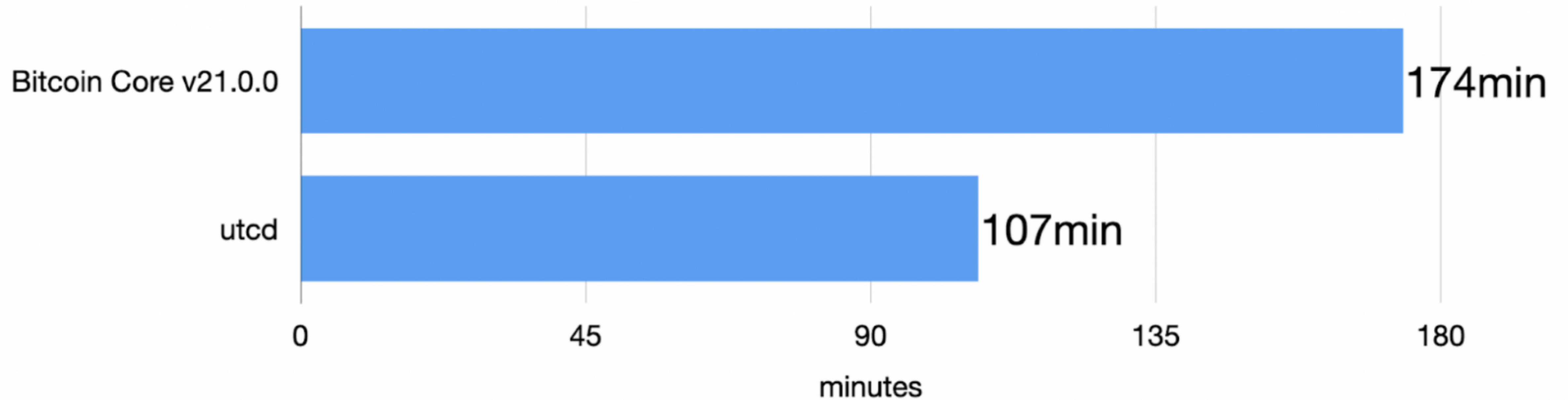




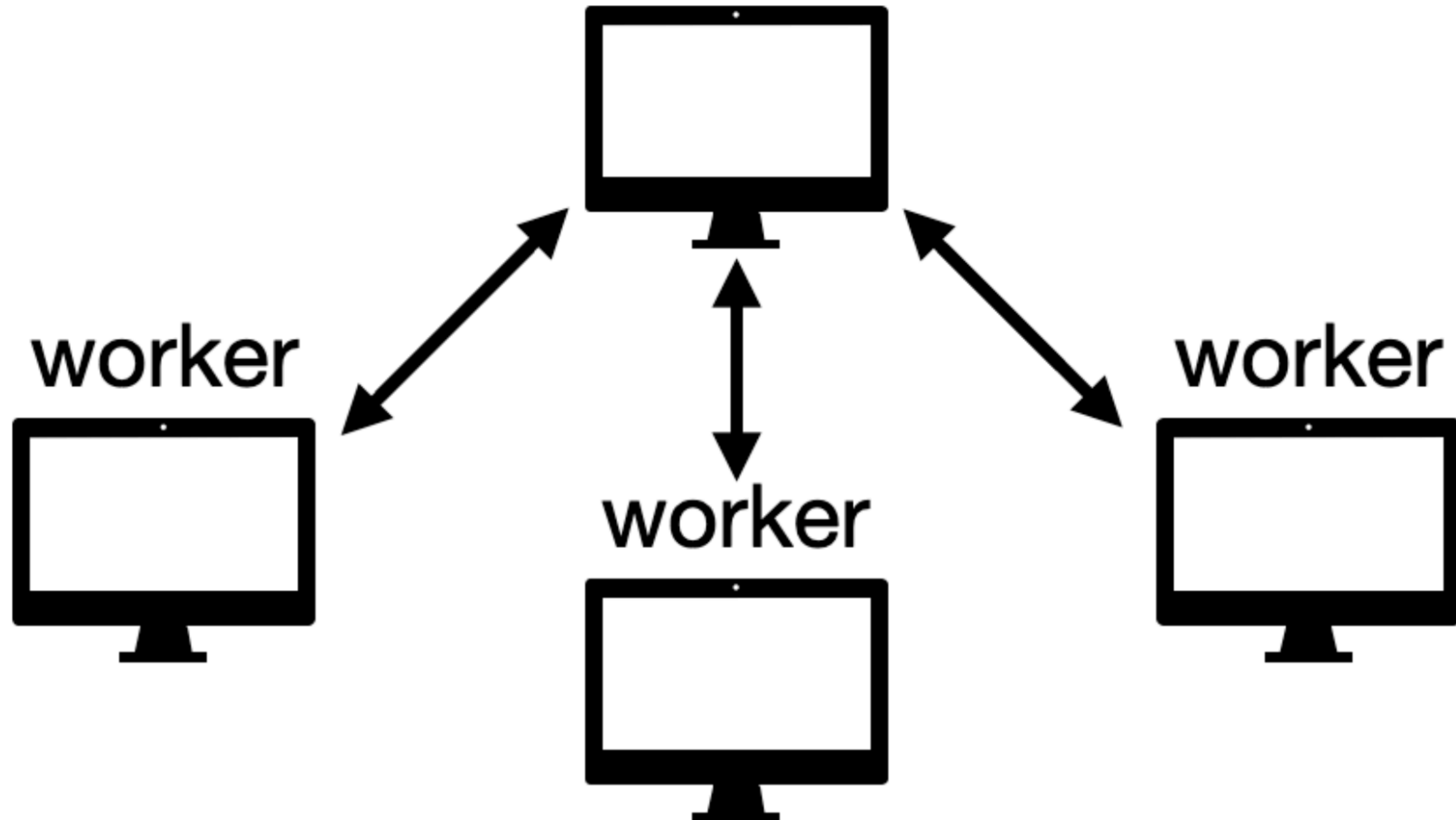
# ***Efficient* Parallel Validation**

**SHA256** 🐰 **>** **Map access** 🐢

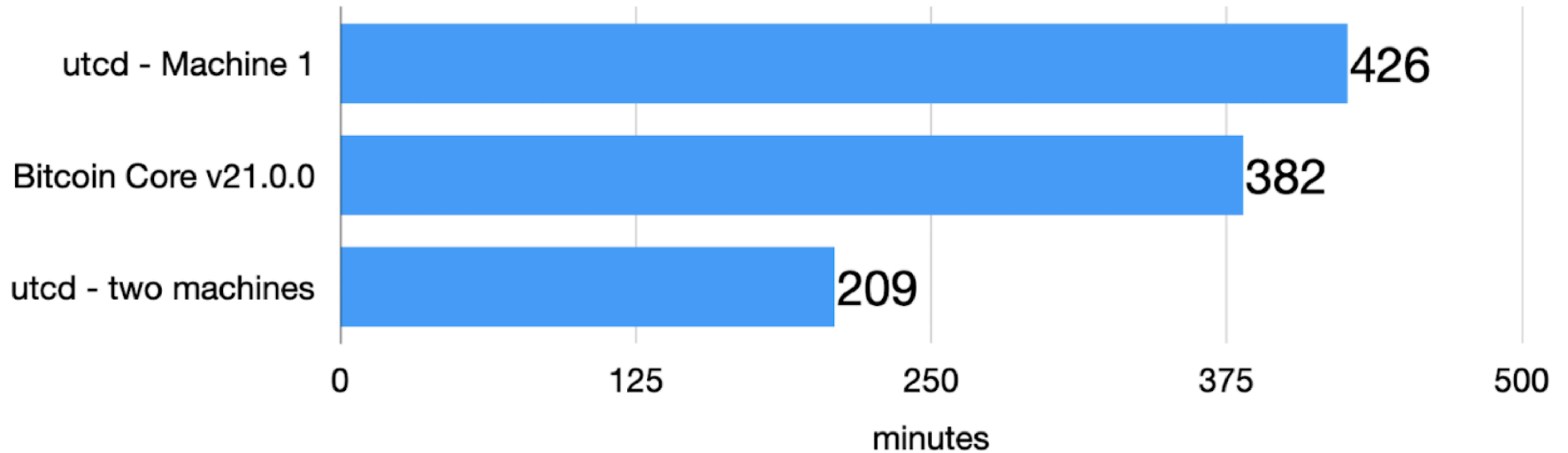
## Initial Block Download Speeds - local nodes (default mode, no signature checks until block 654,683)



# Coordinator/worker



## Initial Block Download Speeds to block 671,000 (Full signature check)



# The UTXO set

## Why merklize it?

- Puts a bound to the UTXO set growth
- Allows for tiny nodes
- Faster block validation
- Define consensus without leveldb

# **The Block validation**

## **Basic steps**

### **1. Check Header (PoW)**

# **The Block validation**

## **Basic steps**

1. Check Header (PoW)
2. Fetch inputs from all TXs (LevelDB)



# The Block validation

## Basic steps

1. Check Header (PoW)
2. Fetch inputs from all TXs (LevelDB)
3. Verify signatures

# **The Block validation**

## **Basic steps**

### **1. Check Header (PoW)**

# The Block validation

## Basic steps

1. Check Header (PoW)
2. Verify Utreexo proof

# The Block validation

## Basic steps

1. Check Header (PoW)
2. Verify Utreexo proof
3. Verify signatures

**How does it work?**

**Replace levelDB**

**What does levelDB do for  
bitcoin?**

# Role of levelDB

It let's you

1. Add a UTXO
2. Delete a UTXO
3. Tell you the existence of a UTXO
4. Provide the data for verification



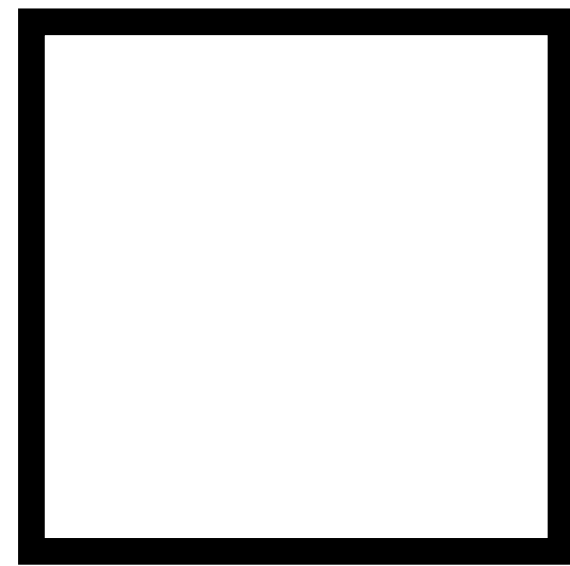
# Role of levelDB

It let's you

1. Add a UTXO
2. Delete a UTXO
3. Tell you the existence of a UTXO
4. Provide the data for verification

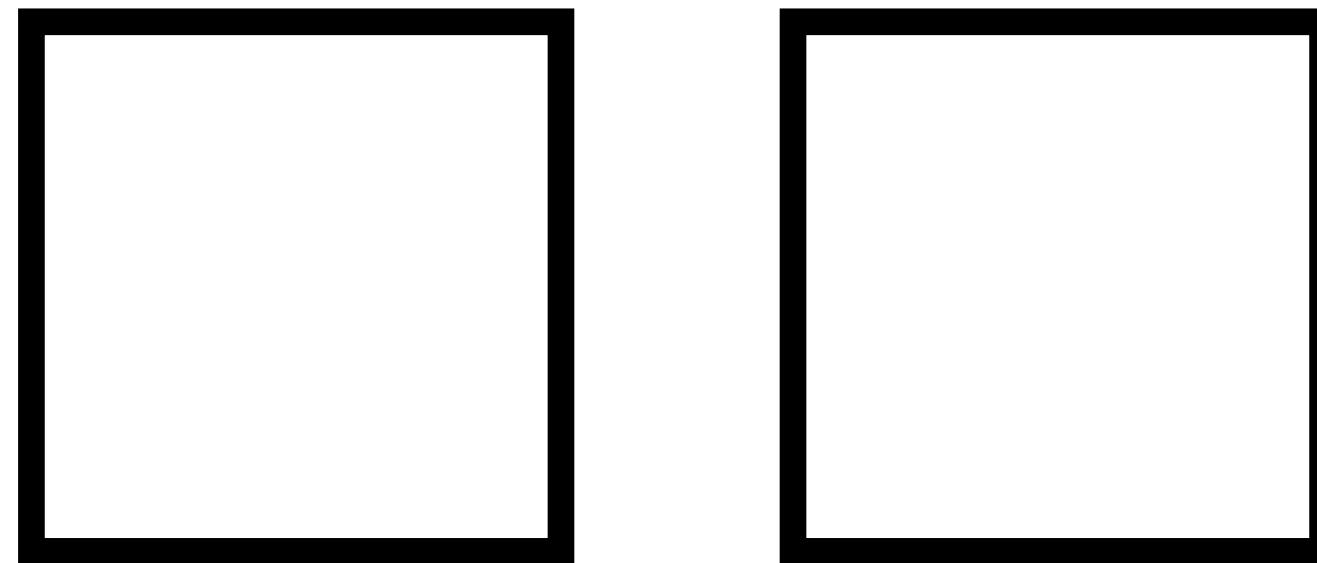
# 1 UTXO

## Single root



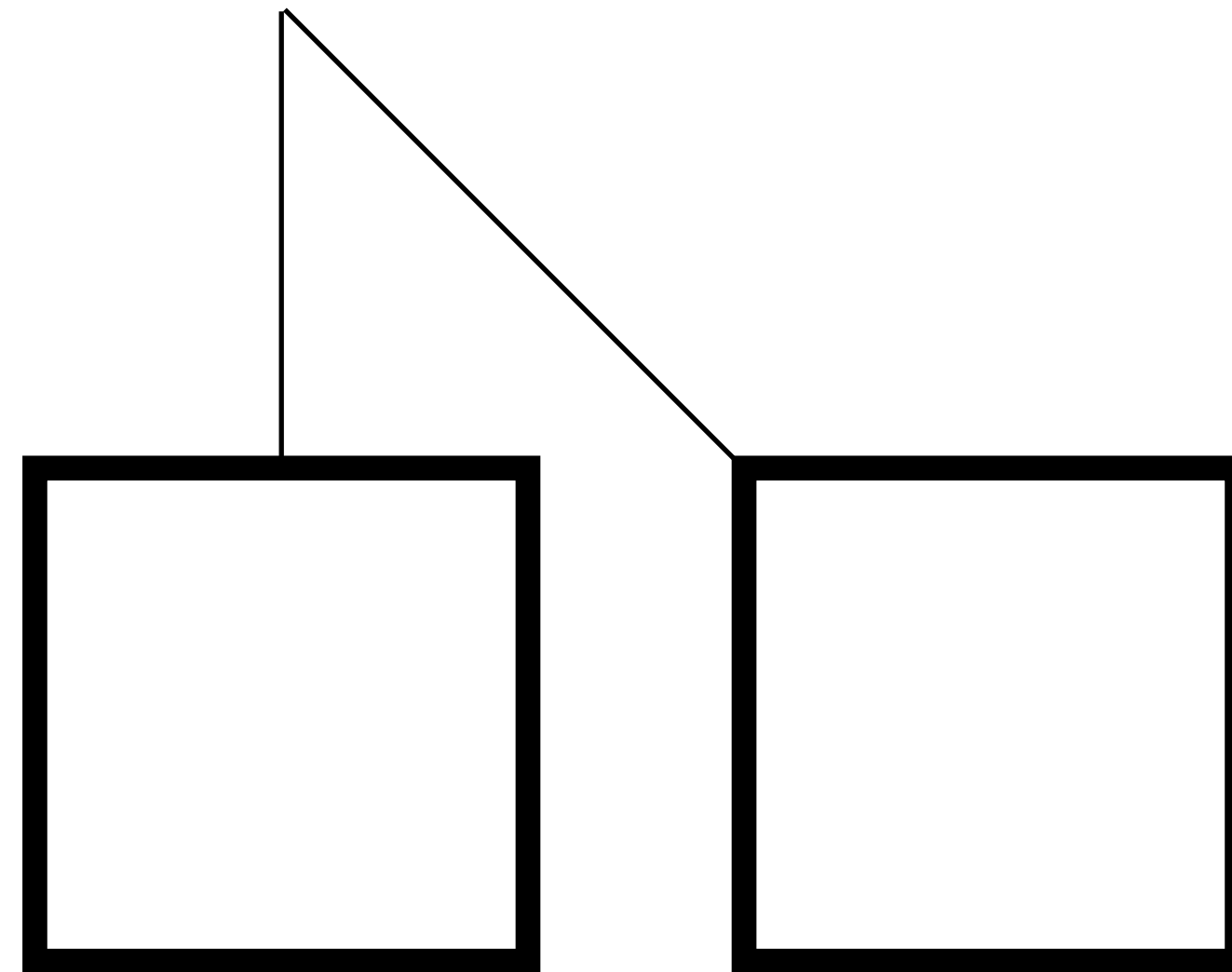
# 2 UTXOs

Add 1 more



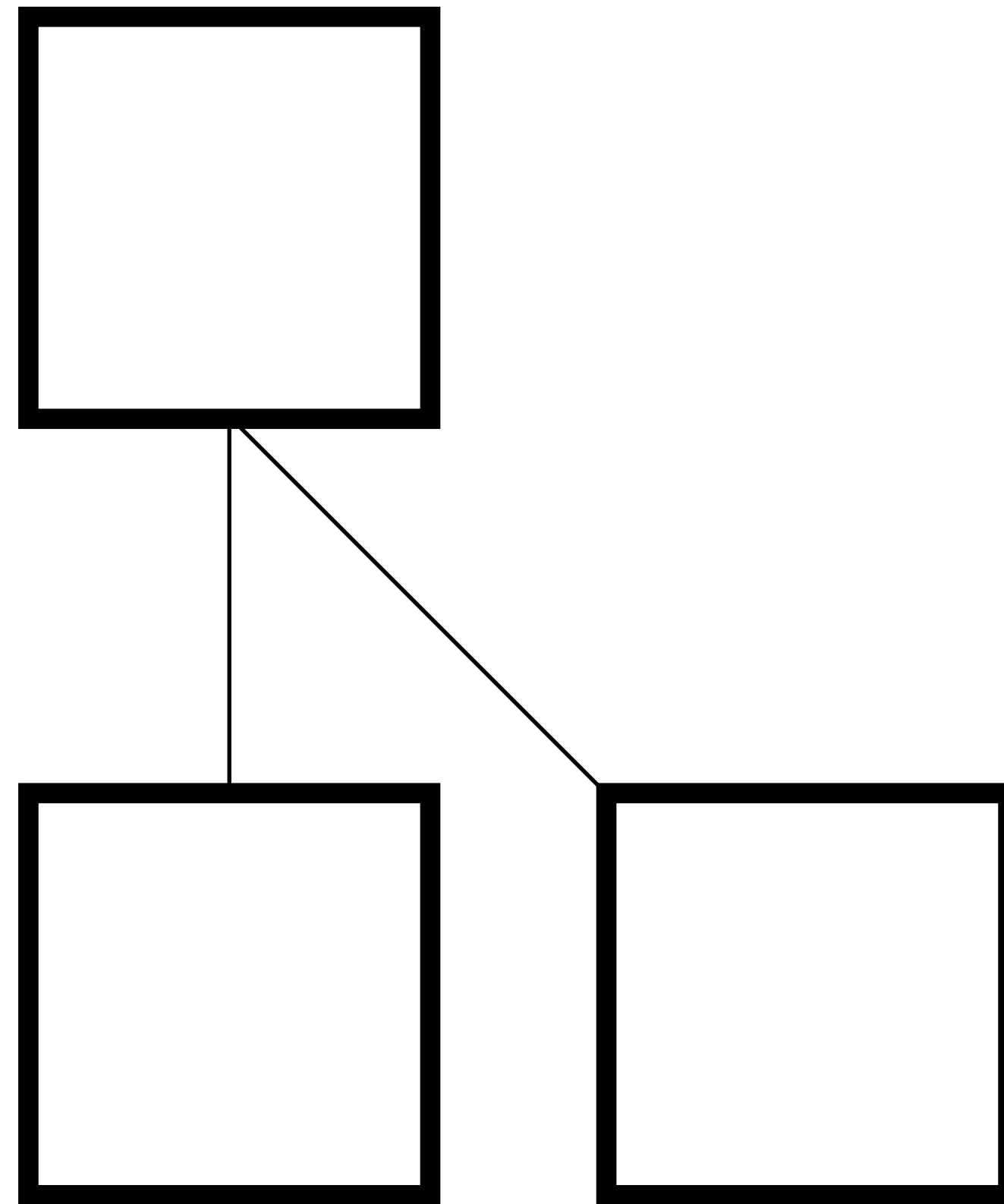
# 2 UTXOs

## Concatenate



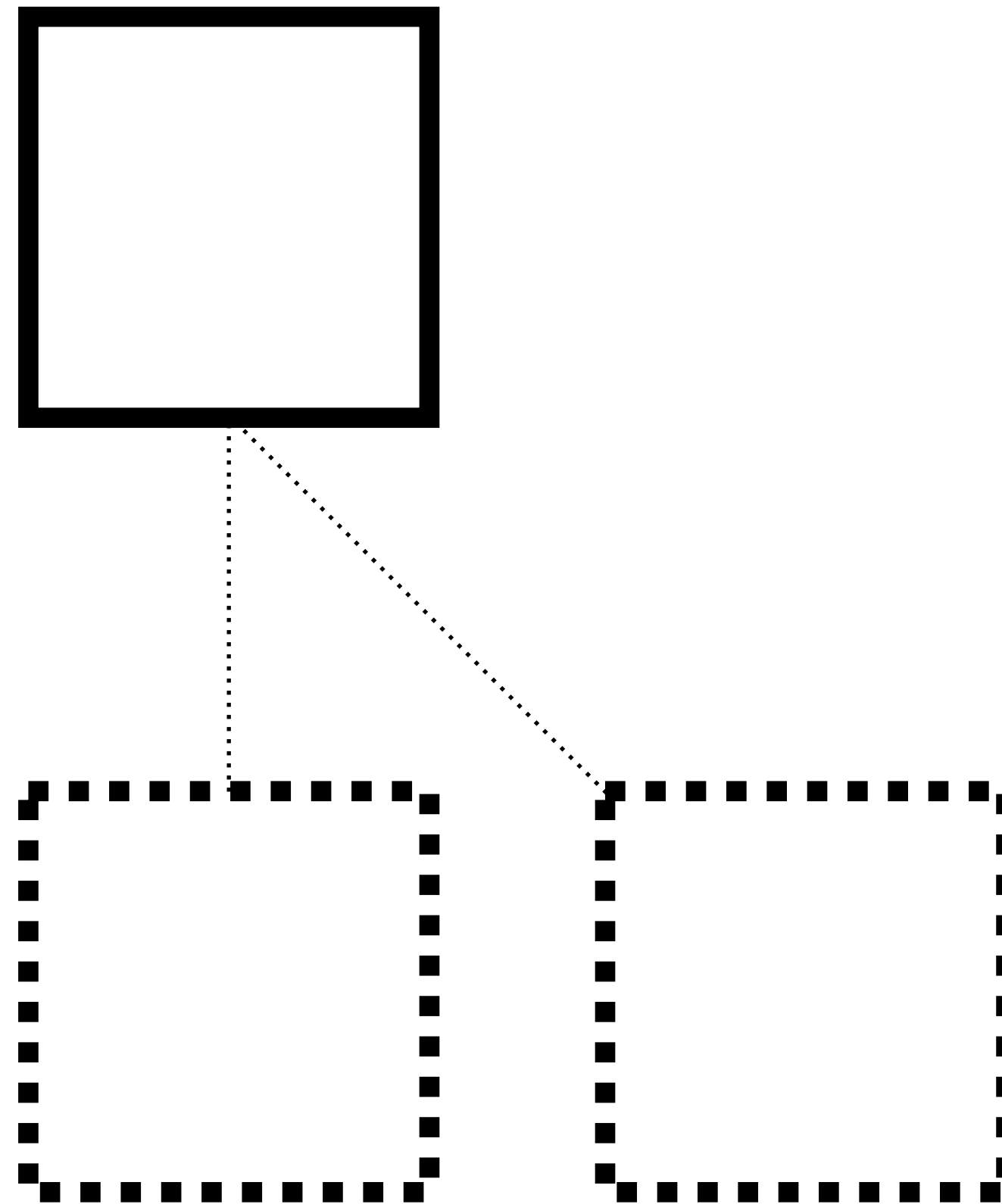
# 2 UTXOs

Hash to create a new root



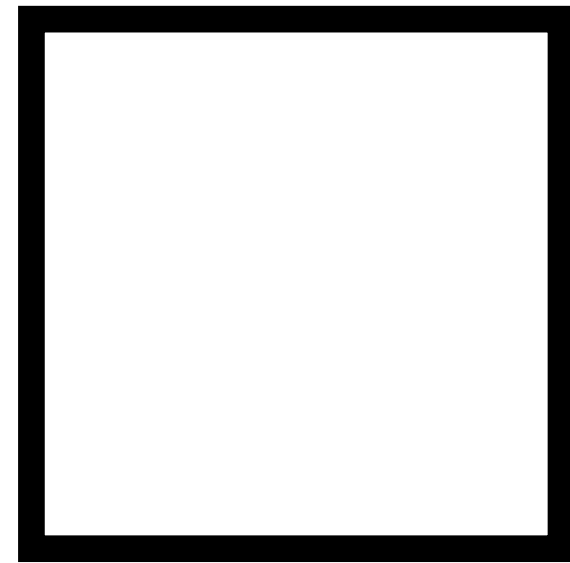
# 2 UTXOs

Can now delete the leaves



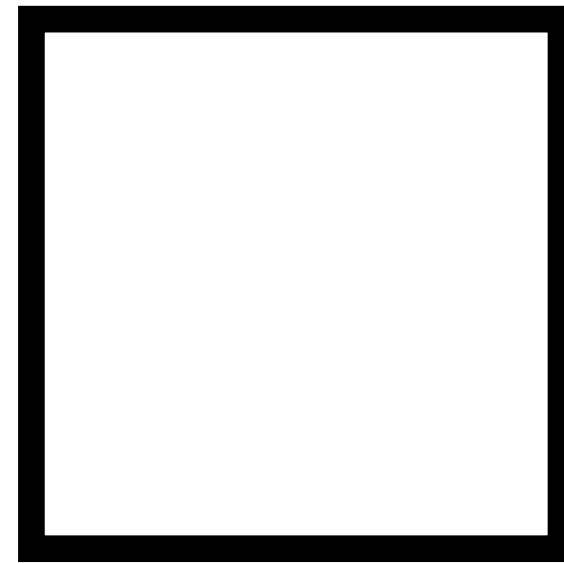
# 2 UTXOs

## Only keep the roots



# 3 UTXOs

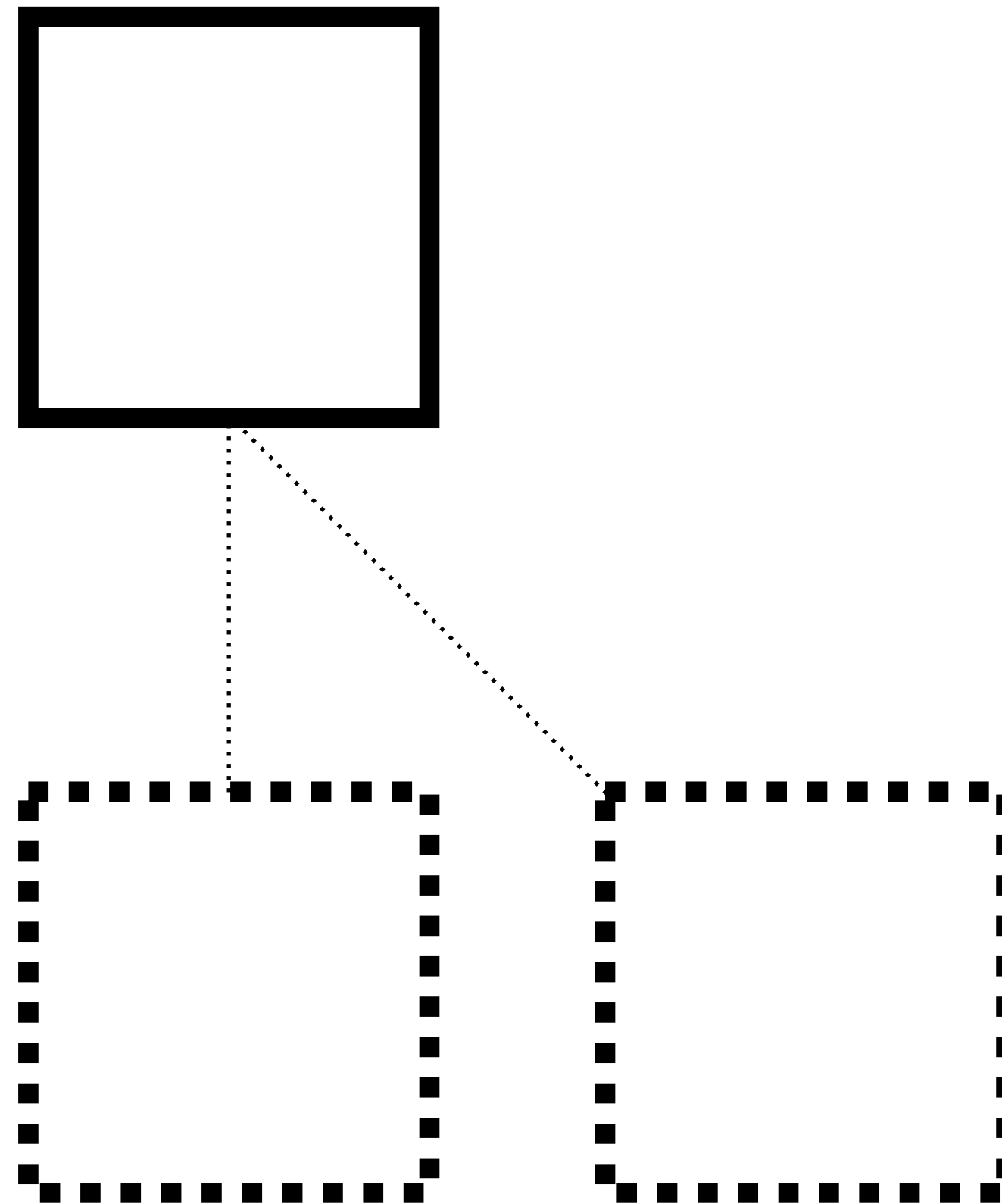
Add 1 more





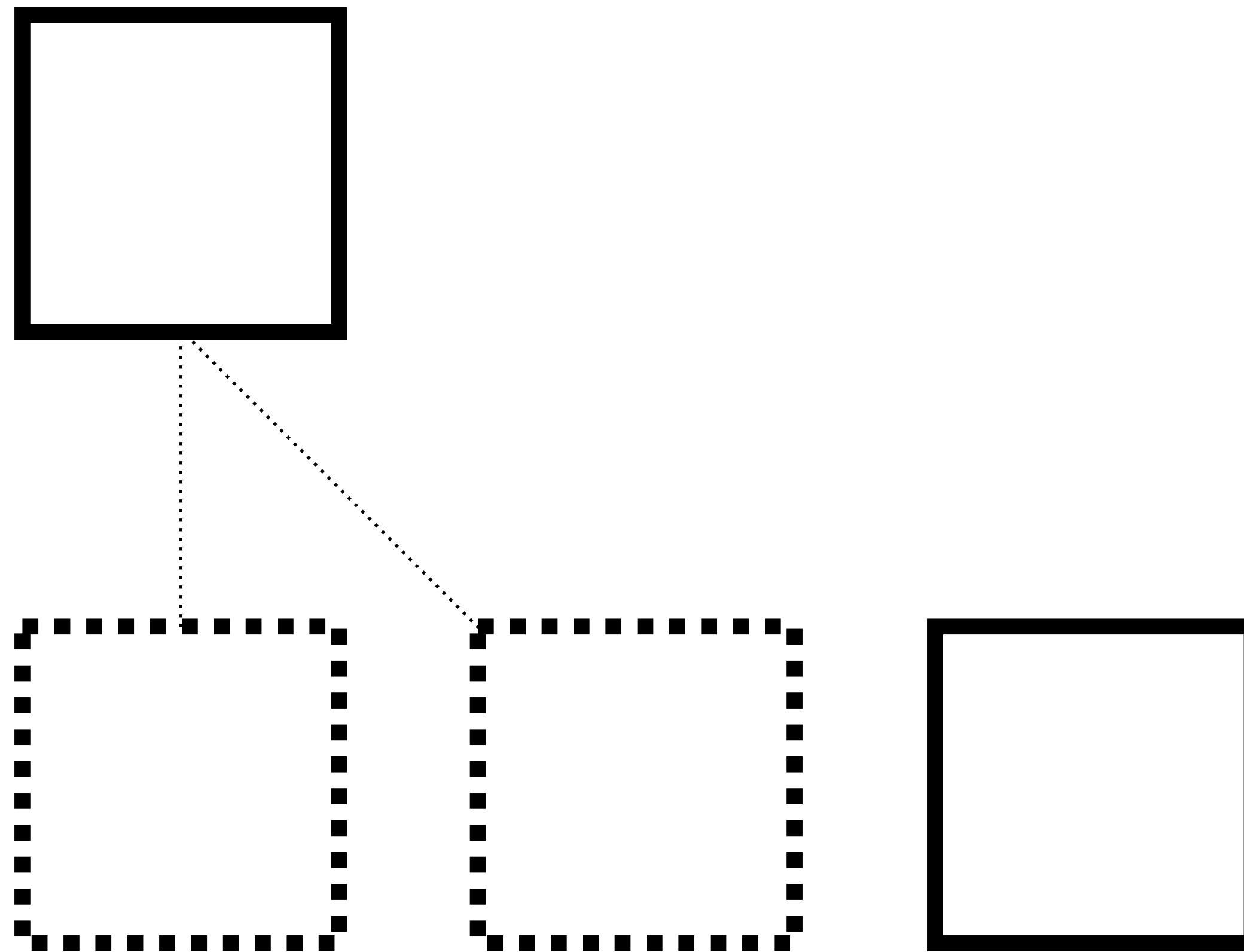
# 3 UTXOs

Add 1 more



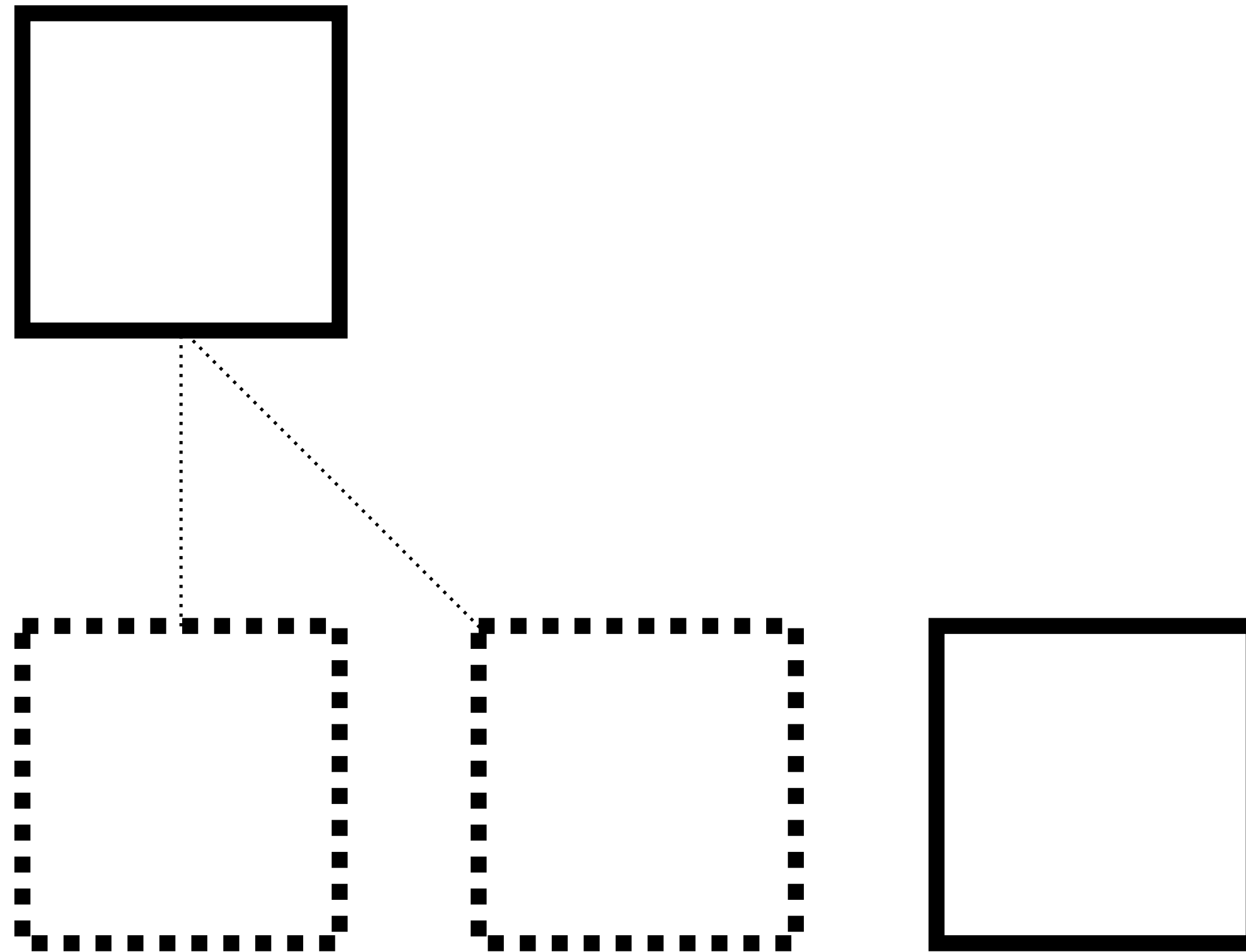
# 3 UTXOs

Add 1 more



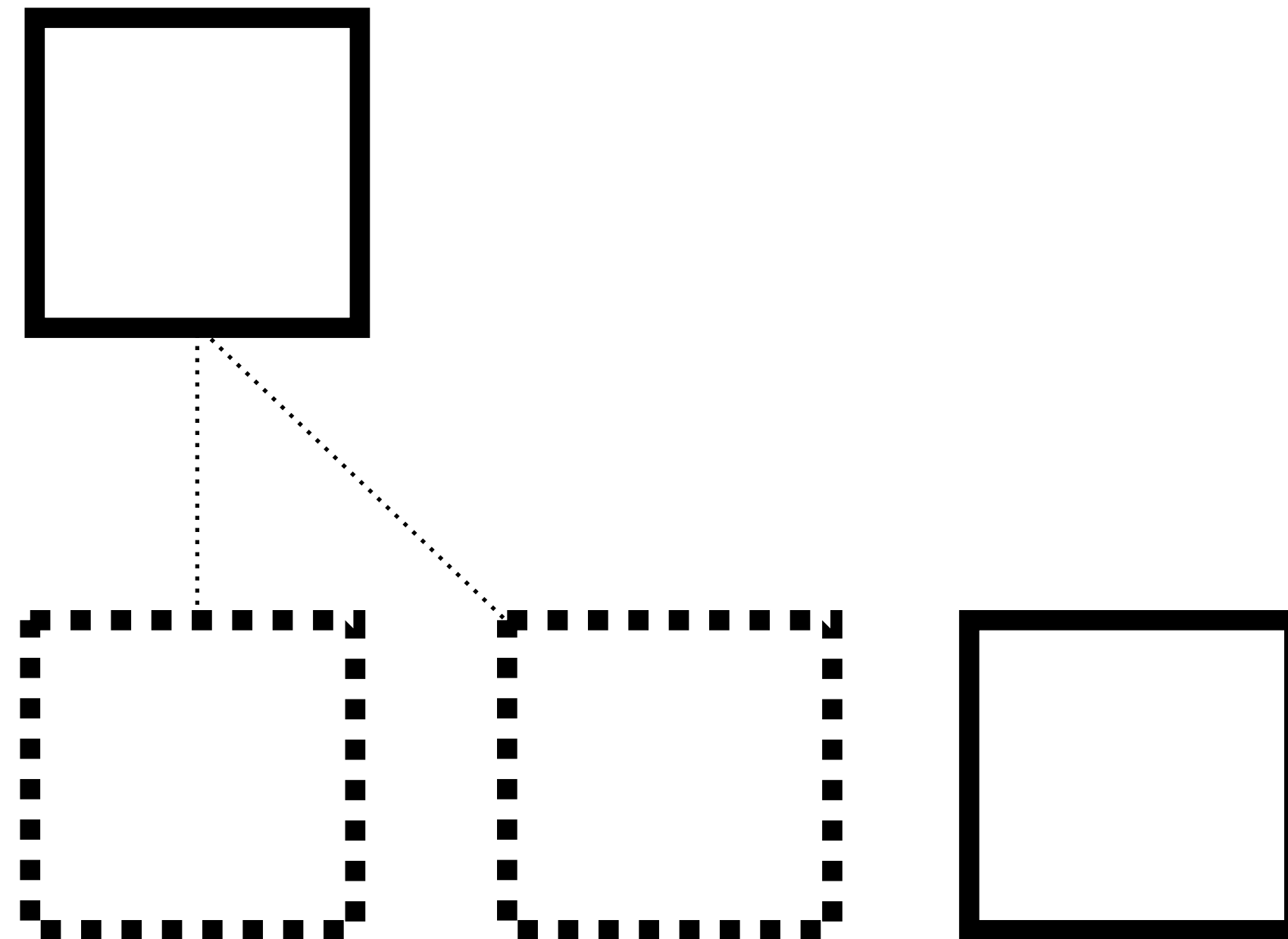
# 3 UTXOs

Nothing to hash with. Becomes a root



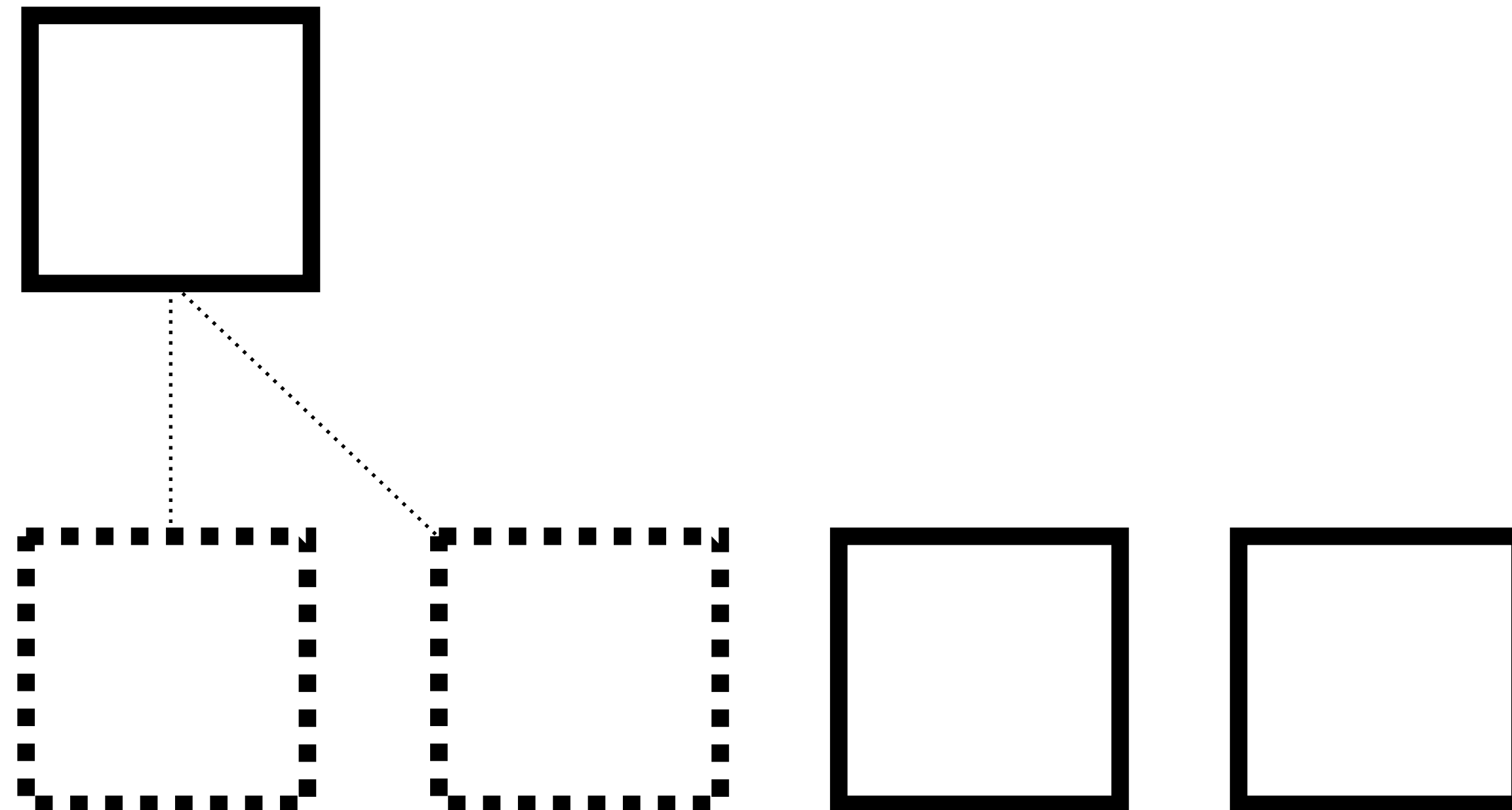
# 4 UTXOs

Another one



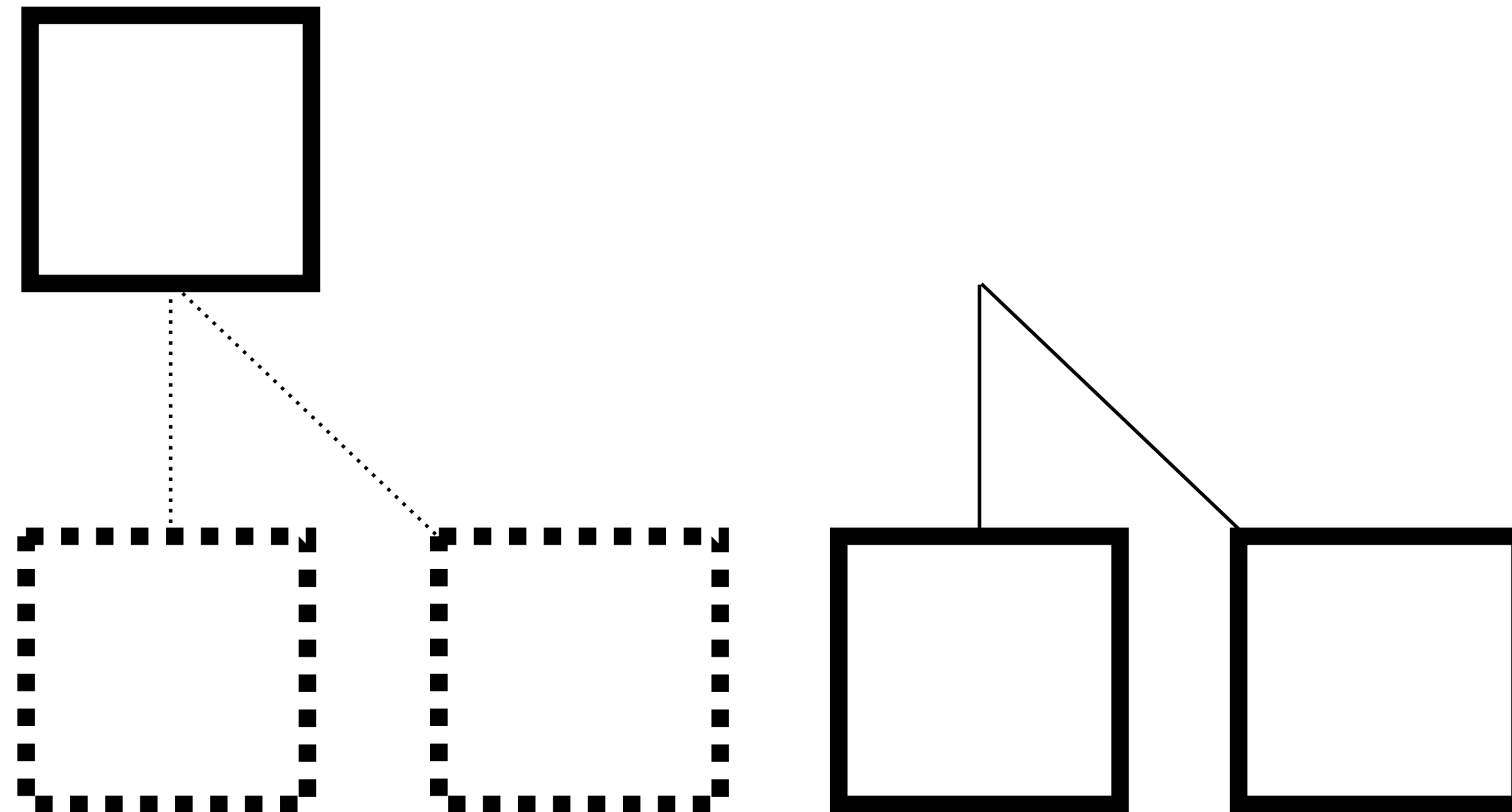
# 4 UTXOs

Another one



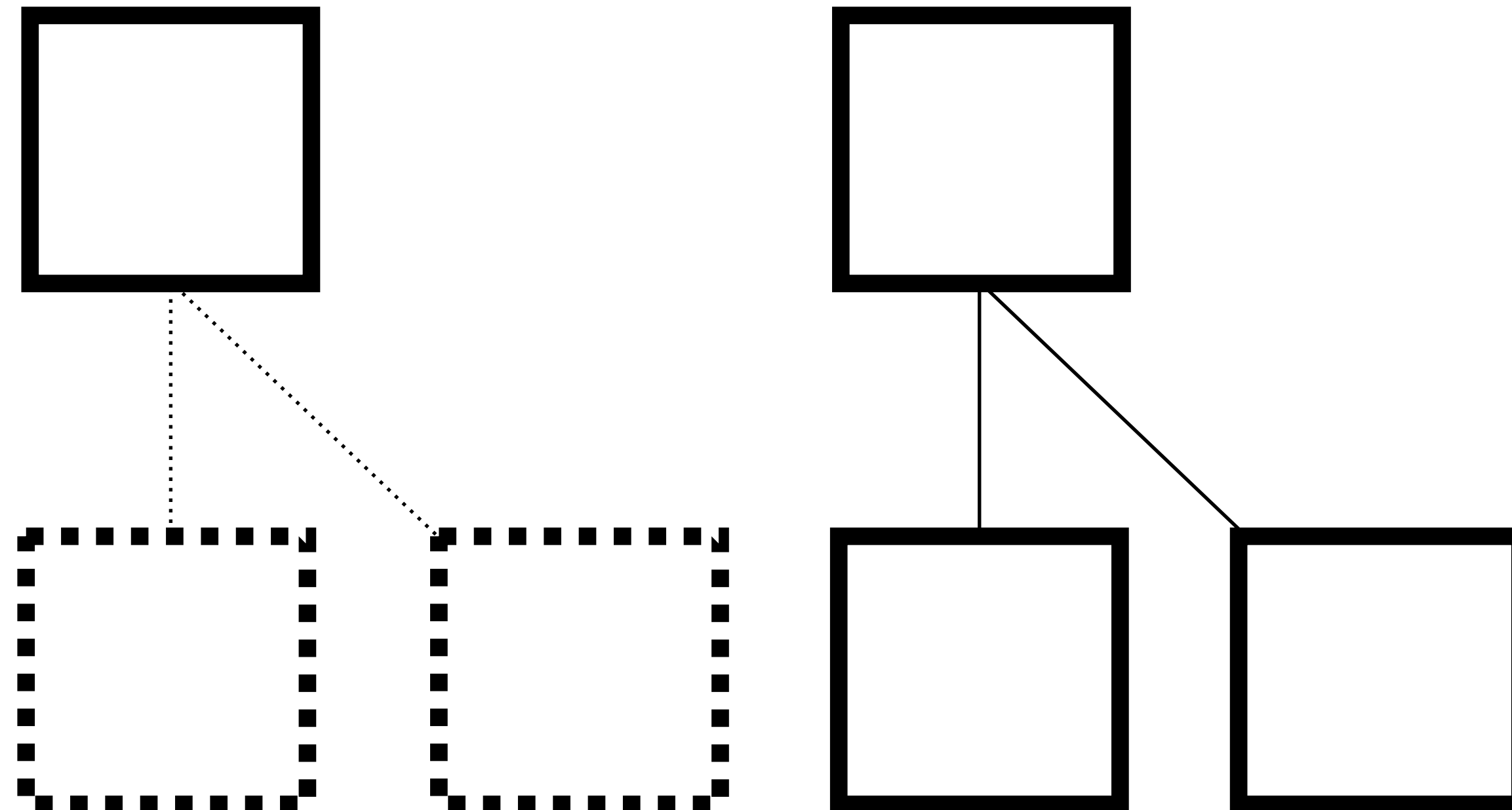
# 4 UTXOs

## Concatenate



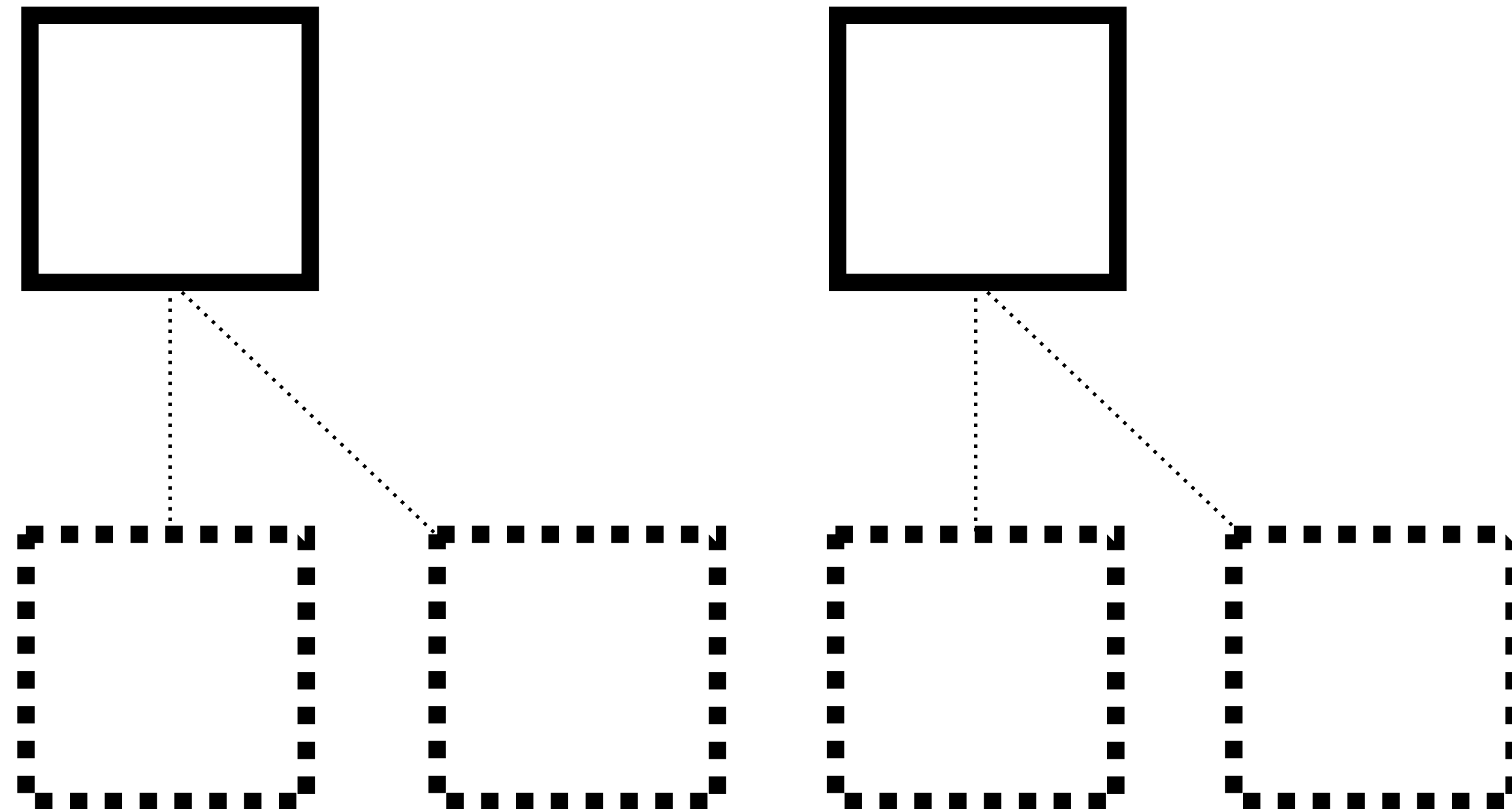
# 4 UTXOs

## Hash



# 4 UTXOs

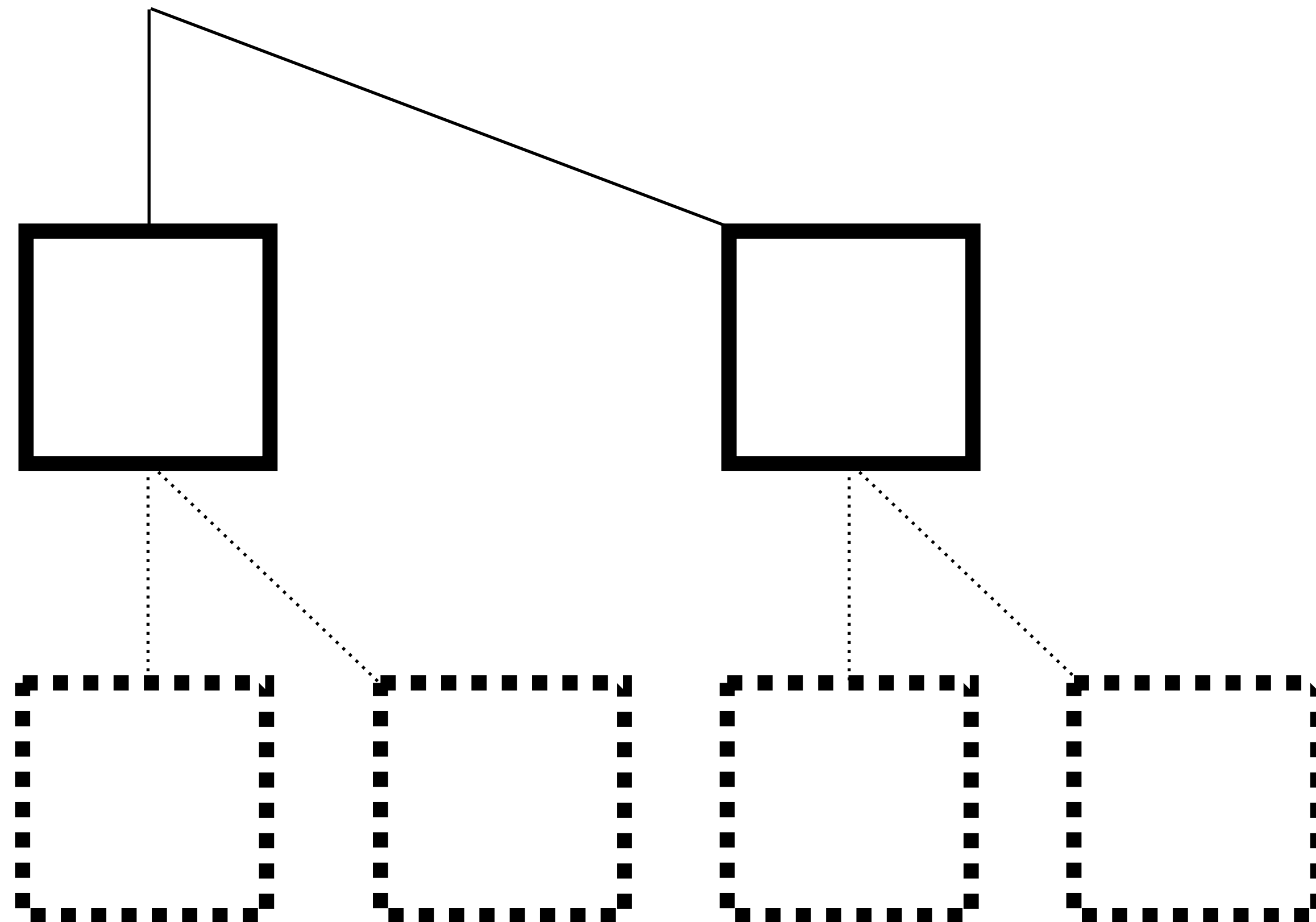
Can now throw away the leaves





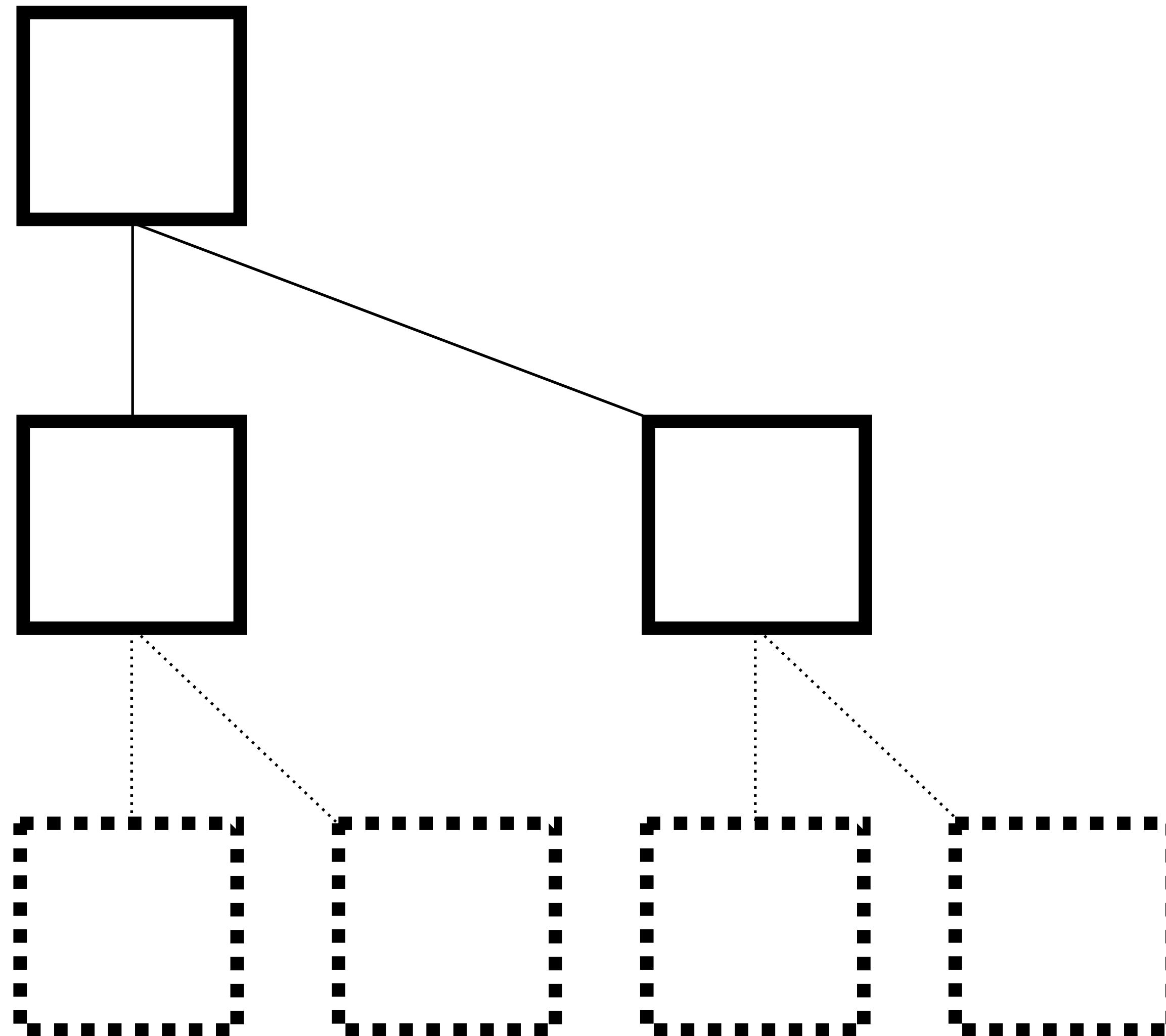
# 4 UTXOs

## Concatenate



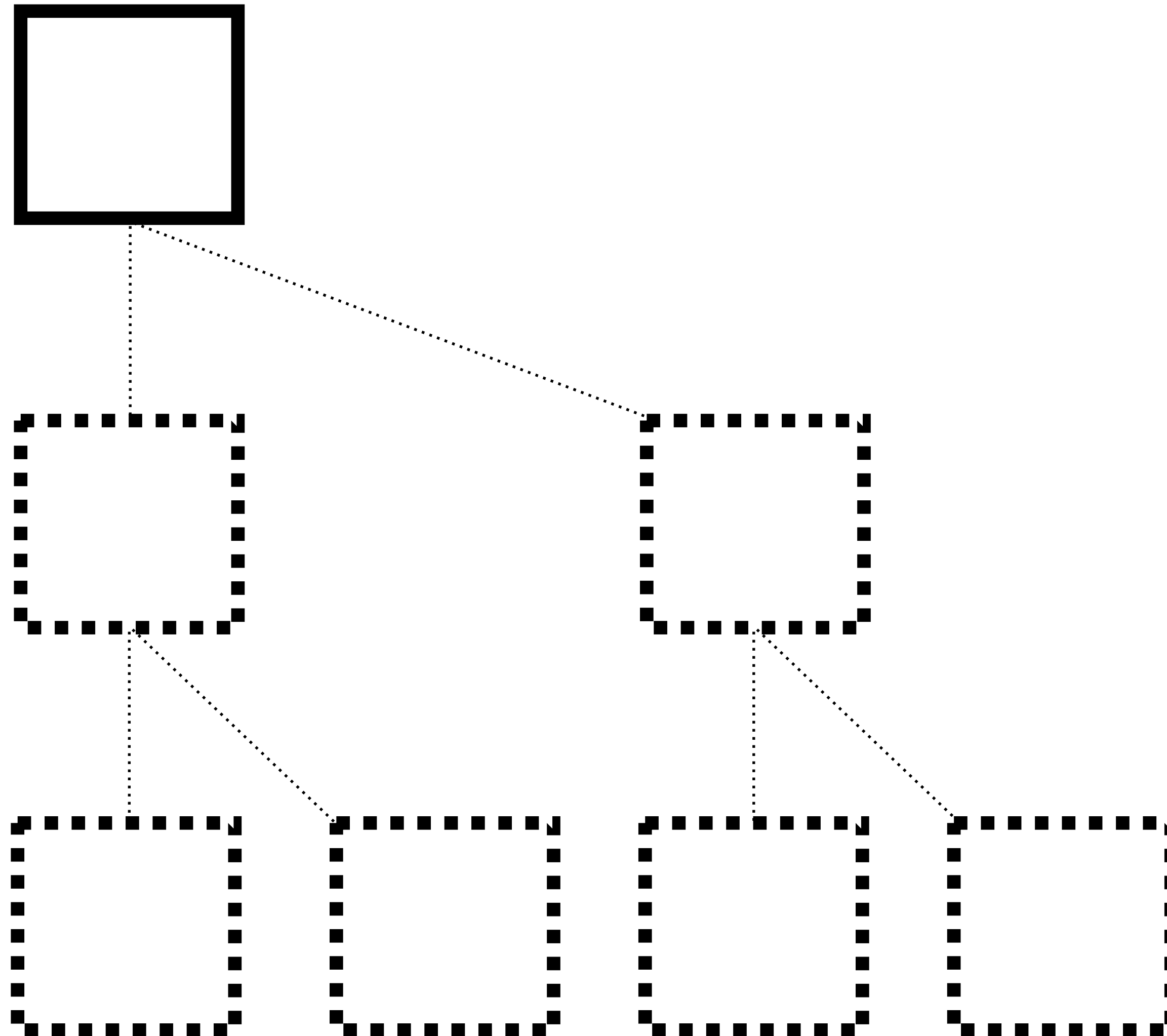
# 4 UTXOs

Hash



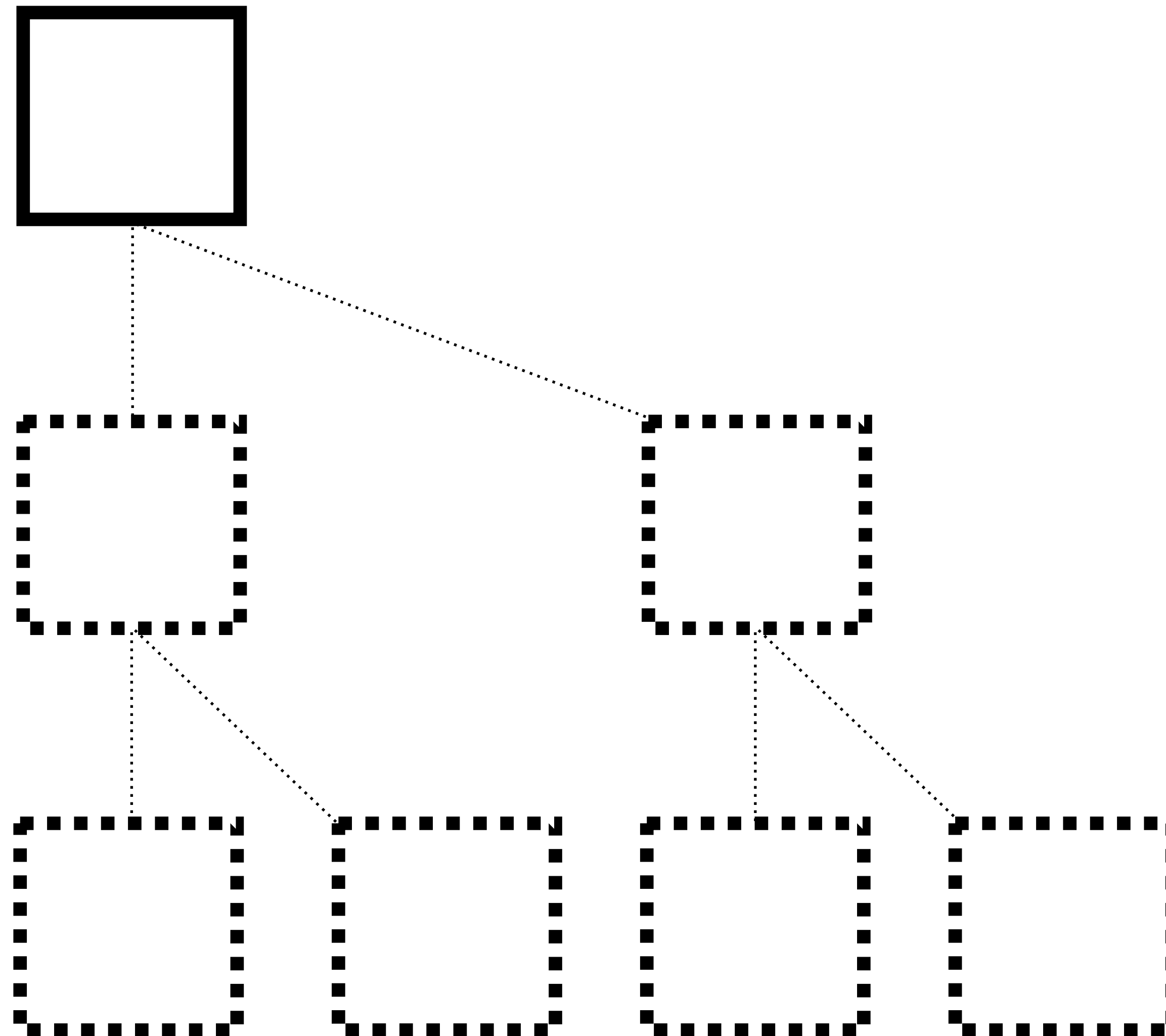
# 4 UTXOs

## Throw away



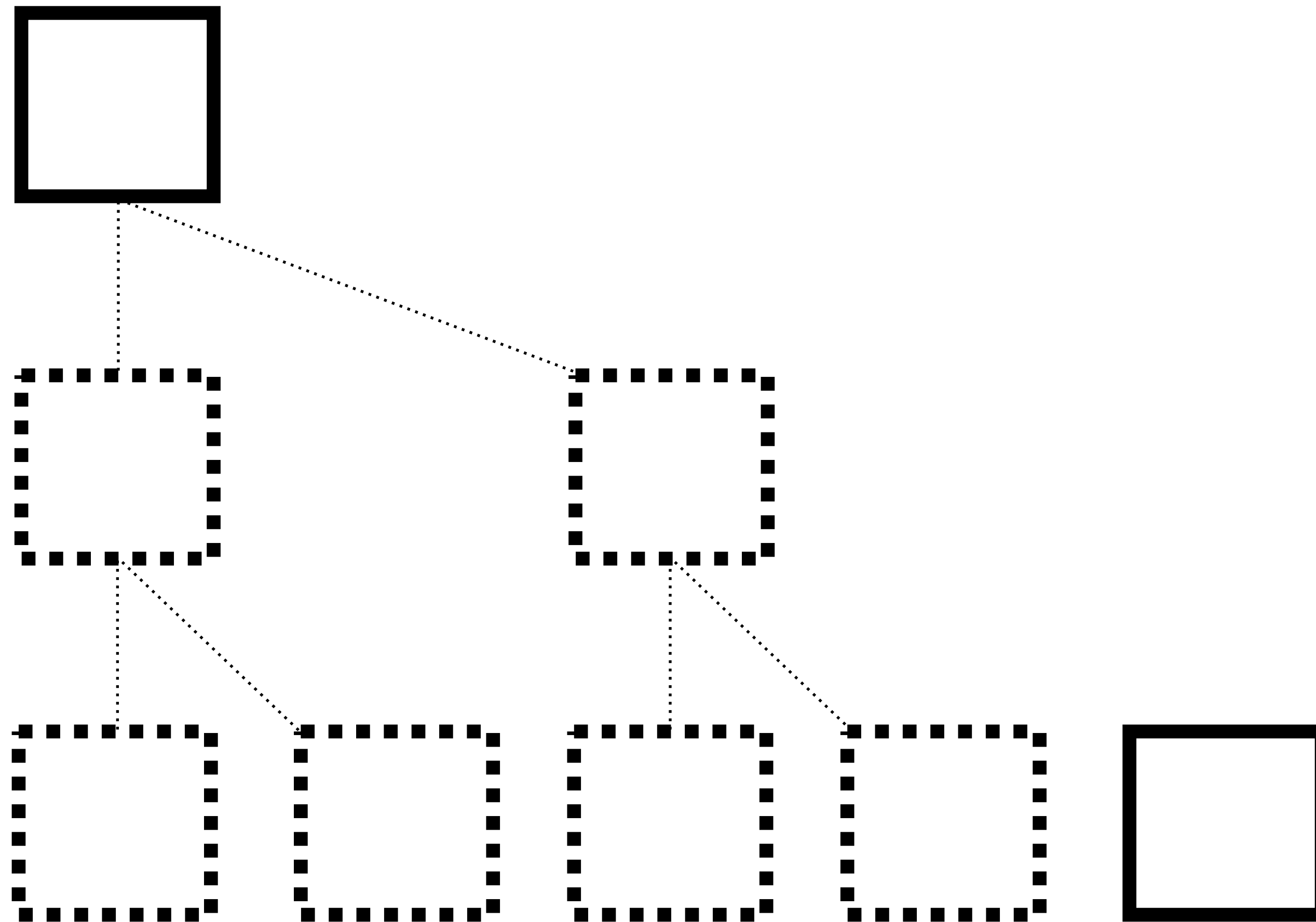
# 4 UTXOs

## Finished tree



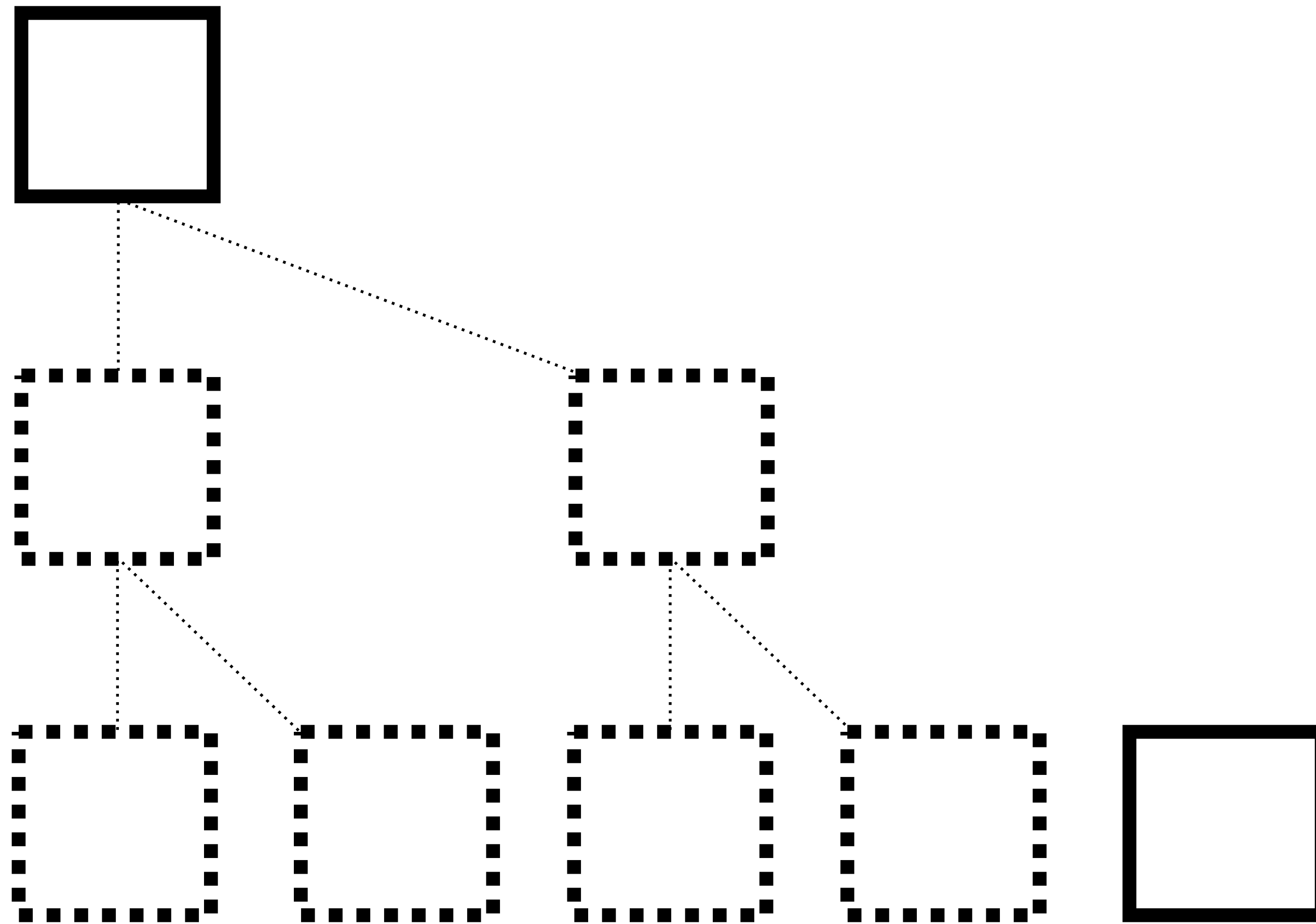
# 5 UTXOs

Add one more



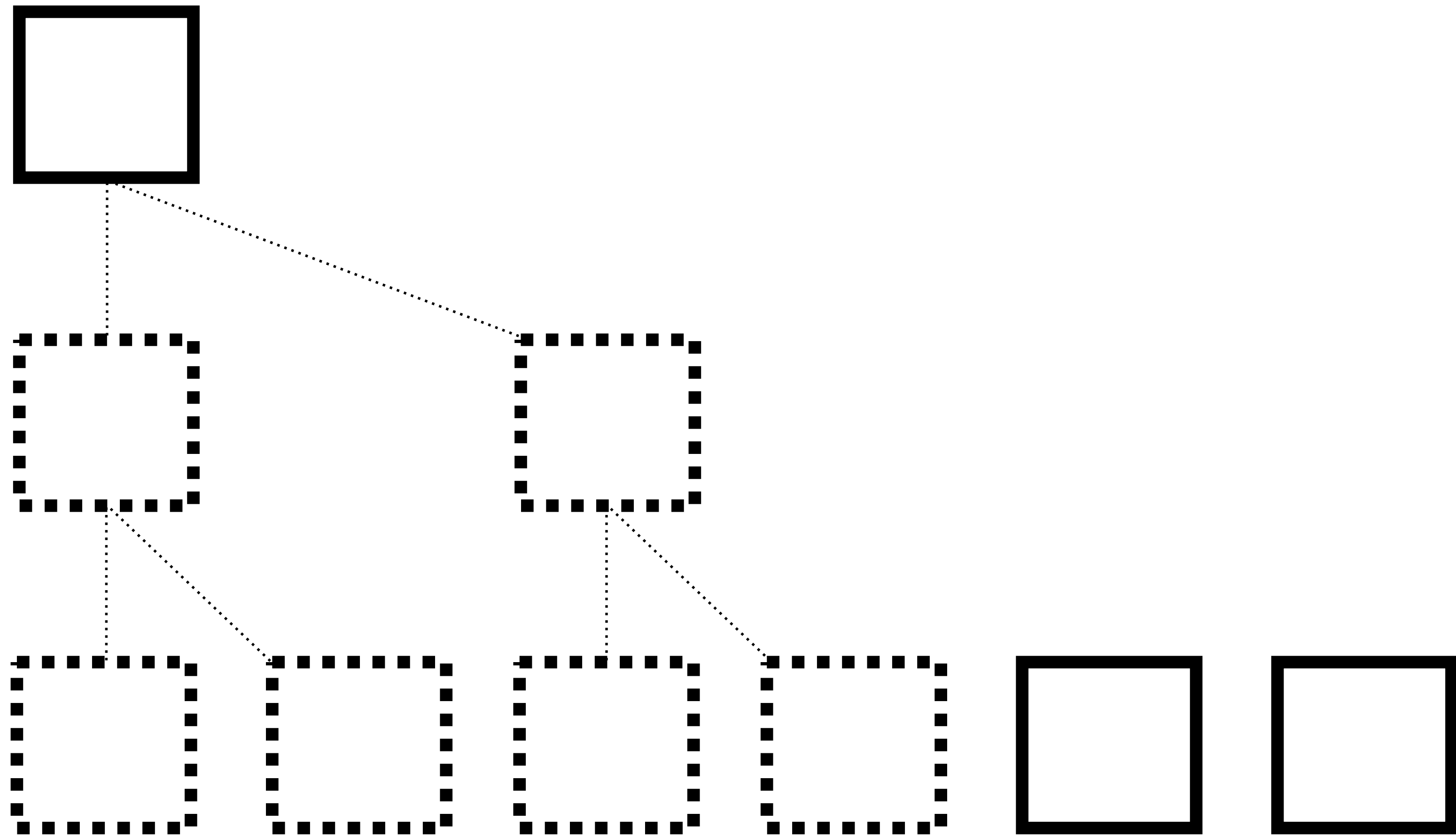
# 5 UTXOs

Nothing to hash with so it becomes a root



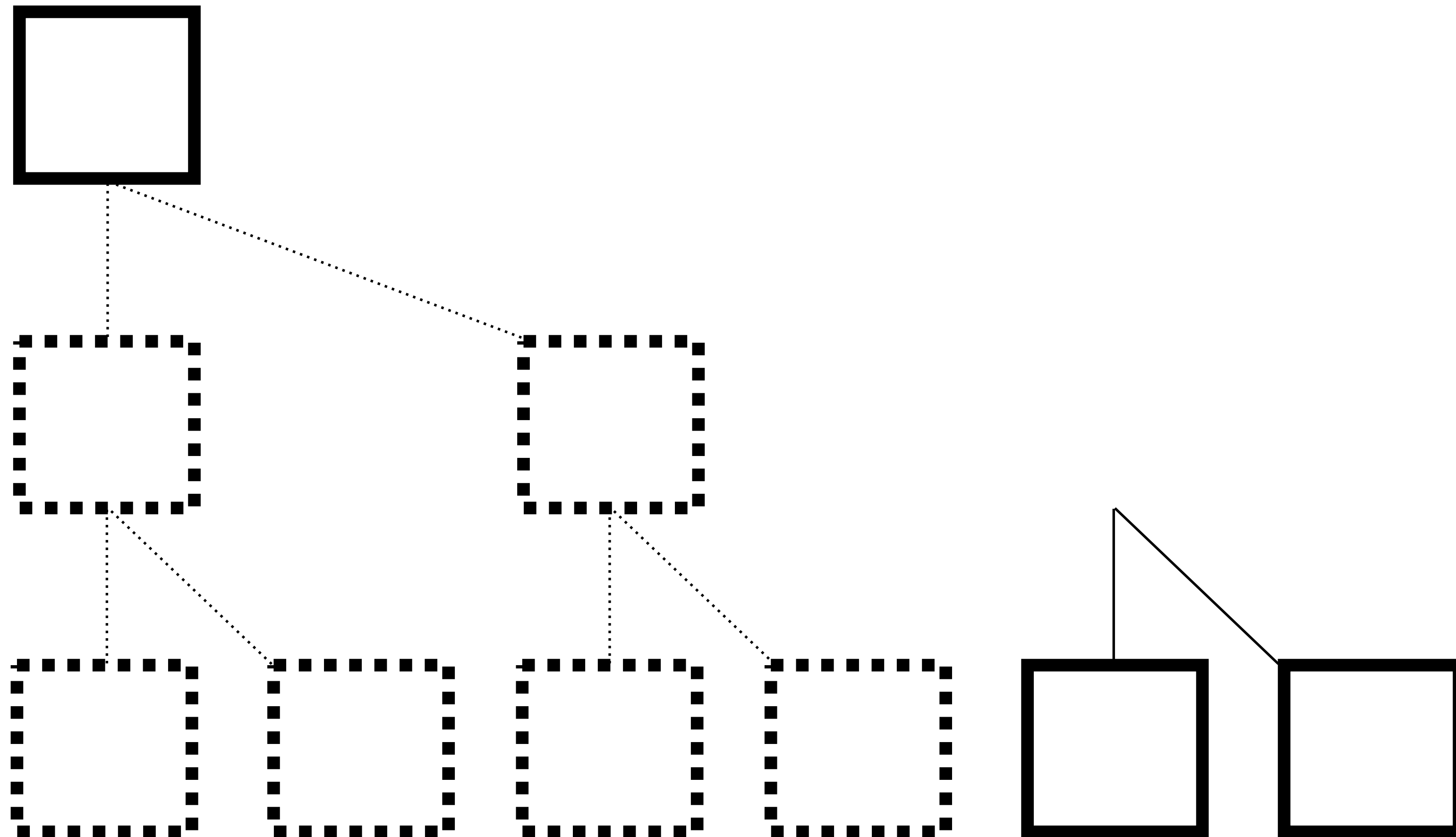
# 6 UTXOs

1 more



# 6 UTXOs

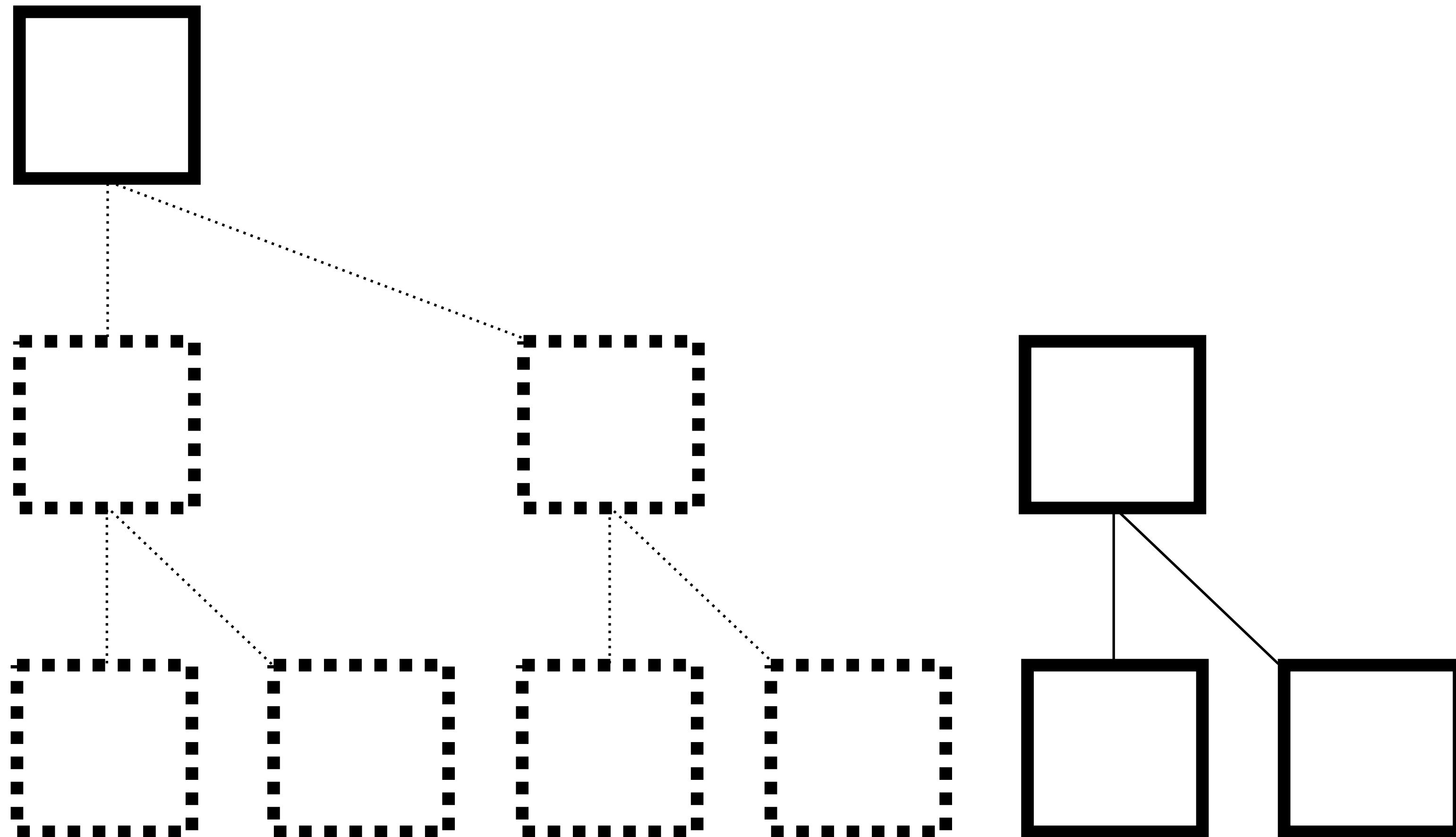
## Concatenate





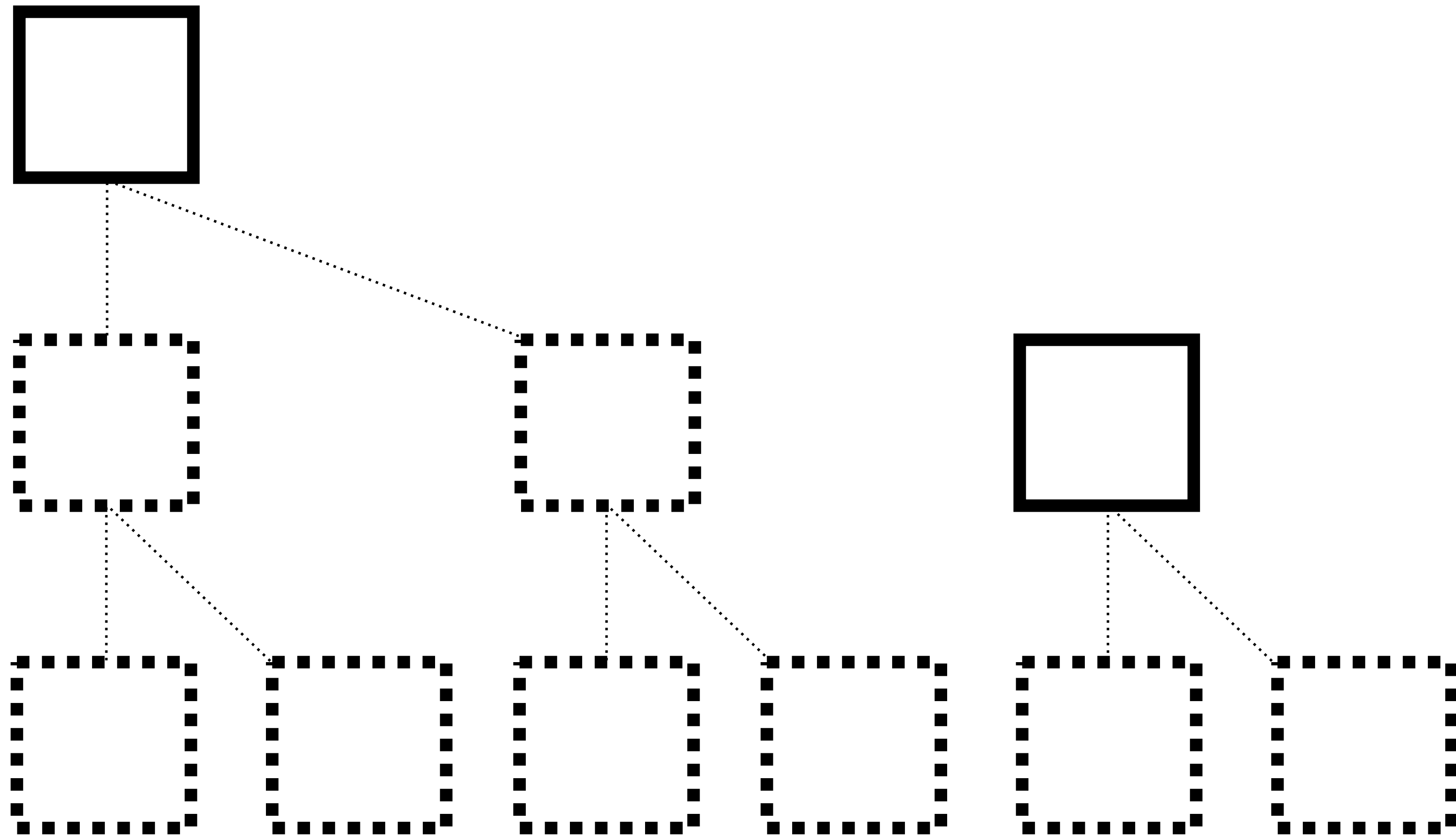
# 6 UTXOs

Hash



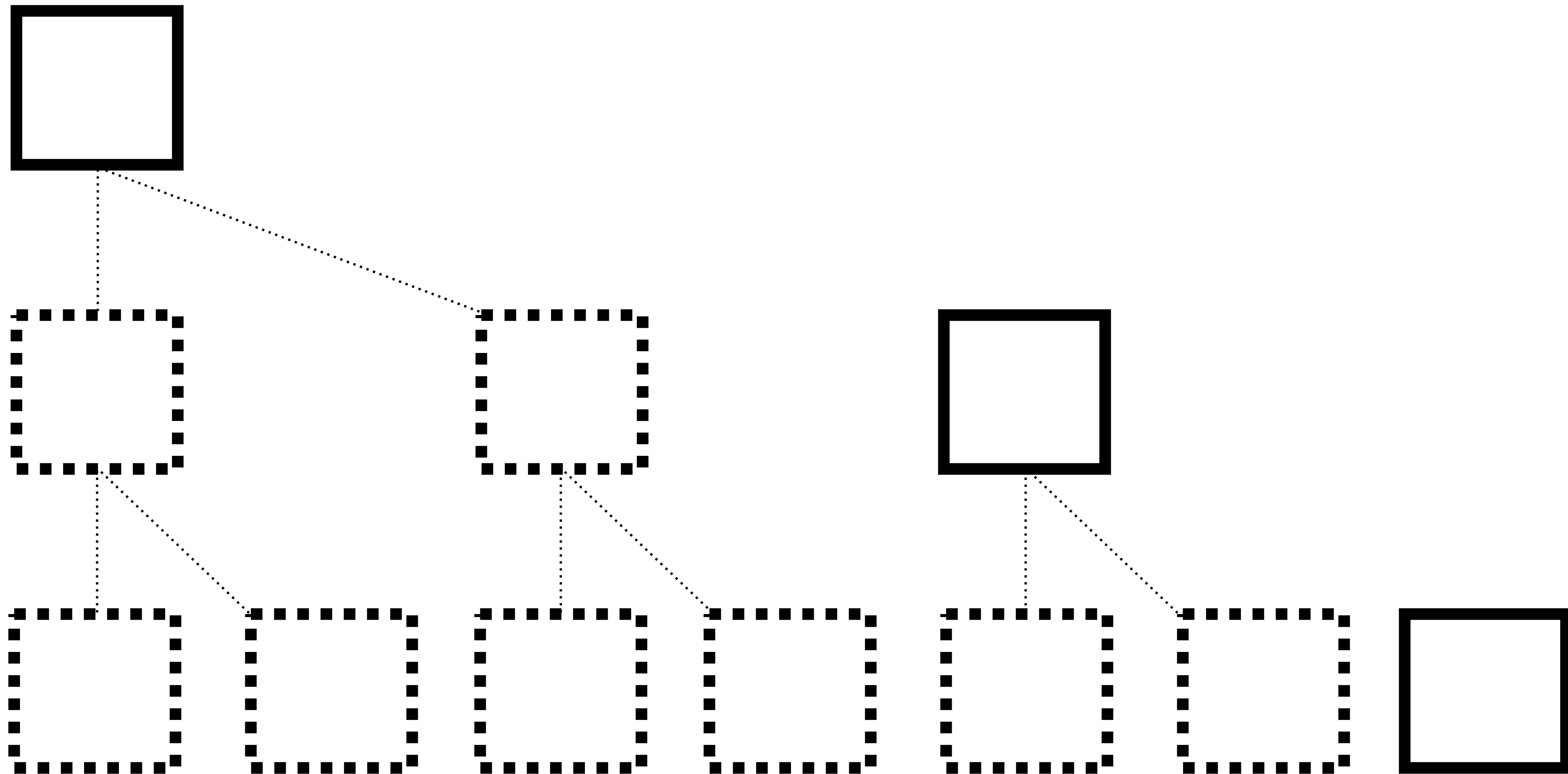
# 6 UTXOs

Throw away



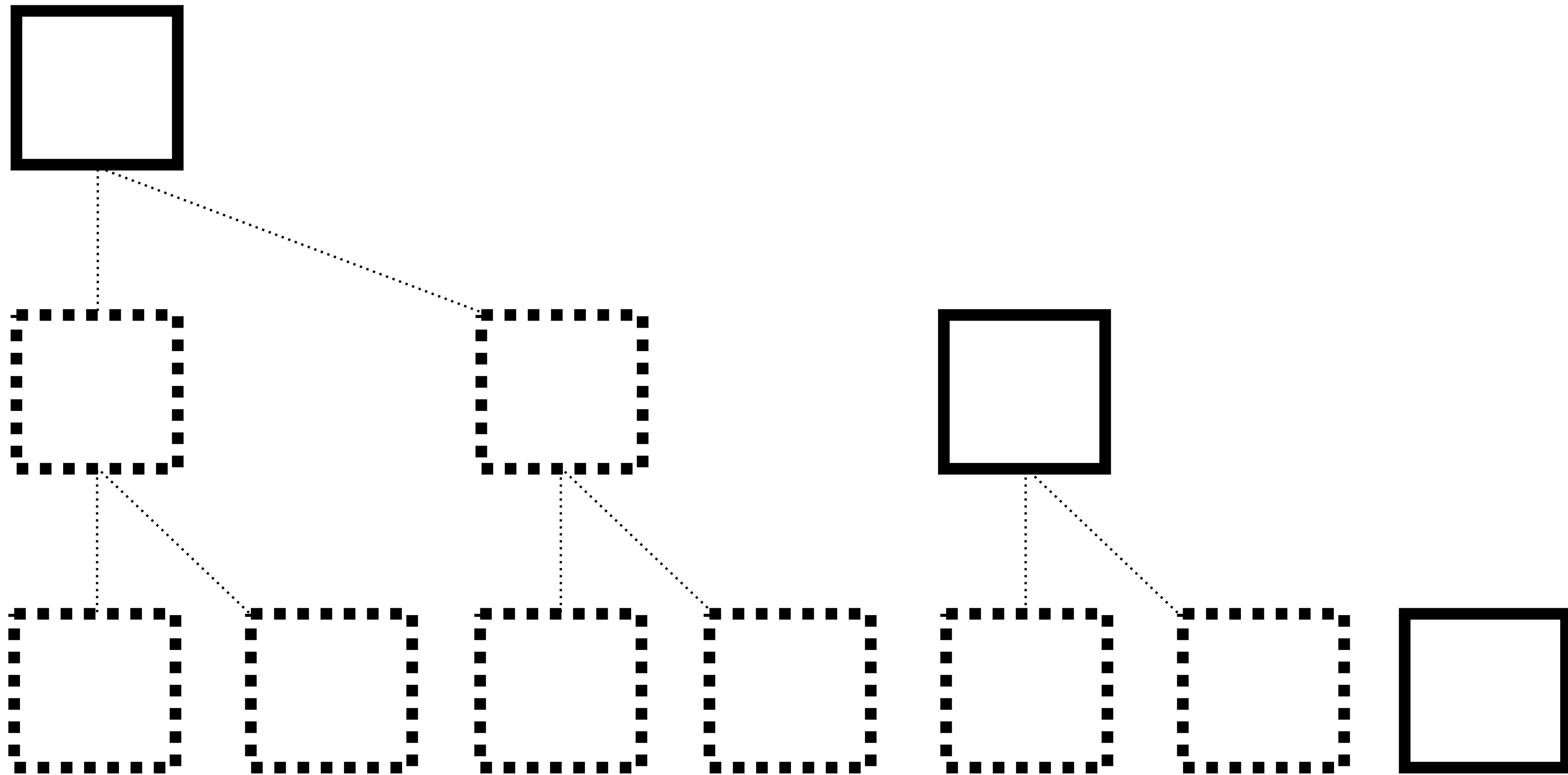
# 7 UTXOs

Another one



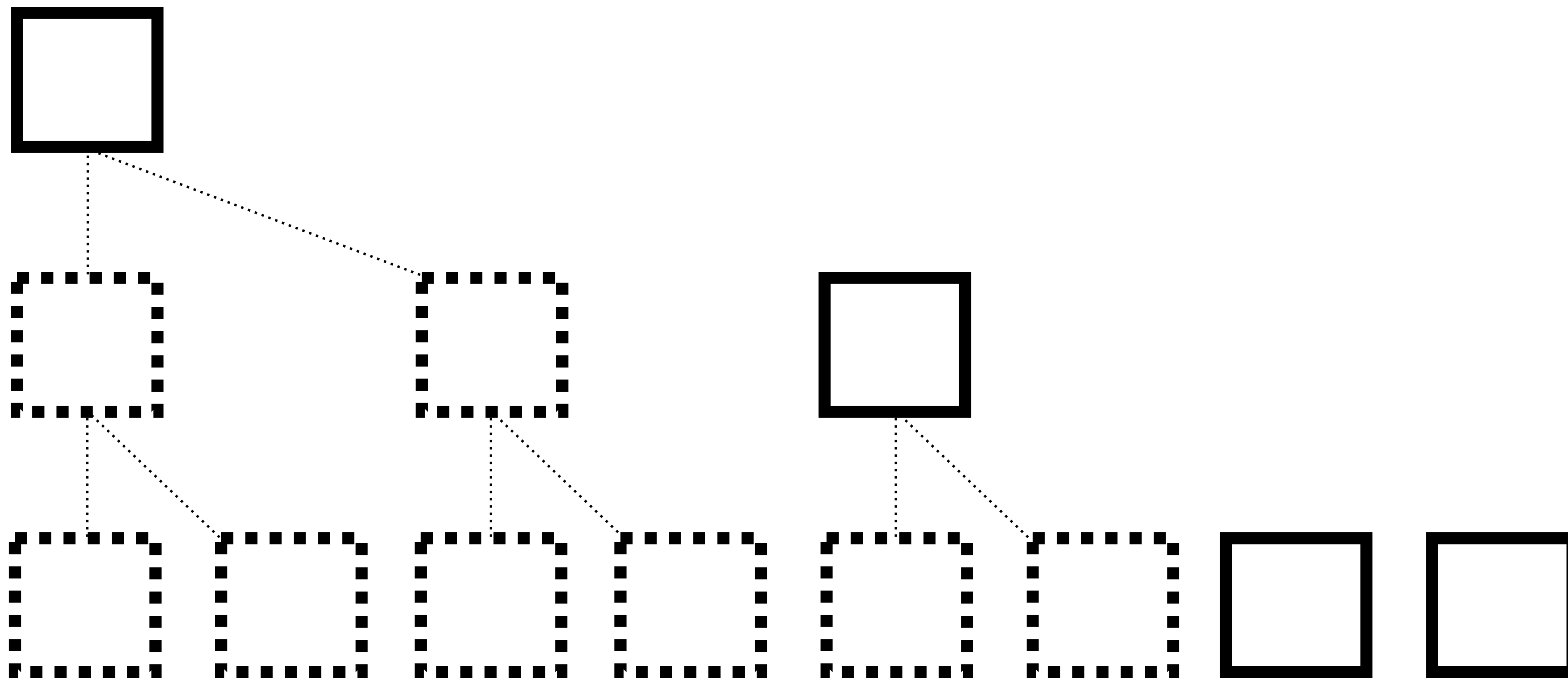
# 7 UTXOs

Nothing to hash with so it becomes a root



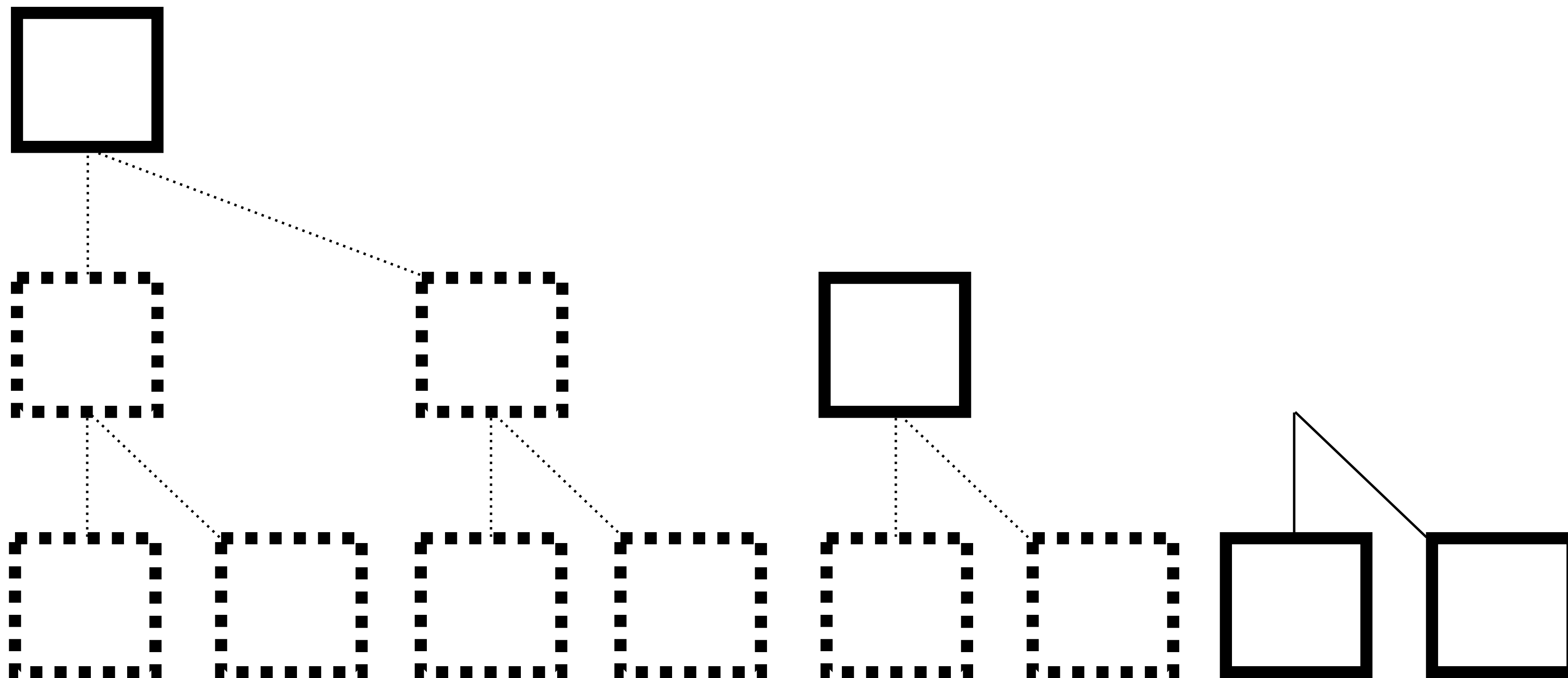
# 8 UTXOs

Another one



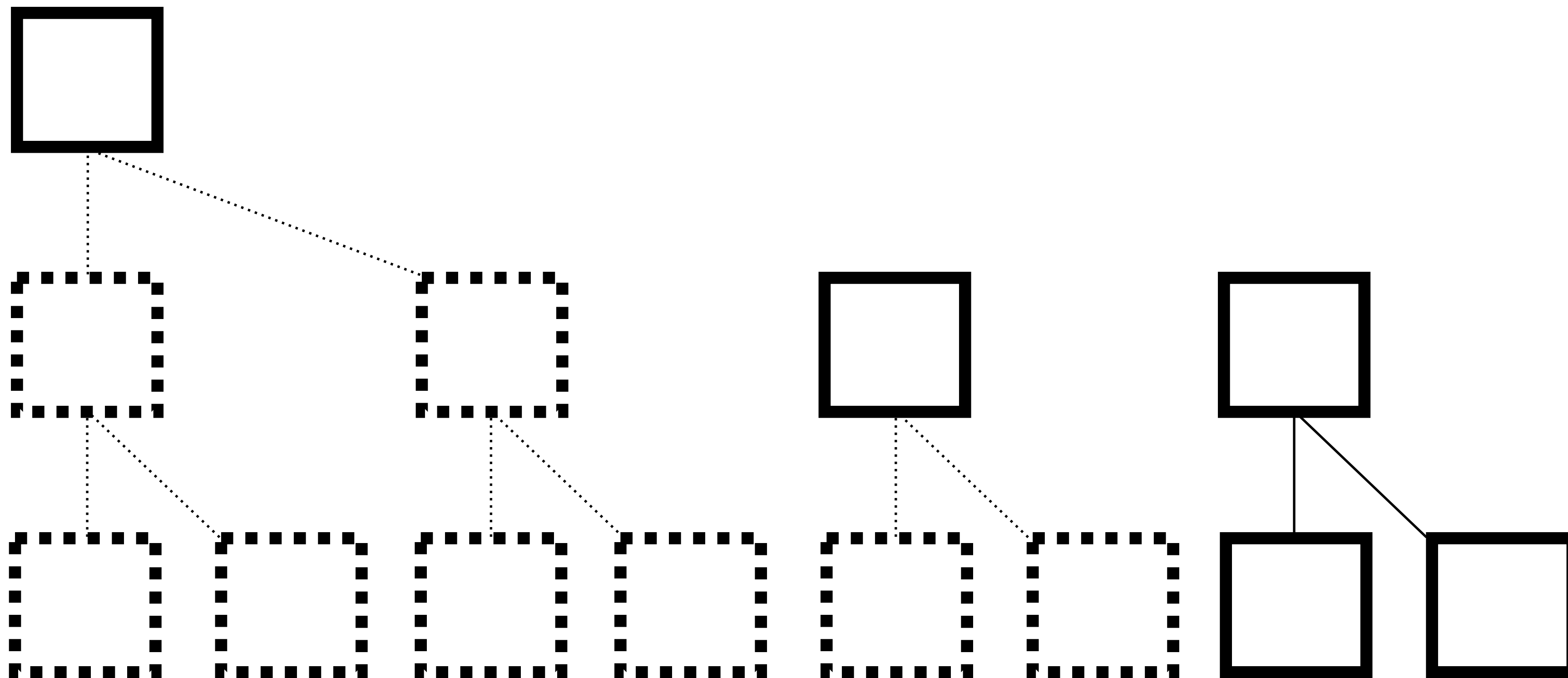
# 8 UTXOs

## Concatenate



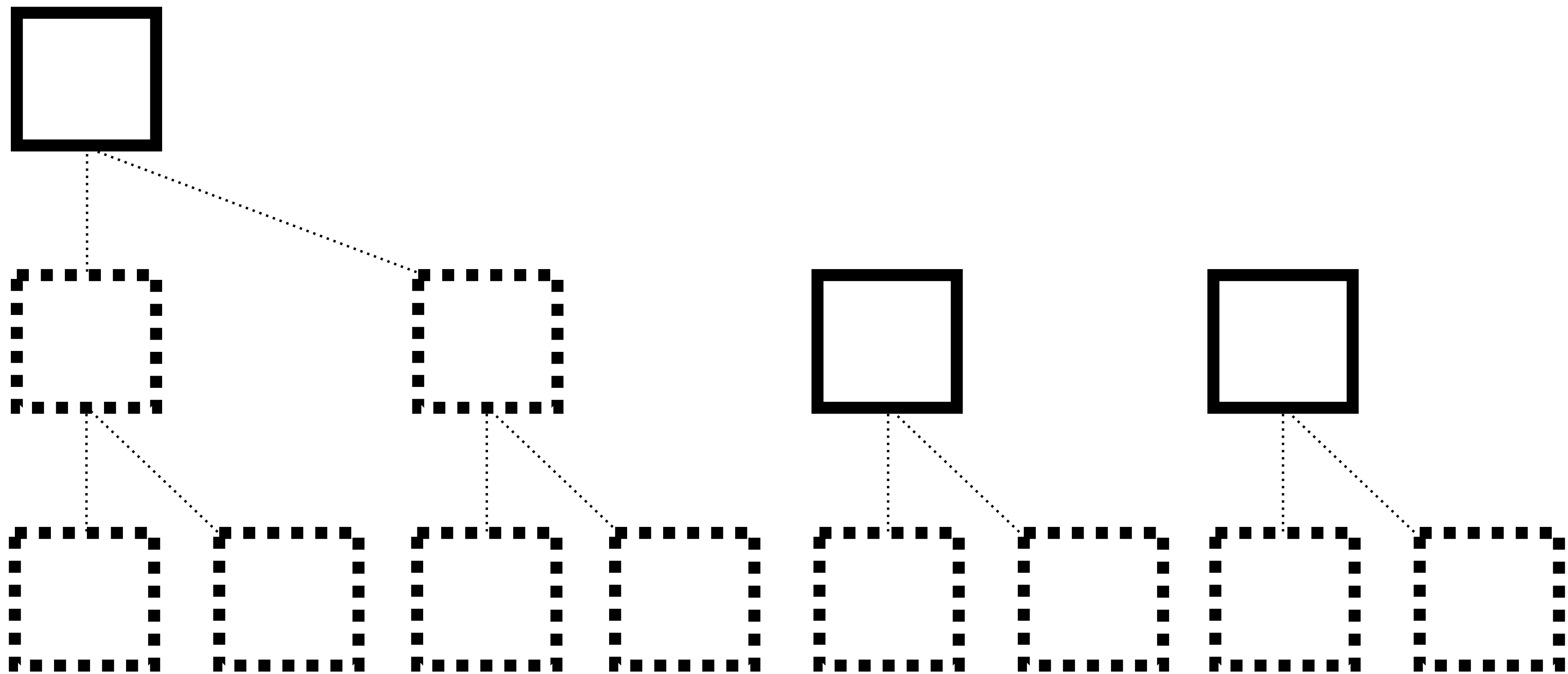
# 8 UTXOs

Hash



# 8 UTXOs

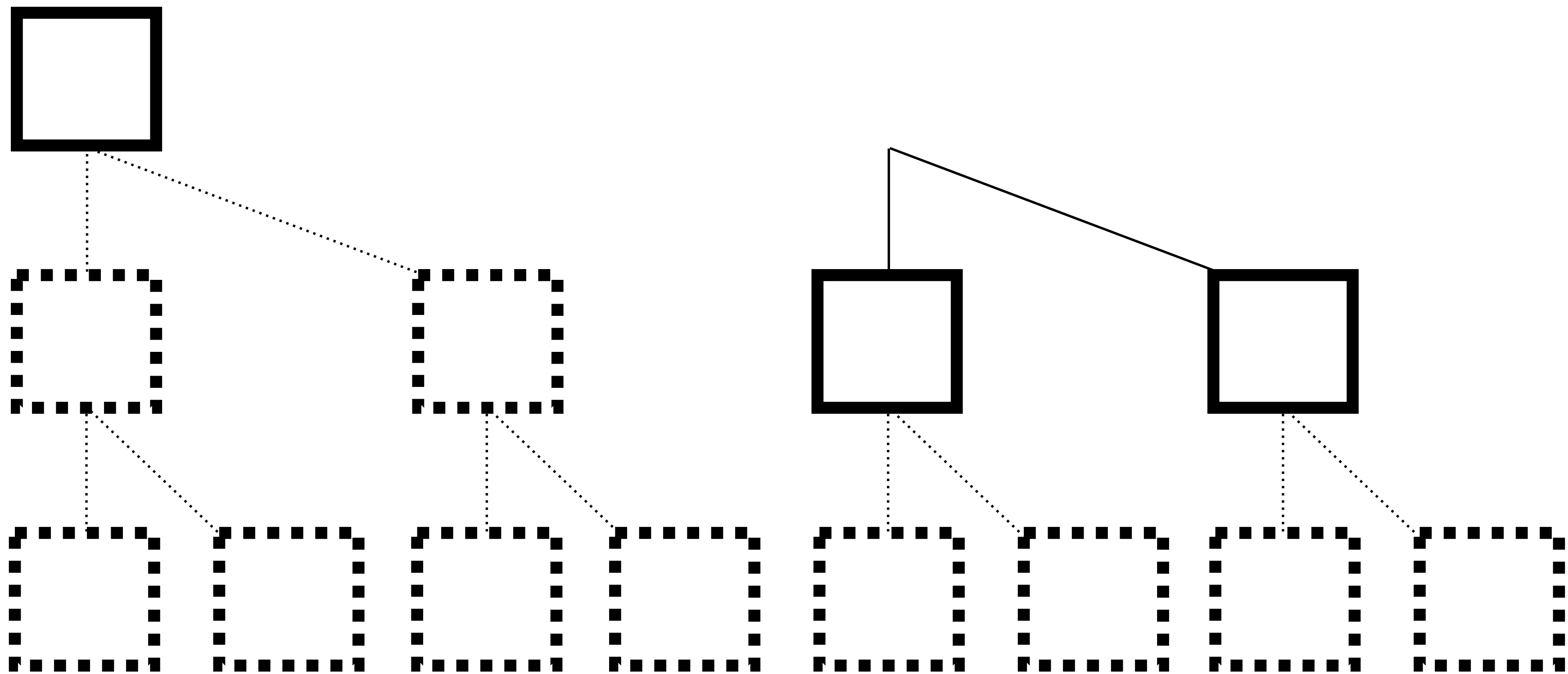
Throw away





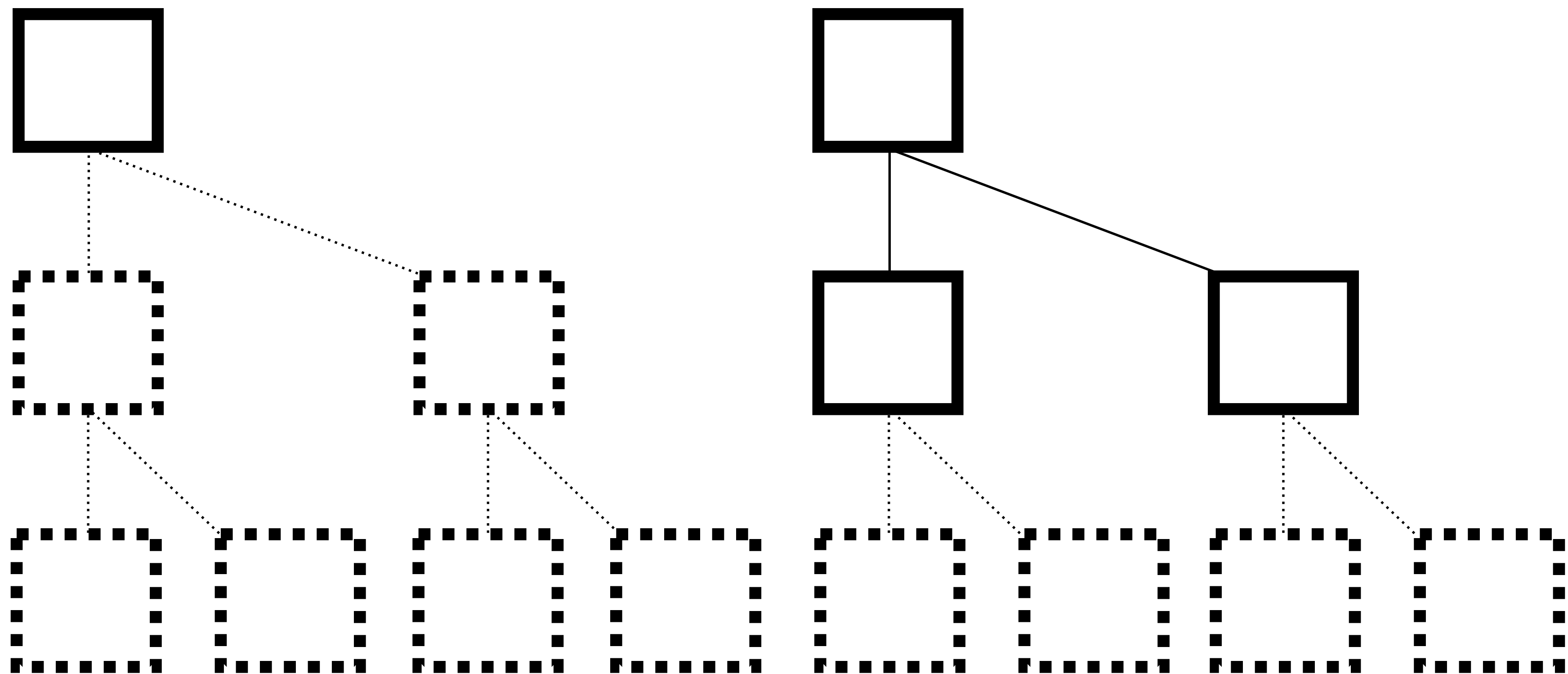
# 8 UTXOs

## Concatenate



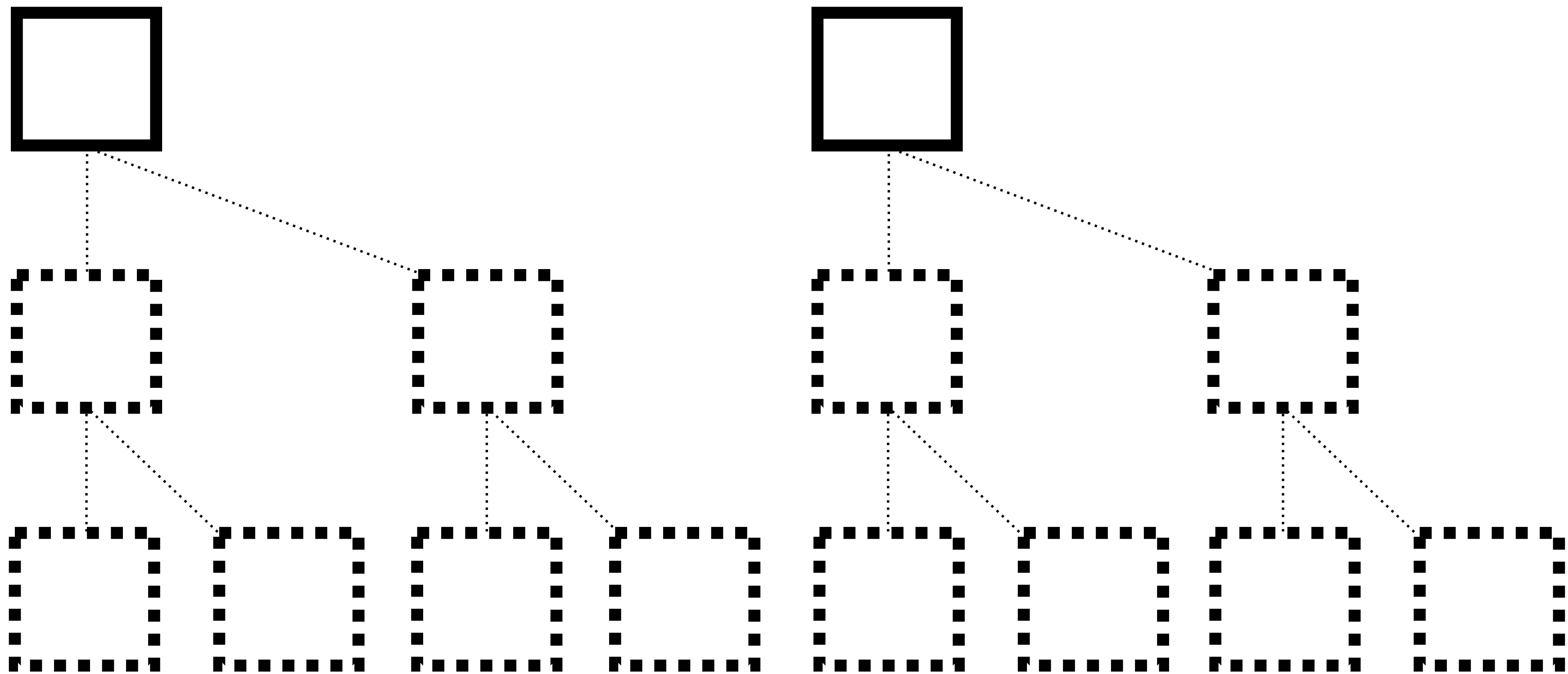
# 8 UTXOs

Hash



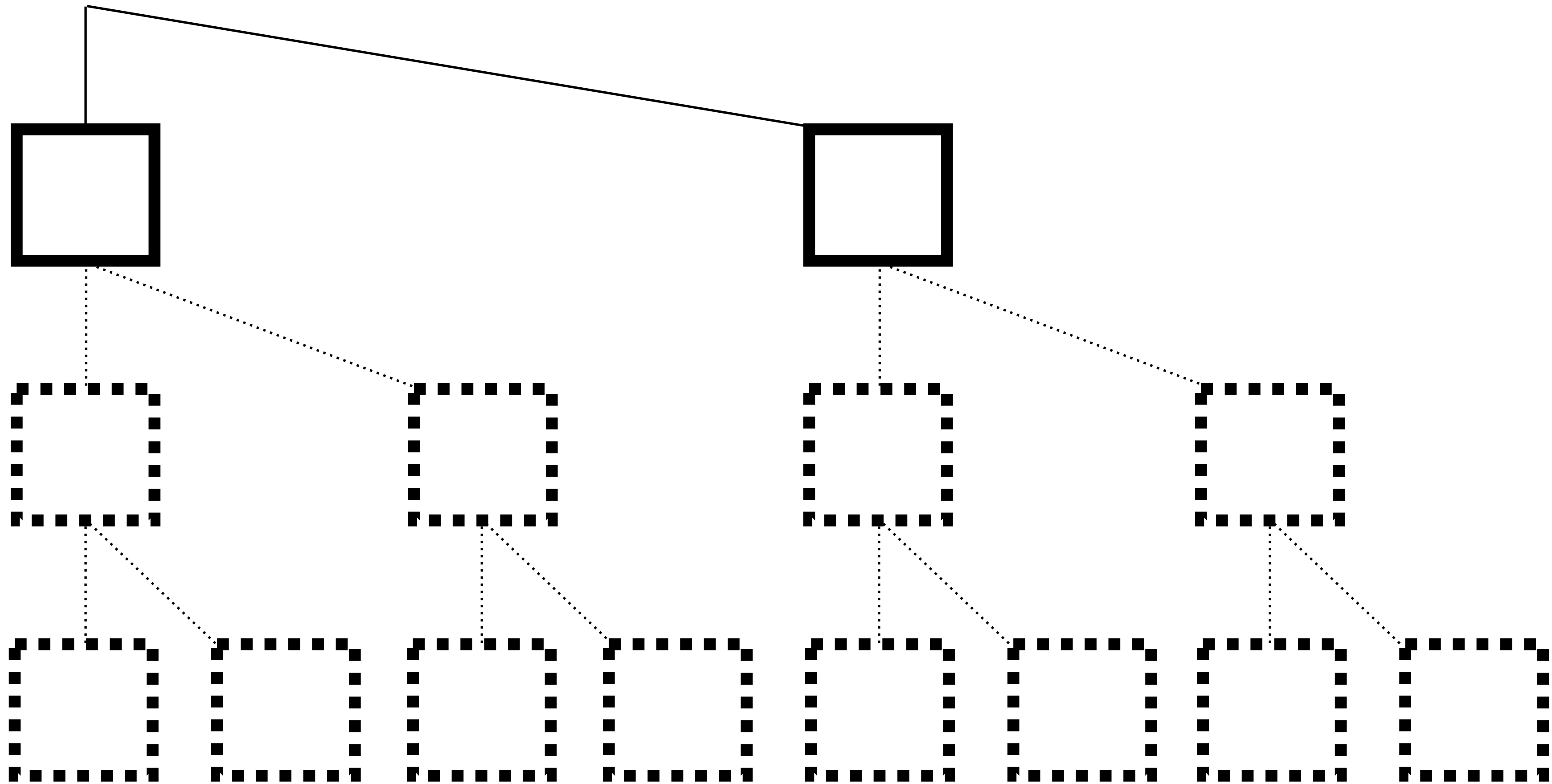
# 8 UTXOs

Throw away

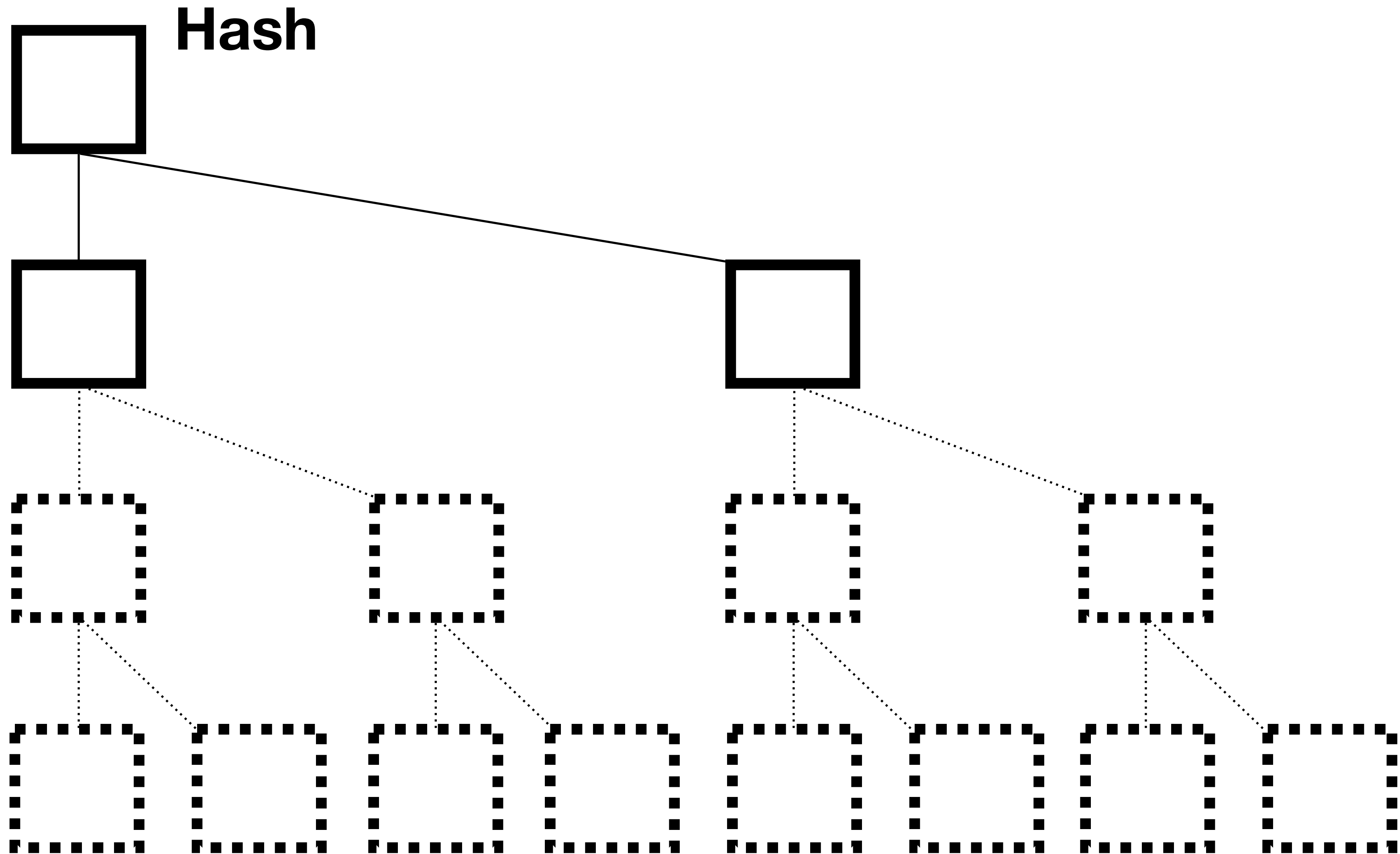


# 8 UTXOs

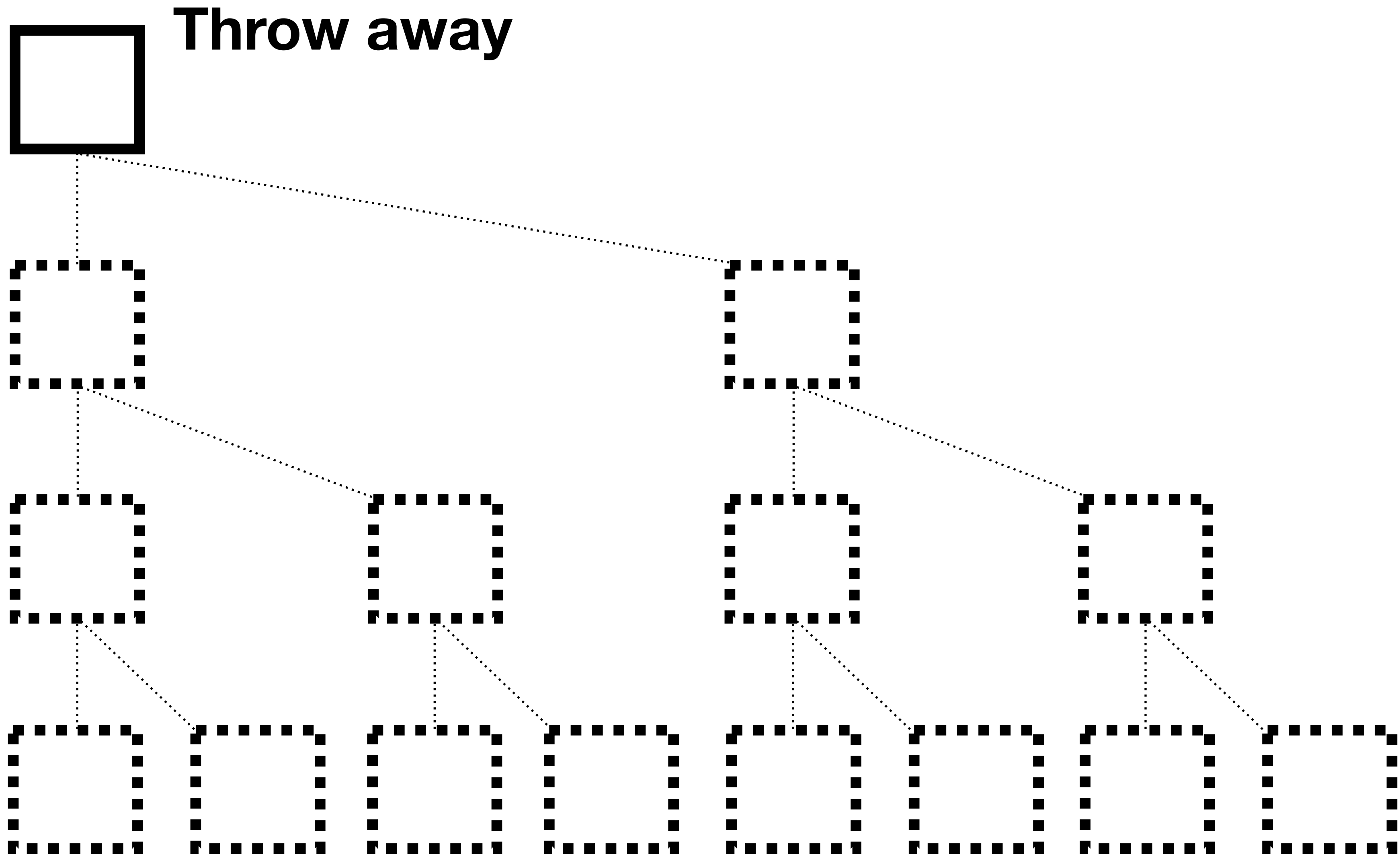
## Concatenate



# 8 UTXOs

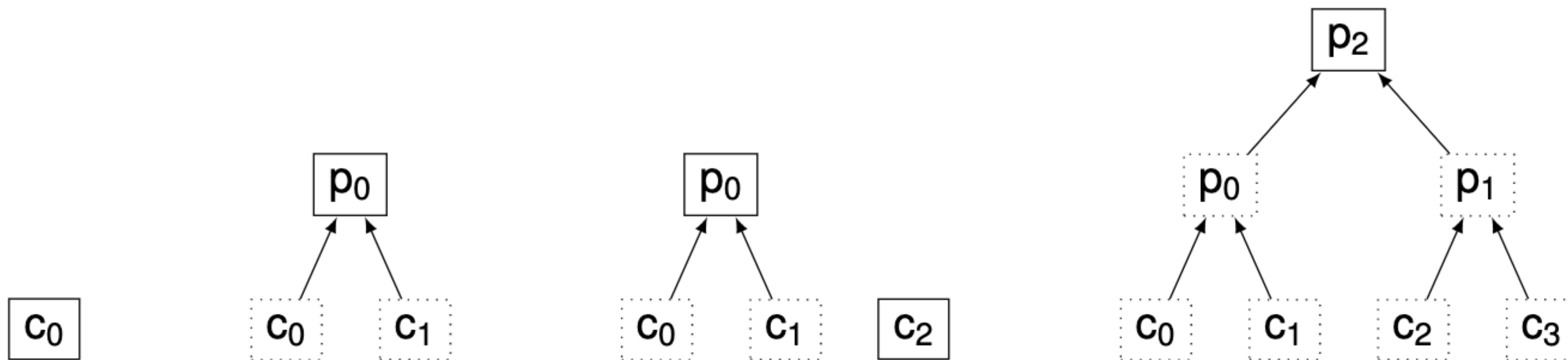


# 8 UTXOs



## 5.1.2 Chaining Value Stack

To help picture the role of the CV stack, Figure 6 shows a growing tree as chunk CVs are added incrementally. As just discussed above, chunk CVs are added to this tree only after the caller has supplied at least 1 byte for the following chunk, so we know that none of these chunks or parent nodes is the root of the tree, and we do not need to worry about the `ROOT` flag yet.



**Figure 6:** An incomplete tree growing incrementally from 1 to 4 chunks. Dotted boxes represent CVs that no longer need to be stored.

# Role of levelDB

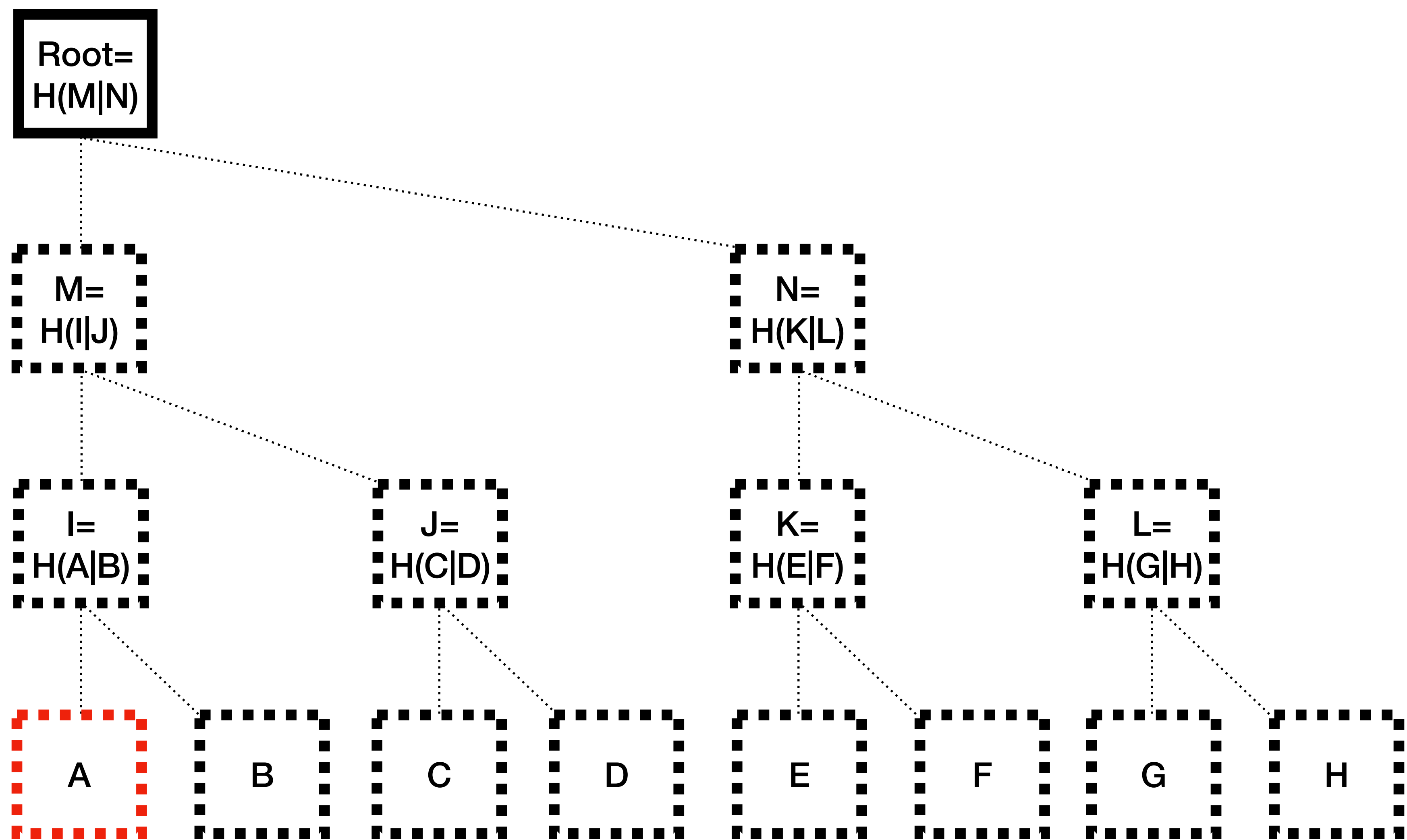
It let's you

1. Add a UTXO
2. Delete a UTXO
3. Tell you the existence of a UTXO
4. Provide the data for verification



**Deletion happens with proving  
existence**

# Deleting A



# The only requirement is the roots

$$\begin{array}{|c|} \hline \text{Root=} \\ \hline H(M|N) \\ \hline \end{array}$$

# Receive the proof for A

Root=  
 $H(M|N)$

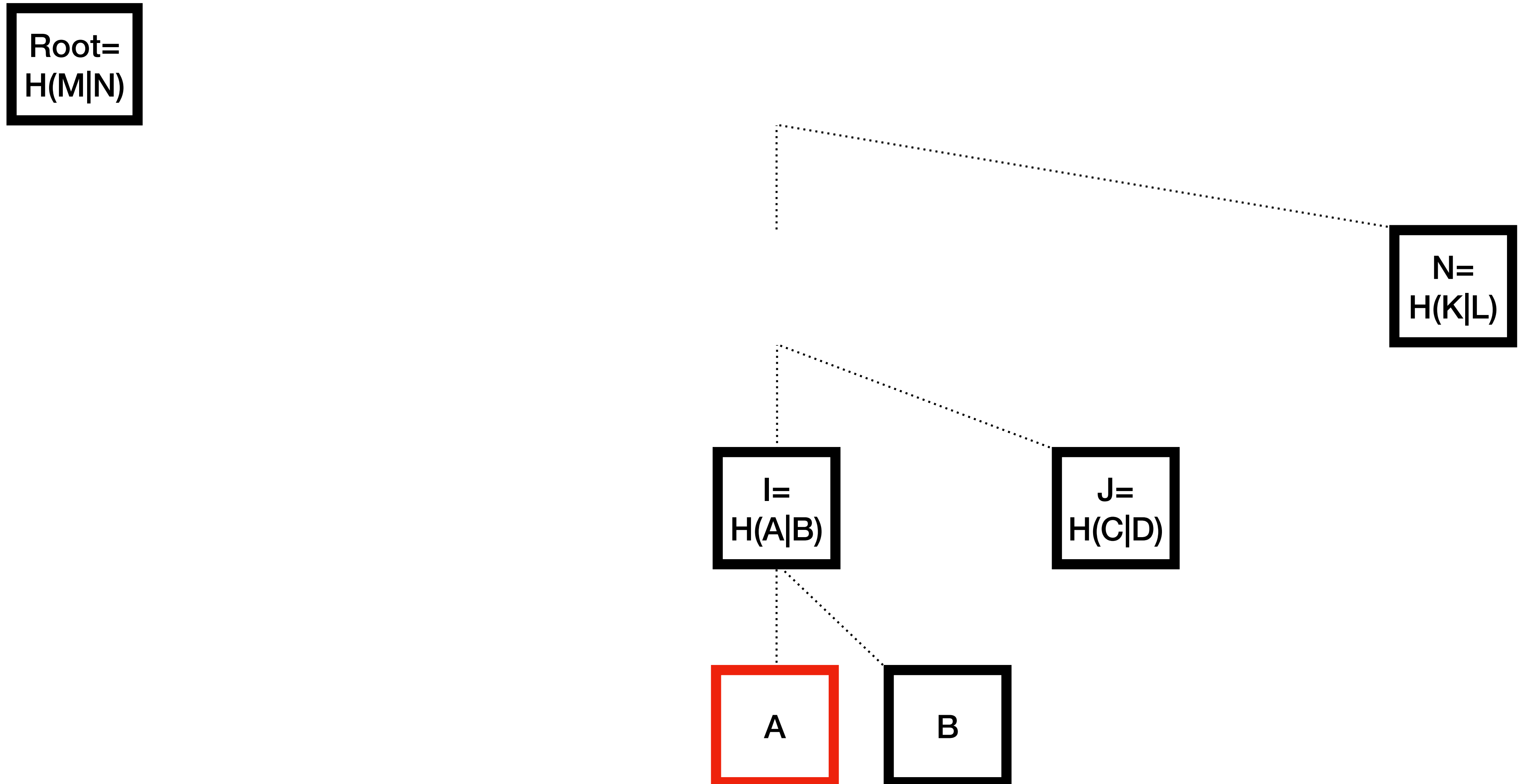
N=  
 $H(K|L)$

J=  
 $H(C|D)$

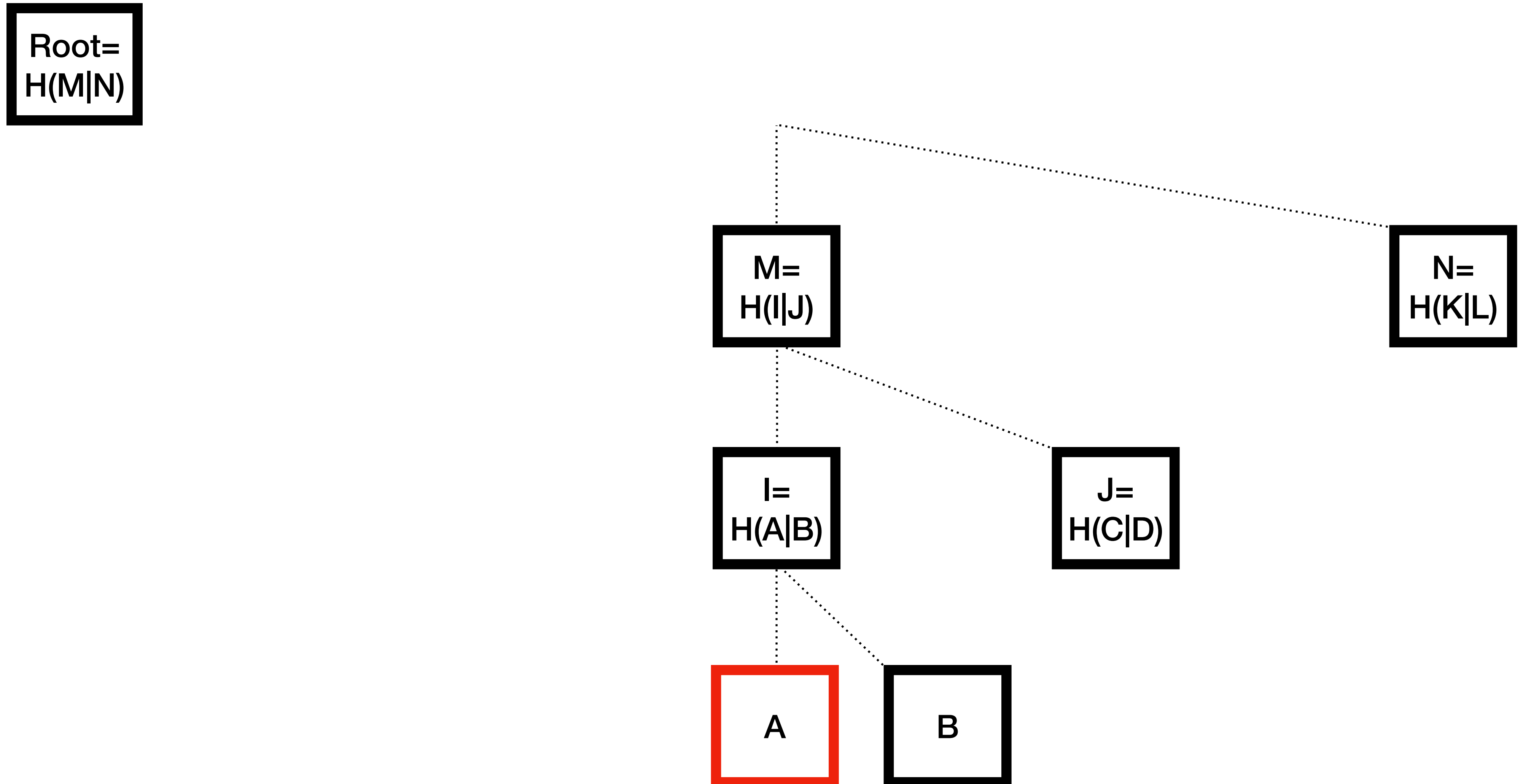
A

B

# Calculate the hash for I

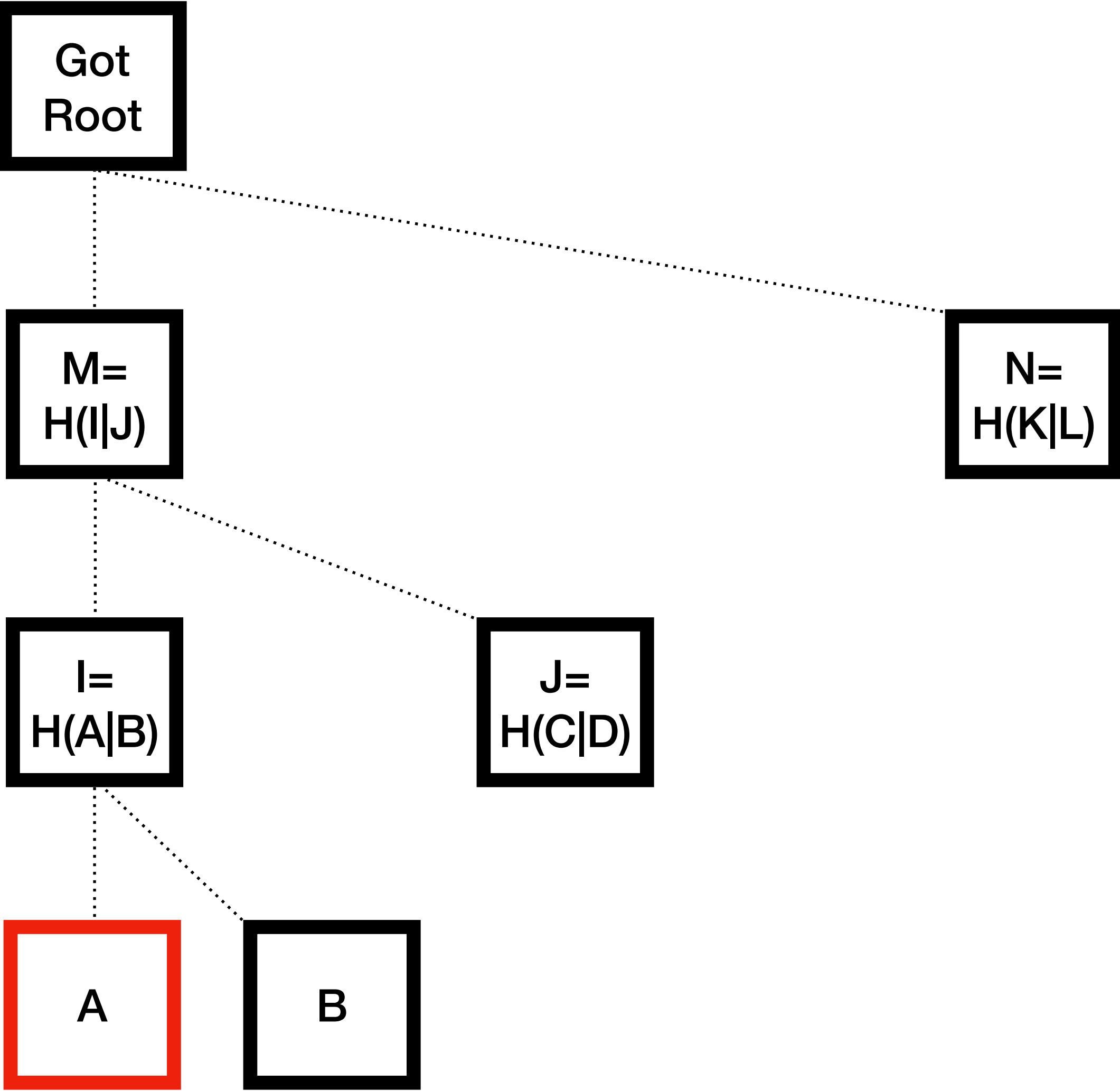


# Calculate the hash for M



# Calculate Root

Root=  
H(M|N)



# Compare roots

Continue if the roots are equal. Ban peer if not equal



Root= $H(M|N)$

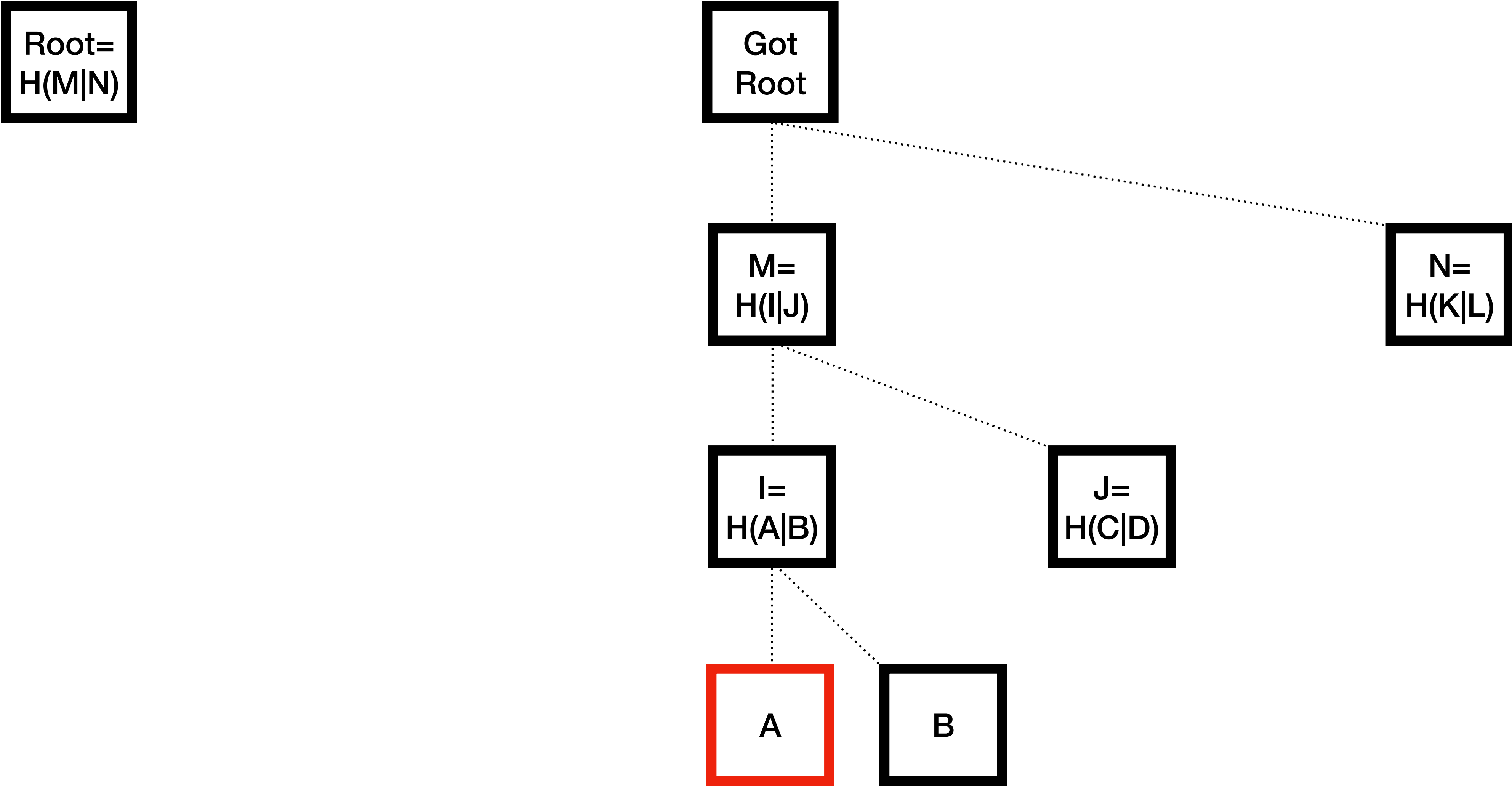
The diagram consists of two square boxes with thick black borders. The left box contains the text 'Root=H(M|N)'. The right box contains the text 'Got Root'. Between the two boxes is an equals sign, represented by two parallel horizontal lines stacked vertically.

=

Got Root

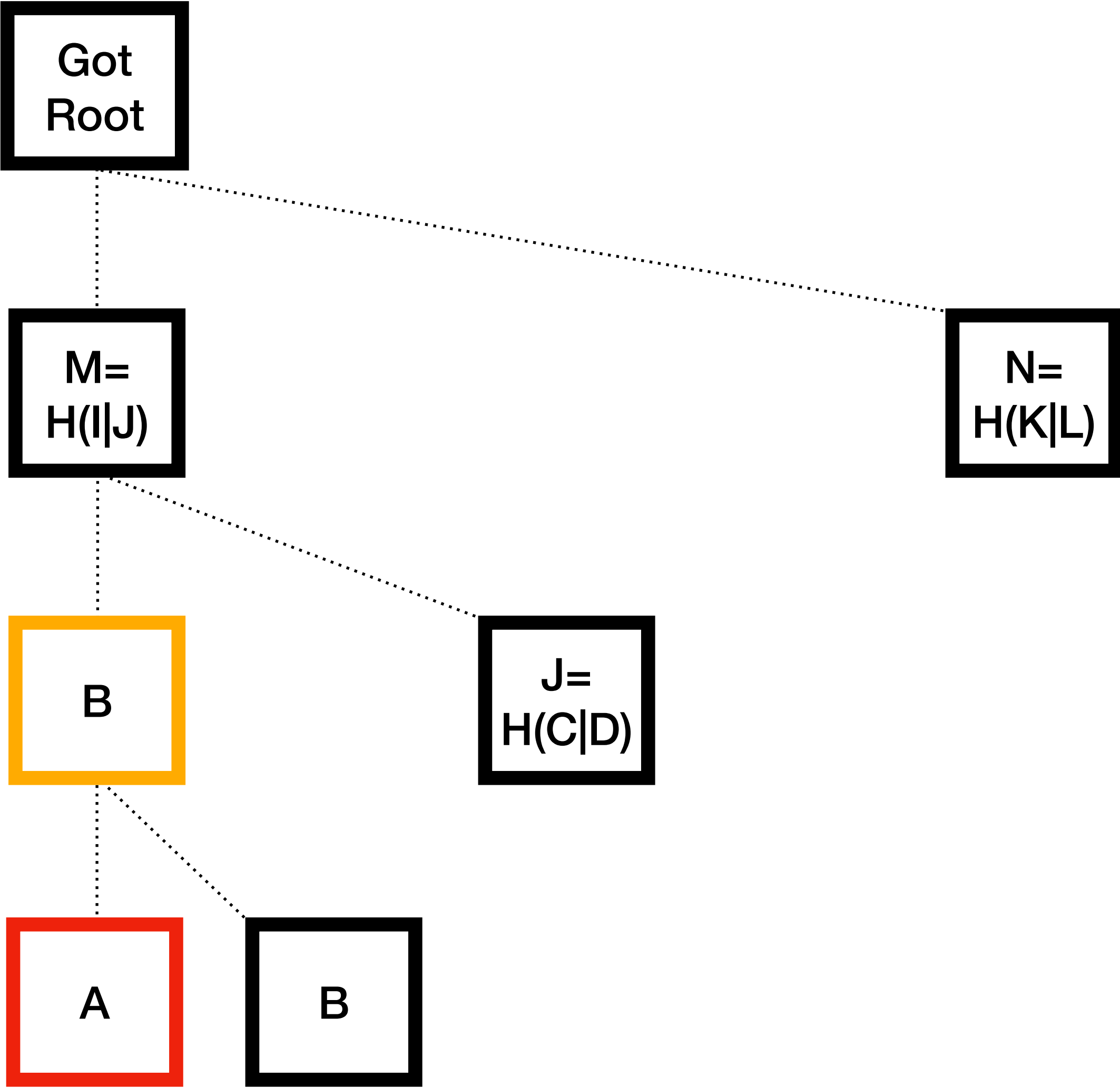


# Calculate root after deletion

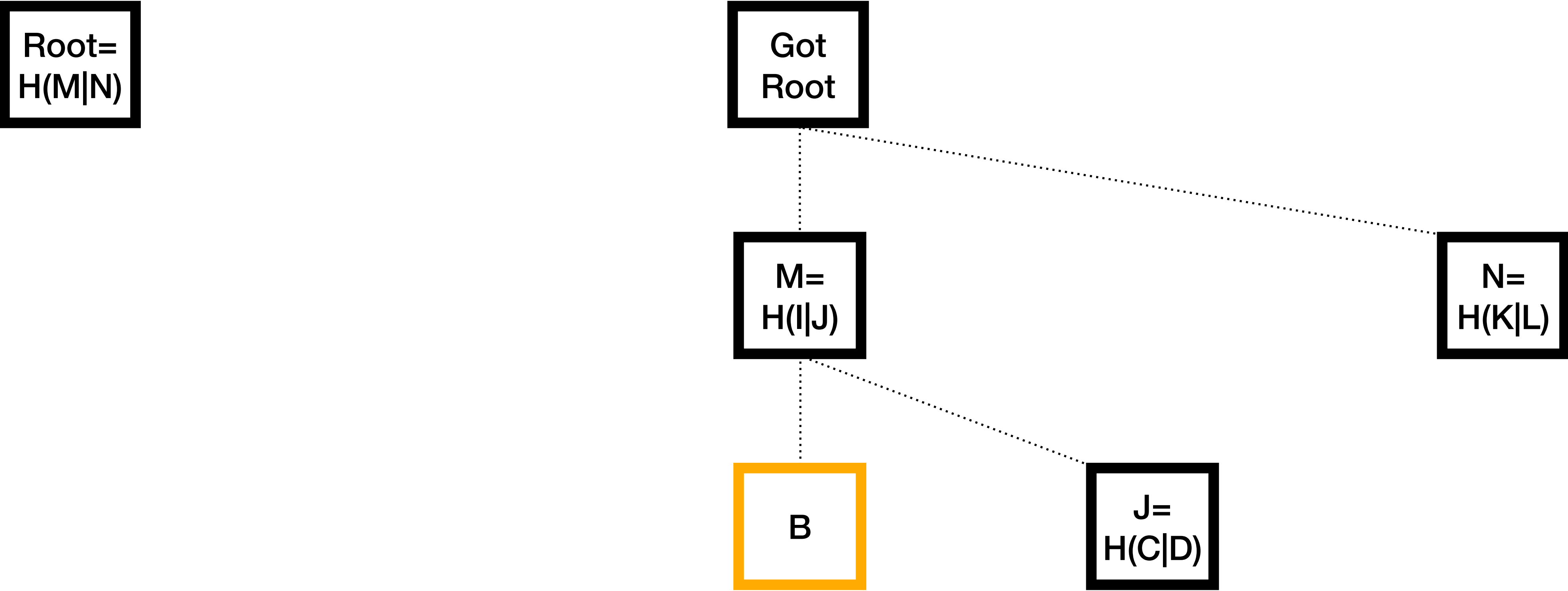


# Move up B to I

Root=  
 $H(M|N)$



# Remove old nodes for A&B



# Calculate $H(B|J)$

Root=  
 $H(M|N)$

Got  
Root

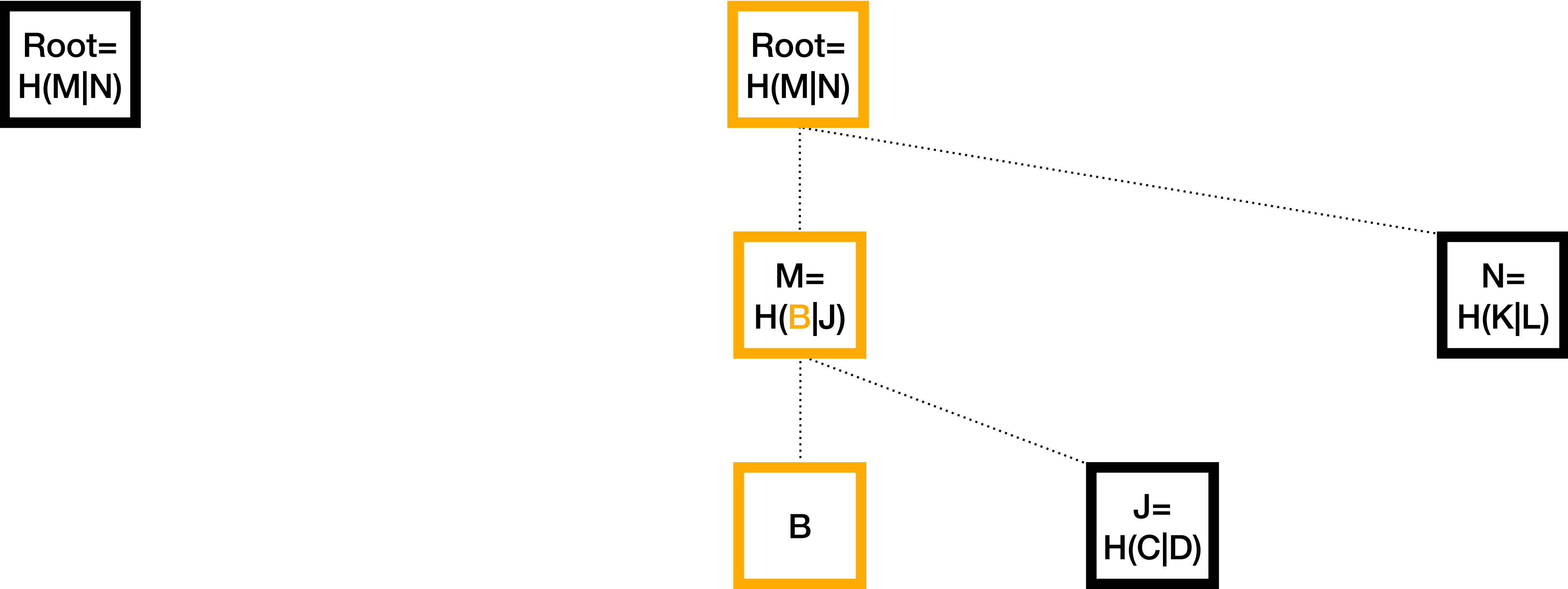
M=  
 $H(B|J)$

N=  
 $H(K|L)$

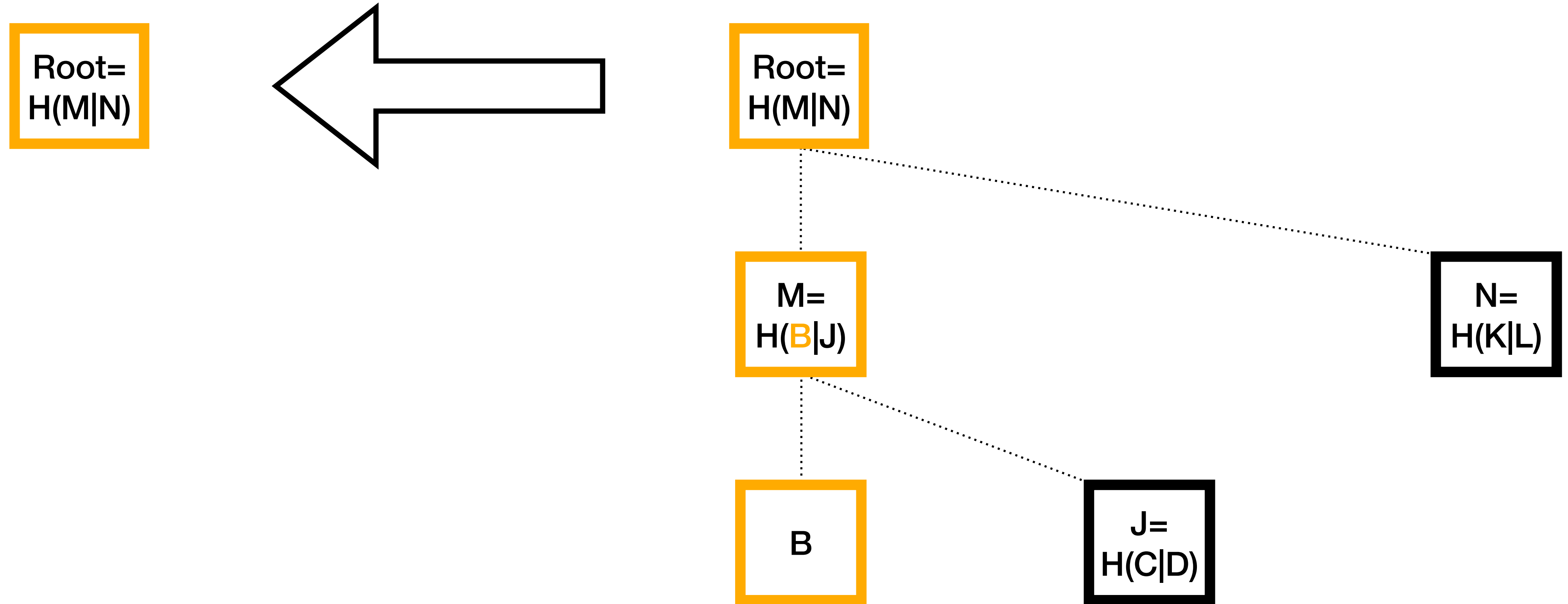
B

J=  
 $H(C|D)$

# Calculate new root



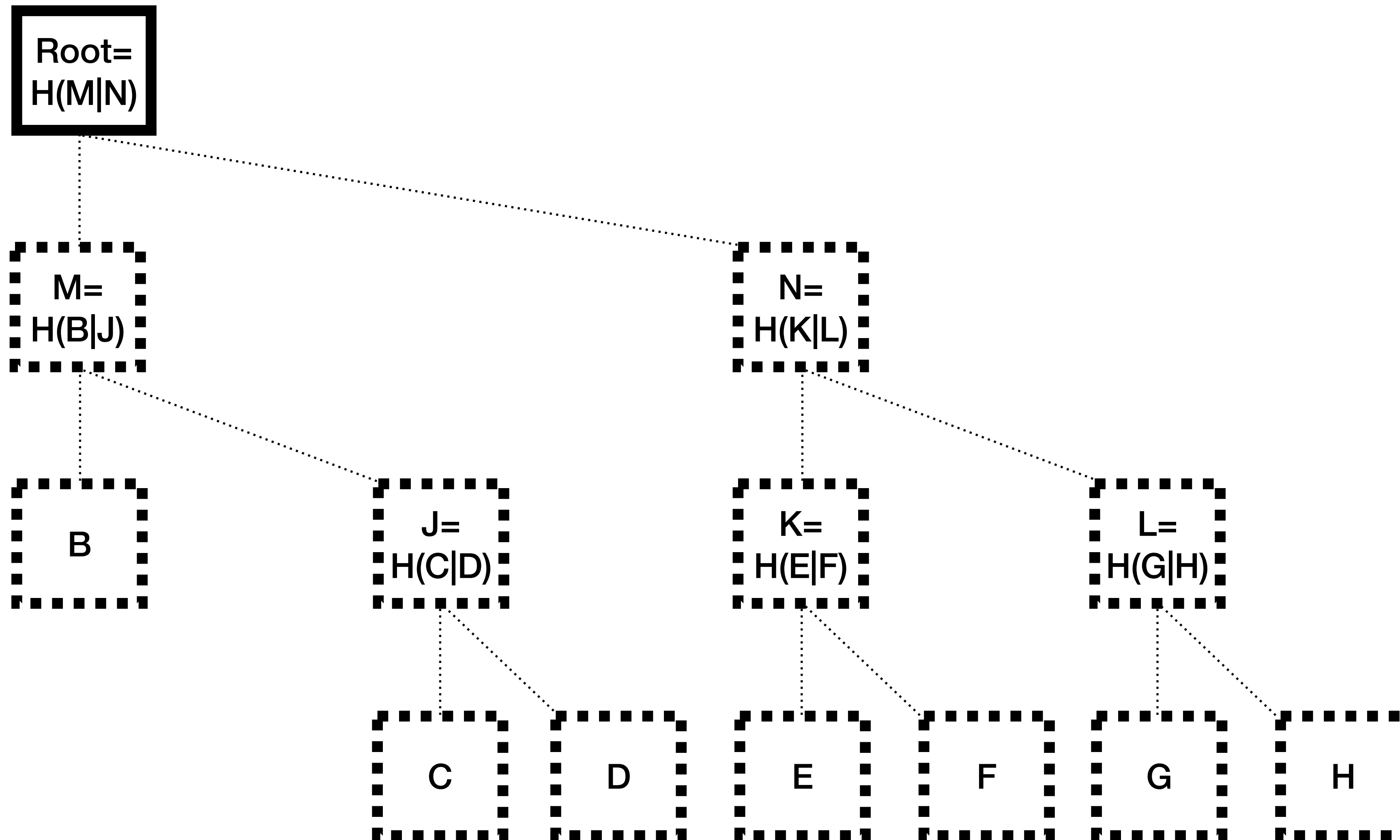
# Copy over the new root to be saved



# Done

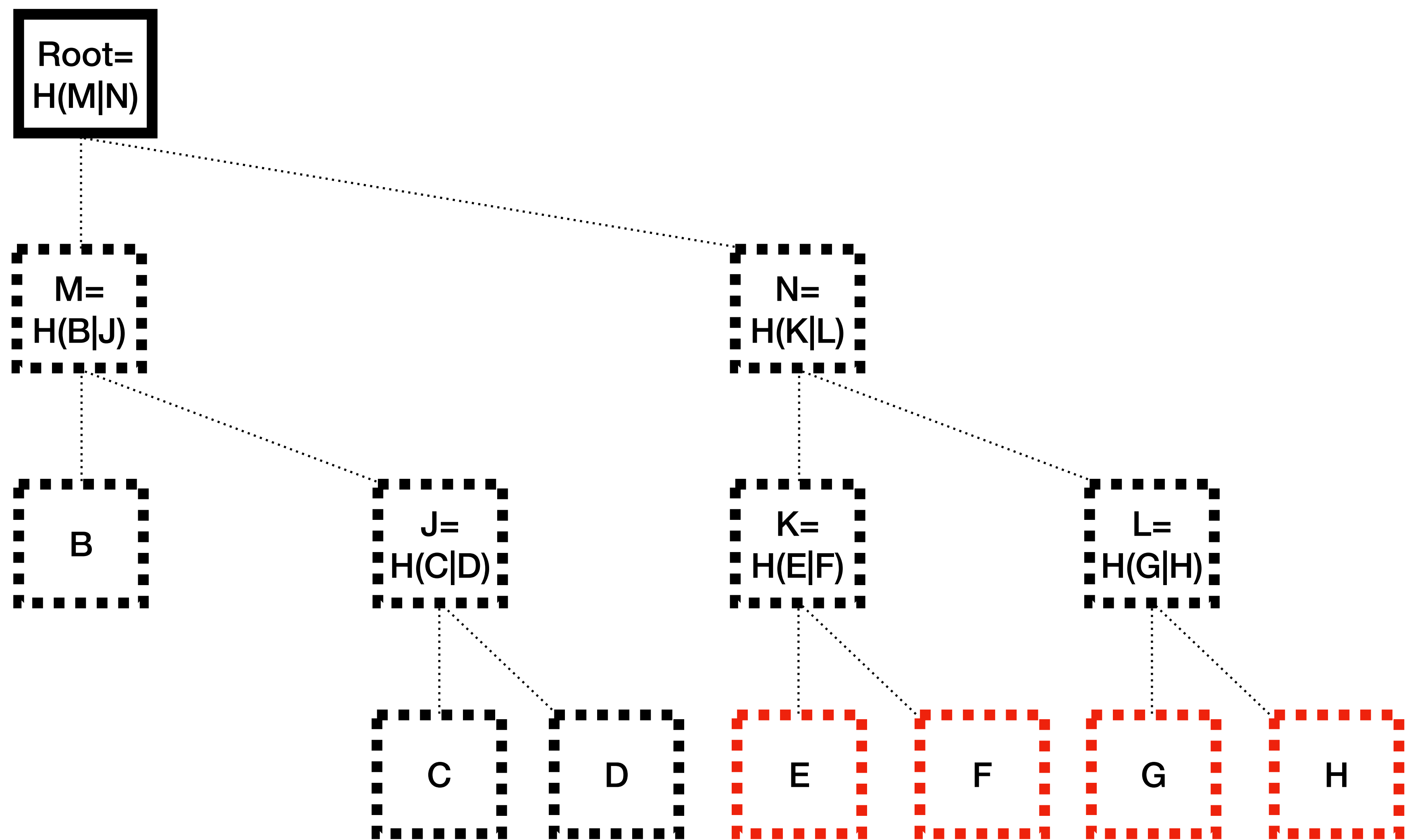
Root=  
 $H(M|N)$

# After deleting A





# Delete E, F, G, H. (Batched deletions)



# Start off with only the root

$$\text{Root} = H(M|N)$$

# Receive the proof

Root=  
 $H(M|N)$

M=  
 $H(B|J)$

E

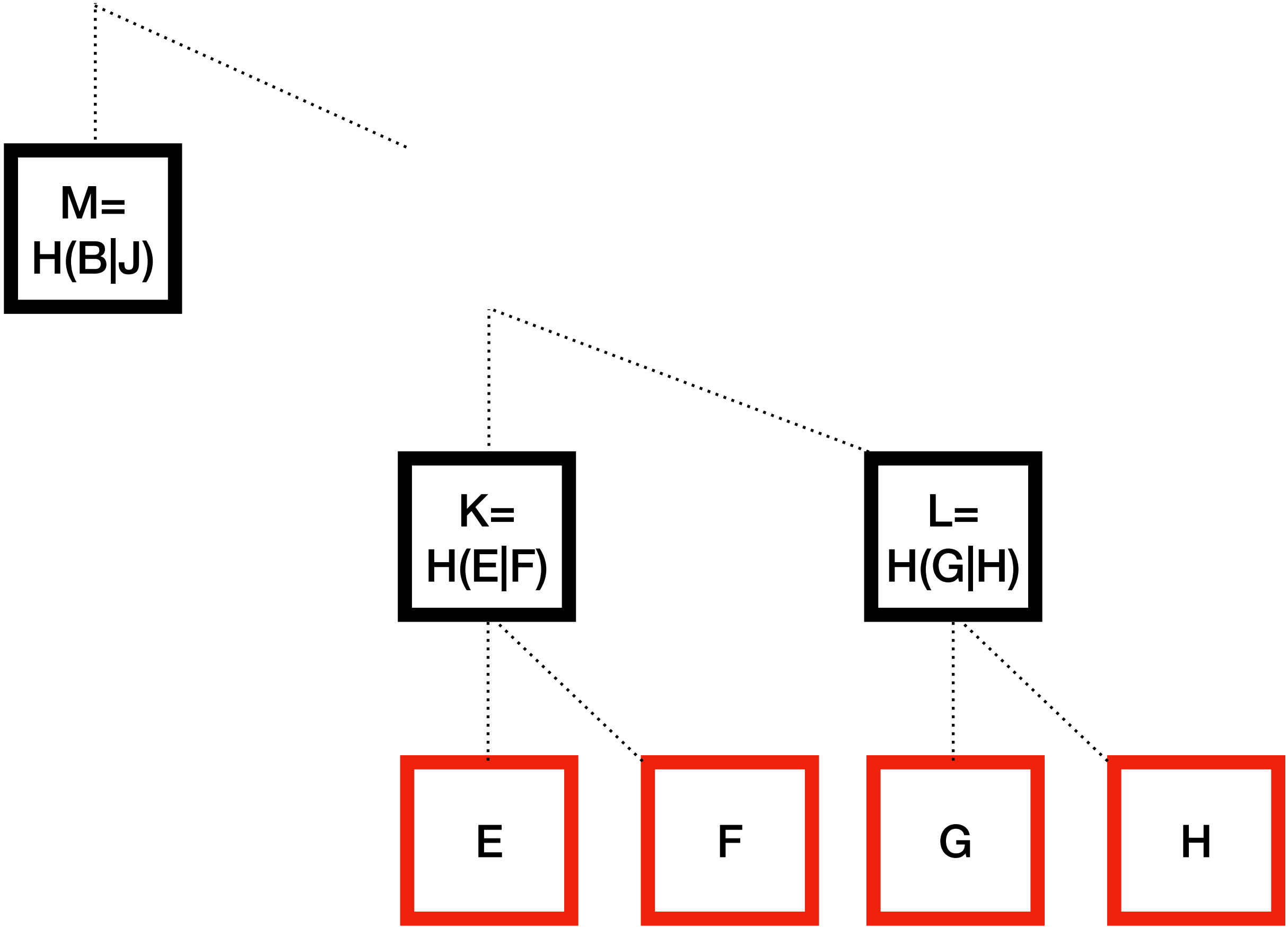
F

G

H

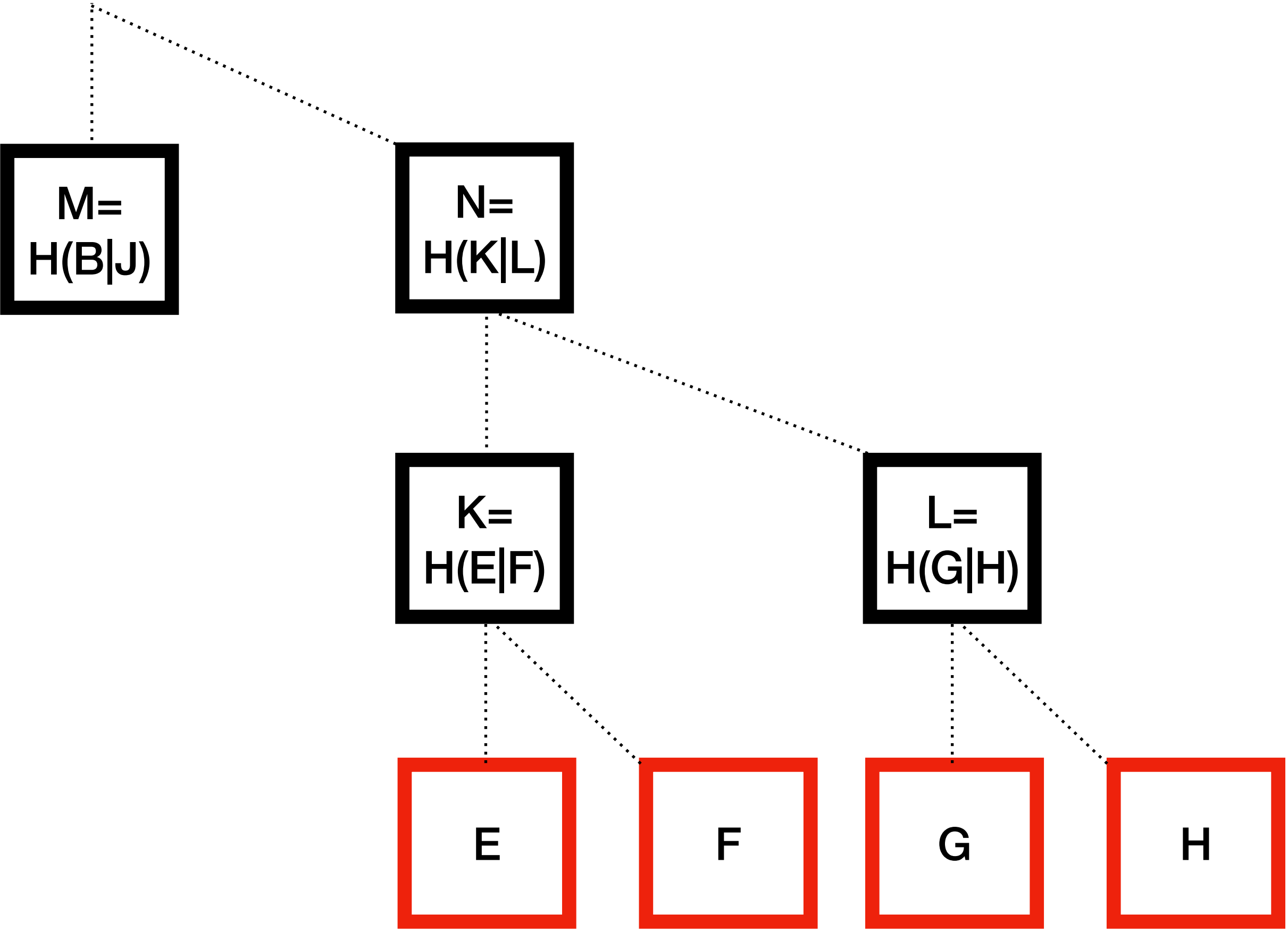
# Calculate K and L

Root=  
H(M|N)



# Calculate N

Root=  
H(M|N)



# Calculate the Root

Root=  
H(M|N)

Got  
Root

M=  
H(B|J)

N=  
H(K|L)

K=  
H(E|F)

L=  
H(G|H)

E

F

G

H

# Compare roots

Continue if the roots are equal. Ban peer if not equal



Root= $H(M|N)$

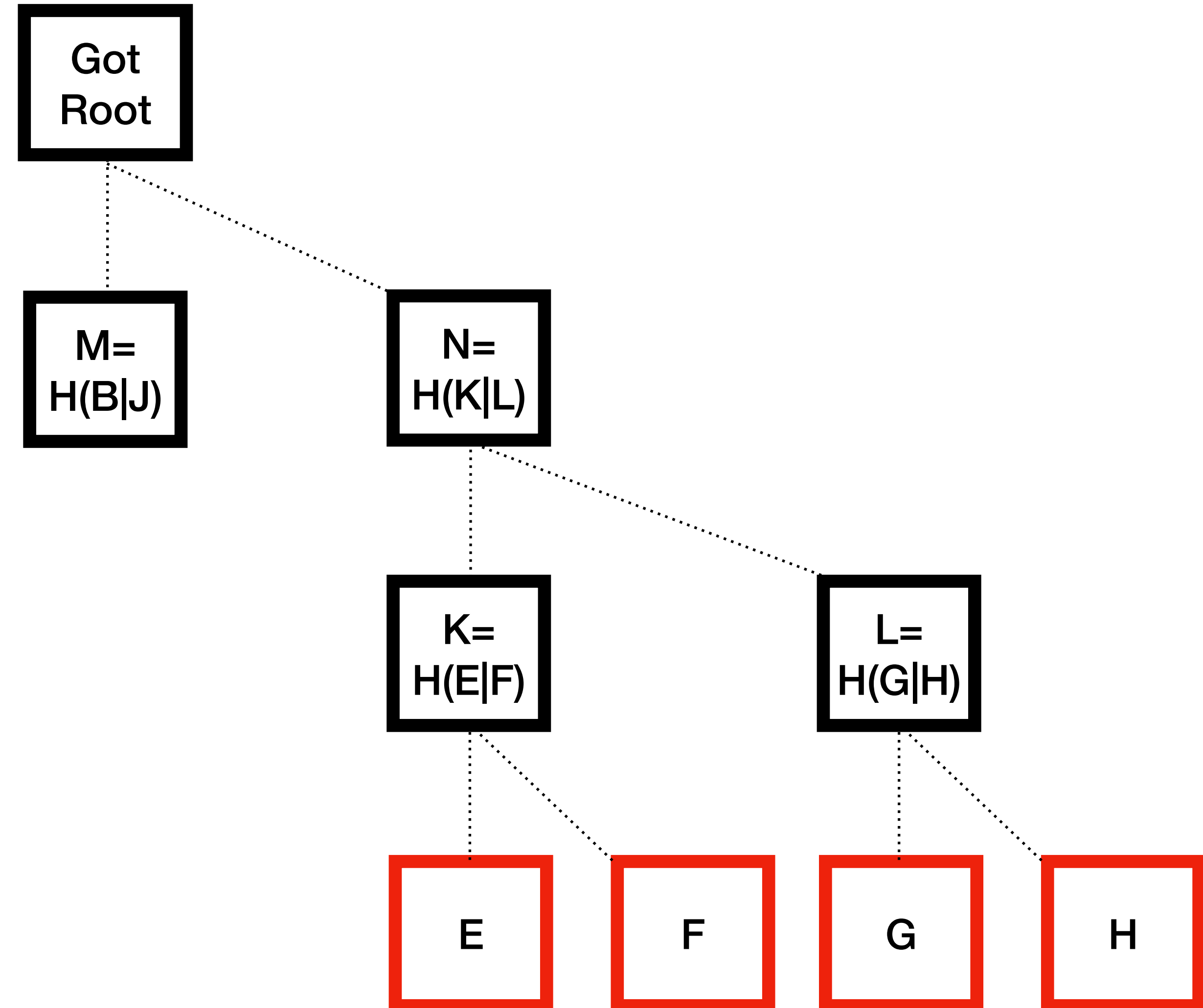
The diagram consists of two square boxes with thick black borders. The left box contains the text 'Root=H(M|N)'. The right box contains the text 'Got Root'. Between the two boxes is an equals sign formed by two parallel horizontal lines.

$=$

Got Root

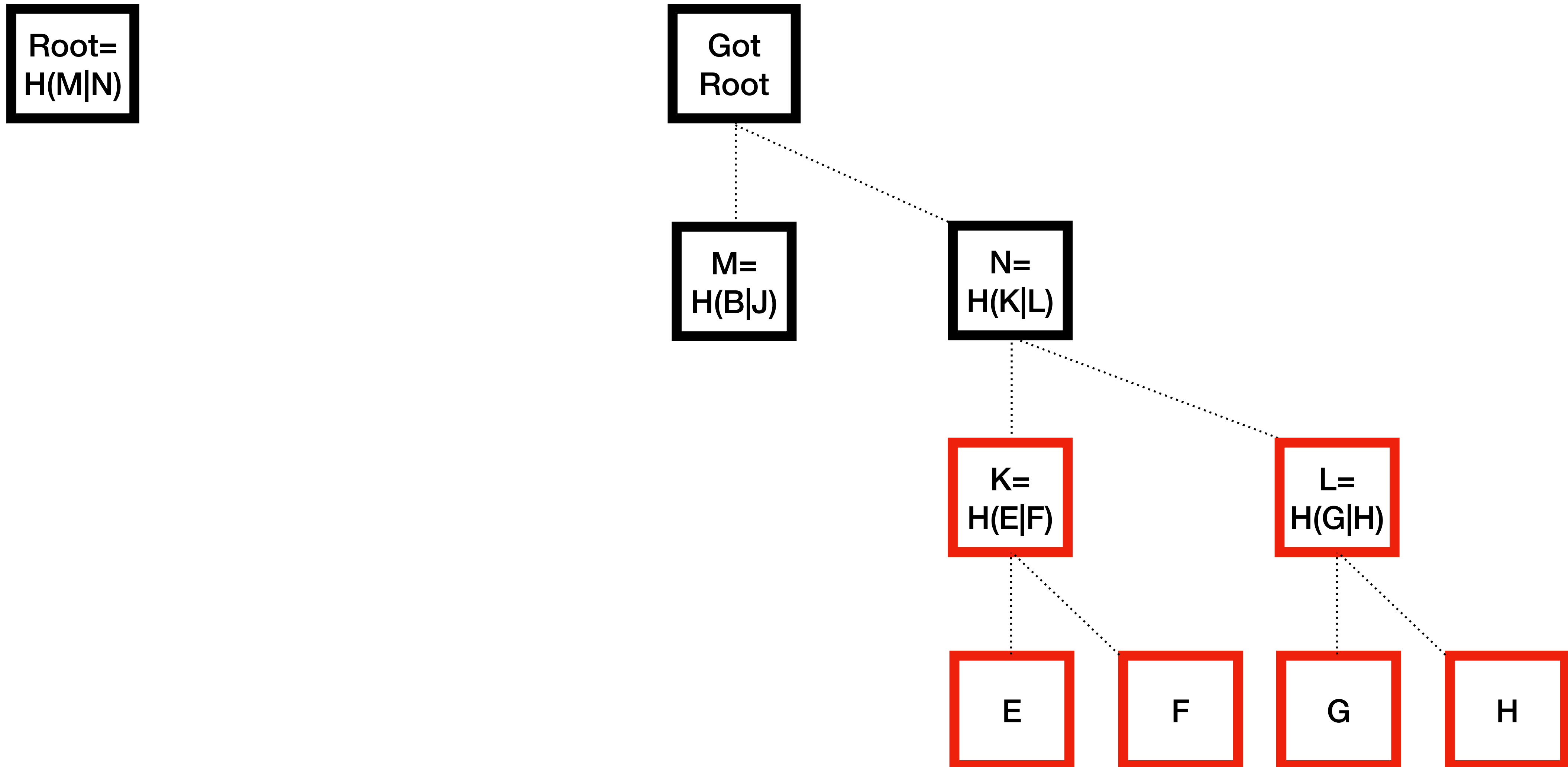
# Calculate new root

Root=  
 $H(M|N)$

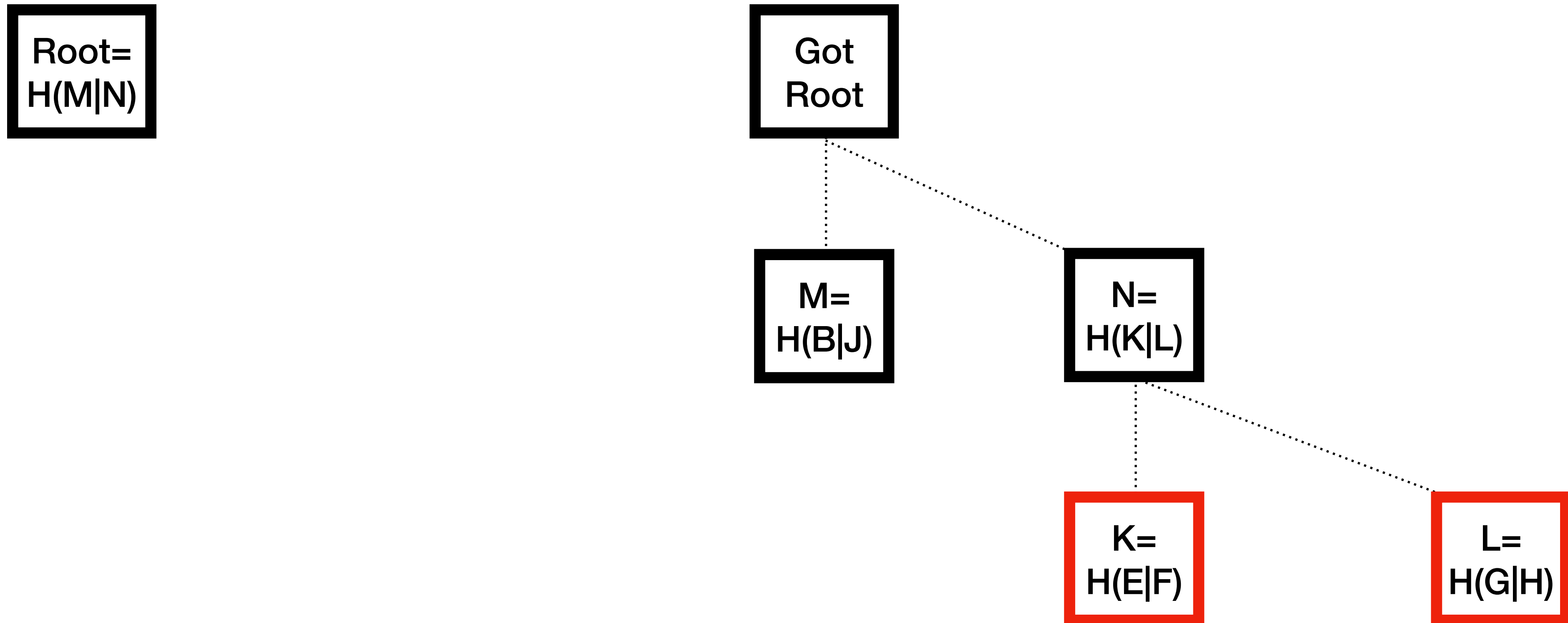




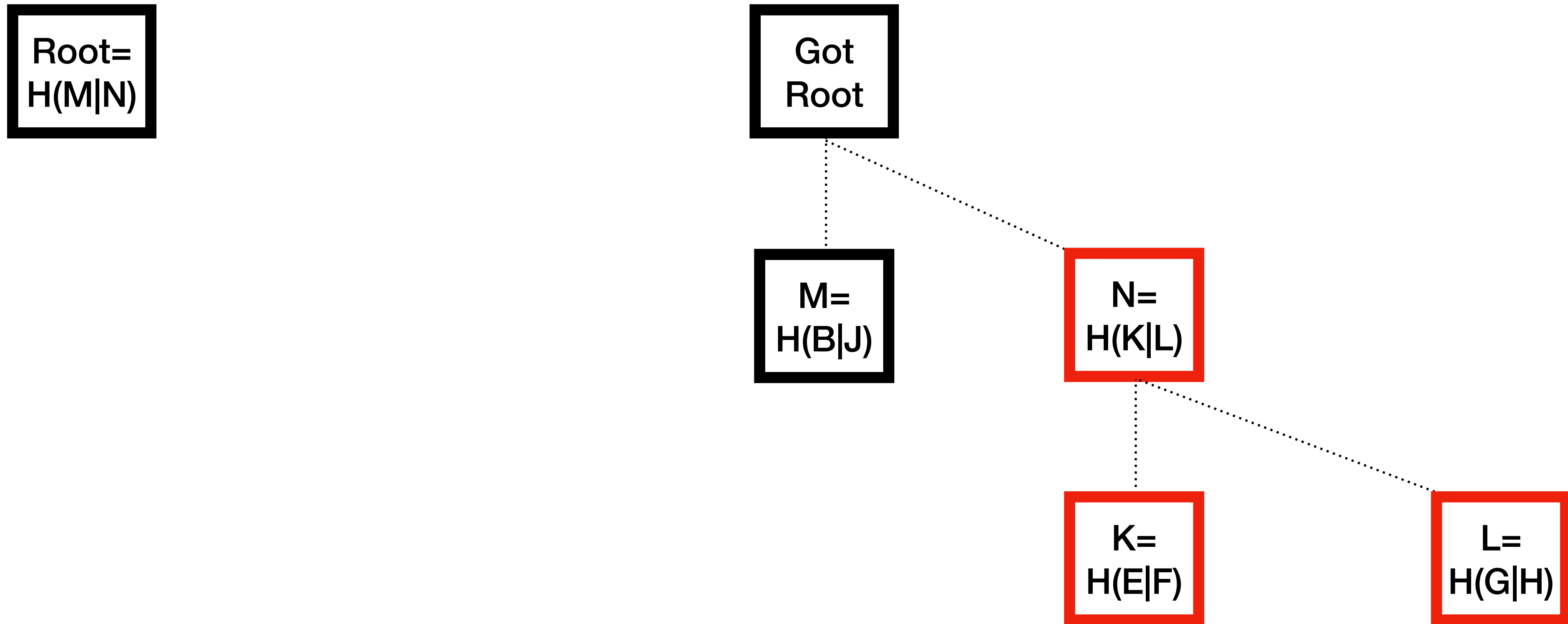
# Mark K & L as node to delete



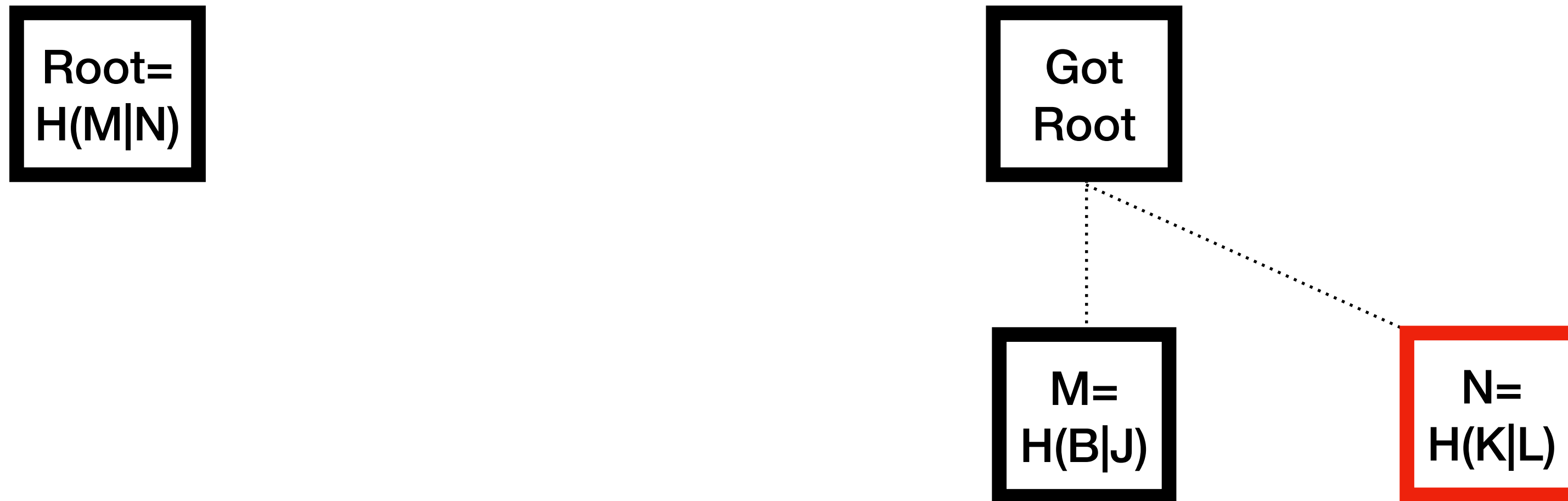
# Remove old nodes for E, F, G, H



# Mark N as node to delete

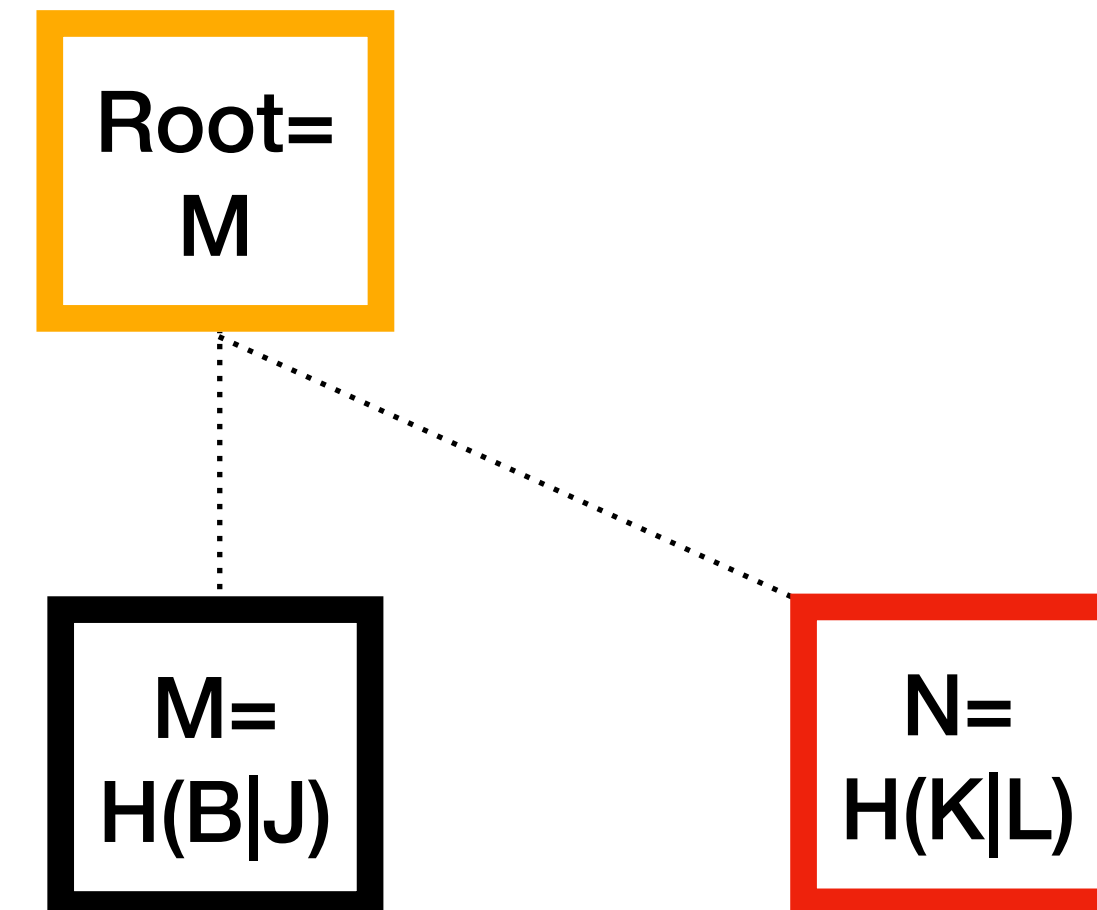


# Remove old nodes for K & L



# Move up M

Root=  
 $H(M|N)$

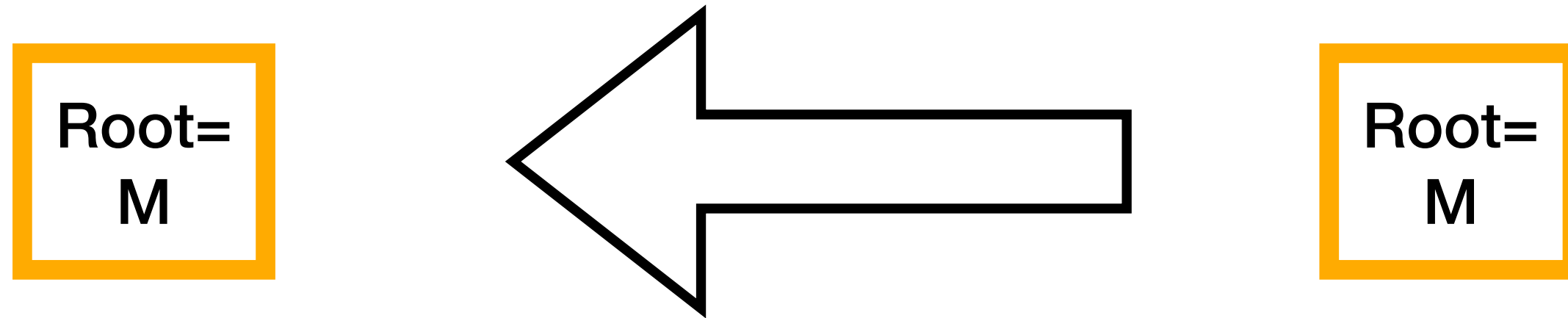


# Remove old nodes for M & N

Root=  
 $H(M|N)$

Root=  
M

# Copy over the new root to be saved

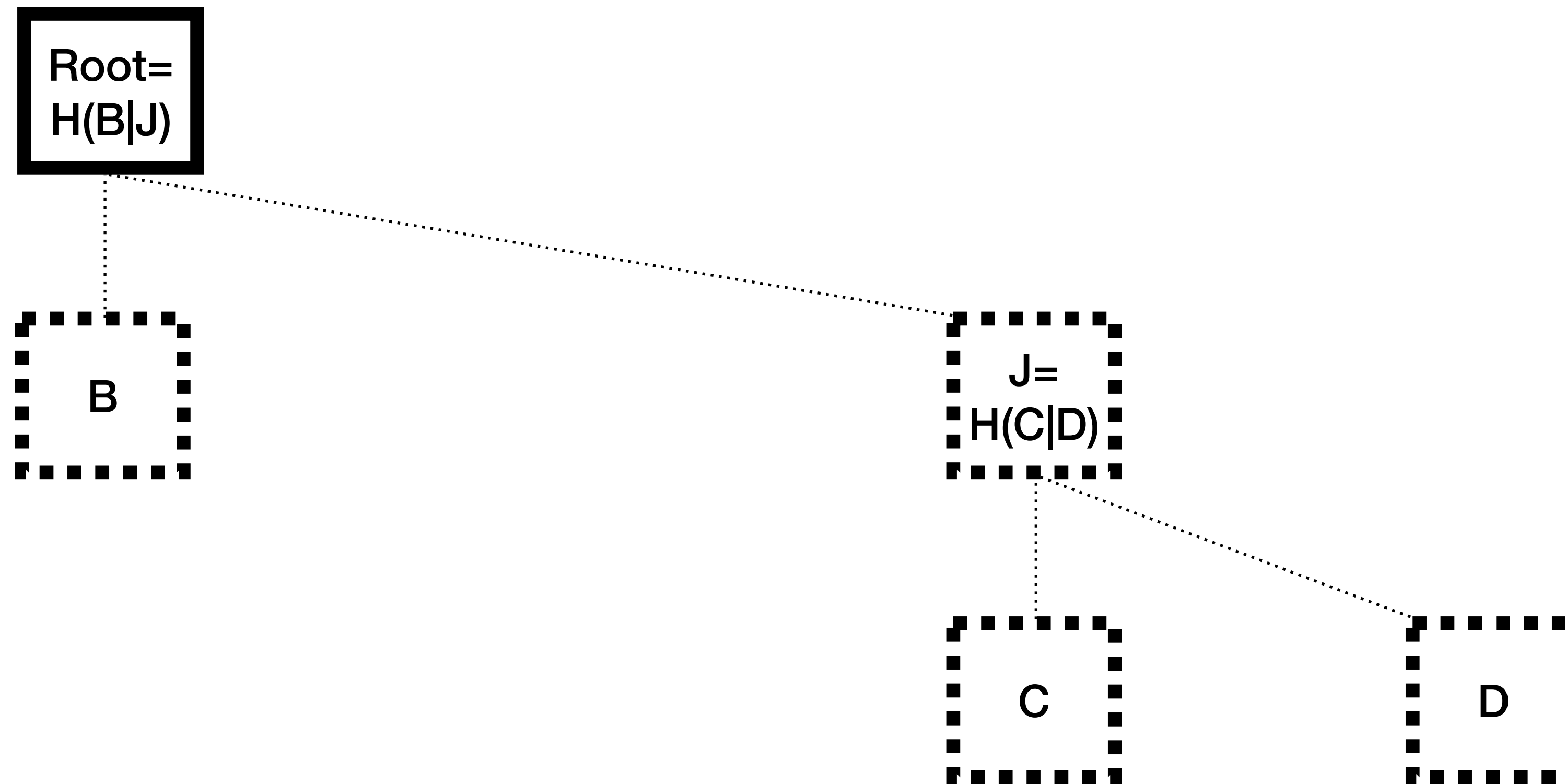


# Done

Root=  
M



# After deleting E, F, G, H



# Consensus

Add, Verify, Delete

- That's all the relevant algorithm for consensus
- Implemented in 174 lines of Python:  
[github.com/utreeexo/pytreeexo](https://github.com/utreeexo/pytreeexo)
- Thanks theStack! (Sebastian Falbesoner)

# Role of levelDB

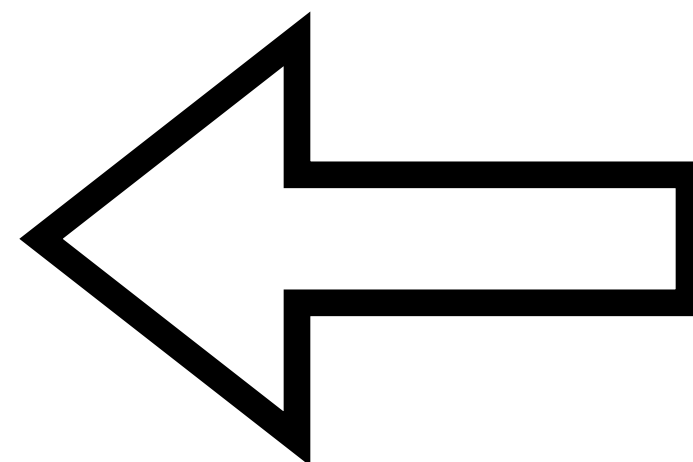
It let's you

1. Add a UTXO
2. Delete a UTXO
3. Tell you the existence of a UTXO
4. Provide the data for verification

# LevelDB

## Fetching a UTXO

LevelDB

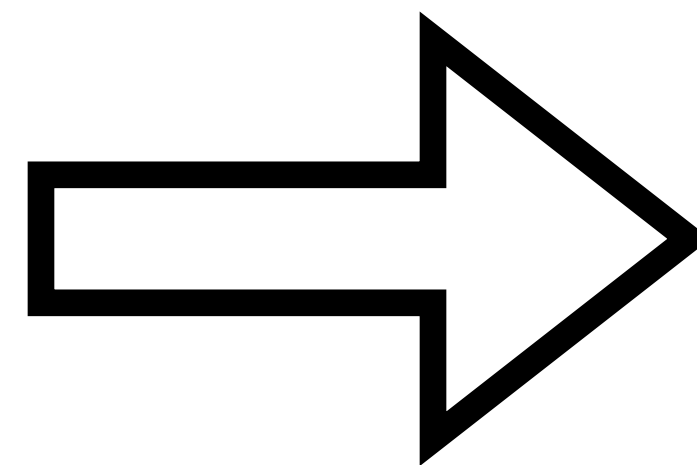


091dd74c2abc5dc5a656450288  
0657037d09f56545b5f08365b6  
5d21dcc19d2f:1

# LevelDB

Fetching a UTXO

LevelDB



Amount: 291312

BlockHeight: 791794

IsCoinbase: False

PkScript: 0014703a38f47197ede65d38c3f79f60093f956d6e43

# No LevelDB with Utreexo



# How the data is provided

What Uteexo nodes download

- Data is provided by the peer
- Sent along with every block or transaction

# TX serialization for Utreexo nodes

Not finalized!

|                    |
|--------------------|
| Version            |
| Flag               |
| TxIn Count         |
| TxIns              |
| TxOut Count        |
| TxOuts             |
| Witness            |
| Locktime           |
| Utreexo Proof Data |



# TX serialization for Utreexo nodes

Not finalized!

|                    |
|--------------------|
| Version            |
| Flag               |
| TxIn Count         |
| TxIns              |
| TxOut Count        |
| TxOuts             |
| Witness            |
| Locktime           |
| Utreexo Proof Data |

# Utreexo Proof Data

Not finalized!

Merkle  
Proof

UTXO  
Data

# Block serialization for Utreexo nodes

Not finalized!

|                            |
|----------------------------|
| Version                    |
| Previous Block Hash        |
| Merkle Root                |
| Timestamp                  |
| Difficulty Bits            |
| Nonce                      |
| Transaction Count          |
| Transactions               |
| Batched Utreexo Proof Data |

# Block serialization for Utreexo nodes

Not finalized!

|                            |
|----------------------------|
| Version                    |
| Previous Block Hash        |
| Merkle Root                |
| Timestamp                  |
| Difficulty Bits            |
| Nonce                      |
| Transaction Count          |
| Transactions               |
| Batched Utreexo Proof Data |

# Batched Utreexo Proof Data

Not finalized!

|                      |
|----------------------|
| Batched Merkle Proof |
| UTXO data Count      |
| UTXO Datas           |

# UTXO Data serialization

## 2 serialization methods

- For calculating the hash to be committed into the accumulator

# UTXO Data serialization

## For hash calculation

|                                                                                     |                                          |
|-------------------------------------------------------------------------------------|------------------------------------------|
| Block Hash                                                                          | The block hash where the tx was included |
| TxHash                                                                              | The transaction hash for the UTXO        |
| Index (Vout)                                                                        | The index within the TX for this UTXO    |
| Block Height + IsCoinBase<br><small>(blockHeight &lt;&lt; 1)&amp;IsCoinBase</small> | Block height and Coinbase indicator      |
| Amount                                                                              | Amount in satoshis                       |
| PkScript length                                                                     | Length of the pkscript in varint         |
| PkScript                                                                            | PkScript itself                          |

# UTXO Data serialization

## For hash calculation

|                                                                                     |                                          |
|-------------------------------------------------------------------------------------|------------------------------------------|
| Block Hash                                                                          | The block hash where the tx was included |
| TxHash                                                                              | The transaction hash for the UTXO        |
| Index (Vout)                                                                        | The index within the TX for this UTXO    |
| Block Height + IsCoinBase<br><small>(blockHeight &lt;&lt; 1)&amp;IsCoinBase</small> | Block height and Coinbase indicator      |
| Amount                                                                              | Amount in satoshis                       |
| PkScript length                                                                     | Length of the pkscript in varint         |
| PkScript                                                                            | PkScript itself                          |



# UTXO Data serialization

## For hash calculation

|                                                            |                                          |
|------------------------------------------------------------|------------------------------------------|
| Block Hash                                                 | The block hash where the tx was included |
| <b>TxHash</b>                                              | <b>The transaction hash for the UTXO</b> |
| Index (Vout)                                               | The index within the TX for this UTXO    |
| Block Height + IsCoinBase<br>(blockHeight << 1)&IsCoinBase | Block height and Coinbase indicator      |
| Amount                                                     | Amount in satoshis                       |
| PkScript length                                            | Length of the pkscript in varint         |
| PkScript                                                   | PkScript itself                          |

# UTXO Data serialization

## For hash calculation

|                                                            |                                          |
|------------------------------------------------------------|------------------------------------------|
| Block Hash                                                 | The block hash where the tx was included |
| TxHash                                                     | The transaction hash for the UTXO        |
| Index (Vout)                                               | The index within the TX for this UTXO    |
| Block Height + IsCoinBase<br>(blockHeight << 1)&IsCoinBase | Block height and Coinbase indicator      |
| Amount                                                     | Amount in satoshis                       |
| PkScript length                                            | Length of the pkscript in varint         |
| PkScript                                                   | PkScript itself                          |

# UTXO Data serialization

## For hash calculation

|                                                                   |                                            |
|-------------------------------------------------------------------|--------------------------------------------|
| Block Hash                                                        | The block hash where the tx was included   |
| TxHash                                                            | The transaction hash for the UTXO          |
| Index (Vout)                                                      | The index within the TX for this UTXO      |
| <b>Block Height + IsCoinBase</b><br>(blockHeight << 1)&IsCoinBase | <b>Block height and Coinbase indicator</b> |
| Amount                                                            | Amount in satoshis                         |
| PkScript length                                                   | Length of the pkscript in varint           |
| PkScript                                                          | PkScript itself                            |

# UTXO Data serialization

## For hash calculation

|                                                                                     |                                          |
|-------------------------------------------------------------------------------------|------------------------------------------|
| Block Hash                                                                          | The block hash where the tx was included |
| TxHash                                                                              | The transaction hash for the UTXO        |
| Index (Vout)                                                                        | The index within the TX for this UTXO    |
| Block Height + IsCoinBase<br><small>(blockHeight &lt;&lt; 1)&amp;IsCoinBase</small> | Block height and Coinbase indicator      |
| Amount                                                                              | Amount in satoshis                       |
| PkScript length                                                                     | Length of the pkscript in varint         |
| PkScript                                                                            | PkScript itself                          |

# UTXO Data serialization

## For hash calculation

|                                                            |                                          |
|------------------------------------------------------------|------------------------------------------|
| Block Hash                                                 | The block hash where the tx was included |
| TxHash                                                     | The transaction hash for the UTXO        |
| Index (Vout)                                               | The index within the TX for this UTXO    |
| Block Height + IsCoinBase<br>(blockHeight << 1)&IsCoinBase | Block height and Coinbase indicator      |
| Amount                                                     | Amount in satoshis                       |
| PkScript length                                            | Length of the pkscript in varint         |
| PkScript                                                   | PkScript itself                          |

# UTXO Data serialization

## For hash calculation

|                                                            |                                          |
|------------------------------------------------------------|------------------------------------------|
| Block Hash                                                 | The block hash where the tx was included |
| TxHash                                                     | The transaction hash for the UTXO        |
| Index (Vout)                                               | The index within the TX for this UTXO    |
| Block Height + IsCoinBase<br>(blockHeight << 1)&IsCoinBase | Block height and Coinbase indicator      |
| Amount                                                     | Amount in satoshis                       |
| PkScript length                                            | Length of the pkscript in varint         |
| PkScript                                                   | PkScript itself                          |

# UTXO Data serialization

## 2 serialization methods

- For calculating the hash to be committed into the accumulator
- For sending data to peers/storage on disk

# UTXO Data serialization

For disk storage/p2p transfer

|                                                            |                                                                                        |
|------------------------------------------------------------|----------------------------------------------------------------------------------------|
| Block Height + IsCoinBase<br>(blockHeight << 1)&IsCoinBase | Block height and Coinbase indicator                                                    |
| Amount                                                     | Amount in satoshis                                                                     |
| IsReconstructablePkScript                                  | True if pkscript can be fetched from the<br>spending TxIn. Nil pkscript for p2pkh&p2sh |
| PkScript length                                            | Length of the pkscript in varint                                                       |
| PkScript                                                   | PkScript itself                                                                        |



# UTXO Data serialization

For disk storage/p2p transfer

|                                                            |                                                                                     |
|------------------------------------------------------------|-------------------------------------------------------------------------------------|
| Block Height + IsCoinBase<br>(blockHeight << 1)&IsCoinBase | Block height and Coinbase indicator                                                 |
| Amount                                                     | Amount in satoshis                                                                  |
| IsReconstructablePkScript                                  | True if pkscript can be fetched from the spending TxIn. Nil pkscript for p2pkh&p2sh |
| PkScript length                                            | Length of the pkscript in varint                                                    |
| PkScript                                                   | PkScript itself                                                                     |

# UTXO Data serialization

For disk storage/p2p transfer

|                                                            |                                                                                     |
|------------------------------------------------------------|-------------------------------------------------------------------------------------|
| Block Height + IsCoinBase<br>(blockHeight << 1)&IsCoinBase | Block height and Coinbase indicator                                                 |
| Amount                                                     | Amount in satoshis                                                                  |
| IsReconstructablePkScript                                  | True if pkscript can be fetched from the spending TxIn. Nil pkscript for p2pkh&p2sh |
| PkScript length                                            | Length of the pkscript in varint                                                    |
| PkScript                                                   | PkScript itself                                                                     |

# UTXO Data serialization

For disk storage/p2p transfer

|                                                            |                                                                                        |
|------------------------------------------------------------|----------------------------------------------------------------------------------------|
| Block Height + IsCoinBase<br>(blockHeight << 1)&IsCoinBase | Block height and Coinbase indicator                                                    |
| Amount                                                     | Amount in satoshis                                                                     |
| IsReconstructablePkScript                                  | True if pkscript can be fetched from the<br>spending TxIn. Nil pkscript for p2pkh&p2sh |
| PkScript length                                            | Length of the pkscript in varint                                                       |
| PkScript                                                   | PkScript itself                                                                        |

# UTXO Data serialization

For disk storage/p2p transfer

|                                                            |                                                                                        |
|------------------------------------------------------------|----------------------------------------------------------------------------------------|
| Block Height + IsCoinBase<br>(blockHeight << 1)&IsCoinBase | Block height and Coinbase indicator                                                    |
| Amount                                                     | Amount in satoshis                                                                     |
| IsReconstructablePkScript                                  | True if pkscript can be fetched from the<br>spending TxIn. Nil pkscript for p2pkh&p2sh |
| PkScript length                                            | Length of the pkscript in varint                                                       |
| PkScript                                                   | PkScript itself                                                                        |

# UTXO Data serialization

For disk storage/p2p transfer

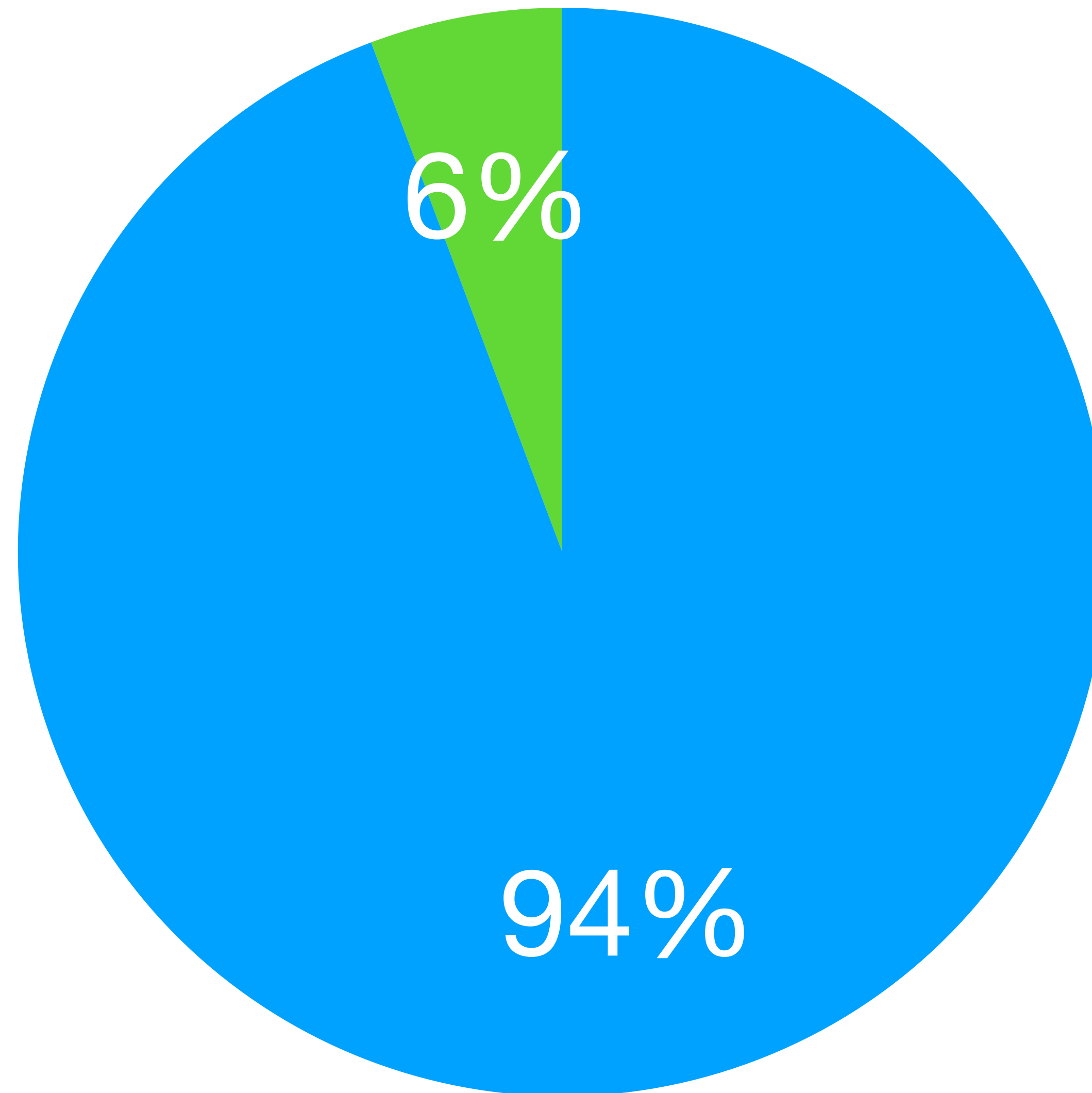
|                                                            |                                                                                     |
|------------------------------------------------------------|-------------------------------------------------------------------------------------|
| Block Height + IsCoinBase<br>(blockHeight << 1)&IsCoinBase | Block height and Coinbase indicator                                                 |
| Amount                                                     | Amount in satoshis                                                                  |
| IsReconstructablePkScript                                  | True if pkscript can be fetched from the spending TxIn. Nil pkscript for p2pkh&p2sh |
| PkScript length                                            | Length of the pkscript in varint                                                    |
| PkScript                                                   | PkScript itself                                                                     |

# Total overhead

Extra data that a utreexo node will download

- 364GB of extra data (as of block 710,000)
- Caching and batch proving multiple blocks is being explored

● Merkle Proof    ● Leaf Data



Measured at block height 710,000. 20GB for LeafData, 343GB for Merkle Proofs

# Current Progress



# [github.com/utreeexo](https://github.com/utreeexo)

Main github organization

- Accumulator implementation in Go
- Full node with Utreeexo
- Full node with electrum-personal-server capability. Full support for protocol v1.4.1.
- Pytreeexo

**[github.com/mit-dci/rustreeexo](https://github.com/mit-dci/rustreeexo)**

**Accumulator in Rust**

- Accumulator implementation in Rust
- Currently missing bridge node capability (2023-05-30)

**[github.com/Davidson-Souza/Floresta](https://github.com/Davidson-Souza/Floresta)**

**Full node in Rust**

- Full node with Utreexo in Rust
- Supports electrum personal server capabilities

# Progress towards end user readiness

## Things done & things left to do

- Accumulator design
- Working full node
- Wallet support
- P2P protocol that supports caching
- Efficient mempool with Utreexo

# Progress towards end user readiness

Things done & things left to do



- Accumulator design
- Working full node
- Wallet support
- P2P protocol that supports caching
- Efficient mempool with Utreexo

# Progress towards end user readiness

Things done & things left to do



- Accumulator design
- Working full node
- Wallet support
- P2P protocol that supports caching
- Efficient mempool with Utreexo

# Progress towards end user readiness

Things done & things left to do



- Accumulator design



- Working full node



- Wallet support

- P2P protocol that supports caching

- Efficient mempool with Utreexo

# Rust implementation

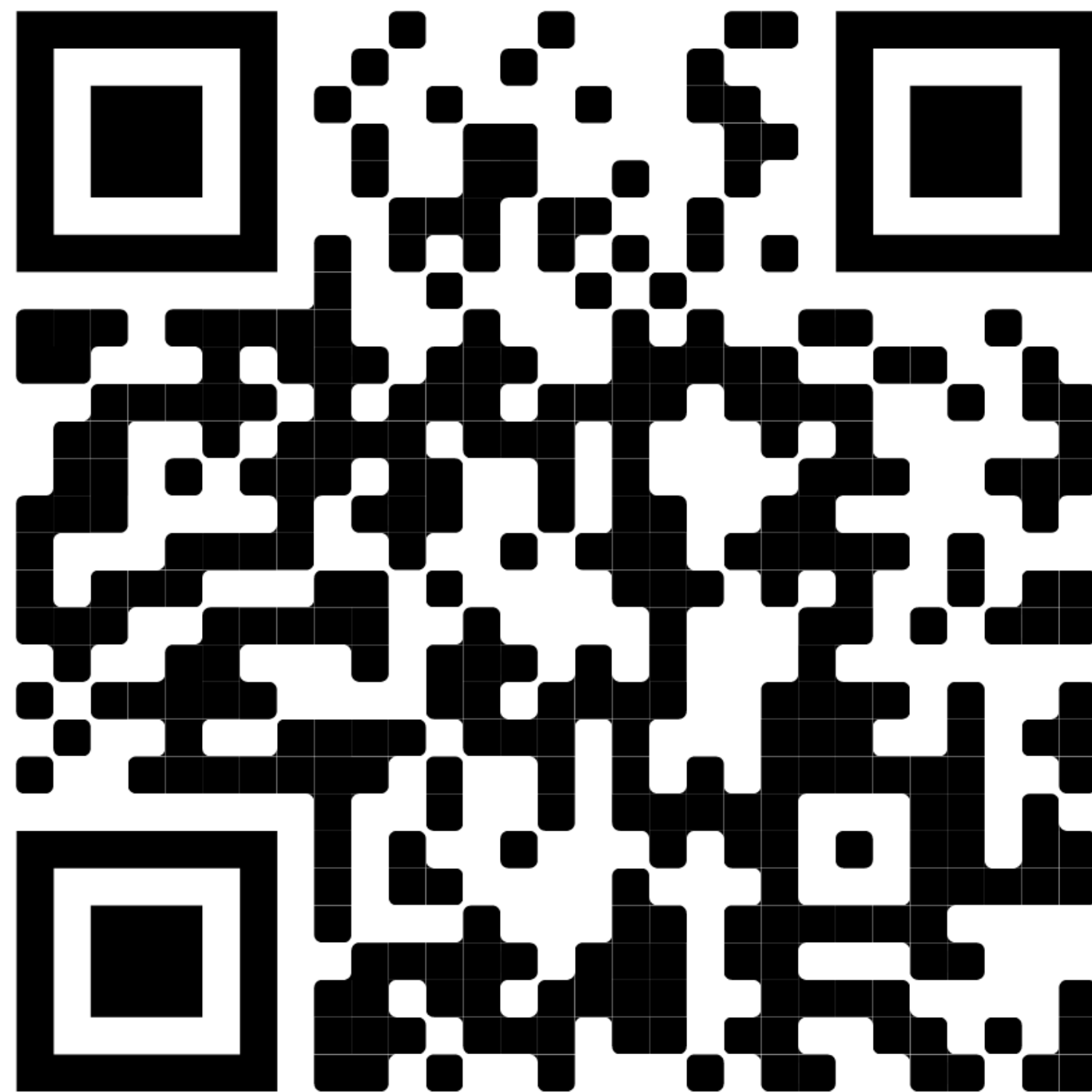
Things going on the Rust side by Davidson



- Working compact utreexo node
- C bindings
- Python bindings
- Javascript bindings



Twitter:  
[@kcalvinalvinn](https://twitter.com/kcalvinalvinn)



Slides:  
[github.com/kcalvinalvin/  
slides-for-btcprague](https://github.com/kcalvinalvin/slides-for-btcprague)

