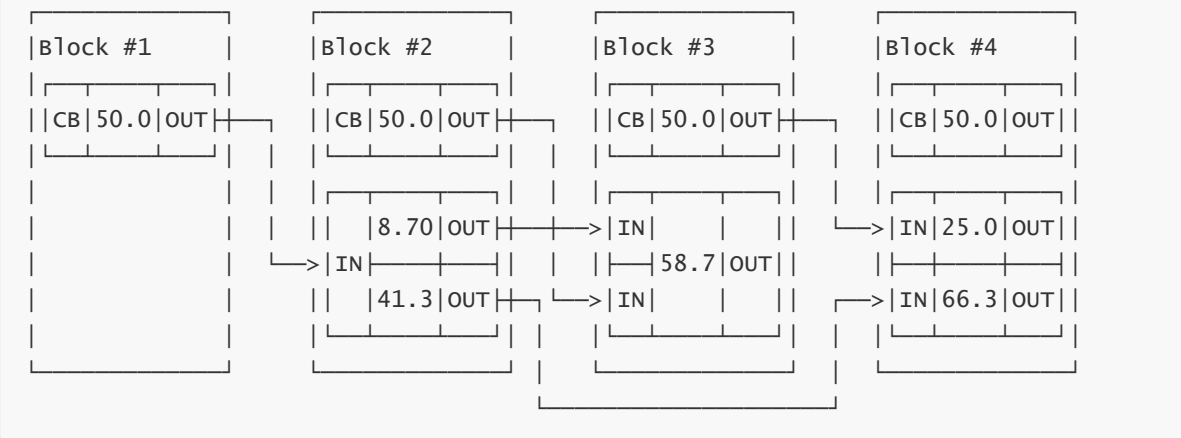
UTXO模型

比特币的区块链由一个个区块串联构成，而每个区块又包含一个或多个交易。

如果我们观察任何一个交易，它总是由若干个输入（Input）和若干个输出（Output）构成，一个Input指向的是前面区块的某个Output，只有Coinbase交易（矿工奖励的铸币交易）没有输入，只有凭空输出。所以，任何交易，总是可以由Input溯源到Coinbase交易。

这些交易的Input和Output总是可以串联起来：



还没有被下一个交易花费的Output被称为UTXO：Unspent TX Output，即未花费交易输出。给定任何一个区块，计算当前所有的UXTO金额之和，等同于自创世区块到给定区块的挖矿奖励之和。

因此，比特币的交易模型和我们平时使用的银行账号有所不同，它并没有账户这个说法，只有UTXO。想要确定某个人拥有的比特币，并无法通过某个账户查到，必须知道此人控制的所有UTXO金额之和。

在钱包程序中，钱包管理的是一组私钥，对应的是一组公钥和地址。钱包程序必须从创世区块开始扫描每一笔交易，如果：

1. 遇到某笔交易的某个Output是钱包管理的地址之一，则钱包余额增加；
2. 遇到某笔交易的某个Input是钱包管理的地址之一，则钱包余额减少。

钱包的当前余额总是钱包地址关联的所有UTXO金额之和。

如果刚装了一个新钱包，导入了一组私钥，在钱包扫描完整个比特币区块之前，是无法得知当前管理的地址余额的。

那么，给定一个地址，要查询该地址的余额，难道要从头扫描几百GB的区块链数据？

当然不是。

要做到瞬时查询，我们知道，使用关系数据库的主键进行查询，由于用了索引，速度极快。

因此，对区块链进行查询之前，首先要扫描整个区块链，重建一个类似关系数据库的地址-余额映射表。这个表的结构如下：

address	balance	lastUpdatedAtBlock
address-1	50.0	0

一开始，这是一个空表。每当扫描一个区块的所有交易后，某些地址的余额增加，另一些地址的余额减少，两者之差恰好为区块奖励：

address	balance	lastUpdatedAtBlock
address-1	50.0	0
address-2	40.0	3
address-3	50.0	3
address-4	10.0	3

这样，扫描完所有区块后，我们就得到了整个区块链所有地址的完整余额记录，查询的时候，并不是从区块链查询，而是从本地数据库查询。大多数钱包程序使用LevelDB来存储这些信息，手机钱包程序则是请求服务器，由服务器查询数据库后返回结果。

如果我们把MySQL这样的数据库看作可修改的，那么区块链就是不可修改，只能追加的只读数据库。但是，MySQL这样的数据库虽然其状态是可修改的，但它的状态改变却是由修改语句（INSERT/UPDATE/DELETE）引起的。把MySQL的binlog日志完整地记录下来，再进行重放，即可在另一台机器上完整地重建整个数据库。把区块链看作不可修改的binlog日志，我们只要把每个区块的所有交易重放一遍，即可重建一个地址-余额的数据库。

可见，比特币的区块链记录的是修改日志，而不是当前状态。

小结

比特币区块链使用UTXO模型，它没有账户这个概念；

重建整个地址-余额数据库需要扫描整个区块链，并按每个交易依次更新记录，即可得到当前状态。