26. März 2020 Studie · Blockchain · 11 Min. Lesezeit

# Analyse des UTXO-Sets von Bitcoin

**Archiviert.** Dieser Inhalt wird nicht mehr aktualisiert. Er bleibt zur Information verfügbar.

**Stand** 2020-03-20
**Quelle** https://technology2.a-sit.at/veroeffentlichungen/analyse-des-utxo-sets-von-bitcoin/

In diesem Projekt wurde das UTXO-Set, d.h. die noch nicht ausgegebenen Transaktions-Outputs von Bitcoin, erklärt und analysiert. Dabei zeigt sich unter anderem, dass das UTXO-Set alle wesentlichen Informationen enthält, um neue Blöcke auf Gültigkeit zu überprüfen, jedoch nur einen Bruchteil der Speicherkapazität der vollen Bitcoin-Blockchain benötigt. Das UTXO-Set erlaubt außerdem, Rückschlüsse über das Alter von Transaktionsoutputs, dem gehaltenem Wert oder die Popularität von einzelnen Transaktionstypen zu ziehen.

## 1. Einleitung

Die Kryptowährung Bitcoin speichert in der Blockchain alle bestätigten (akzeptierten) Transaktionen. Die Transaktionen werden zu Blöcken gruppiert und diese Blöcke werden miteinander verlinkt und bilden die Blockchain. Abbildung 1 zeigt einen Ausschnitt einer Blockchain. Um eine sichere Verlinkung der Blöcke zu gewährleisten, wird der kryptographische Hash des Vorgängerblocks gebildet und in den aktuellen Block aufgenommen. Die Transaktionen sind in der Grafik mittels „Tx“ abgekürzt.

*Abbildung 1: Blockchain: Eine Kette bestehend aus mehreren Blöcken (von links nach rechts gezeichnet).*
Die Blockchain hilft unter anderem, sogenanntes „Double Spending“ zu verhindern. Unter Double Spending versteht man das mehrmalige Ausgeben desselben Tokens. Andererseits ermöglicht die Blockchain allen Teilnehmern alle getätigten Transaktionen abzurufen, zu überprüfen und die ausgebaren Tokens zu errechnen. Die Menge der ausgebaren Tokens wird UTXO-Set genannt. UTXO steht für „Unspent Transaction Output“. Mittels dieses Sets können neu empfangene Blöcke und Transaktionen effizient auf Gültigkeit überprüft werden, ohne die gesamte Blockchain durchsuchen zu müssen. Ein Vorteil dieses Sets gegenüber der gesamten Blockchain ist zudem die Eigenschaft, dass das UTXO Set nicht stetig größer wird, sondern auch schrumpfen kann. Im Vergleich dazu, werden an die Blockchain laufend Blöcke angehängt, wodurch diese mit der Zeit immer größer wird. Mitte März 2020 war die Blockchain bereits ca. 267GB groß [1]. Abbildung 2 zeigt das Wachstum der Bitcoin Blockchain seit ihrem Start. Es ist zu sehen, dass anfangs der benötige Speicherplatz nur sehr langsam stieg. In den letzten Jahren wurde Bitcoin bekannter und beliebter, was sich auch im Wachstum zeigt. Im Vergleich zu den ca. 267GB, benötigt das derzeitige UTXO-Set ca. 3,7GB [2], wobei die durchschnittliche Größe ca. 2,2GB beträgt.

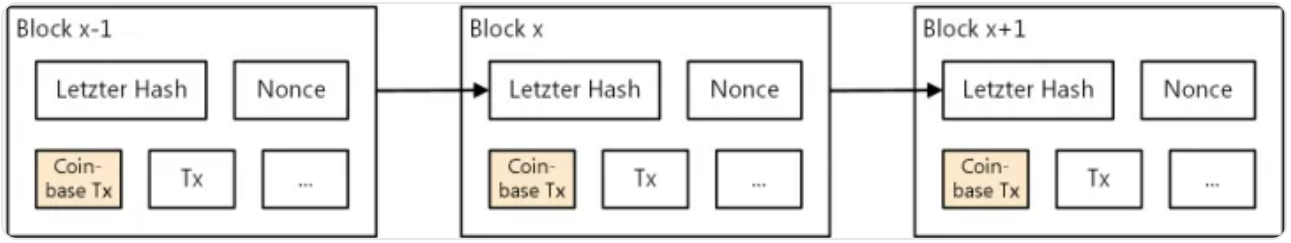



Abbildung 2: Blockchain Wachstum seit dem Start von Bitcoin [1]. Im nächsten Abschnitt wird genauer auf das UTXO-Set eingegangen, sowie das von der Bitcoin Core Software verwendete Speicherformat. Abschnitt 3 stellt die Analyse des UTXO-Sets vor und Abschnitt 4 die Ergebnisse. Abschließend werden Schlussfolgerungen gegeben.

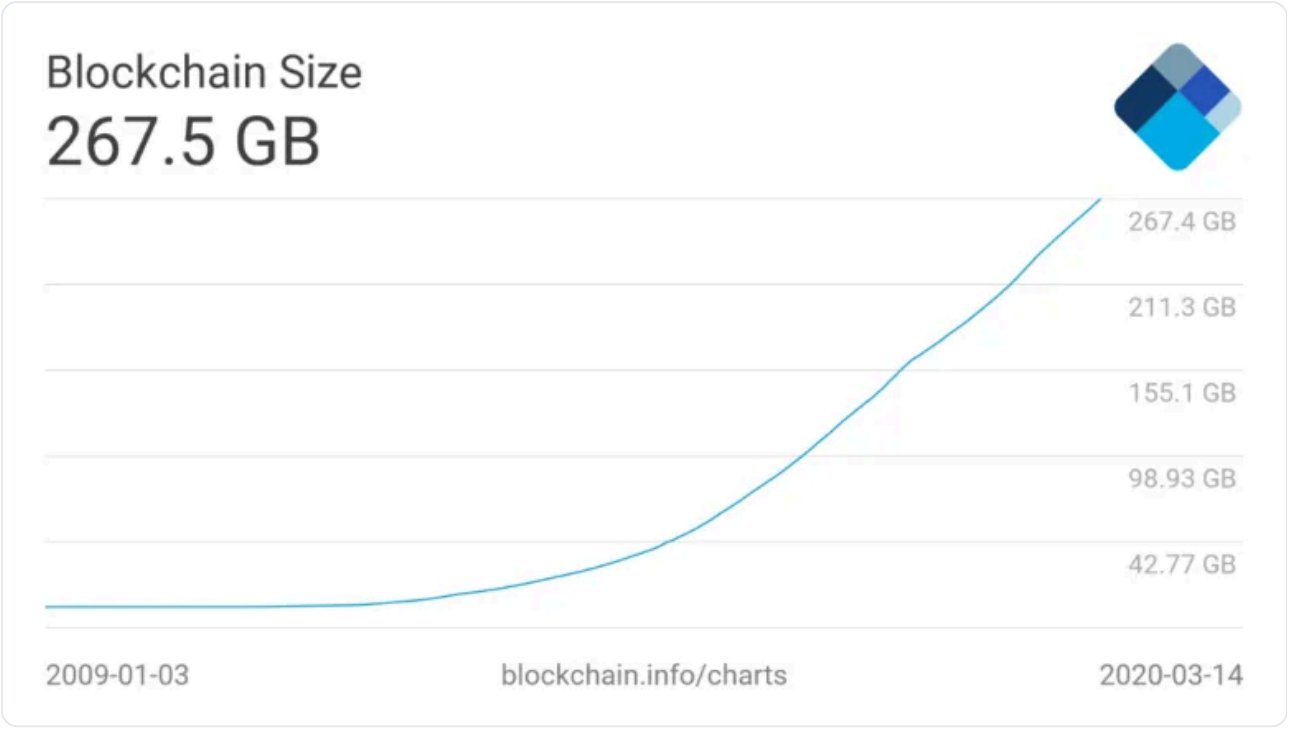



Abbildung 3: Block mit zwei Transaktionen. Bitcoin Core speichert das aktuelle UTXO-Set in einer Schlüssel-Wert Datenbank. Die Daten sind allerdings nicht in für menschenlesbarer Form gespeichert, sondern aus Effizienzgründen kodiert. Im Folgenden wird das seit der Bitcoin Core v0.15 verwendet Format beschrieben. Seit Version 0.15 wird ein Eintrag pro Output verwendet, davor wurden Outputs pro Transaktion gruppiert. In Bitcoin wird das Schlüssel-Wert-Paar „Outpoint-Coin“ genannt. Der Schlüssel „Outpoint“ kodiert den 32-Byte Hashwert der Transaktion, den Index des Outputs in der Transaktion, sowie den Prefix „C“. Der Wert „Coin“ kodiert Metadaten, wie die Blockhöhe, oder ob es sich um eine Coinbase Transaktion handelt, sowie den komprimierten Betrag, den Output-Typ und die Freigabebedingung bzw. das Script. Zudem werden die Einträge in der Datenbank noch mit einem Obfuskierungsschlüssel verborgen. Dadurch können Fehlalarme von Antivirenlösungen vermieden werden, da jede UTXO-Datenbank eine unterschiedliche Dateisignatur erhält. Bei dem Obfuskierungsschlüssel handelt es sich um einen 64-Bit Zufallswert, mit dem allen Daten „verXORed“ werden. So sehen die Einträge für die zwei in Abbildung 3 gezeigten Transaktionen nach der Deobfuskierung aus:

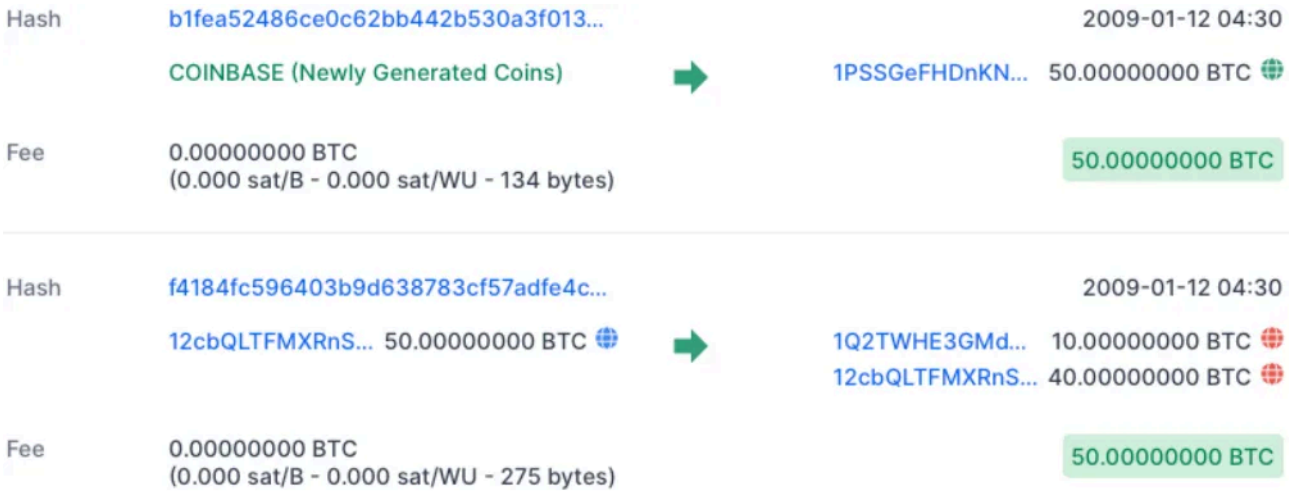
## 2. UTXO-Set

In Bitcoin besteht jede Transaktion aus einem oder mehreren Transaktions-Inputs und Transaktions-Outputs. Im restlichen Dokument werden diese als Inputs und Outputs bezeichnet. Outputs können als Inputs in folgenden Transaktionen verwendet werden, wodurch Währungseinheiten verschoben oder aufgeteilt werden können. Eine Ausnahme sind sogenannte Coinbase-Transaktionen. Diese Transaktionen

haben keinen Input, dürfen aber neue Währungseinheiten erschaffen. Diese Outputs stellen die Belohnung für Miner da, die einen gültigen Block gefunden haben und die Blockchain erweitern. Abbildung 3 zeigt einen Block mit zwei Transaktionen, einer Coinbase Transaktion und einer Transaktion mit einem Input und zwei Outputs. Es handelt sich um die erste nicht Coinbase-Transaktion im Bitcoin-Netzwerk.

```
   4382501c1178fa0b222c1f3d474ec726b832013f0a532b44bb620cce8624a5feb100:8
   1553205d46c4968bde02899d2aa0963367c7a6ce34eec332b32e42e5f3407e052d64ac
   6

   43169e1e83e930853391bc6f35f605c6754cfead57cf8387639d3b4096c54f18f400:8
1540a04ae1a62fe09c5f51b13905f07f06b99a2f7159b2225f374cd378d71302fa2841 4
43169e1e83e930853391bc6f35f605c6754cfead57cf8387639d3b4096c54f18f401:8
154280511db93e1dcdb8a016b49840f8c53bc1eb68a382e97b1482ecad7b148a6909a5 c
```




Hier ist bereits erkennbar, dass drei Outputs erstellt wurden, sowie, dass die letzten beiden Outputs durch dieselbe Transaktion erschaffen wurden, da der Schlüssel fast identisch ist und sich nur im letzten Byte, dem Index, unterscheidet. Nach der Dekodierung bekommt man folgende, für Menschen aufbereitete Daten:

```
{
"index":0,
"tx_id":"82501c1178fa0b222c1f3d474ec726b832013f0a532b44bb620cce8624a5f
eb1",
"coinbase":1,
"amount":5000000000,
"out_type":5,
"data":"05d46c4968bde02899d2aa0963367c7a6ce34eec332b32e42e5f3407e052d6
4ac6"
}{
"index":0,
"tx_id":"169e1e83e930853391bc6f35f605c6754cfead57cf8387639d3b4096c54f1
8f4",
"coinbase":0,
"amount":1000000000,
"out_type":4,
"data":"04ae1a62fe09c5f51b13905f07f06b99a2f7159b2225f374cd378d71302fa2
8414"
}{
"index":1,
"tx_id":"169e1e83e930853391bc6f35f605c6754cfead57cf8387639d3b4096c54f1
8f4",
"coinbase":0,
"amount":4000000000,
"out_type":5,
"data":"0511db93e1dcdb8a016b49840f8c53bc1eb68a382e97b1482ecad7b148a690
9a5c"
}
```


Es ist zu sehen, dass die Byte-Reihenfolge der Transaktions-ID (tx_id) geändert wurde. Zudem wird der Betrag (amount) in Satoshi angegeben. Ein Satoshi entspricht einem Hundertmillionstel Bitcoin.

Um das UTXO-Set aktuell zu halten, muss es nach jedem akzeptierten Block aktualisiert werden. Dabei werden alle Outputs, die in dem neuen Block als Inputs verwenden werden, entfernt und alle neu erstellten Outputs hinzugefügt. Dabei werden die Transaktionen genau in der Reihenfolge bearbeitet, wie sie im Block hinterlegt sind.

Bitcoin unterstützt verschiedene Transaktionstypen, zu den wichtigsten gehören „Pay To Public Key“, „Pay To Public Key Hash“, „Pay To Script Hash“, „Pay To Witness Public Key Hash“, “Pay To Witness Script Hash” und “Pay To MultiSig” Transaktionen. Diese Transaktionstypen werden im Folgenden kurz erklärt:

- **Pay To Public Key (P2PK):** Bei P2PK Transaktionen wird der Output an einen öffentlichen Schlüssel (Public Key) gebunden. Um diesen Output einzulösen, muss eine Signatur mit dem dazugehörigen privaten Schlüssel erstellt werden.

- **Pay To Public Key Hash (P2PKH** ): Dieser Transaktionstyp ähnelt P2PK Transaktionen, der Output wird jedoch nicht an einen öffentlichen Schlüssel, sondern an dessen Hashwert gebunden. Dadurch ergeben sich kleinere Transaktionen wodurch weniger Transaktionsgebühren bezahlt werden müssen.
- **Pay To Script Hash (P2SH):** Bei P2SH Transaktionen wir der Output statt an den Hashwert eines öffentlichen Schlüssels an den Hashwert eines Skriptes gebunden. Um den Output einzulösen, wird das zum Hashwert gehörende Skript benötigt sowie Daten, die das Skript mit dem Rückgabewert „true“ beenden lässt.
- **Pay To Witness Public Key Hash (P2WPKH):** P2WPKH Transaktionen wurden mit dem Protokollupdate „Segregated Witness“ eingeführt und ähneln P2PKH Transaktionen. Im Vergleich zu P2PKH wurde der Beweis des Besitzes vom „scriptSig“ Feld zu einem neuen Feld „witness“ verschoben. Dadurch ergeben sich einige Vorteile, beispielsweise wird dadurch das „Transaction Malleability“ Problem gelöst. „Transaction Malleability“ erlaubt Dritten eine unbestätigte Transaktion so zu verändern, dass sich die Transaktions-ID ändert, wodurch es zu Problemen mit Implementierungen kommen kann, welche sich auf unbestätigte Transaktions-IDs verlassen.
- **Pay To Witness Script Hash (P2WSH):** P2WSH Transaktionen entsprechen im wesentlichen P2SH Transkationen, genauso wie bei P2WPKH wurde jedoch der Beweis des Besitzes vom „scriptSig“ Feld zu einem neuen Feld „witness“ verschoben.
- **Pay To Multisig (P2MS):** P2MS Transaktionen erlauben es, Outputs an mehrere öffentliche Schlüssel zu binden. Um den Output einzulösen wird eine Signatur von einigen oder allen privaten Schlüsseln benötigt. Beispielsweise 2 von 3 Signaturen werden benötigt um einen Output freizugeben.

Im nächsten Abschnitt wird die durchgeführte Analyse des UTXO-Sets vorgestellt.

# 3. Analyse des UTXO-Sets

Das Ziel dieses Projektes ist es, das UTXO-Set zu analysieren. Dabei soll jedoch nicht nur das derzeit aktuelle UTXO-Set analysiert werden, sondern auch die zeitlichen Änderungen. Da der Bitcoin Core Client nur das aktuelle UTXO-Set benötigt, wird nur dieses gespeichert und zur Verfügung gestellt. Im Abschnitt 3.1 wird erklärt, wie man ältere UTXO-Sets bekommen kann und in Anschnitt 3.2 wird erklärt, wie diese Daten aufbereitet wurden.

## 3.1. Zugriff auf ältere UTXO-Sets

Um ein vertrauenswürdiges UTXO-Set zu erhalten, empfiehlt es sich, dieses selbst zu generieren und nicht aus dem Internet zu laden. Im Folgenden wird davon ausgegangen, dass der Bitcoin Core Client [3] verwendet wird.

Wurde die Blockchain noch nicht synchronisiert, bietet der Bitcoin Core Client die Option, nur bis zu einer bestimmten Blockhöhe zu synchronisieren. Mittels des Parameters „stopatheight“ kann die maximal Blockhöhe angegeben werden. Leider eignet sich dieser Ansatz nicht, wenn das UTXO-Set von einer ganz

bestimmten Blockhöhe benötigt wird, da durch die Funktionsweise des Clients teilweise ein paar Blöcke zu viel angehängt werden [4]. Der folgende Befehl startet den Bitcoin Core Daemon (Hintergrundprozess) und synchronisiert die Blockchain bis ca. Block 1.000:

```
./bitcoind -stopatheight=1000
```

Einen anderen Ansatz verfolgen die Entwickler der „Bitcoin Tools“, einer Python Bibliothek für Forschung und Lehre [5]. Sie stellen ein Script zur Verfügung, welches den „Bitcoind“-Prozess beendet, sobald dieser einen bestimmten Block empfängt [6]. Dafür beobachtet das Script die Log-Datei des Prozesses.

Für dieses Projekt wurde jedoch ein dritter Ansatz verfolgt, bei welchem zuerst die Blockchain synchronisiert wird, mindestens bis zu dem gewünschten Block und dann mittels des „invalidateblock“ Befehls gezielt der Nachfolger des gewünschten Blocks auf „Ungültig“ gesetzt wird, wodurch der Client automatisch auf die derzeit längste gültige Kette wechselt. Nachdem der Block ungültig gesetzt wurde, wird die längste Kette beim Startblock (Block 0) anfangen und beim gewünschten Block enden. Dafür muss zuerst der Bitcoin Core Daemon gestartet werden. Da wir an keinen neuen Blöcken interessiert sind, starten wir den Daemon im Offline-Modus (connect=0):

```
./bitcoind -connect=0 -daemon
```

Anschließend kann der Nachfolgeblock des Zielblockes ungültig gesetzt werden:

```
./bitcoin-cli invalidateblock
00000000000080b66c911bd5ba14a74260057311eaeb1982802f7010f1a9f090
```

Bei “00000000000080b66c911bd5ba14a74260057311eaeb1982802f7010f1a9f090” handelt es sich um den Hashwert des Blockes mit der Höhe 100.001. Da Block 100.001 ungültig gesetzt wurde, wird der Block 100.000 der derzeit beste Block und der Bitcoin Core Daemon berechnet das UTXO-Set für diese Blockchain. Durch beenden des Daemons wird das aktuelle UTXO-Set auf die Festplatte geschrieben:

```
./bitcoin-cli stop
```

Im nächsten Abschnitt wird die Analyse vorgestellt.

## 3.2. Analyse

Im Rahmen dieses Projektes wurde zuerst die Blockchain bis zum Block 300000 [1] synchronisiert und danach das UTXO-Set für jeden 100. Block [2] erstellt und in einer Datenbank gespeichert. Da ein UTXO-Set mehrere Gigabyte groß werden kann, würde es enorme Mengen an Speicherplatz benötigen, alle Sets komplett zu speichern. Stattdessen werden nur noch nicht vorhanden Outputs in die Datenbank aufgenommen und zusätzlich wird mitgespeichert, in welchem Block der jeweilige Output zuletzt beobachtet wurde. Daraus kann man dann schließen, ob der Output ausgegeben wurde und falls ja in

welchem Block. Je mehr UTXO-Sets analysiert werden, desto genauer werden die Ergebnisse. Sind die Schritte zu groß, könnten Transaktionen bzw. Output übersehen werden, die kurze Zeit nach der Erstellung wieder ausgegeben wurden. Im nächsten Abschnitt werden die Ergebnisse vorgestellt.

## 4. Ergebnisse

Abbildung 4 zeigt den Verlauf der UTXO-Set Größe von Block 0 bis Block 300000. Im Vergleich dazu zeigt Abbildung 5 die akkumulierte Größe der Blöcke. Die Abbildung zeigt die Größe der Blockchain einmal mit den Zeugendaten und einmal ohne den Zeugendaten. Da die Aufspaltung (Segregated Witnesses) erst seit Blocknummer 477.120 unterstützt wird, sind die Kurven davor ident.

1 Geplant ist die Blockchain vollständig zu synchronisieren und anschließend das UTXO Set für jeden Block zu erstellen. 2 Teilweise wurden die UTXO-Sets auch schon in kleineren Abständen generiert.

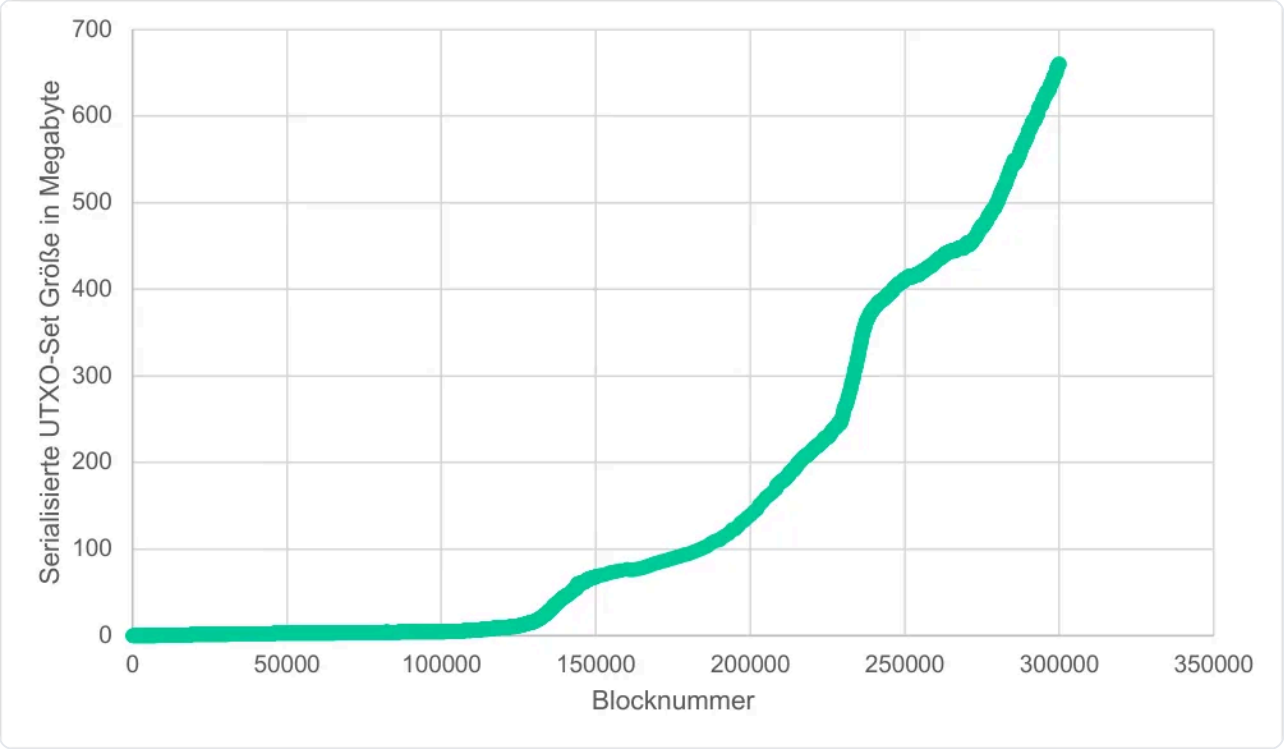



Abbildung 4: Größe des serialisierten UTXO-Sets.

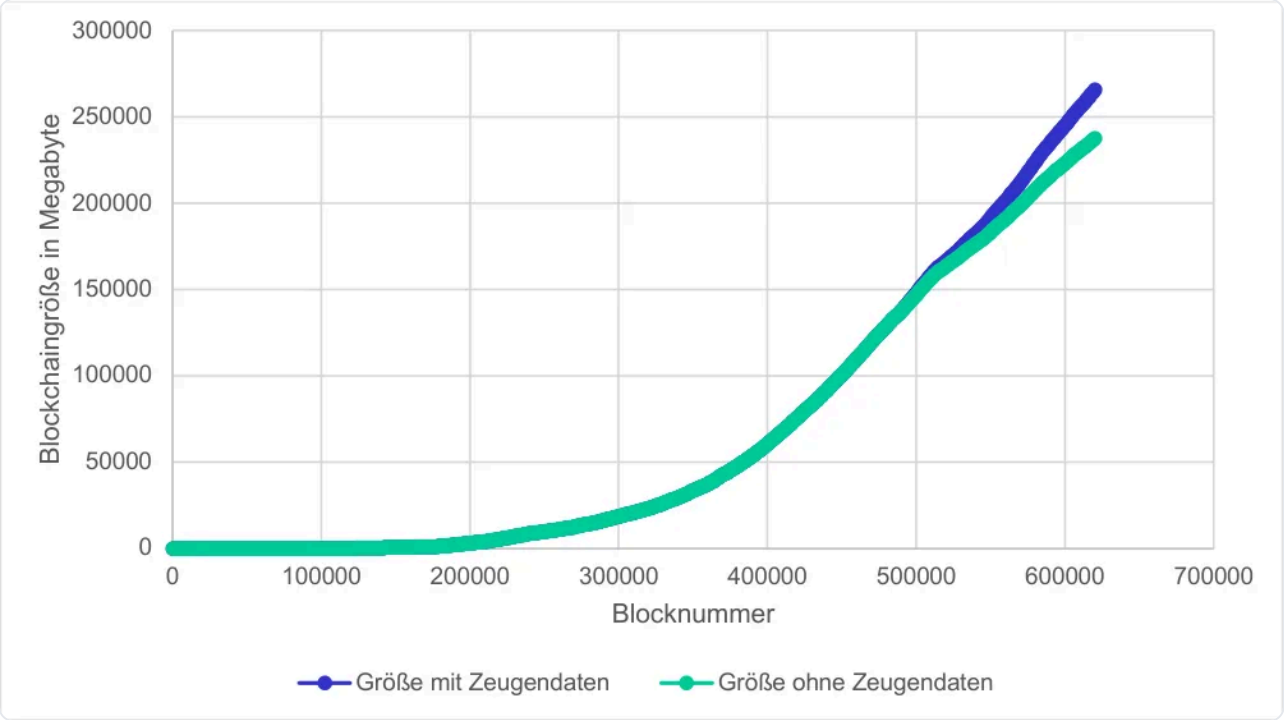



Abbildung 5: Blockchaingröße mit und ohne Zeugendaten. Abbildung 6 zeigt im Vergleich dazu die Größe der Einzelblöcke mit und ohne Zeugendaten. Es ist erkennbar, dass die maximale Blockgröße von einem Megabyte vor der Unterstützung für „Segregated Witnesses“ bereits ausgereizt wurde.

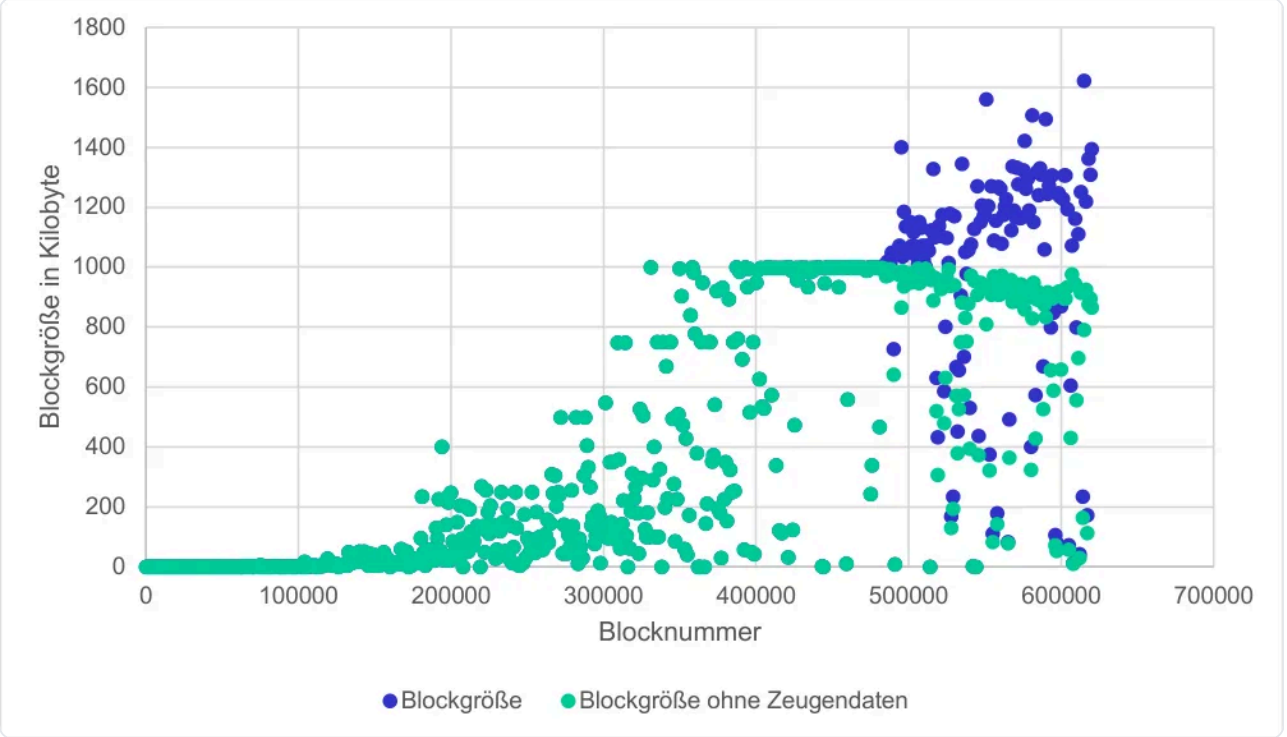



Abbildung 6: Blockgröße mit und ohne Zeugendaten. Abbildung 7 zeigt die Anzahl der Transaktionen pro Block. Hier ist erkennbar, dass anfangs nur wenige Transaktionen getätigt wurden, das Transaktionsvolumen mit der Zeit aber deutlich zugenommen hat.

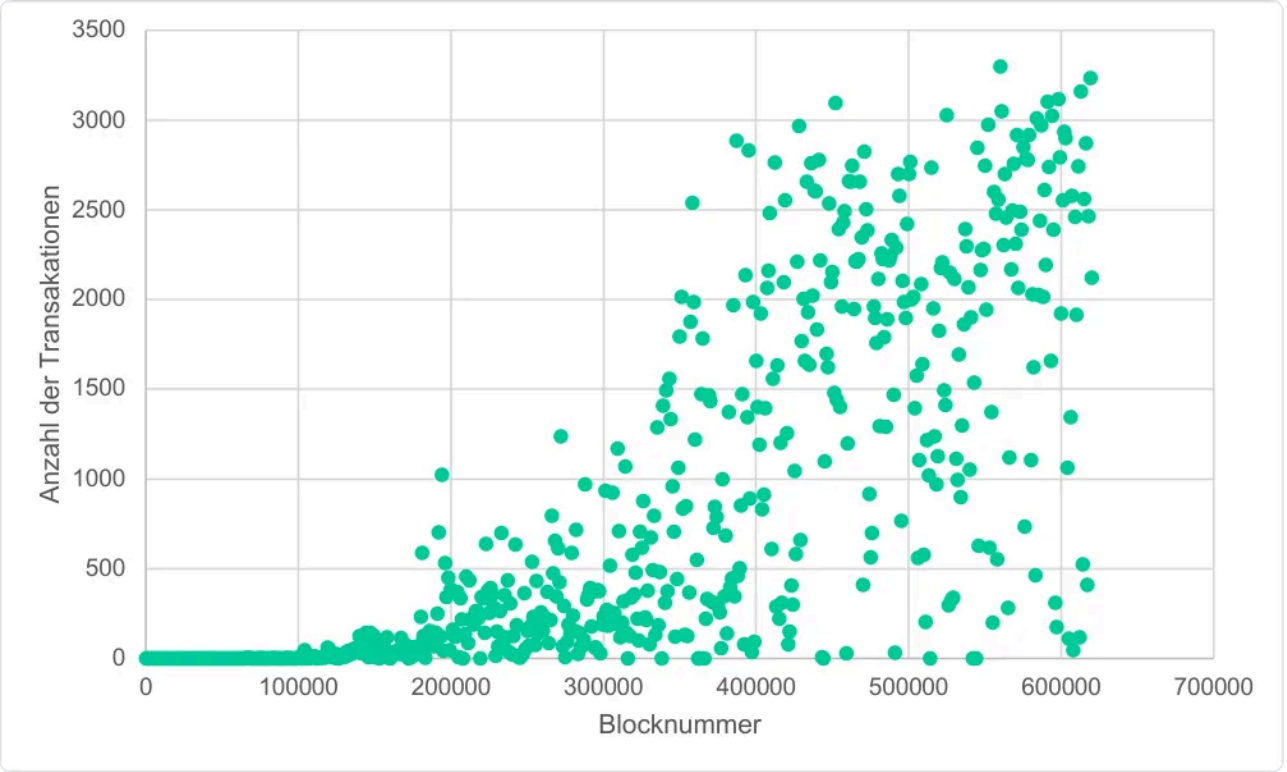



Abbildung 7: Anzahl der Transaktionen pro Block. Mit Stand Block 620.000 enthält das UTXO-Set 66.061.459 Outputs, erzeugt von 39.432.155 verschiedenen Transaktionen. Der wertvollste, nicht ausgegebene Output hält 17.650.099.963.419 Satoshi, oder umgerechnet ca. 176.501 Bitcoins. Der größte Output im gesammelten Datensatz hält 497.976,38 BTC und wurde Ende 2011 erstellt. Dabei handelt es sich jedoch nicht um den größten Output aller Zeiten, da im Zuge dieser Analyse nicht jedes UTXO-Set analysiert wurde. Abbildung 8 zeigt die kumulative Verteilung über die Anzahl der Outputs pro Transaktion. Es ist erkennbar, dass der Großteil der Transaktionen weniger als 3 UTXOs hat. Abbildung 9 zeigt die kumulative Anzahl von UTXOs in Abhängigkeit vom Wert in Satoshi. Es ist beispielsweise erkennbar, dass kaum ein UTXO einen Wert von weniger als 1000 Satoshi hält, aber 95% der UTXOs einen Wert kleiner als 10 [8 ] Satoshi (1 Bitcoin) haben. Abbildung 10 zeigt den zeitlichen Verlauf der verschiedenen Transaktionstypen im UTXO-Set. Die Abbildungen 8-10 wurden mit dem „Statistical Analysis Tool for UTXO Set“ [7] erstellt.

| **Abbildung 8: Kumulative Verteilung über die Anzahl der Outputs pro Transaktion.** | **Abbildung 9: Wert per UTXO in Satoshi.** |
|---|---|
| *Abbildung 10: Zeitlicher Verlauf verschiedener Transaktionstypen im UTXO-Set.* | *Abbildung 10: Zeitlicher Verlauf verschiedener Transaktionstypen im UTXO-Set.* |

Hier ist erkennbar, wie sich die Popularität der einzelnen Transaktionstypen im Laufe der Zeit verändert hat.

## 5. Conclusio

Beim UTXO-Set handelt es sich um eine wichtige Datenstruktur, welche alle ausgebaren Outputs enthält. Dadurch ermöglicht es die effiziente Validierung von neuen empfangenen Transaktionen und Blöcken, ohne dass die gesamte Blockchain benötigt wird. Im Vergleich zur Blockchain benötigt das UTXO-Set zudem deutlich weniger Speicherplatz und wächst bzw. schrumpft mit der Anzahl der enthaltenen Outputs. Mit Stand Block Nummer 620.000 hält das UTXO-Set ca. 66 Millionen Einträge.

## Referenzen

[1] Blockchain.com, „Blockchain Size,“ [Online]. Available:

https://www.blockchain.com/charts/blocks-size. [Zugriff am 16 03 2020].

[2] statoshi.info, „Size of Serialized UTXO Set,“ [Online]. Available:

https://statoshi.info/dashboard/db/unspent-transaction-output-set. [Zugriff am 16 03 2020].

[3] Bitcoin Core, „Bitcoin Core,“ [Online]. Available: https://github.com/bitcoin/bitcoin. [Zugriff am 18 03 2020].

[4] Sjors, „Bitcoin Core -stopatheight is imprecise #13477,“ 15 06 2018. [Online]. Available:

https://github.com/bitcoin/bitcoin/issues/13477. [Zugriff am 18 03 2020].

[5] S. D. Segura und C. Pérez, „Python Bitcoin tools,“ [Online]. Available: https://github.com/srgi/bitcoin_tools. [Zugriff am 18 03 2020].

[6] C. Pérez, „kill_at_heigh.py,“ [Online]. Available: https://github.com/srgi/bitcoin_tools/blob/dev/bitcoin_tools/analysis/status/kill_at_heigh.py. [Zugriff am 18 03 2020].

[7] S. Delgado-Segura, C. Pérez-Solà, G. Navarro-Arribas und J. Herrera-Joancomartí, „STATUS,“ [Online]. Available: https://github.com/srgi/bitcoin_tools/tree/master/bitcoin_tools/analysis/status. [Zugriff am 17 03 2020].