Credible Decentralized Exchange Design via Verifiable Sequencing Rules

2024-2025 学年春夏学期计算经济学讨论班

金政羽

Clovers2333@gmail.com

浙江大学计算机科学与技术学院

2025 年 **5** 月 **16** 日

Contents

- Introduction of Decentralized Finance
  - Blockchain
  - Traditional Finance(Centralized)
  - Liquidity Pool
  - Miner extractable value(MEV)
  - Mechanism design with imperfect commitment.
- Background
  - Quasiconcave Function
  - Potential Functions
  - Model discussion
- Communication Model
  - Sequencing Rule
  - User and miner utility
- Market Manipulation
- Greedy Sequencing Rule

Blockchain

有关区块链的基础可以看 3Blue1Brown 的视频.

为了安全和去中心化，我们默认账本的实现都是使用区块链的，并且无论是 PoW 还是 PoS 的区块链，矿工都是必不可少的.

Traditional Finance(Centralized)

传统金融中，交易双方需要通过中介机构来完成交易，比如银行.

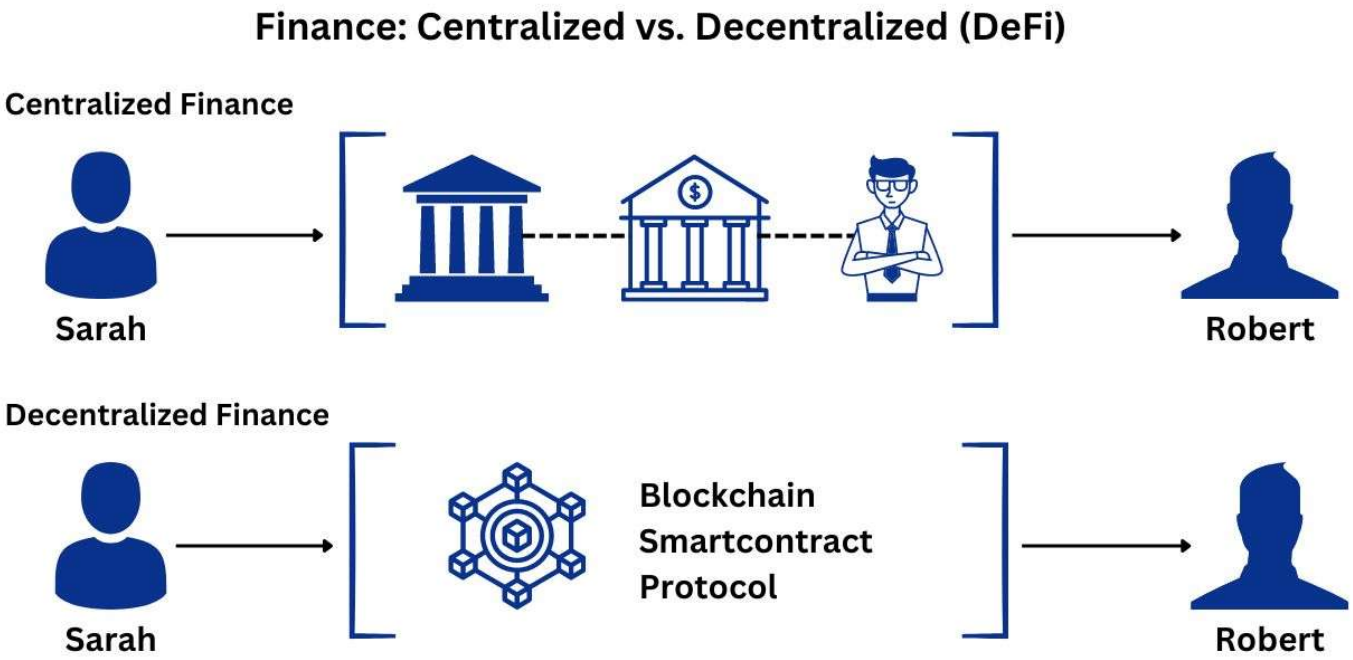
传统的交易方法一般称为" 挂单"，具体的流程如下：

1. 挂单（Limit Order）：用户指定价格和数量（例如" 以 $2000 的价格卖出 1 ETH"）.
2. 撮合交易：交易所需要找到匹配的买卖订单（比如有人挂" 以 $2000 买入 1 ETH"），才能成交.
3. 等待时间：如果当前没有匹配的订单，你的挂单会一直留在订单簿中，直到有人接盘.

区块链的计算和存储资源非常稀缺，订单簿模式会导致：

1. 高内存消耗：每个未成交的挂单都要存储在链上，订单越多，占用空间越大.
2. 高计算成本：每次交易需要扫描订单簿、匹配买卖双方、更新剩余挂单，这些操作在链上非常昂贵（Gas 费高）.
3. 延迟问题：如果没人接盘，交易可能长时间不成交.

Traditional Finance(Centralized)



Liquidity Pool

针对”没有匹配就不会交易”的问题，我们引入流动性池（Liquidity Pool），我们不再是通过中介机构匹配人和人进行交易，而是每个人都可以在池子里进行直接交易.

流动性池的实现方式是:

1. 初始化：设置一个初始价格和初始数量.
2. 交易：当用户想要买入或卖出时，系统根据当前价格和数量计算出交易结果，用户直接从池子中获取货币.
3. 价格调整：交易后，价格会根据买卖双方的力量自动调整.

Liquidity Pool

在本篇论文中，我们假定只有两种货币进行交易，假设两种货币的数量分别为 X_1, X_2，我们流动性池在交易的时候时刻保证 $X_1 \cdot X_2 = k$，其中 k 是一个常数.

因为乘积不变，所以在交易的过程中，会根据当前两种货币的比例来决定两种货币之间的汇率，也间接实现了”需求量大的货币价格更高”的经济学原理.

Miner extractable value(MEV)

在使用区块链进行具体交易的时候，矿工会把若干条交易打包在一个 Block 中，Pool 根据 Block 中的交易顺序进行交易，然后挖出校验码，获得挖矿奖励和手续费.

但是事实上，矿工不仅可以通过上述两种方法盈利，还可以通过操纵交易顺序来获得更多的收益，这就是所谓的 Miner extractable value(MEV).

具体来说，当矿工看到一个大笔的买入时，他可以在这笔买入前后插入自己的交易：自己先在低价买入，然后执行完大额买入以后该货币价格会变高，然后自己再卖出.

这种 MeV 是矿工通过剥削用户的价值来实现的，也是我们接下来要避免的情况. 本文的重心，就是找到一种 Block 的排序和检测方式，试图使得矿工无法通过操纵交易顺序来剥削利益.

Miner extractable value(MEV)

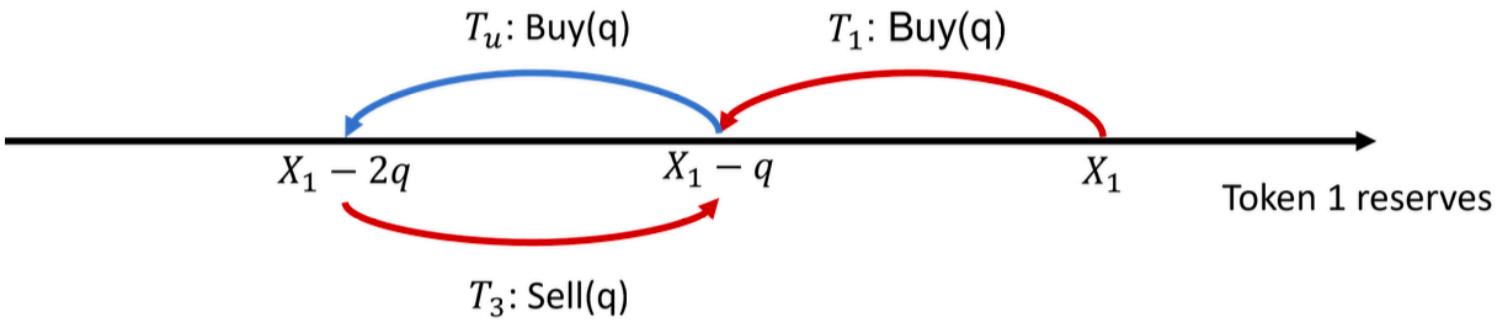


Figure 1: A front-running scheme (a sandwich attack) with execution ordering (T_1, T_u, T_3), where T_u is the user's transaction. The miner's orders, T_1 and T_3, are in red, and the user's order, T_u, is in blue.

Miner extractable value(MEV)

有关 MEV 的解决方案，先前提出过以下几种方法：

1. 批量拍卖（Batch Auctions）
   - 原理：将一段时间内的所有交易打包成一个批次，按统一的市场清算价格结算.
   - 优点：避免交易顺序影响价格，防止矿工操纵.
   - 局限性：区块链本质是顺序执行交易，批量拍卖需改造底层机制，引入延迟和计算开销；不适合高频交易的 DeFi 场景.
2. 可信中继服务（如 Flashbots Protect）
   - 原理：用户将交易私密发送给中继服务，由其中继协调矿工打包，承诺不利用信息优势作恶.
   - 优点：快速落地，减少公开交易被抢跑的风险.
   - 局限性：信任问题——用户无法验证中继服务或矿工是否暗中操纵顺序；中心化风险——依赖少数”可信”中继节点，违背去中心化初衷.

Mechanism design with imperfect commitment

除了 Sandwich attack 以外，矿工还有一种常见的恶意盈利方法——通过制造堵塞来提高手续费.

矿工可以在 Block 中插入大量的垃圾交易（自己买入自己卖出），这样会导致 Block 的剩余空间变小，这样普通用户为了正常完成交易不得不付出更高的手续费（参考经济危机” 倒牛奶”）.

机制设计的目标：在手续费中 Burn 掉一部分，使得矿工自己制造垃圾交易也需要成本，最终使得矿工没有动力这么做.

Quasiconcave Function

定义 (Quasiconcave Function)

我们称函数 $f: \mathbb{R}^n \to \mathbb{R}$ 是拟凹的，如果对于任意的 $x, y \in \mathbb{R}^n$ 和任意的 $\lambda \in [0,1]$，有

$$f(\lambda x + (1-\lambda)y) \geq \min\{f(x), f(y)\}.$$

Potential Functions

定义 (Potential Function)

在货币市场中，我们要定义一个势能函数来衡量 Liquidity Pool 的流动性，$\phi : \mathbb{R}^n \to \mathbb{R}$，我们需要保证，任意一次交易后，势能函数都不能下降，记交易前的状态为 X，交易后的状态为 X'，则有：

$$\phi(X') \geq \phi(X).$$

对于在状态 X 下执行的买单 $\text{Buy}(q)$，$Y(X, \text{Buy}(q))$ 表示用户用多少 2 号代币可以换取 q 单位的 1 号代币，定义为：

$$Y(X, \text{Buy}(q)) := \min\{y \geq 0 : \phi(X - q \cdot e_1 + y \cdot e_2) \geq \phi(X)\}$$

对于在状态 X 下执行的卖单 $\text{Sell}(q)$，$Y(X, \text{Sell}(q))$ 表示用户用 q 单位的 1 号代币可以换取多少 2 号代币，定义为：

$$Y(X, \text{Sell}(q)) := \max\{y \leq X_2 : \phi(X + q \cdot e_1 - y \cdot e_2) \geq \phi(X)\}$$

Potential Functions

为了判断一笔交易是否可行，除了池子的货币量足够以外，用户可能还有一个可以接受的汇率上限，例如 $buy(q,p)$ 就表示用户想要购买 q 单位的 1 号代币，并且愿意支付的 2 号代币数量不超过 $p \cdot q$.

形式化地，买单 $\text{Buy}(q,p)$ 能在 X 成功执行，当且仅当：

$$
\begin{aligned}
&Y(X, \text{Buy}(q)) \leq p \cdot q, \\
&\phi(X_1 - q, X_2 + Y(X, \text{Buy}(q))) = \phi(X), \\
&X_1 \geq q.
\end{aligned}
$$

卖单 $\text{Sell}(q,p)$ 能在 X 成功执行，当且仅当：

$$
\begin{aligned}
&Y(X, \text{Sell}(q)) \geq p \cdot q, \\
&\phi(X_1 + q, X_2 - Y(X, \text{Sell}(q))) = \phi(X), \\
&X_2 \geq Y(X, \text{Sell}(q)).
\end{aligned}
$$

Model discussion

在之前的讨论中，我们发现势能函数的值是不变的，我们宣称，当势能函数满足一定的性质时，交易过程中就可以满足一系列的经济学原理：

定义 (非零支付（non-zero payment）)

如果对于所有 $X \in \mathrm{dom}(\phi)$，都有 $Y(X, \mathrm{Sell}(0)) = 0$，则称势能函数 ϕ 具有非零支付性质.

定义 (开集与向上封闭集)

设 $B_\varepsilon(x) = \{y \in \mathbb{R}^n : \|x - y\| \leq \varepsilon\}$ 表示 x 点的 ε 邻域. 若 $D \subseteq \mathbb{R}^n$，对于任意 $x \in D$，存在 $\varepsilon > 0$，使得 $B_\varepsilon(x) \subseteq D$，则称 D 是开集. 若 $D \subseteq \mathbb{R}^n$，对于任意 $x \in D$，$y \geq x$ 时 $y \in D$，则称 D 是向上封闭集.

Model discussion

Lemma 1

设 $\mathrm{dom}(\phi)$ 是开集且向上封闭．若 ϕ 严格递增，则 ϕ 具有非零支付性质，反之亦然．

Lemma 2

设 X 和 X' 是两个状态，满足 $\phi(X) = \phi(X')$ 且 $X'_1 < X_1$，并假设势能函数 ϕ 是拟凹且严格递增，则有：

- 如果 $\mathrm{Buy}(q)$ 在 X 和 X' 都能成功执行，则 $Y(X, \mathrm{Buy}(q)) \leq Y(X', \mathrm{Buy}(q))$；
- 如果 $\mathrm{Sell}(q)$ 在 X 和 X' 都能成功执行，则 $Y(X, \mathrm{Sell}(q)) \leq Y(X', \mathrm{Sell}(q))$．

Communication Model

区块中的交易执行顺序被称为 execution ordering．在我们的模型中，矿工选择区块（即选择要包含的交易集合），但具体的执行顺序由一个排序规则（sequencing rule）S 决定．

定义 (Sequencing Rule)

排序规则 S 是一个从状态 X 和交易集合 B 映射到 B 的所有非空排列的函数，$S(X,B)$ 是 B 的所有排列的非空子集．

实际应用中，我们希望排序规则能够高效计算，以减轻矿工的计算负担．

定义 (Efficient Sequencer)

如果对于所有初始状态 $X=(X_1,X_2)$ 和区块 B，存在算法能在 $O(\log(X_1+X_2)|B|)$ 时间内，从 $S(X_0,B)$ 输出某个排列 T，则称排序规则 S 是（计算上）高效的．

Communication Model

假设矿工是诚实的，那么我们理想的交易模型如下：

Ideal Trading Game

Input: Initial state X_0; order A_i from each user i; sequencing rule S.
Output: Execution ordering $T = (T_1, \ldots, T_{|T|})$ and states $X_1, X_2, \ldots, X_{|T|}$ where T_t executes at X_{t-1}.
Proceed as follows:

1. The miner initializes the block $B = \emptyset$.
2. For all i, user i privately sends order A_i to the miner.
3. For all i, the miner adds A_i to B.
4. The miner picks some $(T_1, \ldots, T_{|B|}) \in S(X_0, B)$ as the execution ordering.
5. For $t = 1, \ldots |B|$,
   (a) The exchange executes T_t at state X_{t-1}.
   (b) Let X_t be the state after T_t executes on X_{t-1}.

Algorithm 2: Trading game with an honest miner.

Communication Model

在实际交易中，矿工可能会有如下偏离行为：

- 在打包区块时故意不包含某些交易（审查）.
- 在打包区块时插入自己的交易.
- 在执行阶段选择不属于 $S(X_0, B)$ 的执行顺序.

因为矿工可以谎称" 我没有接收到广播/听到了其他广播"，因此我们假设观察者只能看到链上状态，无法观测所有未成交订单，因此只能基于区块链信息检测矿工偏离.

定义 (Safe deviation)

若矿工的偏离策略导致的结果 T 满足存在一组 $A'_i \in \{A_i\}$，使得 $T \in S(X_0, \{A'_i\})$，则称该偏离是安全的.

Communication Model

对于给定的排序规则 S，我们定义其语言：

$$\mathcal{L}(X_0, S) = \{(X_0, T) : \{A_i\},\ T \in S(X_0, \{A_i\})\}$$

即所有初始状态为 X_0，排序器 S 可能输出的结果集合.

一个重要约束是，观察者必须能高效判断 (X_0, T) 是否属于 $\mathcal{L}(X_0, S)$.

定义 (Verifiable sequencing rule)

若存在多项式时间算法 P，对任意 (X_0, T)，判断其是否属于 $\mathcal{L}(X_0, S)$，则称排序规则 S 是可验证的.

User and miner utility

假设结果为 (X_0, T)，$T_{\sigma(i)}$ 表示用户 i 的交易在执行顺序 T 中的位置，若被审查则未定义．定义用户 i 的效用：

$$u_i(X_0, T) = \begin{cases} (0,0) & \text{if } \sigma(i) \text{ is undefined} \\ X_{\sigma(i)} - X_{\sigma(i)-1} & \text{otherwise} \end{cases}$$

设 I 为所有矿工交易执行的时刻，则矿工的效用为：

$$u_0(X_0, T) = \sum_{t \in I} (X_t - X_{t-1})$$

对于两个结果 (X_0, T) 和 (X_0, T')，若 $u_i(X_0, T) \geq u_i(X_0, T')$，称 (X_0, T) 支配 (X_0, T')．若 $u_i(X_0, T) \geq 0$，称为无风险执行（risk-free execution）；若 $u_i(X_0, T) > 0$，且另一种代币数量非负，则为盈利性执行（profitable execution）.

Market Manipulation

Theorem 1

考虑有一个用户买单 $\text{Buy}(q,p)$，即以上限价格 p 买入 q 单位 1 号代币，假设在状态 X 下可行，且交易所保持流动性守恒．则存在一种执行顺序，使得用户用 $q\cdot p$ 单位 2 号代币换取 q 单位 1 号代币，矿工可以无成本获得 $q\cdot p - Y(X,\text{Buy}(q))$ 单位 2 号代币．

证明其实很简单，Miner 只要先买入到价格 p，然后执行 $\text{Buy}(q)$，最后卖出即可．

Market Manipulation

Theorem 2

即使只允许矿工选择区块内容（不能选执行顺序），对于任意排序规则 S，都存在初始状态 X_0 和至少包含三笔用户买卖交易的区块 B，使得矿工仍然可以通过某种执行顺序 $T \in S(X_0, B)$ 获得无风险套利.

Sequencing rule 只能限制买卖的交易顺序，但是不能限制买卖的主体，所以矿工可以针对一个执行顺序 T，把里面的一些买卖人踢掉，变成自己，然后进行 Sandwich attack.

Market Manipulation

假如初始的 B 包含三笔 Buy(2) 和三笔 Sell(1)，则无论 S 如何排序，矿工都可以在 T 中替换自己的交易，实现获利.

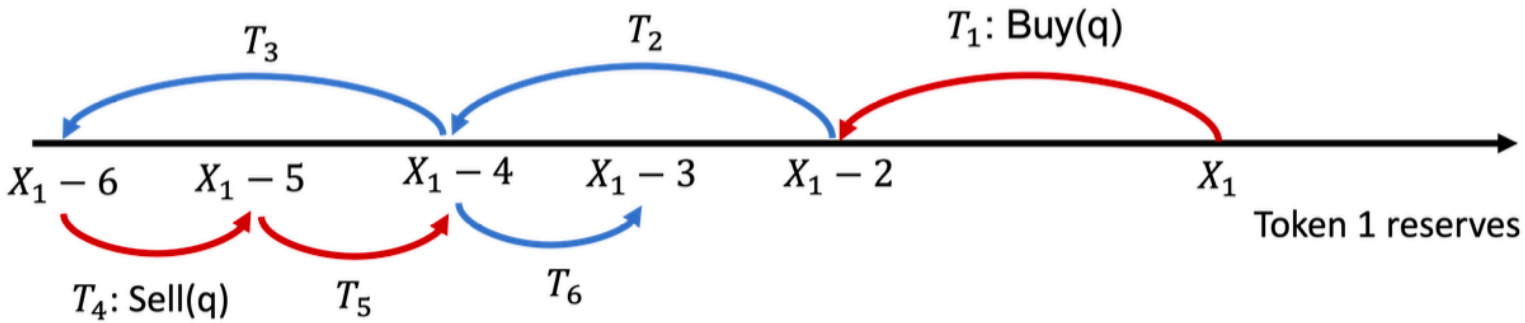


Figure 2: Example of a permutation where the miner obtain risk-free profits. Arrows pointing to the left are buy orders and arrows pointing to the right are sell orders. Miner orders are in red and user orders are in blue.

Greedy Sequencing Rule

Greedy Sequencing Rule

Input: Initial state X_0; set of transactions B.
Output: Execution ordering T.
Proceed as follows:

1. Initialize T as an empty list.
2. Let $B^{\mathbf{buy}} \subseteq B$ be the collection of buy orders in B. Let $B^{\mathbf{sell}} \subseteq B$ be the collection of sell orders in B.
3. While $B^{\mathbf{buy}}$ and $B^{\mathbf{sell}}$ are both non-empty:
   (a) Let $t = |T|$.
   (b) If $X_{t,1} \geq X_{0,1}$:
      i. Let A be any order in $B^{\mathbf{buy}}$.
      ii. Append A to T and remove A from $B^{\mathbf{buy}}$.
   (c) Else:
      i. Let A be any order in $B^{\mathbf{sell}}$.
      ii. Append A to T and remove A from $B^{\mathbf{sell}}$.
   (d) Let X_{t+1} be the state after A executes on X_t.
4. If $B^{\mathbf{buy}} \cup B^{\mathbf{sell}}$ is non-empty, append the remaining transactions in $B^{\mathbf{buy}} \cup B^{\mathbf{sell}}$ to T in any order.

Greedy Sequencing Rule

Theorem 3

在上述的 Greedy Sequencing Rule 下，矿工针对 X 和 B 选择一个 T，一定满足：

1. 可以在多项式时间内判断出矿工不遵循约定，即 $T \notin \mathcal{L}(X, S)$.
2. 如果矿工遵循约定，一定满足用户可以用比不劣于当前汇率的汇率执行交易，或者：
3. 对于劣于当前汇率的交易，矿工不会因为他们的加入而多获得收益.

Greedy Sequencing Rule

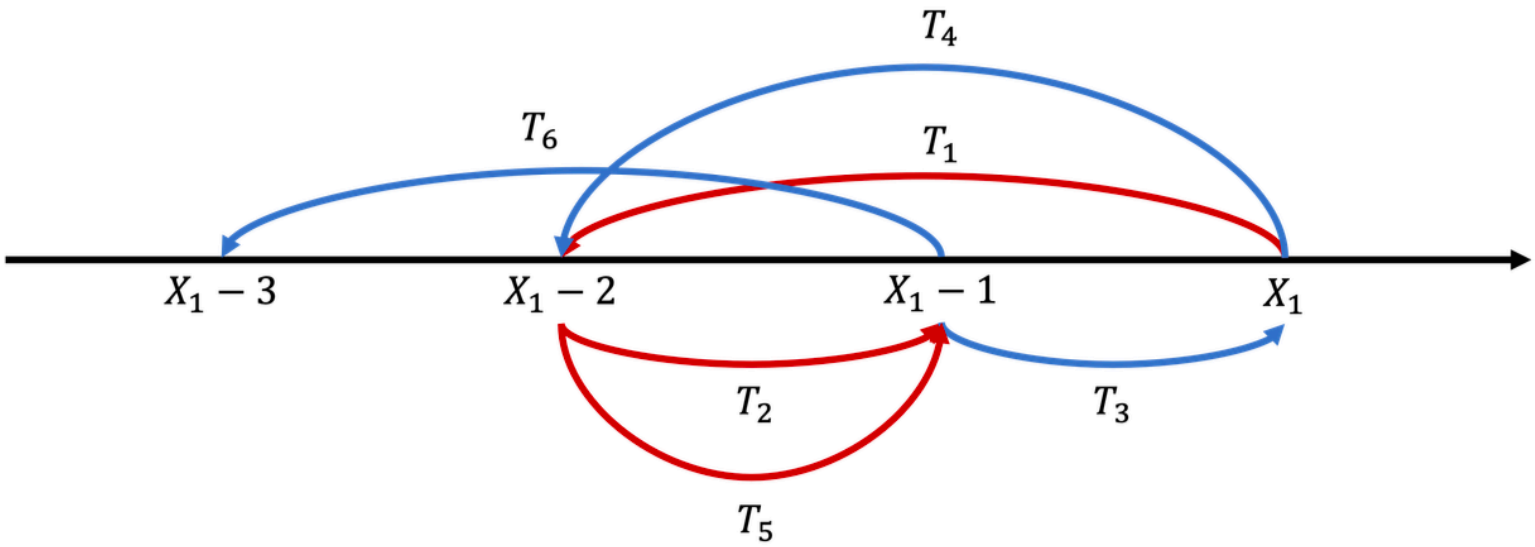


Figure 4: A valid outcome when trading with the Greedy Sequencing Rule for the block that contains three identical buy orders, BUY(2), and three identical sell orders, SELL(2). The miner owns the orders in red. The initial state is (X_1, X_2). Left arrows denote buy orders and right arrows denote sell orders.

Greedy Sequencing Rule

Verifier for the Greedy Sequencing Rule

Input: Outcome (X_0, T).
Output: $True \mid False$.
Proceed as follows:

1. For $t = 1, 2, \ldots, |T|$:
   (a) If $T_t, T_{t+1}, \ldots, T_{|T|}$, are orders of the same type (i.e., all are buy orders or all are sell orders), then output $True$.
   (b) If $X_{t-1,1} \geq X_{0,1}$ and T_t is a buy order, then output $False$.
   (c) If $X_{t-1,1} < X_{0,1}$ and T_t is a sell order, then output $False$.
   (d) Let X_t be the state after T_t executes on X_{t-1}.
2. Output $True$.

Algorithm 4: The Verifier for the Greedy Sequencing Rule.

Reference

Credible Decentralized Exchange Design via Verifiable Sequencing Rules, Matheus V. X. Ferreira, David C. Parkes, 2023.