Verified Double Sided Auctions for Financial Markets

Raja Natarajan
Tata Institute of Fundamental Research, Mumbai, India

Suneel Sarswat
Tata Institute of Fundamental Research, Mumbai, India

Abhishek Kr Singh
Birla Institute of Technology and Science Pilani, Goa, India

Abstract

Double sided auctions are widely used in financial markets to match demand and supply. Prior works on double sided auctions have focused primarily on single quantity trade requests. We extend various notions of double sided auctions to incorporate multiple quantity trade requests and provide fully formalized matching algorithms for double sided auctions with their correctness proofs. We establish new uniqueness theorems that enable automatic detection of violations in an exchange program by comparing its output with that of a verified program. All proofs are formalized in the Coq proof assistant without adding any axiom to the system. We extract verified OCaml and Haskell programs that can be used by the exchanges and the regulators of the financial markets. We demonstrate the practical applicability of our work by running the verified program on real market data from an exchange to automatically check for violations in the exchange algorithm.

2012 ACM Subject Classification Information systems → Online auctions; Software and its engineering → Formal software verification; Theory of computation → Algorithmic mechanism design; Theory of computation → Computational pricing and auctions; Theory of computation → Program verification; Theory of computation → Automated reasoning

Keywords and phrases Double Sided Auction, Formal Verification, Financial Markets, Proof Assistant

Digital Object Identifier 10.4230/LIPIcs.ITP.2021.28

Related Version *Full Version*: `https://arxiv.org/abs/2104.08437` [5]

Supplementary Material *Software*: `https://github.com/suneel-sarswat/dsam`
archived at `swh:1:dir:94b9da19e3cfdb243bd47f1619a2901758fb1243`

Acknowledgements We wish to thank Mohit Garg for his generous contribution to this work.

1 Introduction

Computer algorithms are routinely deployed nowadays by all big stock exchanges to match buy and sell requests. These algorithms are required to abide by various regulatory guidelines. For example, market regulators make it mandatory for trades resulting from double sided auctions at exchanges to be fair, uniform and individual-rational.

In this paper, we introduce a formal framework for analyzing trades resulting from double sided auctions used in the financial markets. To verify the essential properties required by market regulators, we formally define these notions in a theorem prover and then develop important results about matching demand and supply. Finally, we use this framework to verify properties of two important classes of double sided auction mechanisms.

One of the resulting advantages of our work for an exchange or a regulator is that they can check the algorithms deployed for any violations from required properties automatically. This is enabled by the new uniqueness results that we establish in this work. All the definitions and results presented in this paper are completely formalized in the Coq proof assistant

without adding any additional axioms to it. The complete formalization in Coq facilitates automatic program extraction in OCaml and Haskell, with the guarantee that extracted programs satisfy the requirements specified by the market regulator. Consequently, the extracted program could also be deployed directly at an exchange, apart from checking for violations in existing programs. We demonstrate the practical applicability of our work by running the verified program on real market data from an exchange to automatically check for violations in the exchange algorithm.

The rest of this paper is organized as follows: Section 2 provides a brief background and overview of trading at exchanges which is needed to describe our contributions; In Section 3, we briefly state our contributions; Section 4 provides basic definitions and establishes certain combinatorial results; Section 5 describes a fairness procedure; Section 6 describes the uniform matching mechanism used in the financial markets; Section 7 describes the maximum matching mechanism; Section 8 establishes uniqueness results that enables automatic checking for violations in an exchange matching algorithm; Section 9 describes the practical utility of our work through running our verified program on real market data from an exchange; Section 10 concludes the paper with related work and future directions. Parts of some sections which could not be fully accommodated due to space constraints have been moved to the appendix of the full version of the paper [5].

2 Background

Financial trades occur at various types of exchanges. For example, there are exchanges for stocks, commodities and currencies. At any exchange, multiple buyers and sellers participate to trade certain products. Mostly exchanges employ double sided auction mechanisms to match the buyers and sellers. Some exchanges, apart from using double sided auctions, also use an online continuous algorithm for executing trades during certain time intervals, especially for highly traded products.

For conducting trades of a certain product using a double sided auction mechanism, the exchange collects buy and sell requests from the traders for a fixed time period. At the end of this time period, the exchange matches some of the trade requests and outputs trades, all at a single price. This price is sometimes referred to as the equilibrium price and the process as price discovery. A buyer places a buy request, also known as a *bid*, which consists of a quantity indicating the maximum number of units he is interested in buying and a common maximum price (bid's limit price) for each of the units. Similarly, a seller's sell request, an *ask*, consists of a quantity and a minimum price (ask's limit price). Each trade (*transaction*) consists of a bid, an ask, traded quantity, and a trade price. Naturally, the traded quantity should be at most the minimum of the bid and the ask quantities and the trade price should be compatible with the bid and the ask.

Apart from the single price property and compatibility constraint mentioned above, there are other desired properties that the trades (*matching*) should have. The properties that capture these constraints are: *uniform*, *individual-rational*, *fair* and *maximum*. We briefly describe these matching properties:

- **Uniform**: A matching is uniform if all the trades happen at the same price.
- **Individual-rational**: A matching is individual-rational if for each matched bid-ask pair the trade price is between the bid and ask limit prices. In the context of financial markets, the trade price should always be between the limit prices of the matched bid-ask pair.
- **Fair**: A bid b_1 is more competitive than a bid b_2 if b_1 has a higher limit price than b_2 or if their limit prices are the same and b_1 arrives earlier than b_2. Similarly, we can define competitiveness between two asks. A matching is *unfair* if a less competitive bid gets

matched but a more competitive bid is not fully matched. Similarly, it could be unfair if a more competitive ask is not fully matched. If a matching is not unfair, then it is fair.

- **Maximum**: A matching is maximum if it has the maximum possible total traded quantity among all possible matchings.
- **Optimal individual-rational-uniform**: An individual-rational and uniform matching is called optimal individual-rational-uniform if it has the largest total trade volume among all matchings that are individual-rational and uniform.

No single algorithm can possess all the above first four properties simultaneously [12, 4]. In the context of financial markets, regulators insist on the matching being fair and optimal individual-rational-uniform, thus compromising on the maximum property. In other contexts where the matching being maximum is important along with individual rational and fair, uniformity is lost. This gives rise to two different classes of double sided auction mechanisms, each with a different objective. In our work, we consider both these classes of mechanisms.

3 Our Contributions

In this work, we formalize the notion of double sided auctions where trade requests can be of multiple quantities. Prior to our work, similar notions were explicitly defined only for single unit trade requests [8, 6, 13]. In going from formalizing the theory for single unit to the theory of multiple units, the mechanisms and their correctness proofs changed substantially. Due to the possibility of partial trades, the formal analysis of multiple unit trades becomes significantly more involved than in [8]. In this work, we show how to efficiently handle this extra complexity by making the functions and their properties sensitive to the partial trade quantities. This helps us to develop formal proofs of correctness of the recursive mechanisms for double sided auctions.

In addition, we provide new uniqueness results that guarantee that the matching algorithm for the double sided auctions used in the financial markets outputs a unique volume of trades per order if the algorithm is fair and optimal individual-rational-uniform; thus enabling automatic checking of violations in the exchange algorithm by comparing its output with that of a verified program. We demonstrate this by running the extracted OCaml code of our certified mechanism on real data from an exchange and comparing the outputs. Following is a brief description of the key results formalized in this work.

- **Combinatorial result**: We show that the modeling and the libraries we created to obtain our results are also useful in proving other important results on double sided auctions. For example, in Theorem 7, we show that for any p, no matching can achieve a trade volume higher than the sum of the total demand and the total supply in the market at price p.
- **Fairness**: We show that any matching can be converted into a fair matching without compromising on the total traded volume. For this, we design an algorithm, the Fair procedure, which takes a matching M as input, and outputs a matching M'. In Theorem 15, we show that the total traded quantities of M and M' are the same and M' is a fair matching.
- **Uniform mechanism**: We design an algorithm, the UM procedure, that takes as input the bids and the asks and outputs a fair, individual-rational and uniform matching. Furthermore, in Theorem 20, we show that the output matching has the largest total trade volume among all the matchings that are uniform and individual-rational and thus is optimal individual-rational-uniform. This algorithm is used in the exchanges that output trades using double sided auctions.

- **Maximum mechanism**: We design an algorithm, the MM procedure, that takes as input the bids and the asks and outputs an individual-rational, fair and maximum matching. In Theorem 21, we show that the output matching has the largest total trade volume among all the matchings that are individual-rational.
- **Uniqueness theorems**: For any two fair and optimal individual-rational-uniform matchings, Theorem 23 implies that their total trade volume for each order is the same. Thus, if we compare the trade volumes between an exchange's matching output with our verified program's output and for some order they do not match, then the exchange's matching is not fair and optimal individual-rational-uniform. On the contrary, if for each order, the trade volumes match, then Theorem 24 implies that the exchange's matching is also fair and optimal individual-rational-uniform (given that it already is individual-rational and uniform, which can be easily verified by checking the trade prices). Making use of these results, in Section 9, we check violations automatically in real data from an exchange.

The Coq code together with the extracted OCaml and Haskell programs for all the above results is available at [10]. Our Coq formalization consists of approximately 50 new definitions, 750 lemmas and theorems and 12000 lines of code. In the following sections, we provide definitions, procedures and proof sketches that closely follow our actual formalization.

4 Modeling Double Sided Auctions

In a double sided auction multiple buyers and sellers place their orders to buy or sell an underlying product. The auctioneer matches these buy-sell requests based on their *limit prices*, *arrival time*, and the maximum specified *trade quantities*. Note that the limit prices are natural numbers when expressed in the monetary unit of the lowest denomination (like cents in USA). In our presentation, we will be working with lists (of bids, asks and transactions); For ease of readability, we will often use set-theoretic notations like $\in$, $\subseteq$, $\supseteq$, $\emptyset$ on lists whose meanings are easy to guess from the context.

▶ **Definition 1** (Bid). *A bid $b = (id_b, \tau_b, q_b, p_b)$ represents a buy request having four components. Here, the first two components id_b and τ_b are the unique identifier and the timestamp assigned to the buy request b, respectively, whereas the third component q_b represents the quota of b, the maximum quantity of the item the buyer is willing to buy. The last component p_b is the limit price of the buy request, which is the price above which the buyer does not want to buy the item.*

▶ **Definition 2** (Ask). *An ask $a = (id_a, \tau_a, q_a, p_a)$ represents a sell request having four components. Here, the first two components id_a and τ_a are the unique identifier and the timestamp assigned to the sell request a, respectively, whereas the third component q_a represents the quota of a, the maximum quantity of the item the seller is willing to sell. The last component p_a is the limit price of the sell request, which is the price below which the seller does not want to sell the item.*

We say that a bid $b \in B$ is matchable with an ask $a \in A$ if $p_a \leq p_b$.

In a double sided auction, the auctioneer is presented with duplicate-free[1] lists of buy and sell requests (lists B and A, respectively). The auctioneer can match a bid $b \in B$ with an ask

[1] A list of bids or asks is duplicate-free if all the participating orders have distinct ids.

$a \in A$ only if $p_b \geq p_a$. Furthermore, the auctioneer assigns a trade price and a trade quantity to each matched bid-ask pair, which finally results in a *transaction* m. Therefore, we can represent a matching of demand and supply by using a list whose entries are *transactions*.

▶ **Definition 3** (Transaction). *A transaction $m = (b_m, a_m, q_m, p_m)$ describes a trade between the bid b_m and the ask a_m. The next two components q_m and p_m are the traded quantity and the trade price, respectively. For ease of readability, we use the terms $p(b_m)$, $p(a_m)$, $q(b_m)$, and $q(a_m)$ for p_{b_m}, p_{a_m}, q_{b_m} and q_{a_m}, respectively.*

▶ **Definition 4** (Matching M B A). *A list of transactions M is a matching between the duplicate-free lists of bids B and asks A if*

1. *For each transaction $m \in M$, the bid of m is matchable with the ask of m (i.e., $p(a_m) \leq p(b_m)$).*
2. *The list of bids present in M, denoted by B_M, is a subset of B (i.e., $B_M \subseteq B$).*
3. *The list of asks present in M, denoted by A_M, is a subset of A (i.e., $A_M \subseteq A$).*
4. *For each bid $b \in B$, the total traded volume of bid b in the matching M, denoted by $Q(b, M)$, is not more than its maximum quantity (i.e., for all $b \in B, Q(b, M) \leq q_b$).*
5. *For each ask $a \in A$, the total traded volume of ask a in the matching M, denoted by $Q(a, M)$, is not more than its maximum quantity (i.e. for all $a \in A, Q(a, M) \leq q_a$).*

Description. Note that there might be some bids in B which are not matched to any asks in M and some asks in A which are not matched to any bids in M.

▶ Note 5. For simplicity, with slight abuse of notation, we use Q to denote total quantity of various objects which will be clear from the context. So, $Q(b, M)$ and $Q(a, M)$ represent the total quantities of the bid b and the ask a traded in the matching M, respectively. Similarly, the terms $Q(B)$ and $Q(A)$ denote the sum of the quantities of all the bids in B and the sum of the quantities of all the asks in A, respectively. And also, for the total traded quantity in a matching M, we use the term $Q(M)$. However, in the Coq implementation, each of these terms are represented by different names: QMb, QMa, QB, QA and QM.

Formalization notes: We have defined Bid, Ask and Transaction as record types in Coq. We define the proposition *matching_in B A M* to be true if and only if M is a matching between the list of bids B and the list of asks A.

4.1 Matching Demand and Supply

Let $B_{\geq p}$ represents the list of bids in B whose limit prices are at least a given number p. Similarly, $A_{\leq p}$ represents the list of asks in A whose limit prices are at most p. Therefore, the quantities $Q(B_{\geq p})$ and $Q(A_{\leq p})$ represents the total demand and the total supply of the product at the price p in the market, respectively. Although, in general we cannot say much about the relationship between the total demand (i.e. $Q(B_{\geq p})$) and supply (i.e. $Q(A_{\leq p})$) at an arbitrary price p, we can prove the following important results about the traded quantities of the matched bid-ask pairs.

▶ **Lemma 6.** *If M is a matching between the list of bids B and the list of asks A, then*

$$Q(M) = \sum_{b \in B} Q(b, M) \leq \sum_{b \in B} q_b = Q(B) \text{ and } Q(M) = \sum_{a \in A} Q(a, M) \leq \sum_{a \in A} q_a = Q(A)$$

▶ **Theorem 7.** *If M is a matching between the list of bids B and the list of asks A, then for all natural numbers p, we have $Q(M) \leq Q(B_{\geq p}) + Q(A_{\leq p})$*

Theorem 7 states that no matching M can achieve a trade volume higher than the sum of the total demand and supply in the market at any given price.

Proof Idea. We first partition the matching M into two lists: $M_1 = \{m \in M \mid p(b_m) \geq p\}$ and $M_2 = \{m \in M \mid p(b_m) < p\}$. Thus, $Q(M) = Q(M_1) + Q(M_2)$.

It is easy to see that M_1 is a matching between $B_{\geq p}$ and A, and hence from Lemma 6, $Q(M_1) \leq Q(B_{\geq p})$.

Next, we prove that M_2 is a matching between B and $A_{\leq p}$. Consider a transaction m from M_2. Since $m \in M$, $p(b_m) \geq p(a_m)$, and from the definition of M_2, we have $p(b_m) < p$. This implies $p(a_m) < p$, i.e., asks of M_2 come from $A_{\leq p}$. Hence, M_2 is a matching between B and $A_{\leq p}$, and applying Lemma 6, we have $Q(M_2) \leq Q(A_{\leq p})$.

Combining, we have $Q(M) = Q(M_1) + Q(M_2) \leq Q(B_{\geq p}) + Q(A_{\leq p})$, which completes the proof of Theorem 7. □

Formalization notes: The formal proof of Theorem 7 is completed by first proving the Lemmas *Mbgep_bound* ($Q(M_1) \leq Q(B_{\geq p})$) and *Mbltp_bound* ($Q(M_2) \leq Q(A_{\leq p})$) and then combining them in theorem *bound_on_M*. These results can be found in the file "Bound.v".

4.2 Individual-Rational Trades

An auctioneer assigns a trade price to each matched bid-ask pair. In any matching it is desired that the trade price of a bid-ask pair lies between their limit prices. A matching which has this property is called an *individual-rational (IR)* matching.

▶ **Definition 8** (Individual rational). Is_IR(M) := *for all* $m \in M$, $p(b_m) \geq p_m \geq p(a_m)$.

Note that any matching can be converted to individual-rational by changing the price of each transaction to lie between the limit prices of its bid and ask (See Fig 1).

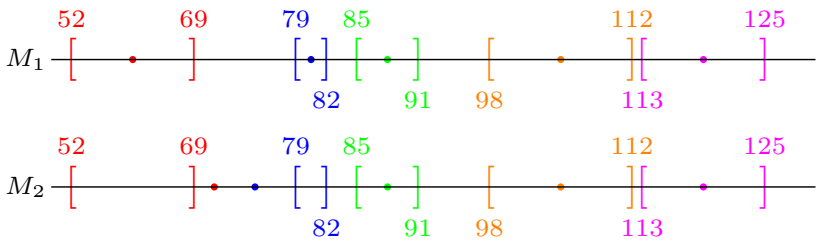

Figure 1 The colored dots represent trade prices for matched bid-ask pairs. Matching M_2 is not IR but M_1 is IR, even though both the matchings contain exactly the same bid-ask pairs.

5 Fairness in Competitive Markets

A double sided auction is a competitive event, where the priority among participating traders is determined by various attributes of the orders. A bid with higher limit price is considered more *competitive* compared to bids with lower limit prices. Similarly, an ask with lower limit price is considered more competitive compared to asks with higher limit prices. Ties are broken in favor of the requests that have an earlier arrival time. A matching which prioritizes more competitive traders is called a *fair* matching.

▶ **Definition 9** (Arrow notation). $\overset{\mathbb{B}}{\uparrow} L$ *denotes that the list* L *is sorted as per the competitiveness of the bids in* L*, with the most competitive bid being on top. Similarly,* $\overset{\mathbb{A}}{\uparrow} L$ *denotes that the*

list L is sorted as per the competitiveness of the asks in L, with the most competitive ask being on top. Similarly we can define $\overset{\mathbb{A}}{\downarrow}$ and $\overset{\mathbb{B}}{\downarrow}$ for sorting lists where the most competitive orders lie at the bottom.

In this section, we show that there exists a procedure Fair that takes a matching M between bids B and asks A as input and outputs a fair matching $M' = \mathsf{Fair}(M, B, A)$ with the same trade volume as that of M. To describe the Fair procedure, we will need the following definitions.

▶ **Definition 10.** *Let M be a matching between bids B and asks A.*

- *M is fair on bids if for all pairs of bids $b_1, b_2 \in B$ such that b_1 is more competitive than b_2 and b_2 participates in the matching M, then b_1 is fully traded in M (i.e., $Q(b_1, M) = q(b_1)$).*
- *Similarly, M is fair on asks if for all pairs of asks $a_1, a_2 \in A$ such that a_1 is more competitive than a_2 and a_2 participates in the matching M, then a_1 is fully traded in M (i.e., $Q(a_1, M) = q(a_1)$).*
- *M is fair if it is both fair on bids and asks.*

The Fair procedure works in two steps: first, it sorts the matching M and the asks A based on the competitiveness of the asks and then runs on them a procedure "fair on asks" FOA that outputs a matching M' that is of the same volume as that of M and is fair on the asks. In the second step, it sorts the resulting matching M' and the bids B based on the competitiveness of the bids and then runs on them a procedure fair on bids FOB that outputs a matching M'' that is of the same volume as that of M' and is fair on the bids. The Fair procedure returns M'' as its output. The procedures FOB and FOA, along with their correctness proofs, mirror each other and we just describe FOB below and show that $\mathsf{FOB}(\overset{\mathbb{B}}{\uparrow} M', \overset{\mathbb{B}}{\uparrow} B)$ outputs a fair on bids matching and has the same trade volume as that of M'. Furthermore, we will show that if M' is fair on asks, then $\mathsf{FOB}(\overset{\mathbb{B}}{\uparrow} M', \overset{\mathbb{B}}{\uparrow} B)$ is fair on asks. This will immediately imply that the procedure $\mathsf{Fair}(M, B, A)$ outputs a fair matching with the same total trade volume as that of M.

5.1 Fair on Bids

To describe the fair on bids FOB procedure, we first need the following notation.

▶ **Definition 11.** *Given a list L and and an element a, $a :: L$ denotes the list whose top element (head) is a and the following elements (tail) are the elements of L (in the same order as they appear in L).*

The FOB procedure takes sorted (based on the bids' competitiveness) lists of transactions M and bids B. Intuitively, when all the bids are of unit quantity, we want to scan the list of transactions in M from top to bottom replacing the bids therein with the bids of B from top to bottom. So, in effect, in the FOB procedure we will implement this intuition apart from taking care of multiple quantity bids; and also make the procedure recursive so that we can provide a formalization friendly inductive proof of correctness. Let $B = b :: B'$ and $M = m :: M'$. In our procedure, we first pick the top bid b of B and the top transaction m of M, and compare q_b with q_m. Now we have three cases. In each of the three cases, the procedure first outputs a transaction between the bid b and the ask of m of quantity $\min\{q_m, q_b\}$. Case I: If $q_b = q_m$, we remove b and m from their respective lists and recursively solve the problem on B' and M'. Case II: If $q_b < q_m$, we remove b from the list B and update

q_m to $q_m - q_b$ and recursively solve the problem on B' and M. Case III: If $q_b > q_m$, we remove m from the list M and set a parameter t to q_m that we will send to the recursive call along with M' and B. The parameter t informs our recursive procedure that the top element b of B has effectively quantity $q_b - t$. Thus, our procedure will take three parameters: the list of transactions, the list of bids and the parameter t (Note that unlike Case II ($q_b < q_m$) where the top transaction is updated, the top bid is not updated in Case III ($q_b > q_m$). This is done for technical reasons: Later we need to prove that the set of bids of FOB is a subset of B, and at the same time we have to ensure that the total traded quantity of the bid b in the matching outputted by FOB remains below its maximum quantity q_b, as required by the matching property. This would not be possible to do if we updated B and hence we take this approach of keeping the total traded quantity of the top bid b in a separate argument: t of f.). Keeping this description in mind, we now formally define the procedure FOB.

▶ **Definition 12** (Fair On Bid (FOB))**.**

$$\mathsf{FOB}(M, B) = f(M, B, 0)$$

where $f(M, B, t) =$

$$\begin{cases} nil & \text{if } M = nil \text{ or } B = nil \\ (b, a_m, q_m, p_m) :: f(M', B', 0) & \text{if } q_m = q_b - t \\ (b, a_m, q_m, p_m) :: f(M', b :: B', t + q_m) & \text{if } q_m < q_b - t \\ (b, a_m, q_b - t, p_m) :: f((b_m, a_m, q_m - (q_b - t), p_m) :: M', B', 0) & \text{if } q_m > q_b - t \end{cases}$$

where $M = m :: M'$ *when* $M \neq nil$ *and* $B = b :: B'$ *when* $B \neq nil$*.*

▶ **Theorem 13.** *Let M be a matching between bids B and asks A where the lists M and B are sorted in the descending order of the competitiveness of their bids (i.e., the most competitive bid and the transaction with the most competitive bid are on top of their respective lists). Let $M_\beta = \mathsf{FOB}(M, B)$, then*

- **(a)** *M_β is a matching between bids B and asks A.*
- **(b)** *For each ask $a \in A$, the total traded quantity of a in M is same as the total traded quantity of a in M_β (i.e., $Q(a, M) = Q(a, M_\beta)$). As a corollary, we get that if M is fair on asks, then M_β is also fair on asks.*
- **(c)** *The total traded quantity of M is equal to the total traded quantity of M_β (i.e., $Q(M) = Q(M_\beta)$).*
- **(d)** *The matching M_β is fair on bids.*

Proof Outline. Here, we briefly describe certain aspects of the proof; more details can be found in the full version of the paper [5] and for the complete formalization see [10]. Note that in each of the recursive calls in f, either the size of the first argument $|M|$ decreases or the size of the second argument $|B|$ decreases. Therefore, we prove the above statements using (well founded) induction on the sum $(|M| + |B|)$. Proof of (a) and (b) is done using induction and case analysis. The proof of (c) follows by combining Lemma 6 with (b). We focus on the proof of (d) below.

Let bid b be the top element of the bids B and $B = b :: B'$. First, we prove two general results:

$$\text{For all } t, \text{ if } Q(M) \geq q_b - t, \text{ then } Q(f(M, b :: B', t), b) = q_b - t,$$

which states that if the total trade volume of the matching M is at least $q_b - t$, then in the matching $f(M, b :: B', t)$ the top bid b has trade quantity $q_b - t$. The proof of this can be

done using induction on the size of M. Intuitively, f tries to match as much quantity of the top bid b with the top transaction in M. When the call $f(M, b :: B', t)$ is made, the top bid b already has t traded quantity and $q_b - t$ of its quantity remains untraded. If the quantity of the top transaction m of M is at least $q_b - t$, then we are done. Otherwise, f matches q_m quantities of b and recursively calls f on a smaller list and then we will be done by applying the induction hypothesis.

Now, we state the second general result.

For all t, if distinct bids b, b' belong to the bids of $f(M, b :: B', t)$, then $Q(M) \geq q_b - t$.

This result can be proved, like the previous result, using induction on the sum $(|M| + |B|)$; see [10] for details. Intuitively, since b' is matched by $f(M, b :: B', t)$ (in particular $b' \in B'$), then $f(M, b :: B', t)$ will completely match b (which has at least $q_b - t$ quantity remaining untraded) before it matches even a single quantity of b'.

Now using the above general results, we prove (d). We need to show the following: for all $b_1, b_2 \in B$, if b_1 is more competitive than b_2 and $Q(\mathsf{FOB}(M, B), b_2) \geq 1$, then $Q(\mathsf{FOB}(M, B), b_1) = q_{b_1}$, i.e., if the bid b_2 participates in the matching M then the bid b_1 is fully traded in M. Fix $b_1, b_2 \in B$ such that b_1 is more competitive than b_2 and $Q(\mathsf{FOB}(M, B), b_2) \geq 1$. Note that the bid b_2 cannot be equal to the bid b since bids B are sorted. Now we analyze three possible cases: $b_1 \neq b$, $b_1 = b$ and $Q(M) \geq q_b$, and $b_1 = b$ and $Q(M) < q_b$.

- In the case when $b_1 \neq b$, we consider the recursive call where b_1 is the top bid in the argument for the first time. In this recursive call the list of bids is smaller than B since the bid b must be fully traded before. Then, we are immediately done by applying the induction hypothesis.
- In the case $b_1 = b$ and $Q(M) \geq q_b$, in the matching $\mathsf{FOB}(M, b :: B') = f(M, b :: B', 0)$ the top bid b has total trade volume $q_b - 0 = q_b$ from the first general result invoked with $t = 0$, and hence $b_1 = b$ is fully traded.
- In the case $b_1 = b$ and $Q(M) < q_b$, we arrive at the contradiction $Q(M) \geq q_b$ by invoking the second general result with $t = 0$, $b = b_1$, $b' = b_2$ and $\mathsf{FOB}(M, b :: B') = f(M, b :: B', 0)$.

□

Similar to the procedure FOB, we have a procedure FOA, that produces a fair matching on asks (see [10]). Combining the FOA and FOB procedures, we have the following definition of the Fair procedure.

▶ **Definition 14.** $\mathsf{Fair}(M, B, A) = \mathsf{FOB}(\overset{\mathbb{B}}{\uparrow} \mathsf{FOA}(\overset{\mathbb{A}}{\uparrow} M, \overset{\mathbb{A}}{\uparrow} A), \overset{\mathbb{B}}{\uparrow} B)$.

We conclude this section by formally summarizing the main fairness result.

▶ **Theorem 15.** *If M is a matching on the list of bids B and the list of asks A, then the matching $M' = \mathsf{Fair}(M, B, A)$ on B and A is a fair matching such that $Q(M) = Q(M')$.*

Formalization notes: The procedure FOB and FOA are implemented in Coq using the Equations plugin which is helpful to write functions involving well-founded recursion [9]. The proof of Theorem 15 is quite extensive and done in several parts. First we prove all the parts of Theorem 13 in the file "mFair_Bid.v". We prove similar theorems for the procedure FOA in "mFair_Ask.v" file. Later all the results are combined in the file "MQFair.v" and the above theorem is proved as *exists_fair_matching*.

6 Uniform Price Matchings in Financial Markets

Liquidity in a market is a measure of how quickly one can trade in that market and maximizing the total trade volume helps increase liquidity. However, to maximize the total trade volume sometimes we have to accept different trade prices to the matched bid-ask pairs (Fig 2).

(a) UM (b) MM

Figure 2 Both the bids and the asks have quantity one. The only individually rational matching of size two is not uniform.

Assigning different trade prices for the same product in the same market simultaneously, might lead to dissatisfaction among some traders. As stated in the introduction, in the financial markets, the matching should be fair and optimal individual-rational-uniform. In this section, we describe the UM process that takes as input a list of bids and a list of asks and produces a fair and optimal individual-rational-uniform matching that can be directly applied in the financial markets for conducting double sided auctions. We present a novel proof of optimality of the UM process.

Before we describe the UM process, we first give some intuition. Observe that in any individual-rational and uniform matching M all the buyers are matched at a single price p and the price p lies between the limit prices of all the matched bid-ask pairs. This means all the matched bids' limit prices are at least p and all the matched asks' limit prices are at most p. In the special case when all the orders are of unit quantity, the matching can be visualized as a fully nested balanced parenthesis (for example, [[[[ ]]]]) where each bid is represented by a closed parenthesis "]" and each ask as an open parenthesis "[" (See Figure 1).

Now, we describe the UM process. We recursively pair the most competitive available bid with the most competitive available ask, if they are matchable. The trade quantity for each matched bid-ask pair is the minimum of the remaining quantities of the respective bid and the ask. The trade price assigned to each pair is the price of the ask in that pair[2]. We terminate the process once there are no more matchable bid-ask pairs remaining. At the end of the process, to produce a uniform matching we have to assign a single trade price to all the matched bid-ask pairs which we choose to be the trade price of the last matched bid-ask pair (which also keeps the individual-rational property intact).

Keeping this description in mind, we now formally define the UM process using recursion.

▶ **Definition 16** (Uniform Matching (UM)). .

$$\mathsf{UM}(B, A) = \mathsf{Replace\_prices}(f_u(\overset{\mathbb{B}}{\uparrow} B, \overset{\mathbb{A}}{\uparrow} A, 0, 0), \mathsf{Last\_trade\_price}(f_u(\overset{\mathbb{B}}{\uparrow} B, \overset{\mathbb{A}}{\uparrow} A, 0, 0)))$$

where $f_u(B, A, t_b, t_a) =$

$$\begin{cases} nil & \text{if } B = nil \text{ or } A = nil \text{ or } p_b < p_a \\ (b, a, q_b - t_b, p_a) :: f_u(B', A', 0, 0) & \text{if } q_a - t_a = q_b - t_b \text{ and } p_b \geq p_a \\ (b, a, q_b - t_b, p_a) :: f_u(B', a :: A', 0, t_a + q_b - t_b) & \text{if } q_a - t_a > q_b - t_b \text{ and } p_b \geq p_a \\ (b, a, q_a - t_a, p_a) :: f_u(b :: B', A', t_b + q_a - t_a, 0) & \text{if } q_a - t_a < q_b - t_b \text{ and } p_b \geq p_a \end{cases}$$

where $B = b :: B'$ *when* $B \neq nil$ *and* $A = a :: A'$ *when* $A \neq nil$.

[2] Observe that any value in the interval of the limit prices of the matched bid-ask pair can be assigned as the trade price and it will not affect any analysis done in this work.

Description. Observe that, similar to the parameter t in the FOB process, we have two parameters t_b and t_a that inform the recursive procedure f_u that the top bid b and the top ask a have effective quantities $q_b - t_b$ and $q_a - t_a$, respectively. In each recursive call, the process f_u outputs a transaction (top bid b, top ask a, quantity $\min\{q_b - t_b, q_a - t_a\}$, price p_a). The process f_u terminates when the top bid is not matchable with the top ask.

Remark 1. It is easy to see that UM outputs a uniform matching: Once the f_u process terminates, Last_trade_price computes the trade price of the last transaction in the output of f_u and Replace_prices replaces the trade prices of each transaction of the output of f_u with the trade price of the last transaction of the output, thus ensuring UM produces a uniform matching. Also, notice that the process Replace_prices does not alter any other information of the output of f_u apart from the trade prices (we will later use this fact in the proof of optimality of UM).

Remark 2. It is easy to see that UM outputs an individual-rational matching: the trade price of a transaction m outputted by a recursive call of f_u is between the limit prices of the bid and the ask of m. Later these prices are altered by Replace_prices, but the individual-rational property is not lost; the trade price of m is also between the limit prices of the transactions of all the previous calls as the bids and the asks are sorted by their competitiveness, and Replace_prices replaces all the trade prices with the trade price of the last transaction.

Now, we discuss the optimality result of the UM process. Throughout this discussion, WLOG, all lists of bids and asks will be sorted by their competitiveness. We make use of the following notation.

▶ **Definition 17.** *Given a matching M, a bid b and an ask a, we use $Q(a \leftrightarrow b, M)$ to denote the total traded quantity between the bid b and the ask a in the matching M.*

Next, we state the main result of this section.

▶ **Theorem 18.** *Given a list of bids B and a list of asks A, let $M_U = \mathsf{UM}(B, A)$ and let M be an arbitrary individual-rational and uniform matching between B and A. Then, $Q(M_U) \geq Q(M)$. In other words,* UM *outputs an optimal individual-rational-uniform matching.*

To prove the above theorem, we need the following lemma.

▶ **Lemma 19.** *If M is an individual-rational and uniform matching between the lists of bids $B = b :: B'$ and asks $A = a :: A'$ such that $Q(M) \geq \min\{q_b, q_a\}$, then there exists another individual-rational and uniform matching M' between the same lists of bids B and asks A such that $Q(M) = Q(M')$ and $Q(a \leftrightarrow b, M') = \min\{q_b, q_a\}$.*

Assuming this lemma, we will first prove Theorem 18 and then later prove the lemma.

Proof of Theorem 18. Note that $f_u(B, A, 0, 0)$ is a specific instance of $f_u(B, A, t_b, t_a)$. So in order to apply the induction hypothesis, we sensitize the theorem statement to incorporate arbitrary values of t_a and t_b. Also, as indicated earlier, the Replace_prices function does not alter the total trade quantity of the output of the f_u, thus $Q(f_u(B, A, 0, 0)) = Q(\mathsf{UM}(B, A))$. Consequently, showing the following suffices.

(∗) Fix an arbitrary list of bids $B = b :: B'$ and an arbitrary list of asks $A = a :: A'$. Fix arbitrarily $t_b < q_b$ and $t_a < q_a$. Let b' be the bid obtained from the bid b by reducing its quantity to $q_b - t_b$. Similarly, let a' be the ask obtained from the ask a by reducing its quantity to $q_a - t_a$. We will show: for all individual-rational and uniform matchings M between $(b' :: B')$ and $(a' :: A')$, $Q(f_u(B, A, t_b, t_a)) \geq Q(M)$.

Clearly, setting $t_b = t_a = 0$ in the above statement $(*)$ gives us Theorem 18.

We prove the above statement using induction on the sum $(|B| + |A|)$. We consider two cases: $p(b') < p(a')$ and $p(b') \geq p(a')$.

In the first case, when $p(b') < p(a')$, since the most competitive bid in $b' :: B'$ is not matchable with the most competitive ask in $a' :: A'$, any matching between $b' :: B'$ and $a' :: A'$ is empty. Thus, $Q(M) = 0$, and we are done.

In the second case, when $p(b') \geq p(a')$, if the total trade quantity of M is less than the quantity of the transaction created by f_u in the first recursive call (i.e., $Q(M) < \min\{q_b - t_b, q_a - t_a\} \leq Q(f_u(B, A, t_b, t_a))$, then we are done. In the case when the total traded quantity of M is more than the quantity of the transaction created by f_u in the first recursive call (i.e., $Q(M) \geq \min\{q_b - t_b, q_a - t_a\}$), we apply Lemma 19 and get another individual-rational and uniform matching M' such that the total volume of M' is equal to the total volume of M and the total traded quantity between the bid b' and the ask a' in M' is equal to $\min\{q_{b'} = q_b - t_b, q_{a'} = q_a - t_a\}$. Now since we have M' such that $Q(M) = Q(M')$, proving the following suffices.

$$Q(f_u(B, A, t_b, t_a)) \geq Q(M') \qquad (**),$$

where M' is an individual-rational and uniform matching between the list of bids $b' :: B'$ and $a' :: A'$ such that $Q(a' \leftrightarrow b', M') = \min\{q_b - t_b, q_a - t_a\}$. We define the matching $M_0 \subseteq M$ as follows: we remove all transactions between b' and a' (of total quantity $Q(a' \leftrightarrow b', M')$) from M' to get M_0. We have $(\dagger)$: $Q(M') = \min\{q_a - t_a, q_b - t_b\} + Q(M_0)$. Also, note that M_0 is individual-rational and uniform (since $M' \supseteq M_0$ is individual-rational and uniform).

Now we argue the proof of $(**)$ in each of the three recursive branches of the function f_u corresponding to $p(b) \geq p(a)$.

- Case: $q_a - t_a = q_b - t_b$. In this case M_0 is a matching between B' and A'. Since $(|B| + |A|) > (|B'| + |A'|)$, we can apply the induction hypothesis to get $Q(f_u(B', A', 0, 0)) \geq Q(M_0)$. Now, applying the definition of f_u we get,

$$\begin{aligned} Q(f_u(B, A, t_b, t_a)) &= \min\{q_a - t_a, q_b - t_b\} + Q(f_u(B', A', 0, 0)) \\ &\overset{\text{I.H.}}{\geq} \min\{q_a - t_a, q_b - t_b\} + Q(M_0) \overset{(\dagger)}{=} Q(M'). \end{aligned}$$

- Case: $q_a - t_a > q_b - t_b$. In this case M_0 is a matching between B' and $\hat{a} :: A'$ (where $q_{\hat{a}} = q_a - t_a - (q_b - t_b) \leq q_a$). Since $(|B| + |A|) > (|B'| + |\hat{a} :: A'|)$, we can apply the induction hypothesis when $B' \neq \emptyset$ to get $Q(f_u(B', A, 0, t_a + (q_b - t_b))) \geq Q(M_0)$. When $B' = \emptyset$, then $Q(f_u(B', A, 0, t_a + (q_b - t_b))) \geq Q(M_0)$ holds trivially as both the sides of the inequality are zeros. Now, applying the definition of f_u we get,

$$\begin{aligned} Q(f_u(B, A, t_b, t_a)) &= (q_b - t_b) + Q(f_u(B', A, 0, t_a + (q_b - t_b))) \\ &\overset{\text{I.H.}}{\geq} (q_b - t_b) + Q(M_0) \overset{(\dagger)}{=} Q(M'). \end{aligned}$$

- Case: $q_a - t_a < q_b - t_b$. This is symmetric to the previous case and the proof follows similarly. □

Having finished the proof of the main result, we now discuss the proof of the lemma that we assumed.

Main proof idea of Lemma 19. Given an individual-rational and uniform matching M with $Q(M) \geq \min\{q_b, q_a\}$ between the list of bids $B = b :: B'$ and the list of asks $A = a :: A'$,

we need to show existence of an individual-rational and uniform matching M' such that $Q(M') = Q(M)$ and the total trade quantity between the bid b and ask a in M' is $\min\{q_b, q_a\}$. We do the following surgery on M in two steps to obtain the desired M'.

Step 1: We first modify M to ensure that bid b and ask a each has at least $\min\{q_b, q_a\}$ total trades in M (not necessarily between each other). This is accomplished by running the Fair procedure on M that outputs a matching which prefers the most competitive orders (b and a) over any other orders. Since $Q(M) \geq \min\{q_b, q_a\}$, we get that $\mathsf{Fair}(M, B, A)$ has at least $\min\{q_b, q_a\}$ trades for each of b and a. Note that Fair does not change the total trade quantity or affect the individual-rational and uniform properties of M. Set $M \leftarrow \mathsf{Fair}(M, B, A)$.

Step 2: In this step, we modify M to ensure that the bid b and ask a have $\min\{q_b, q_a\}$ quantity trade between them. Note that in M individually both b and a have at least $\min\{q_b, q_a\}$ total trade quantity. We will inductively transfer trades of b and a that are not between them to the transaction between b and a, a unit quantity at a time, till they have $\min\{q_b, q_a\}$ quantity trade between them. To better understand this, consider the case when b and a have zero trade quantity between them. Let us say there is a transaction between b and a_1 of quantity q_1 and a transaction between a and b_1 of quantity q_2. We remove these two transactions and replace it with the following four transactions (see Figure 3) that keeps the matching trade volume intact: (1) transaction between b and a_1 of quantity $q_1 - 1$, (2) transaction between a and b_1 of quantity $q_2 - 1$, (3) transaction between b_1 and a_1 of quantity one and (4) transaction between b and a of quantity one. Recall, in a individual-rational and uniform matching with price p, the limit price of each bid is at least p and the limit price of each ask is at most p, implying any bid and ask participating in the matching are matchable. Thus, doing such a replacement surgery is legal and does not affect the individual-rational and uniform properties, and we obtain the desired M' by repeatedly doing this surgery.

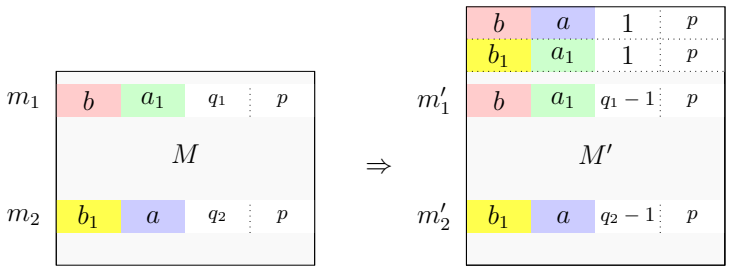

Figure 3 In the above figure the matching M' is obtained from the matching M. Each bid or ask has the same trade quantity in both M and M'. Furthermore, the trade quantity between a and b in M' is one more than that in M.

The proof that UM produces a fair matching follows from inducting on the sum $(|A| + |B|)$ and the fact that B and A are sorted by competitiveness of the participating bids and asks. The argument is similar to the correctness proof of Fair that we saw before. From the discussion above, the next theorem follows immediately.

▶ **Theorem 20.** *For a given list of bids B and the list of asks A, $M = \mathsf{UM}(B, A)$ is a fair and optimal individual-rational-uniform matching on B and A.*

Formalization notes: The formalized proof of the above theorem is done by first proving Lemma 19 (*exists_opt_k*) using induction on the gap $k = \min\{q_b, q_a\} - Q(a \leftrightarrow b, M)$. From this lemma, we get another matching M' such that $Q(a \leftrightarrow b, M) = \min\{q_b, q_a\}$. The

matching M' is altered to M_0 (as described in the proof of Theorem 18 above) by removing all the transactions between the bid b and the ask a. We prove that the altered list M_0 is is a matching between the reduced lists of bids and asks. All the results related to M_0 are in the file "MachingAlter.v". Finally, combining all these we prove the main Theorem 20 as *"UM_main"*.

7 A Maximum Matching Mechanism

In the previous section, we indicated that to achieve maximum trade volume matching we sometimes have to assign different trade prices to the matched bid-ask pairs. An individual rational matching with maximum trade volume is called a maximum matching. In this section we describe a process MM, that takes a list of bids and a list of asks and outputs a fair, individual-rational and maximum matching.

The MM procedure roughly works as follows. In step one, the MM procedure repeatedly pairs the most competitive bid b with the least competitive matchable ask a and outputs a transaction $(b, a, \min\{q_b, q_a\}, p_a)$ and decreases the quantities of b and a by $\min\{q_b, q_a\}$. In step two, the MM procedure applies the Fair procedure on the output of step one.

The detailed MM procedure and the proof of its correctness are similar to that of the UM procedure in spirit. In the proof of optimality, we need to prove a lemma similar to Lemma 19 which states that a given arbitrary individual-rational matching M of sufficiently large trade volume can be altered to obtain a matching M' of the same total trade volume such that the total trade quantity between the most competitive bid and the corresponding least competitive matchable ask in M' is the minimum of their respective quantities. The proof of this requires more surgeries as compared to that in the proof of Lemma 19. Besides this deviation all other arguments of the proof of optimality of MM are similar to that of UM with minor variations.

The proof of MM producing an individual-rational matching is trivial and the proof that it produces a fair matching follows from the fact that MM applies the Fair procedure before it outputs a final matching. We now state the main theorem of this section.

▶ **Theorem 21.** *For a given list of bids B and a list of asks A,* $\mathsf{MM}(B, A)$ *is a fair, individual-rational and maximum trade volume matching between B and A.*

The proof of the above theorem and discussion around the MM procedure can be found in the full version of the paper [5].

Formalization notes: All the formalization details can be found in [10].

8 Uniqueness Theorem

In this section, we establish certain theorems that enable us to automatically check for violations in an exchange matching algorithm by comparing its output with the output of our certified program. Detailed proofs are available in the Coq formalization [10].

Ideally, we would have wanted a theorem that the properties (fair and optimal individual-rational-uniform) imply a unique matching. Such a theorem would enable us to automatically compare a matching produced by an exchange with a matching produced by our certified program to find violations of these properties in the matching produced by the exchange. Unfortunately, such a theorem is not possible; there exists two different matchings M_1 and M_2 on the same list of bids B and asks A, where both are fair and optimal individual-rational-uniform: $M_1 = \{(b_1, a_1, 1, p), (b_2, a_2, 2, p)\}$ and $M_2 = \{(b_1, a_2, 1, p), (b_2, a_2, 1, p), (b_2, a_1, 1, p)\}$

on bids $B = \{b_1 = (*, *, 1, p), b_2 = (*, *, 2, p)\}$ and asks $A = \{a_1 = (*, *, 1, p), a_2 = (*, *, 2, p)\}$ for some arbitrary price p, timestamps and ids. Note that fairness does not require the most competitive bid to be paired with the most competitive ask. For example, assuming a_1 has a lower timestamp than a_2 and b_1 has a lower timestamp than b_2 in the above example, a_1 and b_1 are not matched in the matching M_2, which is a fair matching. Nonetheless, we can show that given a list of bids B and a list of asks A, all matchings that are fair and individual-rational-uniform, must have the same trade volume for each trader. This still allows us to automatically check for violations of the properties in an exchange, by comparing the trades of each trader produced by the exchange against that produced by our certified program.

We have the following lemma which formulates this uniqueness relation on the matchings.

▶ **Theorem 22.** *Let M_1 and M_2 be two fair matchings on the list of bids B and the list of asks A such that $Q(M_1) = Q(M_2)$, then for each order ω, the total traded quantity of ω in M_1 is equal to the total traded quantity of ω in M_2.*

Proof Idea. We now prove the above theorem by using Lemma 6 and deriving a contradiction. Let M_1 and M_2 be fair matchings such that $Q(M_1) = Q(M_2)$. Let b be a buyer whose total trade quantity in M_1 is different (WLOG, more) from his total trade quantity in M_2. It is easy to show that there exists another buyer b' such that her total traded quantity in M_1 is less than her total traded quantity in M_2, i.e., $Q(M_2, b') > Q(M_1, b')$ (since the sum of the total traded quantities of all the bids of B in M_1 is equal to the sum of the total traded quantities of all the bids of B in M_2 from Lemma 6).

Now, there can be two cases: (i) b is more competitive than b' or (ii) b' is more competitive than b, as per price-time priority. In the first case, since $Q(M_1, b) > Q(M_2, b)$, it follows that $Q(M_2, b) < Q(M_1, b) \leq q_b$. This contradicts the fact that M_2 is fair on the bids; this is because a less competitive bid b' is being traded in M_2 (since $Q(M_2, b') > Q(M_1, b') \geq 0$ as noted above), while a more competitive bid b is not fully traded. Similarly, in the second case, we show a contradiction to the fact that M_1 is fair on the bids. □

From the above theorem, we have the following corollary.

▶ **Theorem 23.** *For any two fair and optimal individual-rational-uniform matchings M_1 and M_2 on the list of bids B and the list of asks A, for each order ω, the total traded quantity of ω in M_1 is equal to the total traded quantity of ω in M_2.*

For each trader, we can compare the total traded quantities of the trader in the matching M_1 produced by an exchange with the total traded quantities of the trader in the matching $M_2 = \mathsf{UM}(B, A)$ produced by our certified program. If for some trader, the traded quantities do not match, then from Theorem 20 and Theorem 23 we know that M_1 does not have the desired properties as required by the regulators. On the other hand, if they do match for all traders, then the following theorem states that M_1 is fair (Note that uniform and individual-rational properties can be verified directly from the trade prices and clearly the total trade volume of M_1 and M_2 are the same if the traded quantities are same for each trader).

▶ **Theorem 24.** *Given a list of bids B and a list of asks A, if M_1 is a fair matching and M_2 is an arbitrary matching such that for each order ω, the total traded quantity of ω in M_1 is equal to the total traded quantity of ω in M_2, then M_2 is fair.*

The proof follows immediately from the definition of fairness.

Formalization notes: All the theorems in this section are formalized in the file "Uniqueness.v" using the above proof ideas.

9 Demonstration: Automatic Detection of Violations in Real Data

Please see Appendix A for details on our demonstration, where we automate the process of checking violations in trades using verified programs extracted from our formalization. We then use this to find violations in trades of 100 stocks traded on a real exchange on a particular day. Below, we describe our findings.

Out of the 100 stocks we checked, for three stocks our program outputted "Violation detected!". When we closely examined these stocks, we realized that in all of these stocks, a market ask order (with limit price = 0), was not matched by the exchange in its trading output (and these were the only market ask orders in the entire order-book). On the contrary, market bid orders were matched by them. With further investigation, we observed that corresponding to each of these three violations, in the raw data there was an entry of update request in the order-book with a limit price and timestamp identical to the uniform price and the auction time, respectively. It seems highly unlikely that these three update requests were placed by the traders themselves (to match the microsecond time and also the trade price seems very improbable); we suspect this is an exchange's system generated entry in the order-book. We hope that the exchange is aware of this and doing this consciously. When we delete the market asks in the preprocessing stage, no violations are detected. Even if it is not a violation (but a result of the exchange implementing some unnatural rule that we are not aware of), it is fascinating to see that with the help of verified programs we can identify such minute and interesting anomalies which can be helpful for regulating and improving the exchange's matching algorithm.

10 Related Works and Future Direction

In an earlier work [8], Sarswat and Singh dealt primarily with single unit trade requests and thus provided a proof of concept for obtaining verified programs for financial markets. In the current work, we extend their work to multiple units that results in verified programs which we run on real market data and establish new uniqueness theorems that enable automatic detection of violation in exchanges as demonstrated in this work.

Passmore and Ignatovich in [7] highlight the significance, opportunities and challenges involved in formalizing financial markets. They describe the whole spectrum of financial algorithms that need to be verified for ensuring safe and fair markets. Iliano *et al.* [1] use concurrent linear logic (CLF) to outline two important properties of a continuous trading system. There are also some works formalizing various concepts from auction theory [2, 3, 11], particularly focusing on the Vickrey auction mechanism.

In our opinion, future works should focus on developing a theory for continuous double auctions for financial markets. Currently the specifications for continuous double auctions are vague and this is an obstacle for obtaining verified programs.

References

1 Iliano Cervesato, Sharjeel Khan, Giselle Reis, and Dragisa Zunic. Formalization of automated trading systems in a concurrent linear framework. In *Linearity-TLLA@FLoC*, volume 292 of *EPTCS*, pages 1–14, 2018. URL: `http://arxiv.org/abs/1904.06159`.

2 Cezary Kaliszyk and Julian Parsert. Formal microeconomic foundations and the first welfare theorem. In *Proceedings of the 7th ACM SIGPLAN International Conference on Certified Programs and Proofs*, pages 91–101. ACM, 2018.

3 Stéphane Le Roux. Acyclic preferences and existence of sequential nash equilibria: a formal and constructive equivalence. In *International Conference on Theorem Proving in Higher Order Logics*, pages 293–309. Springer, 2009.

4 R Preston McAfee. A dominant strategy double auction. *Journal of economic Theory*, 56(2):434–450, 1992.

5 Raja Natarajan, Suneel Sarswat, and Abhishek Kr Singh. Verified double sided auctions for financial markets, 2021. `arXiv:2104.08437`.

6 Jinzhong Niu and Simon Parsons. Maximizing matching in double-sided auctions. In *International conference on Autonomous Agents and Multi-Agent Systems, AAMAS '13, Saint Paul, MN, USA, May 6-10, 2013*, pages 1283–1284, 2013.

7 Grant Olney Passmore and Denis Ignatovich. Formal verification of financial algorithms. In *26th International Conference on Automated Deduction, Proceedings*, volume 10395 of *Lecture Notes in Computer Science*, pages 26–41. Springer, 2017.

8 Suneel Sarswat and Abhishek Kr Singh. Formally verified trades in financial markets. In Shang-Wei Lin, Zhe Hou, and Brendan Mahoney, editors, *Formal Methods and Software Engineering - 22nd International Conference on Formal Engineering Methods, ICFEM 2020, Singapore, Singapore, March 1-3, 2021, Proceedings*, volume 12531 of *Lecture Notes in Computer Science*, pages 217–232. Springer, 2020. `doi:10.1007/978-3-030-63406-3_13`.

9 Matthieu Sozeau and Cyprien Mangin. Equations reloaded: High-level dependently-typed functional programming and proving in coq. *Proceedings of the ACM on Programming Languages*, 3(ICFP):1–29, 2019.

10 Suneel Sarswat. Coq formalization of mdsa. `https://github.com/suneel-sarswat/dsam`.

11 Emmanuel M. Tadjouddine, Frank Guerin, and Wamberto Weber Vasconcelos. Abstracting and verifying strategy-proofness for auction mechanisms. In *DALT*, volume 5397 of *Lecture Notes in Computer Science*, pages 197–214. Springer, 2008.

12 Peter R. Wurman, William E. Walsh, and Michael P. Wellman. Flexible double auctions for electronic commerce: theory and implementation. *Decision Support Systems*, 24(1):17–27, 1998.

13 Dengji Zhao, Dongmo Zhang, Md Khan, and Laurent Perrussel. Maximal matching for double auction. In *Australasian Conference on Artificial Intelligence*, volume 6464 of *Lecture Notes in Computer Science*, pages 516–525. Springer, 2010.

A Demonstration on real data

In this section, we demonstrate the practical applicability of our work. For this, we procured real data from a prominent stock exchange. This data consists of order-book and trade-book of everyday trading for a certain number of days. For our demonstration, we considered trades for the top 100 stocks (as per their market capitalizations) of a particular day. For privacy reasons, we conceal the real identity of the traders, stocks and the exchange by masking the stock names (to s1 to s100) and the traders' identities. We also converted the timestamps appropriately into natural numbers (which keeps the time in microseconds, as in the original data). Furthermore, the original data has multiple requests with the same order id; this is because some traders update or delete an existing order placed by them before the double sided auction is conducted. In our preprocessing, we just keep the final lists of bids and asks in the order-book that participate in the auction. Furthermore, there are certain market orders, i.e., orders that are ready to be traded at any available price, which effectively means a limit price of zero for an ask and a limit price of infinity for a bid; in the preprocessing we set these limit prices to zero and the largest OCaml integer, respectively.

We then extracted the verified OCaml programs and ran them on the processed market data. The output trades of the verified code were then compared with the actual trades

in the trade-book from the exchange. From the uniqueness theorems in the Section 8, we know that if the total trade quantity of each order in these two matchings are equal, then the matching produced by the exchange has the desired properties (if it is uniform and IR which can be checked trivially by looking at the prices in the trade-book). We also know that if they are not equal for some trader, then the matching algorithm of the exchange does not have the requisite desired properties (or there is some error in storing or reporting the order-book or the trade-book accurately).

The processed data and the relevant programs for this demonstration are available at [10]. The extracted OCaml programs of the functions required for this demonstration are stored in a separate file named "certified.ml". The input bids, asks and trades of each stock are in "s.bid", "s.ask" and "s.trade" files, where "s" is the masked id for that stock. For example, file "s1.bid" contains all the bids for the stock "s1". To feed the inputs to the verified program and to print the output of the certified program, we have written two OCaml scripts: create.ml and compare.ml. The create.ml script feeds inputs (lists of bids and asks) to the UM process, and then prints its output matching M. The compare.ml script compares the matching produced by the UM process M with the actual trades M_{EX} in the exchange trade-book. If the total trade quantity for all the traders in M matches with that of the total trade quantity in M_{EX}, then the compare.ml script outputs "Matching does not violate the guidelines". If for some bid (or ask) the total trade quantity of M and M_{EX} does not match, then the program outputs "Violation detected!".