Betting Boolean-Style: A Framework for Trading in Securities Based on Logical Formulas

Lance Fortnow [1]

Department of Computer Science, University of Chicago, 1100 E. 58th St., Chicago, IL 60637

Joe Kilian

NEC Laboratories America, 4 Independence Way, Princeton, NJ 08540

David M. Pennock [1]

Yahoo! Labs, 74 N. Pasadena Ave., 3rd floor, Pasadena, CA 91103

Michael P. Wellman

University of Michigan, AI Laboratory, 1101 Beal Avenue, Ann Arbor, MI 48109

Abstract

We develop a framework for trading in *compound securities*: financial instruments that pay off contingent on the outcomes of arbitrary statements in propositional logic. Buying or selling securities—which can be thought of as betting on or against a particular future outcome—allows agents both to hedge risk and to profit (in expectation) on subjective predictions. A compound securities market allows agents to place bets on arbitrary boolean combinations of events, enabling them to more closely achieve their optimal risk exposure, and enabling the market as a whole to more closely achieve the social optimum. The tradeoff for allowing such expressivity is in the complexity of the agents' and auctioneer's optimization problems.

We develop and motivate the concept of a compound securities market, presenting the framework through a series of formal definitions and examples. We then analyze in detail the auctioneer's matching problem. We show that, with n events, the matching problem is worst-case intractable: specifically, the problem is co-NP-complete in the divisible case and Σ_2^p-complete in the indivisible case. We show that the latter hardness result holds even under severe language restrictions on bids. With $\log n$ events, the problem is tractable (polynomial) in the divisible case and worst-case intractable (NP-complete) in the indivisible case. We briefly discuss matching algorithms and tractable special cases.

Preprint submitted to Elsevier Science *31 December 2003*

Key words: Compound securities markets, computational complexity of matching, combinatorial betting, trading in financial instruments based on logical formulas, risk allocation, information aggregation, hedging, speculating, betting, gambling

1 Introduction

Securities markets effectively allow traders to place bets on the outcomes of uncertain future propositions. Examples include stock markets like NASDAQ, options markets like CBOE [1], futures markets like CME [2], other derivatives markets, insurance markets, political stock markets [3,4], sports betting markets [5–7], horse racing markets [8], idea futures markets [9], decision markets [10] and even market games [11–13]. The economic value of securities markets is two-fold. First, they allow traders to *hedge risk*, or to insure against undesirable outcomes. For example, the owner of a stock might buy a *put option* (the right to sell the stock at a particular price) in order to insure against a stock downturn. Or the owner of a house may purchase an insurance contract to hedge against unforeseen damage to the house. Second, securities markets allow traders to *speculate*, or to obtain a subjective expected profit when market prices do not reflect their assessment of the likelihood of future outcomes. For example, a trader might buy a call option if he believes that the likelihood is high that the price of the underlying stock will go up, regardless of risk exposure to changes in the stock price. Because traders stand to earn a profit if they can make effective probability assessments, often prices in financial markets yield very accurate aggregate forecasts of future events [14–17].

Real securities markets have complex payoff structures with various triggers. However, these can all be modeled as collections of more basic or atomic *Arrow-Debreu securities* [18–20]. One unit of one Arrow-Debreu security pays off one dollar if and only if (iff) a corresponding binary event occurs; it pays nothing if the event does not occur. So, for example, one unit of a security denoted $\langle$Acme100$\rangle$ might pay \$1 iff Acme's stock is above \$100 on January 4, 2004. An Acme stock option as it would be defined on a financial exchange can be though of as a portfolio of infinitely many such atomic securities, or can be approximated to any degree with a finite number of atomic securities.

In this paper, we develop and analyze a framework for trading in *compound securities markets* with payoffs contingent on arbitrary logical combinations of events, including conditionals. For example, given binary events A, B, and C, one trader might bid to buy three units of a security denoted $\langle A \wedge (\bar{B} \vee C)\rangle$

[1] Research done while at the NEC Research Institute, Princeton, New Jersey.

that pays off \$1 iff the compound event $A \wedge (\bar{B} \vee C)$ occurs for thirty cents each. Another trader may bid to sell six units of a security $\langle A|C \rangle$ that pays off \$1 iff A occurs for fifty-five cents each, *conditional* on event C occurring, meaning that the transaction is revoked if C does not occur (i.e., no payoff is given and the price of the security is refunded) [21]. Bids may also be *divisible*, meaning that bidders are willing to accept less than the requested quantity, or *indivisible*, meaning that bids must be fulfilled either completely or not at all. Given a set of such bids, the auctioneer faces a complex *matching problem* to decide which bids are accepted for how many units at what price. Typically, the auctioneer seeks to take on no risk of its own, only matching up agreeable trades among the bidders, but we also consider alternative formulations where the auctioneer acts as a market maker willing to accept some risk.

We examine the computational complexity of the auctioneer's matching problem. Let the length of the description of all the available securities be $\mathrm{O}(n)$. With n events, the matching problem is co-NP-complete in the divisible case and Σ_2^p-complete in the indivisible case. This Σ_2^p-complete hardness holds even when the bidding language is significantly restricted. These complexity results mean that, although identifying a match may not always be computationally difficult, in the worst case finding a match will take computing time that is exponential in n, assuming (as most computer scientists do) that $\mathrm{P} \neq \mathrm{NP}$. With $\log n$ events, the problem is polynomial in the divisible case—meaning that the problem is computationally feasible. With $\log n$ events and indivisible bids, the problem is NP-complete, again implying worst-case intractability.

Section 2 presents some necessary background information, motivation, and related work, including a review of the meaning of—and the distinctions between—the various computational complexity classes cited above. Section 3 formally describes our framework for compound securities, and defines the auctioneer's matching problem. Section 4 briefly discusses natural algorithms for solving the matching problem. Section 5 proves our central computational complexity results. Section 6 discusses the possibility of tractable special cases. Section 7 concludes with a summary and some ideas of future directions.

2 Preliminaries

2.1 Background and notation

Imagine a world where there are only two future uncertain events of any consequence: (1) the event that one's house is struck by lightning by December 31, 2003, denoted *struck*, and (2) the event that Acme's stock price goes above \$100 by January 4, 2004, denoted *acme100*. In this simple world there are

four possible future *states*—all possible combinations of the binary events' outcomes:

$$\begin{aligned}
&\mathit{struck} \wedge \mathit{acme100},\\
&\overline{\mathit{struck}} \wedge \mathit{acme100},\\
&\mathit{struck} \wedge \overline{\mathit{acme100}},\\
&\overline{\mathit{struck}} \wedge \overline{\mathit{acme100}}.
\end{aligned}$$

Hedging risk can be thought of as an action of moving money between various possible future states. For example, insuring one's house transfers money from future states where *struck* is not true to states where it is. Selling a security denoted $\langle$acme100$\rangle$—that pays off \$1 iff the event *acme100* occurs—transfers money from future states where Acme's price is above \$100 on January 4 to states where it's not. Speculating is also an act of transferring money between future states, though usually associated with maximizing expected return rather than reducing risk. For example, betting on a football team moves money from the "team loses" state to the "team wins" state. In practice, agents engage in a mixture of hedging and speculating, and there is no clear dividing line between the two [22].

All possible future outcomes form a *state space* Ω, consisting of mutually exclusive and exhaustive states $\omega \in \Omega$. Often a more natural way to think of possible future outcomes is as an *event space* $\mathcal{A}$ of linearly independent events $A \in \mathcal{A}$ that may overlap arbitrarily. So in our toy example *struck* $\wedge$ *acme100* is one of the four disjoint states, while *struck* is one of the two events. Note that a set of n linearly independent events defines a state space Ω of size 2^n consisting of all possible combinations of event outcomes. Conversely, any state space Ω can be factored into $\lceil \log |\Omega| \rceil$ events.

Suppose that $\mathcal{A}$ exhaustively covers all meaningful future outcomes (i.e., covers all eventualities that agents may wish to hedge against and/or speculate upon). Then the existence of 2^n linearly independent securities—called a *complete* market—allows agents to distribute their wealth arbitrarily across future states. [2] An agent may create any hedge or speculation it desires. Under classical conditions, agents trading in a complete market form an equilibrium where risk is allocated Pareto optimally. If the market is incomplete, meaning it consists of fewer than 2^n linearly independent securities, then in general agents cannot construct arbitrary hedges and equilibrium allocations may be nonoptimal [18,19,23,20].

In real-world settings, the number of meaningful events n is large and thus the number of securities required for completeness is intractable. No truly

[2] By linearly independent securities, we mean that the vectors of payoffs in all future states of these securities are linearly independent.

complete market exists or will ever exist. One motivation behind compound securities markets is to provide a mechanism that supports the most transfer of risk using the least number of transactions possible. Compound securities allow a high degree of expressivity in constructing bids. The tradeoff for increased expressivity is increased computational complexity, from both the bidder's and auctioneer's point of view.

2.2 Computational complexity

Computer scientists have developed a procedure for characterizing how difficult or time-consuming a particular problem will be to solve on a computer. Problems are organized according to their *computational complexity*, or relative difficulty. Complexity is *not* described in seconds or any other unit of time, since that would vary greatly depending on the particulars of the computer hardware architecture used. Complexity is instead described as a growth function that increases as the size of the input, or the number of bits required to describe the problem, increases. If the size of a problem is n, then a complexity that is linear in n, denoted $O(n)$, is considered computationally feasible. Generally, any complexity that is polynomial in n is considered feasible, while a complexity that is exponential in n is considered intractable. If the problem's complexity grows exponentially in n, then the time required to solve the problem will become prohibitive even for fairly small-sized problems, almost regardless of the hardware used.

Computational complexity is a *worst case* notion: saying that a problem falls into a computationally intractable class means only that, *for some* instances of the problem, solutions will require an inordinate amount of time. The characterization says nothing about the set of all instances of the problem, or even about the average case. It may well be that most instances of the problem are easy to solve, though some will be extremely hard. Also, the complexity of solving a problem exactly may differ from the complexity of solving the problem to within some degree of approximation.

The two main complexity classes are denoted P and NP. Problems that lie in the class P are solvable in polynomial time and are considered to be computationally feasible. Problems that lie in the class NP include all of the polynomial-solvable problems in P, as well as a huge number of other common problems for which no known polynomial-time algorithm is known, and for which most believe require exponential time in the worst case. [3] Although no one has proven that P $\neq$ NP, most computer scientists believe that the two

[3] NP stands for *non-deterministic polynomial time*, for reasons that are beyond the scope of this paper. NP does *not* mean non-polynomial: in fact all problems in P are also in NP.

classes are not the same, and that many more problems lie in NP than in P. A problem is said to be NP-hard if it is provable as hard or harder than every other problem in NP. A problem is said to be NP-complete if it is both NP-hard and is itself a member of NP. Since it is widely believed that there exist problems in NP that require exponential time in the worst case, [4] any problem that can be designated as NP-hard or NP-complete can be considered, for all intents and purposes, to require exponential time to solve in the worst case.

In this paper, we make use of two other less common complexity classes. A binary decision problem, whose solution is either YES or NO, is said to be co-NP-complete if the complementary problem, with precisely reversed solutions (YES=NO and NO=YES), is NP-complete. A co-NP-complete problem is NP-hard, and for all practical purposes can be considered equally as hard as its complementary NP-complete problem. The complexity class Σ_2^p includes all problems in NP as well as many other problems that are thought to be even harder. A problem is considered Σ_2^p-complete if it is a member of Σ_2^p and is provably at least as hard as all other problems in Σ_2^p.

Problems are proven to be NP-complete in two steps: (1) showing that the problem is a member of the class NP, and (2) providing a polynomial-time *reduction* from a problem that is already known to be NP-complete to the new problem. Armed with the reduction, anyone who develops a fast algorithm for the new problem will immediately have a fast algorithm (within a polynomial factor) for the known NP-complete problem; therefore, the new problem is at least as hard as the known problem in terms of computational complexity. Proving that a problem is Σ_2^p-complete is analogous, with the reduction working from a known Σ_2^p-complete problem. We employ this reduction technique below in Section 5 to derive our main complexity results.

Although Σ_2^p-complete problems are theoretically and provably harder than NP-complete problems (assuming P $\neq$ NP), in practice the distinction is usually not that great. The practical difference between P and NP is greater by far than the difference between NP and Σ_2^p. For example, algorithms for Bayesian inference—a problem that is even harder than Σ_2^p-complete [5]—are commercially available and work reasonably well in a variety of real applications. Still, the complexity results for Bayesian inference, as well as for the matching problem described in this paper, insure that, for some problem instances, all the computing power in world operating over the lifetime of the universe will not be enough to solve the problem.

[4] Indeed, after several decades of trying, no one has ever found any algorithm to solve any of the thousands of NP-hard problems in less than exponential time.
[5] Specifically, Bayesian inference is #P-complete [24].

The quest to reduce the number of financial instruments required to support an optimal allocation of risk dates to Arrow's original work [18]. The requirement stated above of "only" 2^n linearly-independent securities is itself a reduction from the most straightforward formulation. In an economy with k standard goods, the most straightforward complete market contains $k \cdot 2^n$ securities, each paying off in one good under one state realization. Arrow [18] showed that a market where securities and goods are essentially separated, with 2^n securities paying off in a single numeraire good plus k *spot* markets in the standard goods, is also complete. For our purposes, we need consider only the securities market.

Varian [25] shows that a complete market can be constructed using fewer than 2^n securities, replacing the missing securities with options. Still, the number of linearly independent financial instruments—securities plus options—must be 2^n to guarantee completeness.

Though the requirement of 2^n financial instruments cannot be relaxed if one wants to guarantee completeness in all circumstances, Pennock and Wellman [26] explore conditions under which a smaller securities market may be *operationally complete*, meaning that its equilibrium is Pareto optimal *with respect to the agents involved*, even if the market contains less than 2^n securities. The authors show that in some cases the market can be structured and "compacted" in analogy to Bayesian network representations of joint probability distributions [27]. They show that, if all agents' risk-neutral independencies agree with the independencies encoded in the market structure, then the market is operationally complete. For collections of agents all with constant absolute risk aversion, agreement on Markov independencies is sufficient.

Bossaerts, Fine, and Ledyard [28] develop a mechanism they call *combined-value trading* (CVT) that allows traders to order an arbitrary *portfolio* of securities in one bid, rather than breaking up the order into a sequence of bids on individual securities. If the portfolio order is accepted, all of the implied trades on individual securities are executed simultaneously, thus eliminating so-called *execution risk* that prices will change in the middle of a planned sequence of orders. The authors conduct laboratory experiments showing that, even in thin markets where ordinary sequential trading breaks down, CVT supports efficient pricing and allocation. Note that CVT differs significantly from compound securities trading. CVT allows instantaneous trading of any *linear* combination of securities, while compound securities allow more expressive securities that can encode *nonlinear* boolean combinations of events. For example, CVT may allow an agent to order securities $\langle A \rangle$ and $\langle B \rangle$ in a bundle

that pays off as a linear combination of A and B,[6] but CVT won't allow the construction of a compound security $\langle A \wedge B \rangle$ that pays off \$1 iff *both* A and B occur, or a compound security $\langle A|B \rangle$.

Related to CVT are *combinatorial auctions* [29,30] and *exchanges* [31], mechanisms that have recently received quite a bit of attention in the economics and computer science literatures. Combinatorial auctions allow bidders to place distinct values on all possible bundles of goods rather than just on individual goods. In this way bidders can express substitutability and complementarity relationships among goods that cannot be expressed in standard parallel or sequential auctions. Compound securities differ from combinatorial auctions in concept and complexity. Compound securities allow bidders to construct an arbitrary bet on any of the 2^{2^n} possible compound events expressible as logical functions of the n base events, conditional on any other of the 2^{2^n} compound events. Agents optimize based on their own subjective probabilities and risk attitude (and in general, their beliefs about other agents' beliefs and utilities, ad infinitum). The central auctioneer problem is identifying arbitrage opportunities: that is, to match bets together without taking on any risk. Combinatorial auctions, on the other hand, allow bids on any of the 2^n bundles of n goods. Typically, uncertainty—and thus risk—is not considered. The central auctioneer problem is to maximize social welfare. Also note that the problems lie in different complexity classes. While clearing a combinatorial auction is polynomial in the divisible case and NP-complete in the indivisible case, matching in a compound securities market is NP-complete in the divisible case and Σ_2^p-complete in the indivisible case. In fact, even the problem of deciding whether *two* bids on compound securities match, even in the divisible case, is NP-complete (see Section 5.2).

There is a sense in which it is possible to translate our matching problem for compound securities into an analogous problem for clearing *two-sided* combinatorial exchanges [31] of exponential size. Specifically, if we regard payoff in a particular state as a good, then compound securities can be viewed as bundles of (fractional quantities of) such goods. The material balance constraint facing the combinatorial auctioneer corresponds to a restriction that the compound-security auctioneer be disallowed from assuming any risk. Note that this translation is not at all useful for addressing the compound-security matching problem, as the resulting combinatorial exchange has an exponential number of goods.

Hanson [32] develops a market mechanism called a *market scoring rule* that is especially well suited for allowing bets on a combinatorial number of outcomes. The mechanism maintains a joint probability distribution over all 2^n

[6] Specifically, one unit of each pays off \$2 iff both A and B occur, \$1 iff A or B occurs (but not both), and \$0 otherwise.

states, either explicitly or implicitly using a Bayesian network or other compact representation. At any time any trader who believes the probabilities are wrong can change any part of the distribution by accepting a lottery ticket that pays off according to a scoring rule (e.g., the logarithmic scoring rule) [33], as long as that trader also agrees to pay off the most recent person to change the distribution. The market interface can be made to look to traders like a continuous double auction with a market maker who is always willing to accept a bid on any boolean proposition at some price. In the limit of a single trader, the mechanism behaves like a scoring rule, suitable for polling a single agent for its probability distribution. In the limit of many traders, it produces a combined estimate. Since the market essentially always has a complete set of posted prices for all possible outcomes, the mechanism avoids the problem of thin markets, or illiquidity, that necessarily plagues any market containing an exponential number of alternative investments. The mechanism requires a patron to pay off the final person to change the distribution, though the patron's payment is bounded. Though Hanson offers some initial suggestions, several open problems remain, including efficient methods for representing and updating the joint distribution and recording traders positions and portfolios, without resorting to exponential time and space algorithms.

Fagin, Halpern, and Megiddo [34] give a sound and complete axiomatization for deciding whether sets of *probabilistic inequalities* are consistent. Bids for compound securities can be thought of as expressions of probabilistic inequalities: for example, a bid to buy $\langle A \wedge B \rangle$ at price 0.3 is a statement that the probability of $A \wedge B$ is greater than 0.3. If a set of single-unit bids correspond to a set of inconsistent probabilistic inequalities, then there is a match. However, because they are interested in a much different framework, Fagin *et al.* do not consider several complicating factors specific to the securities market framework: namely, handling multi-unit or fractional bid quantities, identifying matches, choosing among multiple matches, and optimizing based on probabilities and risk attitudes. We address these issues below.

3 Framework for trading in compound securities

3.1 High-level description

Common knowledge among agents is the set of events $\mathcal{A}$. There are no predefined securities. Instead, agents offer to buy or sell securities of their own design that pay off contingent on logical combinations of events and event negations. Combination operators may include conjunctions, disjunctions, and conditionals.

For all practical purposes, it is impossible for agents to trade in enough securities (2^n) to form a complete market, so agents must devise their best tradeoff between the number and complexity of their bids, and the extent to which their risks are hedged and desirable bets are placed. In its most general form, the problem is game-theoretic in nature, since what an agent should offer depends on what it believes other agents will accept. At the other end of the spectrum, a simplified version of the problem is to optimize bids only on currently available securities at current prices. In between these two formulations are other possible interesting optimization problems, including approximation techniques.

The auctioneer faces a nontrivial problem of matching buy and sell orders to maximize surplus (the cash and securities left over after accepted bids are fulfilled). For example, offers to sell $\langle A_1 A_2 \rangle$ at \$0.2 and $\langle A_1 \bar{A}_2 \rangle$ at \$0.1 can match with an offer to buy $\langle A_1 \rangle$ at \$0.4, with surplus \$0.1. Or an offer to sell $\langle A_1 \rangle$ at \$0.3 can match with an offer to buy $\langle A_1 A_2 \rangle$ at \$0.4, with surplus \$0.1 in cash and $\langle A_1 \bar{A}_2 \rangle$ in securities. In general, a single security might qualify for multiple matches, but only one can be transacted. So the auctioneer must find the optimal (or approximately optimal) set of matches that maximizes surplus, which could be measured in a number of ways. In another formulation, the auctioneer functions as a market maker willing to take on a certain amount of risk.

Informally, our motivation is to provide a mechanism that allows a very high degree of expressivity in placing hedges and bets, and is also capable of approximating the optimal (complete-market) allocation of risk, trading off the number and complexity of securities and transactions needed.

3.2 Formal description

3.2.1 Securities

We use ϕ and ψ to denote arbitrary boolean formulas, or logical combinations of events in $\mathcal{A}$. We denote securities $\langle \phi | \psi \rangle$. Securities pay off \$1 if and only if (iff) ϕ and ψ are true, pay off \$0 iff ϕ is false and ψ is true, and are canceled (i.e., any price paid is refunded) iff ψ is false. We define $T \equiv \Omega$ to be the event "true" and $F \equiv \emptyset$ to be the event "false". We abbreviate $\langle \phi | T \rangle$ as $\langle \phi \rangle$.

3.2.2 Orders

Agents place orders, denoted o, of the form "q units of $\langle \phi | \psi \rangle$ at price p per unit", where $q > 0$ implies a buy order and $q < 0$ implies a sell order. We assume agents submitting buy (sell) orders will accept any price $p^* \leq p$ ($p^* \geq p$).

We distinguish between *divisible* and *indivisible* orders. Agents submitting divisible orders will accept any quantity αq where $0 < \alpha \leq 1$. Agents submitting indivisible orders will accept only exactly q units, or none at all. We believe that, given the nature of what is being traded (state-contingent dollars), most agents will be content to trade using divisible orders.

Every order o can be translated into a payoff vector $\boldsymbol{\Upsilon}$ across all states $\omega \in \Omega$. The payoff $\Upsilon^{\langle\omega\rangle}$ in state ω is $q \cdot 1_{\omega\in\psi}(1_{\omega\in\phi} - p)$, where $1_{\omega\in E}$ is the indicator function equaling 1 iff $\omega \in E$ and zero otherwise. Recall that the 2^n states correspond to the 2^n possible combinations of event outcomes. We index multiple orders with subscripts (e.g., o_i and $\boldsymbol{\Upsilon}_\mathbf{i}$). Let the set of all orders be $\mathcal{O}$ and the set of all corresponding payoff vectors be $\mathcal{P}$.

Example 1 *(Translating orders into payoff vectors) Suppose that* $|\mathcal{A}| = 3$. *Consider an order to buy two units of* $\langle A_2 \vee A_3 | A_1\rangle$ *at price \$0.8. The corresponding payoff vector is:*

$$\begin{aligned}\boldsymbol{\Upsilon} &= \langle \Upsilon^{\langle A_1 A_2 A_3\rangle}, \Upsilon^{\langle A_1 A_2 \bar{A_3}\rangle}, \Upsilon^{\langle A_1 \bar{A_2} A_3\rangle}, \ldots, \Upsilon^{\langle \bar{A_1}\bar{A_2}\bar{A_3}\rangle}\rangle \\ &= 2 \cdot \langle 0.2, 0.2, 0.2, -0.8, 0, 0, 0, 0\rangle\end{aligned}$$

□

3.2.3 The matching problem

The auctioneer's task, called the *matching problem*, is to determine which orders to accept among all orders $o \in \mathcal{O}$. Let α_i be the fraction of order o_i accepted by the auctioneer (in the indivisible case, α_i must be either 0 or 1; in the divisible case, α_i can range from 0 to 1). If $\alpha_i = 0$, then order o_i is considered rejected and no transactions take place concerning this order. For accepted orders ($\alpha_i > 0$), the auctioneer receives the money lost by bidders and pays out the money won by bidders, so the *auctioneer's payoff vector* is:

$$\boldsymbol{\Upsilon}_{\text{auc}} = \sum_{\boldsymbol{\Upsilon}_\mathbf{i}\in\mathcal{P}} -\alpha_i \boldsymbol{\Upsilon}_\mathbf{i}.$$

We also call the auctioneer's payoff vector the *surplus vector*, since it is the (possibly state-contingent) money left over after all accepted orders are filled.

Assume that the auctioneer wants to choose a set of orders so that it is guaranteed not to lose any money in any future state, but that the auctioneer does not necessarily insist on obtaining a positive benefit from the transaction (i.e., the auctioneer is content to break even).

Definition 1 *(Matching problem, indivisible case) Given a set of orders $\mathcal{O}$, does there exist $\alpha_i \in \{0,1\}$ with at least one $\alpha_i = 1$ such that*

$$\forall \omega, \Upsilon_{\text{auc}}^{\langle\omega\rangle} \geq 0?$$

In other words, does there exist a nonempty subset of orders that the auctioneer can accept without risk? □

If $\forall \omega, \Upsilon_{\text{auc}}^{\langle\omega\rangle} = c$ where c is nonnegative, then the surplus leftover after processing this match is c dollars. Let $m = \min_\omega[\Upsilon_{\text{auc}}^{\langle\omega\rangle}]$. In general, processing a match leaves m dollars in cash and $\Upsilon_{\text{auc}}^{\langle\omega\rangle} - m$ in state-contingent dollars, which can then be translated into securities.

Example 2 *(Indivisible order matching) Suppose $|\mathcal{A}| = 2$. Consider an order to buy one unit of $\langle A_1 A_2\rangle$ at price 0.4 and an order to sell one unit of $\langle A_1\rangle$ at price 0.3. The corresponding payoff vectors are:*

$$\begin{aligned}
\boldsymbol{\Upsilon_1} &= \langle \Upsilon_1^{\langle A_1 A_2\rangle}, \Upsilon_1^{\langle A_1 \bar{A}_2\rangle}, \Upsilon_1^{\langle \bar{A}_1 A_2\rangle}, \Upsilon_1^{\langle \bar{A}_1 \bar{A}_2\rangle} \rangle \\
&= \langle \quad 0.6, \quad -0.4, \quad -0.4, \quad -0.4 \rangle \\
\boldsymbol{\Upsilon_2} &= \langle \quad -0.7, \quad -0.7, \quad 0.3, \quad 0.3 \rangle
\end{aligned}$$

The auctioneer's payoff vector (the negative of the component-wise sum of the above two vectors) is:

$$\boldsymbol{\Upsilon}_{\text{auc}} = -\boldsymbol{\Upsilon_1} - \boldsymbol{\Upsilon_2} = \langle 0.1, 1.1, 0.1, 0.1 \rangle.$$

Since all components are nonnegative, the two orders match. The auctioneer can process both orders, leaving a surplus of \$0.1 in cash and one unit of $\langle A_1 \bar{A}_2\rangle$ in securities. □

Now consider the divisible case, where order can be partially filled.

Definition 2 *(Matching problem, divisible case) Given a set of orders $\mathcal{O}$, does there exist $\alpha_i \in [0,1]$ with at least one $\alpha_i > 0$ such that*

$$\forall \omega, \Upsilon_{\text{auc}}^{\langle\omega\rangle} \geq 0,$$

□

Example 3 *(Divisible order matching) Suppose $|\mathcal{A}| = 2$. Consider an order to sell one unit of $\langle A_1\rangle$ at price \$0.5, an order to buy one unit of $\langle A_1 A_2 | A_1 \vee A_2\rangle$*

at price \$0.5, and an order to buy one unit of $\langle A_1|\bar{A}_2\rangle$ *at price \$0.4. The corresponding payoff vectors are:*

$$\begin{aligned}
\boldsymbol{\Upsilon_1} &= \langle \Upsilon_1^{\langle A_1 A_2\rangle}, \Upsilon_1^{\langle A_1 \bar{A}_2\rangle}, \Upsilon_1^{\langle \bar{A}_1 A_2\rangle}, \Upsilon_1^{\langle \bar{A}_1 \bar{A}_2\rangle}\rangle \\
&= \langle \quad -0.5, \quad -0.5, \quad 0.5, \quad 0.5\rangle \\
\boldsymbol{\Upsilon_2} &= \langle \quad 0.5, \quad -0.5, \quad -0.5, \quad 0\rangle \\
\boldsymbol{\Upsilon_3} &= \langle \quad 0, \quad 0.6, \quad 0, \quad -0.4\rangle
\end{aligned}$$

It is clear by inspection that no non-empty subset of whole orders constitutes a match: in all cases where $\alpha_i \in \{0, 1\}$ *(other than all* $\alpha_i = 0$*), at least one state sums to a positive amount (negative for the auctioneer). However, if* $\alpha_1 = \alpha_2 = 3/5$ *and* $\alpha_3 = 1$*, then the auctioneer's payoff vector is:*

$$\boldsymbol{\Upsilon}_{\text{auc}} = -\frac{3}{5}\boldsymbol{\Upsilon_1} - \frac{3}{5}\boldsymbol{\Upsilon_2} - \boldsymbol{\Upsilon_3} = \langle 0, 0, 0, 0.1\rangle,$$

constituting a match. The auctioneer can process 3/5 of the first and second orders, and all of the third order, leaving a surplus of 0.1 units of $\langle \bar{A}_1 \bar{A}_2\rangle$*. In this example, a divisible match exists even though an indivisible match is not possible; we examine the distinction in detail in Section 5, where we separate the two matching problems into distinct complexity classes.* □

The matching problems defined above are decision problems: the task is only to show the existence or nonexistence of a match. However, there may be multiple matches from which the auctioneer can choose. Sometimes the choices are equivalent from the auctioneer's perspective; alternatively, an objective function can be used to find an optimal match according to that objective.

Example 4 *(Auctioneer alternatives I) Suppose* $|\mathcal{A}| = 2$*. Consider an order to sell one unit of* $\langle A_1\rangle$ *at price \$0.7, an order to sell one unit of* $\langle A_2\rangle$ *at price \$0.7, an order to buy one unit of* $\langle A_1 A_2\rangle$ *at price \$0.4, an order to buy one unit of* $\langle A_1 \bar{A}_2\rangle$ *at price \$0.4, and an order to buy one unit of* $\langle \bar{A}_1 A_2\rangle$ *at price \$0.4. The corresponding payoff vectors are:*

$$\begin{aligned}
\boldsymbol{\Upsilon_1} &= \langle -0.3, -0.3, \quad 0.7, \quad 0.7\rangle \\
\boldsymbol{\Upsilon_2} &= \langle -0.3, \quad 0.7, -0.3, \quad 0.7\rangle \\
\boldsymbol{\Upsilon_3} &= \langle \quad 0.6, -0.4, -0.4, -0.4\rangle \\
\boldsymbol{\Upsilon_4} &= \langle -0.4, \quad 0.6, -0.4, -0.4\rangle \\
\boldsymbol{\Upsilon_5} &= \langle -0.4, -0.4, \quad 0.6, -0.4\rangle
\end{aligned}$$

Consider the indivisible case. The auctioneer could choose to accept bids 1, 3, and 4 together, or the auctioneer could choose to accept bids 2, 3, and 5 together. Both constitute matches, and in fact both yield identical payoffs ($\boldsymbol{\Upsilon}_{\text{auc}} = \langle 0.1, 0.1, 0.1, 0.1 \rangle$, or \$0.1 in cash) for the auctioneer. □

Example 5 *(Auctioneer alternatives II) Suppose $|\mathcal{A}| = 2$. Consider an order to sell two units of $\langle A_1 \rangle$ at price \$0.6, an order to buy one unit of $\langle A_1 A_2 \rangle$ at price \$0.3, and an order to buy one unit of $\langle A_1 \bar{A}_2 \rangle$ at price \$0.5. The corresponding payoff vectors are:*

$$\boldsymbol{\Upsilon}_{\mathbf{1}} = \langle -0.4, -0.4, \ \ 0.6, \ \ 0.6 \rangle$$

$$\boldsymbol{\Upsilon}_{\mathbf{2}} = \langle \ \ 0.7, -0.3, -0.3, -0.3 \rangle$$

$$\boldsymbol{\Upsilon}_{\mathbf{3}} = \langle -0.5, \ \ 0.5, -0.5, -0.5 \rangle$$

Consider the divisible case. The auctioneer could choose to accept one unit each of all three bids, yielding a payoff to the auctioneer of \$0.2 in cash ($\boldsymbol{\Upsilon}_{\text{auc}} = \langle 0.2, 0.2, 0.2, 0.2 \rangle$). Alternatively, the auctioneer could choose to accept 4/3 units of bid 1, and one unit each of bids 2 and 3, yielding a payoff to the auctioneer of 1/3 units of security $\langle A_1 \rangle$. Both choices constitute matches (in fact, accepting any number of units of bid 1 between 1 and 4/3 can be part of a match), though depending on the auctioneer's objective, one choice might be preferred over another. For example, if the auctioneer believes that A_1 is very likely to occur, it may prefer to accept 4/3 units of bid 1. □

There are many possible criteria for the auctioneer to decide among matches, all of which seem reasonable in some circumstances. One natural quantity to maximize is the volume of trade among bidders; another is the auctioneer's utility, either with or without the arbitrage constraint.

Definition 3 *(Trade maximization problem) Given a set of indivisible (divisible) orders $\mathcal{O}$, choose $\alpha_i \in \{0, 1\}$ ($\alpha_i \in [0, 1]$) to maximize*

$$\sum_i \alpha_i q_i,$$

under the constraint that

$$\forall \omega, \Upsilon_{\text{auc}}^{\langle \omega \rangle} \geq 0.$$

□

Another reasonable variation is to maximize the total percent of orders filled, or $\sum_i \alpha_i$, under the same (risk-free) constraint that $\forall \omega, \Upsilon_{\text{auc}}^{\langle \omega \rangle} \geq 0$.

Definition 4 *(Auctioneer risk-free utility-maximization problem) Let the auctioneer's subjective probability for each state ω be $\Pr(\omega)$, and let the auctioneer's utility for y dollars be $u(y)$. Given a set of indivisible (divisible) orders $\mathcal{O}$, choose $\alpha_i \in \{0, 1\}$ ($\alpha_i \in [0, 1]$) to maximize*

$$\sum_{\omega \in \Omega} \Pr(\omega) u(\Upsilon_{\text{auc}}^{\langle \omega \rangle}),$$

under the constraint that

$$\forall \omega, \Upsilon_{\text{auc}}^{\langle \omega \rangle} \geq 0.$$

□

Definition 5 *(Auctioneer standard utility-maximization problem) Let the auctioneer's subjective probability for each state ω be $\Pr(\omega)$, and let the auctioneer's utility for y dollars be $u(y)$. Given a set of indivisible (divisible) orders $\mathcal{O}$, choose $\alpha_i \in \{0, 1\}$ ($\alpha_i \in [0, 1]$) to maximize*

$$\sum_{\omega \in \Omega} \Pr(\omega) u\left(\Upsilon_{\text{auc}}^{\langle \omega \rangle}\right).$$

□

This last objective function drops the risk-free (arbitrage) constraint. In this case, the auctioneer is a market maker with beliefs about the likelihood of outcomes, and the auctioneer may actually lose money is some outcomes.

Still other variations and other optimization criteria seem reasonable, including social welfare, etc. It also seems reasonable to suppose that the surplus be shared among bidders and the auctioneer, rather than retained solely by the auctioneer. This is analogous to choosing a common transaction price in a double auction (e.g., the midpoint between the bid and ask prices), rather than the buyer paying the bid price and the seller receiving the ask price, with the difference going to the auctioneer. The problem becomes more complicated when dividing surplus securities, in part because they are valued differently by different agents. Formulating reasonable sharing rules and examining the resulting incentive properties seems a rich and promising avenue for further investigation.

4 Matching algorithms

The straightforward algorithm for solving the divisible matching problem is linear programming; we set up an appropriate linear program in Section 5.1. The straightforward algorithm for solving the indivisible matching problem is integer programming. With n events, to set up the appropriate linear or integer programs, simply writing out the payoff vectors in the straightforward way requires $O(2^n)$ space.

There is some hope that specialized algorithms that exploit structure among bids can perform better in terms of average-case time and space complexity. For example, in some cases matches can be identified using logical reduction techniques, without writing down the full payoff vectors. So a match between the following bids:

- sell 1 of $\langle A_1 A_2 \rangle$ at \$0.2
- sell 1 of $\langle A_1 \bar{A}_2 \rangle$ at \$0.1
- buy 1 of $\langle A_1 \rangle$ at \$0.4

can be identified by reducing the first two bids to an equivalent offer to sell $\langle A_1 \rangle$ at \$0.3 that clearly matches with the third bid. Formalizing a logical-reduction algorithm for matching, or other algorithms that can exploit special structure among the bids, is a promising avenue for future work.

5 The computational complexity of matching

In this section we examine the computational complexity of the auctioneer's matching problem. Here n refers to the problem's input size that includes descriptions of all of the buy and sell orders. We also assume that n bounds the number of base securities.

We consider four cases based on two parameters:

(1) Whether to allow divisible or indivisible orders.
(2) The number of securities. We consider two possibilities:
   (a) $O(\log n)$ base securities yielding a polynomial number of states.
   (b) An unlimited number of base securities yielding an exponential number of states.

We show the following results.

Theorem 1 *The matching problem is*

(1) computable in polynomial-time for $O(\log n)$ base securities with divisible orders.
(2) co-NP-complete for unlimited securities with divisible orders.
(3) NP-complete for $O(\log n)$ base securities with indivisible orders.
(4) Σ_2^p-complete for unlimited securities with indivisible orders.

5.1 Small number of securities with divisible orders

We can build a linear program based on Definition 2. We have variables α_i. For each i, we have

$$0 \leq \alpha_i \leq 1$$

and for each state ω in Ω we have the constraint

$$\Upsilon_{\text{auc}}^{\langle\omega\rangle} = \sum_i -\alpha_i \Upsilon_i^{\langle\omega\rangle} \geq 0.$$

Given these constraints we maximize

$$\sum_i \alpha_i.$$

A set of orders has a matching exactly when $\sum_i \alpha_i > 0$.

With $O(\log n)$ base securities, we have $|\Omega| = O(n)$, so we can solve this linear program in polynomial time.

One could also maximize some linear combination of the $-\Upsilon_i^{\langle\omega\rangle}$s to maximize the surplus. Note however that this approach may not find matchings that have precisely zero surplus.

5.2 Large number of securities with divisible orders

With unlimited base securities, the linear program given in Section 5.1 has an exponential number of constraint equations. Each constraint is short to describe and easily computable given ω.

Let $m \leq n$ be the total number of buy and sell orders. By the theory of linear programming, an upper bound on the objective function can be forced by a collection of $m+1$ constraints. So if no matching exists there must exist $m+1$ constraints that force all the α_i to zero. In nondeterministic polynomial-time

we can guess these constraints and solve the reduced linear program. This shows that matching is in co-NP.

To show co-NP-completeness we reduce the NP-complete problem of Boolean formula satisfiability to the nonexistence of a matching. Fix a formula ϕ. Let the base securities be the variables of ϕ and consider the single security $\langle\phi\rangle$ with a buy order of 0.5. If the formula ϕ is satisfiable then there is some state with payoff 0.5 (auctioneer payoff -0.5) and no fractional unit of the security $\langle\phi\rangle$ is a matching. If the formula ϕ is not satisfiable then every state has an auctioneer's payoff of 0.5 and a single unit of the security $\langle\phi\rangle$ is a matching.

One could argue that if the formula ϕ is not satisfiable then no rational buyer would want to buy $\langle\phi\rangle$ for a cost of 0.5. We can get around this problem by adding auxiliary base securities, A and B, and defining two securities

$$\langle\tau\rangle = (\phi \wedge A) \vee (\overline{A} \wedge B)$$
$$\langle\tau'\rangle = (\phi \wedge A) \vee (\overline{A} \wedge \overline{B})$$

with separate buy orders of 0.5 on each.

If ϕ were satisfiable then in the state corresponding to the satisfying assignment and both A and B to be true, $\langle\tau\rangle$ and $\langle\tau'\rangle$ both have an auctioneer's payoff of -0.5 so no divisible matching can exist.

If ϕ were not satisfiable then one unit of each would be a matching since in every state at least one of $\langle\tau\rangle$ or $\langle\tau'\rangle$ are false.

5.3 Small number of securities with indivisible orders

This case is easily seen to be in NP: Just nondeterministically guess a nonempty subset S of orders and check for each state ω in Ω that

$$\Upsilon_{\text{auc}}^{\langle\omega\rangle} = \sum_{i \in S} -\Upsilon_i^{\langle\omega\rangle} \geq 0.$$

Since $|\Omega| = O(n)$ and $|S|$ is bounded by a polynomial in n, the verification can be done in polynomial time.

To show that matching is NP-complete we reduce the NP-complete problem EXACT COVER BY 3-SETS (X3C) to a matching of securities.

The input to X3C consists of a set X and a collection C of 3-element subsets of X. The input (X, C) is in X3C if C contains an exact cover of X, i.e., there

is a subcollection C' of C such that every element of X occurs in exactly one member of C'. Karp [35] showed that X3C is NP-complete.

Informally we will reduce X3C to matching by a set of states for the elements of X and a security $\langle\phi_i\rangle$ for each 3-element subset c_i in C. We set up a market such that a match can only be achieved by a set of securities that correspond to an exact cover. To achieve this we also need to add some extra states and securities to balance out the possibilities.

Suppose we have an instance (X, C) with the vector $X = \{x_1, \ldots, x_{3q}\}$ and $C = \{c_1, \ldots, c_\ell\}$.

We set $\Omega = \{e_1, \ldots, e_{3q}, r, s\}$ to be the underlying state space. Since $|\Omega| = O(n)$, we could in principle define a set of $O(\log n)$ base events whose power set spans Ω; however, we don't explicitly define the base events here. We define $O(n)$ compound securities—again, the $O(\log n)$ set of generating base securities is implicit—labeled $\langle\phi_1\rangle, \ldots, \langle\phi_\ell\rangle$, $\langle\psi_1\rangle, \ldots, \langle\psi_q\rangle$ and $\langle\tau\rangle$, as follows:

- Security $\langle\phi_i\rangle$ is true in state r, and is true in state e_k if x_k is not in c_i.
- Security $\langle\psi_j\rangle$ is true only in state s.
- Security $\langle\tau\rangle$ is true in each state e_k but not r or s.

We have buy orders on each $\langle\phi_i\rangle$ and $\langle\psi_j\rangle$ security for $0.5 - \frac{1}{8q}$ and a buy order on $\langle\tau\rangle$ for 0.5.

We claim that a matching exists if and only if (X, C) is in X3C.

If (X, C) is in X3C, let C' be the subcollection that covers each element of X exactly once. Note that $|C'| = q$.

We claim the collection consisting of $\langle\phi_i\rangle$ for each c_i in C', every $\langle\psi_j\rangle$ and $\langle\tau\rangle$ has a matching.

In each state e_k we have an auctioneer's payoff of

$$\begin{aligned}(.5 - \frac{1}{8q}) + (q-1)(-.5 - \frac{1}{8q}) + q(.5 - \frac{1}{8q}) - .5 \\ = .5 - 2q\frac{1}{8q} = .25 \geq 0.\end{aligned}$$

In states r and s the auctioneer's payoffs are

$$-q(.5 + \frac{1}{8q}) + q(.5 - \frac{1}{8q}) + .5 = .5 - 2q\frac{1}{8q} = .25 \geq 0.$$

Suppose now that (X, C) is not in X3C but there is a matching. Consider the number q' of the $\langle \phi_i \rangle$ in that matching and q'' the number of $\langle \psi_j \rangle$ in the matching. Since a matching requires a nonempty subset of the orders and $\langle \tau \rangle$ by itself is not a matching we have $q' + q'' > 0$.

We have three cases.

$q' > q$: In state r, the auctioneer's payoff is

$$-q'(.5 + \frac{1}{8q}) - q(-.5 + \frac{1}{8q}) + .5 \leq -(q' + q)\frac{1}{8q} < 0.$$

$q'' > q'$: In state s, the auctioneer's payoff is

$$-q''(.5 + \frac{1}{8q}) - q'(-.5 + \frac{1}{8q}) + .5 \leq -(q'' + q')\frac{1}{8q} < 0.$$

$q'' \leq q' \leq q$: Consider the set C' consisting of the c_i where $\langle \phi_i \rangle$ is in the matching. There must be some state e_k not in any of the c_i or C' would be an exact cover. The auctioneer's payoff in e_k is at most

$$-q'(.5 + \frac{1}{8q}) - q''(-.5 + \frac{1}{8q}) \leq -(q'' + q')\frac{1}{8q} < 0.$$

5.4 Large Number of Securities with Indivisible Orders

For the case of $O(n)$ base securities and indivisible orders, we will show that computing a matching is Σ_2^p-complete, remaining so even for quite restricted types of securities, and hence is (likely) harder than any problem in NP or co-NP. While it may seem that being NP-complete or co-NP-complete is "hard enough", there are certain practical consequences of being outside of NP and co-NP. If the matching problem were in NP, one could use heuristics to search for and verify a match if it exists; even if such heuristics fail in the worst case, they may succeed for most examples in practice. Similarly, if the matching problem were in co-NP, one might hope to at least heuristically rule out the possibility of matching. But for problems outside of NP or co-NP, there is no framework for verifying that a heuristically derived answer is correct. Less formally, for NP (or co-NP)-complete problems, you have to be lucky; for Σ_2^p-complete problems, you can't even tell if you've been lucky.

We note that the existence of a matching is in Σ_2^p. The class Σ_2^p is the second level of the *polynomial-time hierarchy*. One can view NP problems like satisfiability as an existential question, showing there exists a satisfying assignment. Co-NP problems like tautology ask universal questions, do all assignments

make the formula true? The class Σ_2^p has an existential and a universal quantifier, in that order. Showing a matching exists has this flavor: there exists a subset of the securities such that for all future states the market maker does not lose money.

Formally, a language L is in Σ_2^p if there exists a polynomial p and a polynomial-time computable set A such that x is in L if and only if there is a y with $|y| = p(|x|)$ such that for all z, with $|z| = p(|x|)$, (x, y, z) is in A.

To show that the matching problem is in Σ_2^p, we use y to choose a subset of the orders and z to represent a state ω, with (x, y, z) in A if the set of orders has a total nonpositive auctioneer's payoff in state ω.

We prove a stronger theorem which implies that matching is Σ_2^p-complete. Let $\langle\phi_1\rangle, \ldots, \langle\phi_n\rangle$ be a set of securities. We discuss the problem from the perspective of the auctioneer as a *buyer* deciding from among all bids which to accept or buy. In relation to the auctioneer, and in this section only, we use the words "buy" and "accept" interchangeably, and we refer to the auctioneer as "the buyer". All payoffs and costs quoted in this section are from the auctioneer's point of view. Security $\langle\phi_i\rangle$ pays off \$1 (or more generally, $\$p_i$) when ϕ_i is satisfied, assuming it is a sell offer that is accepted (opposite but symmetric to the bidder's payoff). The cost to the auctioneer for buying a security is denoted c_i (again, opposite but symmetric to the bidder's costs). The $0-1$-*matching problem* asks whether the auctioneer can, by accepting either 0 or 1 of each security, guarantee a worst-case payoff strictly larger than the total cost. For convenience, in some places in this section we will denote securities by S rather than $\langle\phi\rangle$.

Theorem 2 *The $0-1$-matching problem is Σ_2^p-complete. Furthermore, the problem remains Σ_2^p-complete under the following two special cases:*

(1) For all i, ϕ_i is a conjunction of 3 base events (or their negations), $p_i = 1$, and $c_i = c_j$ for all i and j.
(2) For all i, ϕ_i is a conjunction of at most 2 base securities (or their negations).

These hardness results hold even if there is a promise that no subset of the securities guarantees a worst-case payoff identical to their cost.

To prove Theorem 2, we reduce from the "standard" Σ_2^p problem that we call T$\exists\forall$BF. Given a boolean formula f with variables $x_1, \ldots, x_n$ and $y_1, \ldots, y_n$ is the following fully-quantified formula true

$$\exists x_1 \ldots \exists x_n \forall y_1 \ldots \forall y_n \; f(x_1, \ldots, x_n, y_1, \ldots, y_n)?$$

The problem remains Σ_2^p-complete when

$$f(x_1, \ldots, x_n, y_1, \ldots, y_n)$$

is restricted to being a disjunction of conjunctions of at most 3 variables (or their negations), e.g.,

$$f(x_1, \ldots, x_n, y_1, \ldots, y_n) = \\ (x_1 \wedge \bar{x}_3 \wedge y_2) \vee (x_2 \wedge y_3 \wedge y_7) \vee \cdots.$$

This form, without the bound on the conjunction size, is known as *disjunctive normal form* (DNF); the restriction to conjunctions of 3 variables is 3-DNF.

We reduce T∃∀BF to finding a matching. For the simplest reduction, we consider the matching problem where one has a set of Arrow-Debreu securities whose payoff events are conjunctions of the base events or their negations. The auctioneer has the option of accepting either 0 or 1 of each of the given securities.

We first reduce to the case where the payoff events are conjunctions of arbitrarily many base events (or their negations). By a standard trick we can reduce the number of base events in each conjunction to 3, and with a slight twist we can even ensure that all securities have the same price as well as the same payoff. Finally, we show that the problem remains hard even if only conjunctions of 2 variables are allowed, though with securities that deviate slightly from Arrow-Debreu securities in that they may have varying, non unit payoffs.

5.4.1 The basic reduction

Before describing the securities, we give some intuition. The T∃∀BFproblem may be viewed as a game between a *selector* and an *adversary*. The selector sets the x_i variables, and then the adversary sets the y_i variables so as to falsify the formula f. We can view the $0-1$-matching problem as one in which the auctioneer is a *buyer* who buys securities corresponding to disjunctions of the base events, and then the adversary sets the values of the base events to minimize the payoff from the securities.

We construct our securities so that the optimal buying strategy is to buy n "expensive" securities along with a set of "cheap" securities, of negligible cost (for some cases we can modify the construction so that all securities have the same cost). The total cost of the securities will be just under 1, and each security pays off 1, so the adversary must ensure that none of the securities pays off. Each expensive security forces the adversary to set some variable, x_i

to a particular value to prevent the security from paying off; this corresponds to setting the x_i variables in the original game. The cheap securities are such that preventing every one of of these securities from paying off is equivalent to falsifying f in the original game.

Among the technical difficulties we face is how to prevent the buyer from buying conflicting securities, e.g., one that forces $x_i = 0$ and the other that forces $x_i = 1$, allowing for a trivial arbitrage. Secondly, for our analysis we need to ensure that a trader cannot spend more to get more, say by spending 1.5 for a set of securities with the property that at least 2 securities pay off under all possible events.

For each of the variables $\{x_i\}, \{y_i\}$ in f, we add a corresponding base security (with the same labels). For each existential variable x_i we add additional base securities, n_i and z_i. We also include a base security Q.

In our basic construction, each expensive security costs C and each cheap security costs ϵ; all securities pay off 1. We require that $Cn+\epsilon(|\text{cheap securities}|) < 1$ and $C(n+1) > 1$. That is, one can buy n expensive securities and all of the cheap securities for less than 1, but one cannot buy $n+1$ expensive securities for less than 1. We at times refer to a security by its payoff clause.

Remark: We may loosely think of ϵ as 0. However, this would allow one to buy a security for nothing that pays (in the worst case) nothing. By making $\epsilon > 0$, we can show it hard to distinguish portfolios that guarantee a positive profit from those that risk a positive loss. Setting $\epsilon > 0$ will also allow us to show hardness results for the case where all securities have the same cost.

For $1 \leq i \leq n$, we have two expensive securities with payoff clauses $(\bar{x}_i \wedge Q)$ and $(\bar{n}_i \wedge Q)$ and two cheap securities with payoff clauses $(x_i \wedge \bar{z}_i)$ and $(n_i \wedge \bar{z}_i)$.

For each clause $\mathcal{C} \in f$, we convert every negated variable $\bar{x}_i$ into n_i and add the conjunction $z_1 \wedge \cdots \wedge z_n$. Thus, for a clause $\mathcal{C} = (x_2 \wedge \bar{x}_7 \wedge \bar{y}_5)$ we construct a cheap security with payoff clause

$$(z_1 \wedge \cdots \wedge z_n \wedge x_2 \wedge n_7 \wedge \bar{y}_5). \tag{1}$$

Finally, we have a cheap security with payoff clause $(\bar{Q})$.

We now argue that a matching exists iff

$$\exists x_1 \ldots \exists x_n \forall y_1 \ldots \forall y_n \; f(x_1, \ldots, x_n, y_1, \ldots, y_n).$$

We do this by successively constraining the buyer and the adversary, elim-

inating behaviors that would cause the other player to win. The resulting "reasonable" strategies correspond exactly to the game version of T∃∀BF.

First, observe that if the adversary sets all of the base securities to false (0), then only the $(\bar{Q})$ security will pay off. Thus, no buyer can buy more than n expensive securities and guarantee a profit. The problem is thus whether one can buy n expensive securities and all the cheap securities, so that at for any setting of the base events at least one security will pay off.

Clearly, the adversary must make Q hold, or the $(\bar{Q})$ security will pay off. Next, we claim that for each i, $1 \leq i \leq n$, the auctioneer must buy at least one of the $(\bar{x}_i \wedge Q)$ and $(\bar{n}_i \wedge Q)$ securities. This follow from the fact that if the adversary sets x_i, n_i and z_i to be false, and every other base event to be true, then only the $(\bar{x}_i \wedge Q)$ and $(\bar{n}_i \wedge Q)$ securities will pay off. As no auctioneer can buy more than n expensive securities, it must therefore buy exactly one of $(\bar{x}_i \wedge Q)$ or $(\bar{n}_i \wedge Q)$, for each i, $1 \leq i \leq n$. For the rest of the analysis, we assume that the auctioneer follows this constraint.

Suppose that the buyer buys $(\bar{x}_i \wedge Q)$. Then the adversary must set x_i to be true (since it must set Q to be true), or the security will pay off. It must then set z_i to be true or $(x_i \wedge \bar{z}_i)$ will pay off. Since the buyer doesn't buy $(\bar{n}_i \wedge Q)$ (by the above constraint), and all the other securities pay the same or less when n_i is made false, we can assume without loss of generality that the adversary sets n_i to be false. Similarly, if the buyer buys $(\bar{n}_i \wedge Q)$, then the adversary must set n_i and z_i to be true, and we can assume without loss of generality that the adversary sets x_i to be false. Note that the adversary must in all cases set each z_i event to be true.

Summarizing the preceding argument, there is an exact correspondence between the rational strategies of the buyer and settings for the x_i variables forced on the adversary. Furthermore, the adversary is also constrained to set the variables $Q, z_1, \ldots, z_n$ to be true, and without loss of generality may be assumed to set $n_i = \bar{x}_i$. Under these constraints, those securities not corresponding to clauses in f are guaranteed to not pay off.

The adversary also decides the value of the $y_1, \ldots, y_m$ base events. Recall that for each clause $\mathcal{C} \in f$ there is a corresponding security constructed as in (1). Given that z_i is true and $n_i = \bar{x}_i$ (without loss of generality), it follows from the security's construction that the setting of the y_is causes the security to pay off iff it satisfies C. This establishes the reduction from T∃∀BF to the matching problem, when the securities are constrained to be a conjunction of polynomially many base events or their negations.

5.4.2 Reducing to 3-variable conjunctions

There are standard methods for reducing DNF formulae to 3-DNF formulae, which are trivially modifiable to our securities framework; we include the reduction for completeness. Given a security $\langle \mathcal{C} \rangle$ whose payoff clause is

$$\mathcal{C} = (v_1 \wedge v_2 \wedge \cdots \wedge v_k)$$

(variable negations are irrelevant to this discussion), cost c and payoff p, introduce a new auxiliary variable, w, and replace the security with two securities, $\langle \mathcal{C}_1 \rangle$ and $\langle \mathcal{C}_1 \rangle$, with payoff clauses,

$$\mathcal{C}_1 = (v_1 \wedge v_2 \wedge w) \text{ and}$$
$$\mathcal{C}_2 = (\bar{w} \wedge v_3 \wedge \cdots \wedge v_k).$$

The securities both have payoff p, and their costs can be any positive values that sum to c. Note that at most one of the securities can pay off at a time. If only one security is bought, then the adversary can always set w so that it won't pay off; hence the auctioneer will buy either both or neither, for a total cost of c (here we use the fact that one is only allowed to buy either 0 or 1 shares of each security). Then, it may be verified that, given the ability to set w arbitrarily, the adversary can cause $\mathcal{C}$ to be unsatisfied iff it can cause both $\mathcal{C}_1$ and $\mathcal{C}_2$ to be unsatisfied. Hence, owning one share each of $\langle \mathcal{C}_1 \rangle$ and $\langle \mathcal{C}_2 \rangle$ is equivalent to owning one share of $\langle \mathcal{C} \rangle$.

Note that $\mathcal{C}_1$ has three variables and $\mathcal{C}_2$ has $k - 1$ variables. By applying the transformation successively, one obtains an equivalent set of securities, of polynomial size, whose payoff clauses have at most 3 variables.

We note that in the basic construction, all of the clauses with more than 3 variables are associated with cheap securities (cost ϵ). Instead of subdividing costs, we can simply make all of the resulting securities have cost ϵ; the constraints on C and ϵ must reflect the new, larger number of cheap securities.

One can ensure that all of the payoff clauses have exactly 3 variables, with a similar construction. A security $\mathcal{S}$ with cost c, payoff p and defining clause $(x \wedge y)$ can be replaced by securities $\mathcal{S}_1$ and $\mathcal{S}_2$ with cost $c/2$, payoff p and defining clauses $(x \wedge y \wedge w)$ and $(x \wedge y \wedge \bar{w})$, where w is a new auxiliary variable. Essentially the same analysis as given above applies to this case. The case of single-variable payoff clauses is handled by two applications of this technique.

5.4.3 Reducing to equi-cost securities

By setting C and ϵ appropriately, one can ensure that in the basic reduction every security costs a polynomially bounded integer multiple of ϵ; call this ratio r. We now show how to reduce this case to the case where every security costs ϵ. Recall that the expensive securities have payoff clauses $(\bar{x}_i \wedge Q)$ or $(\bar{n}_i \wedge Q)$. Assume that security S has payoff clause $(\bar{x}_i \wedge Q)$ (the other case is handled identically). Replace S with security S', with payoff clause $(\bar{x}_i \wedge Q \wedge w_1)$ ($w_1, \ldots, w_{r-1}$ are auxiliary variables; fresh variables are chosen for each clause), and also $S_1, \ldots, S_{r-1}$, with payoff clauses

$$(\bar{w}_1 \wedge w_2), (\bar{w}_2 \wedge w_3), \ldots, (\bar{w}_{r-2} \wedge w_{r-1}), \text{ and} (\bar{w}_{r-1} \wedge \bar{w}_1).$$

Clearly, buying none of the new securities is equivalent to not buying the original security. We show that buying all of the new securities is equivalent to buying the original security, and that buying a proper, nonempty subset of the securities is irrational.

We first note that if the auctioneer buys securities $S_1, \ldots, S_{r-1}$, then the adversary must set w_1 to be true, or one of the securities will pay off. To see this, note that if w_i is set to false, then $(\bar{w}_i \wedge w_{i+1})$ will be true unless w_{i+1} is set to false; thus, setting w_1 to false forces the adversary to set w_{r-1} to false, causing the final clause to be true. Having set w_1 true, the adversary can set $w_2, \ldots, w_{r-1}$ to false, ensuring that none of the securities $S_1, \ldots, S_{r-1}$ pays out. If w_i is true, then $(\bar{x}_i \wedge Q \wedge w_1)$ is equivalent to $(\bar{x}_i \wedge Q)$. So buying all of the replacement securities for ϵ each is equivalent to buying the original security for ϵr.

It remains to show that buying a proper, nonempty subset of the securities is irrational. If one doesn't buy S', then the adversary can set the w variables so that none of $S_1, \ldots, S_{r-1}$ will pay off; any money spent on these securities is wasted. If one doesn't buy S_{r-1}, the adversary can set all w to false, in which case none of the new clauses will pay off, regardless of the value of x_i and Q. Similarly, if one doesn't buy S_i, for $1 \leq i \leq r-2$, the adversary can set w_{i+1} to be true, all the other w variables to be false, and again there is no payoff, regardless of the value of x_i and Q. Thus, buying a proper subset of these securities will not increase ones payoff.

We note that this reduction can be combined trivially with the reduction that ensures that all of the defining clauses have 3 or fewer variables. With a slightly messier argument, all of the defining clauses can be set up to have exactly 3 variables.

5.4.4 Reducing to clauses of at most 2 variables

If we allow securities to have variable payoffs and prices, we can reduce to the case where each security's payoff clause is a conjunction of at most 2 variables or their negations.

Given a security s with payoff clause $(X \wedge Y \wedge Z)$, cost c and payoff 1, we introduce fresh auxiliary variables, w_1, w_2 and w_3 (new variables are used for each clause) and replace S with the following securities:

- Securities S_1, S_2 and S_3, each with cost $c/3$ and payoff 1, with respective payoff clauses $(X \wedge w_1)$, $(Y \wedge w_2)$ and $(Z \wedge w_3)$.
- Securities $S'_1, \ldots, S'_6$, each with cost 4 and payoff $24-\epsilon_2$, with payoff clauses,

$$(w_1 \wedge w_2)\ (w_1 \wedge w_3)\ (w_2 \wedge w_3)$$
$$(\bar{w}_1 \wedge \bar{w}_2)\ (\bar{w}_1 \wedge \bar{w}_3)\ (\bar{w}_2 \wedge \bar{w}_3)$$

Here, ϵ_2 is a tiny positive quantity, described later. By a simple case analysis, we have the following.

Observations:

(1) For any i, there exists a setting of w_1, w_2 an w_3 such that of the S' securities only S'_i pays off.
(2) For any setting of w_1, w_2 and w_3, at least one of the S' securities will pay off.
(3) If w_1, w_2 and w_3 are all false, all of the S' securities will pay off.
(4) Setting one of w_1, w_2 or w_3 to be true, and the others to be 0, will cause exactly one of the S' securities to pay off.

By Observation 1, there is no point in buying a nonempty proper subset of the S' securities: The adversary can ensure that none of the bought securities will pay off, and even if all the S securities pay off, it will not be sufficient to recoup the cost of buying a single S' security. By Observation 2, if one buys all the S' securities, one is guaranteed to almost make back ones investment (except for ϵ_2), in which case by Observations 3 and 4, the adversaries optimal strategy is to make exactly one of w_1, w_2 or w_3 true. We set C, ϵ and ϵ_2 so that

$$Cn + \epsilon(|\text{cheap securities}|) + \epsilon_2(|\text{clauses}|) < 1.$$

Thus, the accumulated losses of ϵ_2 can never spell the difference between making a guaranteed profit and making no profit at all. Note also that by making ϵ_2 positive we prevent the existence of "break-even" buying strategies in which the buyer only purchases S' securities.

Summarizing the previous argument, we may assume without loss of generality that the buyer buys all of the S' securities (for all clauses), and that for each clause the adversary sets exactly one of that clause's auxiliary variables w_1, w_2 or w_3 to be true. For the rest of the discussion, we assume that the players follow these constraints.

We next claim that a rational buyer will either buy all of S_1, S_2 or S_3, or none of them. If the buyer doesn't buy S_1, then if the adversary makes w_1 true and w_2 and w_3 false, neither S_2 nor S_3 will pay off, regardless of how the adversary sets X, Y and Z. Hence, there is no point in buying either S_2 or S_3 if one doesn't buy S_1. Applying the same argument to S_2 and S_3 establishes the claim.

Clearly, buying none of S_1, S_2 and S_3 has, up to negligible ϵ_2 factors, the same price/payoff behavior as not buying S. We next argue that, subject to the established constraints put on the players' behaviors, buying all of S_1, S_2 and S_3 has the same price/payoff behavior (again ignoring ϵ_2 factors) as buying S, regardless of how the adversary sets X, Y and Z. First, in both cases, the cost is c. If the adversary makes X, Y and Z true, then S pays off 1, and (assuming that exactly one of w_1, w_2 and w_3 is true), exactly one of S_1, S_2 or S_3 will pay off 1. If X is false, then S doesn't pay off, and the adversary can set w_1 true (and w_2 and w_3 false), ensuring that none of S_1, S_2 and S_3 pays off. The same argument holds if Y or Z are false.

6 Tractable Cases

The logical question to ask in light of these complexity results is whether further, more severe restrictions on the space of securities can enable tractable matching algorithms. Although we have not systematically explored the possibilities, the potential for useful tractable cases certainly exists.

Suppose, for example, that bids are limited to unit quantities of securities of the following two forms:

(1) Disjunctions of positive events: $\langle A_1 \vee \cdots \vee A_k \rangle$.
(2) Single negative events: $\langle \bar{A}_i \rangle$.

Let p be the price offered for a disjunction $\langle A_1 \vee \cdots \vee A_k \rangle$, and q_i the maximal price offered for the respective negated disjuncts. This disjunction bid is part of a match iff $p + \sum_i q_i \geq k$. Evaluating whether this condition is satisfied by a subset of bids is quite straightforward.

Although this example is contrived, its application is not entirely implausible.

For example, the disjunctions may correspond to insurance customers, who want an insurance contract to cover all the potential causes of their asset loss. The atomic securities are sold by insurers, each of whom specialize in a different form of disaster cause.

7 Conclusions and future directions

We have analyzed the computational complexity of matching for securities based on logical formulas. Many possible avenues for future work exist, including

(1) Analyzing the agents' optimization problem:
   - How to choose quantities and bid/ask prices for a collection of securities to maximizes one's expected utility, both for linear and nonlinear utility functions.
   - How to choose securities; that is, deciding on what collection of boolean formulas to offer to trade, subject to constraints or penalties on the number or complexity of bids.
   - How to make the above choices in a game theoretically sound way, taking into account the choices of other traders, their reasoning about other traders, etc.

(2) Although matching is likely intractable, are there good heuristics that achieve matches in many cases or approximate a matching?

(3) Exploring sharing rules for dividing the surplus, and incentive properties of the resulting mechanisms.

(4) Study the incremental problem of finding a matching between a single new order and a set of old orders known to have no matches between them. The objective function would be to satisfy as much of the new order as possible, giving the advantage of any price differences to the new order. (This is the standard double auction rule.)

(5) We may consider a market to be in *computational equilibrium* if no computationally-bounded player can find a strategy that increases utility. With few exceptions [36,37], little is known about computational equilibriums. A natural question is to determine whether a market can achieve a computational equilibrium that is not a true equilibrium, and under what circumstances this may occur.

Acknowledgments

We thank Rahul Sami for his help with Section 5.4.4. We thank Rahul, Joan Feigenbaum and Robin Hanson for useful discussions and the anonymous ref-

erees for many helpful comments on the presentation and for the question on the incremental problem.

References

[1] J. C. Jackwerth, M. Rubinstein, Recovering probability distributions from options prices, Journal of Finance 51 (5) (1996) 1611–1631.

[2] R. Roll, Orange juice and weather, American Economic Review 74 (5) (1984) 861–880.

[3] R. Forsythe, F. Nelson, G. R. Neumann, J. Wright, Anatomy of an experimental political stock market, American Economic Review 82 (5) (1992) 1142–1161.

[4] R. Forsythe, T. A. Rietz, T. W. Ross, Wishes, expectations, and actions: A survey on price formation in election stock markets, Journal of Economic Behavior and Organization 39 (1999) 83–110.

[5] S. Debnath, D. M. Pennock, C. L. Giles, S. Lawrence, Information incorporation in online in-game sports betting markets, in: Fourth ACM Conference on Electronic Commerce (EC'03), 2003, pp. 258–259.

[6] J. M. Gandar, W. H. Dare, C. R. Brown, R. A. Zuber, Informed traders and price variations in the betting market for professional basketball games, Journal of Finance LIII (1) (1998) 385–401.

[7] C. Schmidt, A. Werwatz, How accurate do markets predict the outcome of an event? the Euro 2000 soccer championships experiment, Tech. Rep. 09-2002, Max Planck Institute for Research into Economic Systems (2002).

[8] R. H. Thaler, W. T. Ziemba, Anomalies: Parimutuel betting markets: Racetracks and lotteries, Journal of Economic Perspectives 2 (2) (1988) 161–174.

[9] R. D. Hanson, Could gambling save science? Encouraging an honest consensus, Social Epistemology 9 (1) (1995) 3–33.

[10] R. Hanson, Decision markets, IEEE Intelligent Systems 14 (3) (1999) 16–19.

[11] K.-Y. Chen, L. R. Fine, B. A. Huberman, Forecasting uncertain events with small groups, in: Third ACM Conference on Electronic Commerce (EC'01), 2001, pp. 58–64.

[12] D. M. Pennock, S. Lawrence, C. L. Giles, F. Å. Nielsen, The real power of artificial markets, Science 291 (2001) 987–988.

[13] D. M. Pennock, S. Lawrence, F. Å. Nielsen, C. L. Giles, Extracting collective probabilistic forecasts from web games, in: Seventh International Conference on Knowledge Discovery and Data Mining, 2001, pp. 174–183.

[14] R. Forsythe, R. Lundholm, Information aggregation in an experimental market, Econometrica 58 (2) (1990) 309–347.

[15] C. R. Plott, S. Sunder, Rational expectations and the aggregation of diverse information in laboratory security markets, Econometrica 56 (5) (1988) 1085–1118.

[16] C. R. Plott, J. Wit, W. C. Yang, Parimutuel betting markets as information aggregation devices: Experimental results, Tech. Rep. Social Science Working Paper 986, California Institute of Technology (Apr. 1997).

[17] C. R. Plott, Markets as information gathering tools, Southern Economic Journal 67 (1) (2000) 1–15.

[18] K. J. Arrow, The role of securities in the optimal allocation of risk-bearing, Review of Economic Studies 31 (2) (1964) 91–96.

[19] J. H. Dreze, Market allocation under uncertainty, in: Essays on Economic Decisions under Uncertainty, Cambridge University Press, 1987, pp. 119–143.

[20] A. Mas-Colell, M. D. Whinston, J. R. Green, Microeconomic Theory, Oxford University Press, New York, 1995.

[21] B. de Finetti, Theory of Probability: A Critical Introductory Treatment, Vol. 1, Wiley, New York, 1974.

[22] J. B. Kadane, R. L. Winkler, Separating probability elicitation from utilities, Journal of the American Statistical Association 83 (402) (1988) 357–363.

[23] M. Magill, M. Quinzii, Theory of Incomplete Markets, Vol. 1, MIT Press, 1996.

[24] G. Cooper, The computational complexity of probabilistic inference using Bayes belief networks, Artificial Intelligence 42 (1990) 393–405.

[25] H. R. Varian, The arbitrage principle in financial economics, J. Economic Perspectives 1 (2) (1987) 55–72.

[26] D. M. Pennock, M. P. Wellman, Compact securities markets for Pareto optimal reallocation of risk, in: Sixteenth Conference on Uncertainty in Artificial Intelligence, 2000, pp. 481–488.

[27] J. Pearl, Probabilistic Reasoning in Intelligent Systems, Morgan Kaufmann, 1988.

[28] P. Bossaerts, L. Fine, J. Ledyard, Inducing liquidity in thin financial markets through combined-value trading mechanisms, European Economic Review 46 (2002) 1671–1695.
URL `citeseer.nj.nec.com/bossaerts00inducing.html`

[29] S. de Vries, R. V. Vohra, Combinatorial auctions: A survey, INFORMS J. of Computing.
URL `citeseer.nj.nec.com/devries01combinatorial.html`

[30] N. Nisan, Bidding and allocation in combinatorial auctions, in: Second ACM Conference on Electronic Commerce (EC'00), 2000, pp. 1–12.

[31] T. Sandholm, S. Suri, A. Gilpin, D. Levine, Winner determination in combinatorial auction generalizations, in: First International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS), 2002.

[32] R. Hanson, Combinatorial information market design, Information Systems Frontiers 5 (1).

[33] R. L. Winkler, A. H. Murphy, Good probability assessors, J. Applied Meteorology 7 (1968) 751–758.

[34] R. Fagin, J. Y. Halpern, N. Megiddo, A logic for reasoning about probabilities, Information and Computation 87 (1/2) (1990) 78–128.
URL citeseer.nj.nec.com/fagin90logic.html

[35] R. Karp, Reducibility among combinatorial problems, in: R. E. Miller, J. W. Thatcher (Eds.), Complexity of Computer Computations, Plenum Press, 1972, pp. 85–103.

[36] P. J. Brewer, Decentralized computation procurement and computational robustness in a smart market, Economic Theory 13 (1999) 41–92.

[37] N. Nisan, A. Ronen, Computationally feasible VCG mechanisms, in: Second ACM Conference on Electronic Commerce (EC'00), 2000, pp. 242–252.
URL citeseer.nj.nec.com/nisan00computationally.html