University of Toronto
Department of Economics

Working Paper 484

Implementation with Securities

By Rahul Deb and Debasis Mishra

April 22, 2013

Implementation with Securities [*]

Rahul Deb [†] and Debasis Mishra [‡]

April 15, 2013

Abstract

We study mechanism design in a setting where agents know their types but are uncertain about the utility from any alternative. The final realized utility of each agent is observed by the principal and can be contracted upon. In such environments, the principal is not restricted to using only transfers but can employ *security* contracts which determine each agent's payoff as a function of their realized utility and the profile of announced types. We show that using security contracts instead of transfers expands the set of (dominant strategy) implementable social choice functions. Our main result is that in a finite type space, every social choice function that can be implemented using a security contract can also be implemented using a *royalty* contract. Royalty contracts are simpler and commonly used security contracts, in which agents initially pay a transfer and keep a fraction of their realized utility. We also identify a condition called *acyclicity* that is necessary and sufficient for implementation in these environments.

JEL Classification Codes: D44, D47, D71, D82, D86

Keywords: dominant strategy implementation, acyclicity, security contracts, royalty contracts, cycle monotonicity

[*] We are grateful to Juan Carlos Carbajal, Arunava Sen, Andrzej Skrzypacz, Rakesh Vohra and seminar audiences at the 2012 Midwest Economic Theory Meeting, University of Toronto, Indian Statistical Institute for valuable suggestions and feedback.

[†] University of Toronto. Email: `rahul.deb@utoronto.ca`

[‡] Indian Statistical Institute. Email: `dmishra@isid.ac.in`

1 Introduction

The classic setting in mechanism design with quasi-linear utilities is the following. Agents privately observe their type and make reports to the mechanism designer. Based on these reports, the mechanism designer chooses an alternative and transfer amounts. Agents then realize their utility from the chosen alternative and their final payoff is this utility less their transfer amount. We refer to such mechanisms as quasilinear mechanisms. An important aspect of this setting is that the mechanism is a function only of the reports and not of the realized utilities of the agents. This could either be because the principal cannot observe these utilities or that they are not verifiable by third parties and hence contracts based on them cannot be enforced.

However, in many practical settings principals can and do offer contracts which are functions of both the agents' reports and their realized utilities. Consider the example of an author selling the publishing rights of his upcoming book. Here, the publisher (the principal) chooses the terms of the contract and an amount to invest in research, publicity and marketing for the author's work (the alternative). The revenue from sales is uncertain but it depends on the quality of the author's writing (the private type) and the investment by the publishing house. Contracts often consist of an advance payment to the author and, in return, the publisher gets to keep a percentage of the revenue from future sales of the book. The terms (the advance and the royalty rate) depend on the author's reported type but the final payoffs from the contract to both parties depend on the true type and chosen alternative. Such contracts are ubiquitous, and settings where they are used include musicians seeking record labels, entrepreneurs selling their firms to acquirers or soliciting venture capital, and sports associations selling broadcasting rights. In addition, auctions are often conducted in which buyers bid using such contracts as opposed to simply making cash bids. Examples include the sale of private companies and divisions of public companies, government sales of oil leases, wireless spectrum and highway building contracts.

In these settings, the payoffs to both the principal and the agents from the mechanism differ from the classic quasilinear setting in two important respects. Firstly, at the interim stage (after realizing the type but before the alternative is chosen), agents are uncertain about the utility they will get from any alternative. Secondly, the ex-post utility from the chosen alternative is observable to both the agents and the principal, and the payoffs from the mechanism to both can depend on this realized utility.

In this paper, we study the problem of dominant strategy implementation in such a general environment. With this implementation criterion, it is not necessary to assume that either the principal or the agents have prior beliefs over the types of all agents. A mechanism in this context consists of a social choice function (scf) and a security contract which determines each agent's payoff as a function of their realized utility and the profile of announced types. An important (and commonly used) example of a security contract is a royalty contract, where (as a function of the reports) the mechanism designer specifies an upfront transfer and the percentage of the realized utility that is awarded to the agents. We say that an scf is implementable using a security (royalty) contract if there exists a security (royalty) contract such that truthful reporting of type is a dominant strategy for each agent in the mechanism.

The primary objectives of this paper are (a) to characterize the set of scfs implementable using security contracts, (b) to characterize the set of scfs that are implementable using royalty contracts, and (c) to examine how much these two sets expand the set of implementable scfs from the classic quasilinear environment.

Under a reasonable assumption on the distribution of utilities, our main result shows that if the type space is finite, then any scf implementable using a security contract can be implemented using a royalty contract. Additionally, we show that the set of scfs implementable by a security contract in this environment can be characterized using a condition called *acyclicity*, which is simple to interpret and apply.

Thus, as far as implementability is concerned, the mechanism designer can focus solely on the simpler class of royalty contracts. Put differently, our main result shows that the set of implementable scfs does not expand when we go from royalty contracts to more complicated security contracts. This provides one explanation for the ubiquity of royalty contracts in practical applications. However, we show using an example that the payoff achieved from a security contract that implements an scf may not be achievable using any royalty contract that implements the same scf. Thus, revenue/payoff equivalence does not hold across these two classes of contracts.

As an application of our results, consider a planner trying to implement alternatives that maximize social welfare. There are two commonly used criteria for evaluating welfare. The first is the utilitarian scf in which the planner chooses an alternative to maximize the sum of the agents' utilities. It is well known that the utilitarian scf can be implemented in dominant strategies using the quasilinear Vickrey-Clarke-Groves transfers. The second is the max-min or Rawlsian scf in which the planner chooses an alternative that maximizes the minimum utility of agents. It has been shown that the Rawlsian scf may not be implementable using quasilinear transfers. A contribution of this paper is to show that the Rawlsian scf is acyclic, and hence, by our result, implementable using a royalty contract. Thus, security contracts allow the principal to achieve certain important welfare objectives which may not be possible to implement using quasilinear transfers.

Our main result has a connection with the classic results of revealed preference in consumer theory.[1] The acyclicality condition we use to characterize implementability is analogous to the Generalized Axiom of Revealed Preference (Varian, 1982) which is a necessary and sufficient condition for a finite price consumption data set to be rationalized by a utility maximizing consumer. Additionally, the celebrated Afriat's theorem (Afriat, 1967; Varian, 1982) shows that a data set can be rationalized by a utility function if and only if it can be rationalized by a concave utility function. Analogously, our main result shows that acyclicity is necessary and sufficient for implementability using either security or royalty contracts. By contrast, implementability by quasilinear transfers is characterized by cycle monotonicity (Rochet, 1987; Rockafellar, 1970), which is a stronger condition than acyclicality.[2]

The rest of the paper is organized as follows. Section 2 introduces the model. In Section 3, we present our main result - the implementability equivalence between security and royalty mechanisms. In Section 4, we present the characterization of implementable scfs using the acyclicity condition. Section 5 shows that the Rawlsian scf can be implemented using royalty contracts. Then, we present some extensions and discussions of our result in Section 6. In order to make formal connections with our model, we defer the discussion of the related literature to Section 7. Finally, in our concluding remarks in Section 8, we provide a few avenues for future research.

2 The Model

There is a set of agents $N := \{1, \ldots, n\}$, who face a mechanism designer (principal). The set of alternatives is A. The type of an agent i is denoted by θ_i and the set of types for this agent is denoted by Θ_i. Each agent knows his type but the mechanism designer does not know the types of the agents. The ex-post utility of agent i with type θ_i for an alternative $a \in A$ is given by $u_i(\theta_i, a)$, and is observed by both the agent and the mechanism designer.[3] At the interim stage (i.e., after realization of the type and before an alternative is chosen), this utility is not known to the agent and the mechanism designer. It is assumed that when agent i has type θ_i, his ex-post utility from alternative a follows some distribution on $\mathbb{R}$ with cumulative distribution $G_{\theta_i,a}$. Note that since the utility is a random variable, its realization

[1] Since the earliest work on implementation (Rochet, 1987), there have been analogies made with revealed preference theory.

[2] Implementability of an scf by quasilinear transfers can be considered to be analogous to rationalizability of choice data by quasilinear utility functions (Brown and Calsamiglia, 2007). Additionally, Rahman (2011) considers a very general quasilinear setting with stochastic signals and shows the relation to revealed stochastic preference.

[3] A more general model that we do not consider is to allow interdependence in utilities, i.e., u_i is a function of types of all the agents.

need not reveal the type of the agent.[4] We will impose the following restriction on the distribution of utilities.

Definition 1 *The distributions of utilities is ordered by first order stochastic dominance or simply* **ordered** *if for all i, $\theta_i, \theta'_i \in \Theta_i$ and for all $a \in A$, we have*

$$\textit{either } G_{\theta_i,a} \succeq_{FOSD} G_{\theta'_i,a} \textit{ or } G_{\theta'_i,a} \succeq_{FOSD} G_{\theta_i,a},$$

where $\succeq_{FOSD}$ is the first-order stochastic dominance relation.

The above ordering requirement says that for every agent i and every alternative $a \in A$, the types in Θ_i can be ordered using the FOSD relation. To the best of our knowledge, most of the theoretical work on mechanism design with securities requires this assumption.[5] Importantly, the standard deterministic mechanism design environment (where the distribution of utilities corresponding to each alternative is degenerate) is ordered in the above sense.

A **social choice function (scf)** is a mapping $f : \Theta_1 \times \cdots \times \Theta_n \to A$. This mapping specifies the chosen alternative for every profile of reported types.

We now define a security contract. A **security contract** of agent i is a mapping $s_i : \mathbb{R} \times \Theta_1 \times \cdots \times \Theta_n \to \mathbb{R}$, which is strictly increasing in the first argument. The security contract for agent i assigns a payoff to i given his *realized* utility and the profile of reported types. The interpretation here is that the final utility of the agent is not known ex-ante when the agent reports his type but it is revealed to the agent and the mechanism designer once it is realized at which point, the security contract assigns a payoff to the agent. For instance, if an alternative $a \in A$ is chosen and the true type of the agent is θ_i, the security contract assigns payoff $s_i(u_i(\theta_i, a), \theta'_i, \theta'_{-i})$ to i when utility $u_i(a, \theta_i)$ is realized, and the reported type profile is $(\theta'_i, \theta'_{-i})$. Since s_i is strictly increasing in the first argument, the mechanism awards the agent a strictly greater payoff whenever the realized utility is strictly higher.

While the security contracts we consider are very general and model many real world securities, they are with loss of generality. Requiring s_i to be strictly increasing in the first argument is not completely innocuous as it rules out certain commonly used securities which are weakly increasing such as call options and convertible debt.[6] The main reason that this

[4] Of course, if the principal knew the prior distribution over the agents' types, the realized utility would allow him to update the prior. By contrast, if the principal does not know the type distribution, he will not be able to make inference (dominant strategy implementation is appropriate for these cases). That said, we allow the supports of the distributions of utilities to vary over different alternatives. Hence, even without prior knowledge of type distributions, there may be certain realizations of utility from which the principal can back out the type of the agent.

[5] Often, the stronger assumption of affiliation (Milgrom and Weber, 1982) is made instead (DeMarzo et al., 2005; Gorbenko and Malenko, 2010).

[6] Later, we will show that we can strengthen our ordering condition on distribution of utilities such that some of these weakly increasing contracts can also be covered in our results.

assumption is made is to prevent the principal from "buying" the agents, thereby making them indifferent amongst reports and trivializing the implementation problem.[7] Additionally, notice that we do not allow the payoff to agent i from the security contract to depend on the realized utilities of the other agents but only on their announced types. This is true in most real world securities and, to the best of our knowledge, this simplifying assumption is made in all of the papers in the literature.

A **security mechanism** is $(f, s_1, \ldots, s_n)$, where f is an scf and $(s_1, \ldots, s_n)$ are the security contracts of the agents. A special case of the security mechanism is the standard **quasi-linear mechanism** $(f, t_1, \ldots, t_n)$, in which the contracts just specify transfer rules $(t_1, \ldots, t_n)$, where $t_i : \Theta_1 \times \cdots \times \Theta_n \to \mathbb{R}$ is the transfer rule of agent i. The payoff assigned to agent i by such a quasi-linear mechanism is $u_i(\theta_i, f(\theta'_i, \theta'_{-i})) - t_i(\theta'_i, \theta'_{-i})$ if the agent's true type is θ_i and the profile of reported types is $(\theta'_i, \theta'_{-i})$.

Another class of security mechanisms is the class of **royalty mechanisms**, which consists of an scf and a simpler security contract for every agent, which we call the **royalty contract**. A royalty contract for agent i consists of two mappings, a royalty rule $r_i : \Theta_1 \times \cdots \times \Theta_n \to (0, 1]$ and a transfer rule $t_i : \Theta_1 \times \cdots \times \Theta_n \to \mathbb{R}$. The payoff assigned to agent i by such a royalty contract is

$$r_i(\theta'_i, \theta'_{-i}) u_i(\theta_i, f(\theta'_i, \theta'_{-i})) - t_i(\theta'_i, \theta'_{-i}),$$

if the true type of agent i is θ_i and the profile of reported types is $(\theta'_i, \theta'_{-i})$. In words, a royalty contract specifies a transfer amount and a fraction of the utility to be shared. As we pointed out in the introduction, such contracts are ubiquitous in practice. Once again, by not allowing $r_i(\theta_i, \theta_{-i}) = 0$ for all θ_i, θ_{-i}, we are excluding the possibility that the mechanism designer can "buy" the agent.

We will now define the notion of (dominant strategy) implementation that we consider. Because this is an environment with uncertainty about the utility of the agent, the agent must compute his expected utility before reporting his type. In particular, if the true type of agent i is θ_i and the profile of reports is $(\theta'_i, \theta'_{-i})$, the (security) mechanism $(f, s_1, \ldots, s_n)$, provides him expected utility given by

$$\mathbb{E}_{u_i}[s_i(u_i(\theta_i, f(\theta'_i, \theta'_{-i})), \theta'_i, \theta'_{-i})],$$

where the expectation is taken using the cdf $G_{\theta_i, f(\theta'_i, \theta'_{-i})}$.

Definition 2 *An scf f is* **implementable by a security contract** *if there exist security*

[7] When the supports of the utility distributions are such that realized utilities reveal types, this restriction may also prevent the mechanism designer from punishing detectable misreports which result in higher realized utilities.

contracts $(s_1, \ldots, s_n)$ *such that*

$$\mathbb{E}_{u_i}[s_i(u_i(\theta_i, f(\theta_i, \theta_{-i})), \theta_i, \theta_{-i})] \geq \mathbb{E}_{u_i}[s_i(u_i(\theta_i, f(\theta'_i, \theta_{-i})), \theta'_i, \theta_{-i})] \qquad \forall\ i,\ \theta_i, \theta'_i \in \Theta_i \text{ and } \theta_{-i} \in \Theta_{-i}.$$

In this case, we say that the security contracts $(s_1, \ldots, s_n)$ *implement* f *and the security mechanism* $(f, s_1, \ldots, s_n)$ *is incentive compatible.*

In case of implementation by a royalty mechanism, the incentive constraints look simpler.

Definition 3 *An scf* f *is* **implementable by a royalty contract** *if there exist royalty contracts* $((r_1, t_1), \ldots, (r_n, t_n))$*, such that*

$$r_i(\theta_i, \theta_{-i})\mathbb{E}_{u_i}[u_i(\theta_i, f(\theta_i, \theta_{-i}))] - t_i(\theta_i, \theta_{-i}) \geq r_i(\theta'_i, \theta_{-i})\mathbb{E}_{u_i}[u_i(\theta_i, f(\theta'_i, \theta_{-i}))] - t_i(\theta'_i, \theta_{-i}) \qquad \forall\ i,\ \theta_i, \theta'_i \in \Theta_i \text{ and } \theta_{-i} \in \Theta_{-i}.$$

In this case, we say that the royalty contracts $((r_1, t_1), \ldots, (r_n, t_n))$ *implement* f *and the royalty mechanism* $(f, (r_1, t_1), \ldots, (r_n, t_n))$ *is incentive compatible.*

2.1 A Motivating Example - The Max-min Social Choice Function

The ability to contract on the realized utilities in the ex-post stage has a nontrivial impact on the set of scfs that can be implemented using securities in such environments. We illustrate this with a simple example. Consider an environment with two agents $N := \{1, 2\}$ and three alternatives $A := \{a_1, a_2, a_3\}$. Assume that the expected utility is $\bar{u}_i(\theta_i, a) := \mathbb{E}_{u_i}[u_i(\theta_i, a)]$ for agent i if his true type is θ_i and an alternative $a \in A$ is chosen.

Definition 4 *An scf* f *is a* **max-min scf** *if for every type profile* $\theta \equiv (\theta_1, \theta_2)$

$$f(\theta_1, \theta_2) \in \arg\max_{a \in A} \min_{i \in \{1,2\}} \mathbb{E}_{u_i}[u_i(\theta_i, a)].$$

Thus, the max-min scf chooses an alternative that maximizes the minimum utility of the two agents. In social choice theory literature, the max-min scf is also referred to as the Rawlsian rule - see Arrow and Sen (2002) for a comprehensive discussion on various aspects of this rule. If an alternative is a welfare scheme and the utilities of the agents reflect the utility derived from these welfare schemes, then the max-min scf chooses a welfare scheme that maximizes the minimum utility derived by any agent. Put differently, the Rawlsian rule reflects a society which aims to maximize the well-being of its worst off member. Bikhchandani et al. (2006) (see supplemental material) showed that the max-min scf is not implementable using quasi-linear transfers. The following example demonstrates this.

Example 1

Consider a type space $\Theta_1 \supseteq \{\theta_1, \theta_1'\}$ and $\Theta_2 \supseteq \{\theta_2\}$. Suppose $A = \{a_1, a_2, a_3\}$. The expected utilities at the two type profiles (θ_1, θ_2) and (θ_1', θ_2) are as shown in Table 1.

	a_1	a_2	a_3		a_1	a_2	a_3
$\mathbb{E}_{u_1}[u_1(\theta_1, \cdot)] = \bar{u}_1(\theta_1, \cdot)$	2	3	5	$\mathbb{E}_{u_1}[u_1(\theta_1', \cdot)] = \bar{u}_1(\theta_1', \cdot)$	0	1	2
$\mathbb{E}_{u_2}[u_2(\theta_2, \cdot)] = \bar{u}_2(\theta_2, \cdot)$	1	4	2	$\mathbb{E}_{u_2}[u_2(\theta_2, \cdot)] = \bar{u}_2(\theta_2, \cdot)$	1	4	2

Table 1: Utilities

If f is the max-min scf, then $f(\theta_1, \theta_2) = a_2$ but $f(\theta_1', \theta_2) = a_3$. To see why this is not implementable by quasi-linear transfers, assume for contradiction that there exists a transfer rule $t_1 : \Theta_1 \times \Theta_2 \to \mathbb{R}$ for agent 1, such that

$$\bar{u}_1(\theta_1, f(\theta_1, \theta_2)) - t_1(\theta_1, \theta_2) \geq \bar{u}_1(\theta_1, f(\theta_1', \theta_2)) - t_1(\theta_1', \theta_2)$$
$$\bar{u}_1(\theta_1', f(\theta_1', \theta_2)) - t_1(\theta_1', \theta_2) \geq \bar{u}_1(\theta_1', f(\theta_1, \theta_2)) - t_1(\theta_1, \theta_2).$$

Then, adding these constraints, we get

$$\left[\bar{u}_1(\theta_1, f(\theta_1, \theta_2)) - \bar{u}_1(\theta_1, f(\theta_1', \theta_2))\right] + \left[\bar{u}_1(\theta_1', f(\theta_1', \theta_2)) - \bar{u}_1(\theta_1', f(\theta_1, \theta_2))\right] \geq 0.$$

Plugging in these values from Table 1, we notice that $-2 + 1 \geq 0$, a contradiction.

As an application of our main characterization, we will show that such a max-min scf can be implemented using a royalty contract. This shows that there are important welfare objectives that cannot be achieved using quasi-linear transfers but can be implemented using royalty contracts.

3 The Main Result

Our main result shows that for finite type spaces, the set of scfs that can be implemented using any (security) contract can also be implemented using a royalty contract.

Theorem 1 *Suppose the type space is finite and distributions of utilities are ordered. Then, an scf is implementable using a security contract if and only if it is implementable using a royalty contract.*

The proof of Theorem 1 involves identifying a necessary condition for implementation using a security mechanism, and then showing that this condition is sufficient for implementation using a royalty mechanism. Before we state this condition, we will like to point out some simplification in notation that we will use. For clarity of exposition, we choose to conduct the analysis that follows in a single agent environment. We can do so because the incentive compatibility requirement is for each agent i and all possible reports θ_{-i} of the other agents, and so we can conduct our analysis for an arbitrary agent i and take θ_{-i} as fixed. By taking θ_{-i} as fixed, social choice functions essentially become a function of i's type alone and so we can drop the subscript on type space Θ_i so that $f : \Theta \to A$. Then, we can also drop the subscript from the utility function u_i and represent it as u. Finally, the subscripts in the notation for the contracts can also be simplified and we can use $s : \mathbb{R} \times \Theta \to \mathbb{R}$ instead of s_i, $r : \Theta \to (0, 1]$ and $t : \Theta \to \mathbb{R}$ instead of r_i and t_i respectively. Mechanisms can now simply be expressed as (f, s), (f, r, t) instead of $(f, s_1, \ldots, s_n)$, $(f, (r_1, t_1), \ldots, (r_n, t_n))$ respectively. Except for Section 5, we will use this one agent notation everywhere from now on.

The proof of the theorem utilizes the following *acyclicity* condition. For every scf f, define the preference relation $\succeq^f$ over types as follows: for every $\theta, \theta' \in \Theta$,

$$\theta' \succeq^f \theta \text{ if } G_{\theta', f(\theta)} \succeq_{FOSD} G_{\theta, f(\theta)}.$$

Also, let $\theta' \succ^f \theta$ if the above FOSD relation is strict. Note that $\succeq^f$ may not be a complete relation and may not be transitive.[8]

Definition 5 *An scf f is* **acyclic** *if for every $\theta, \theta' \in \Theta$ such that there is a sequence of types $\theta^1, \ldots, \theta^k$ with $\theta' \succeq^f \theta^1$, $\theta^1 \succeq^f \theta^2$, …, $\theta^{k-1} \succeq^f \theta^k$, $\theta^k \succeq^f \theta$, we have $\theta \not\succ^f \theta'$.*

The next proposition establishes that acyclicity is a necessary condition for implementation by a security contract.

Proposition 1 *Suppose the distributions of utilities are ordered. If an scf is implementable by a security contract, then it is acyclic.*

Proof: Suppose scf f is implementable by a security contract s. Consider $\theta, \theta' \in \Theta$ such that there is a sequence of types $\theta^1, \ldots, \theta^k$ with $\theta' \succeq^f \theta^1$, $\theta^1 \succeq^f \theta^2$, …, $\theta^{k-1} \succeq^f \theta^k$, $\theta^k \succeq^f \theta$. Let $\theta' \equiv \theta^0$ and $\theta \equiv \theta^{k+1}$. Pick any $j \in \{0, 1, \ldots, k\}$. Now, since $G_{\theta^j, f(\theta^{j+1})} \succeq_{FOSD} G_{\theta^{j+1}, f(\theta^{j+1})}$, we have

$$\begin{aligned}\mathbb{E}_u[s(u(\theta^j, f(\theta^j)), \theta^j)] &\geq \mathbb{E}_u[s(u(\theta^j, f(\theta^{j+1})), \theta^{j+1})] && \big(\text{since } f \text{ is implementable}\big)\\ &\geq \mathbb{E}_u[s(u(\theta^{j+1}, f(\theta^{j+1})), \theta^{j+1})] && \big(\text{since } G_{\theta^j, f(\theta^{j+1})} \succeq_{FOSD} G_{\theta^{j+1}, f(\theta^{j+1})}\big),\end{aligned}$$

[8] In contrast to our notion of acyclicality over types, Rochet (1987) provides an acyclicality condition over alternatives as a necessary (but not sufficient) condition for implementation in the deterministic quasilinear environment. Of course, our condition will also be a necessary for implementation in the deterministic quasilinear environment.

where we also used the fact that s is strictly increasing in the first argument for the second inequality. This shows that $\mathbb{E}_u[s(u(\theta^j, f(\theta^j)), \theta^j)] \geq \mathbb{E}_u[s(u(\theta^{j+1}, f(\theta^{j+1})), \theta^{j+1})]$ for any $j \in \{0, 1, \ldots, k\}$. Summing over $j \in \{0, 1, \ldots, k\}$, and telescoping, we get

$$\begin{aligned}\mathbb{E}_u[s(u(\theta', f(\theta')), \theta')] &\geq \mathbb{E}_u[s(u(\theta, f(\theta)), \theta)] \\ &\geq \mathbb{E}_u[s(u(\theta, f(\theta')), \theta')] \qquad \big(\text{since } f \text{ is implementable}\big).\end{aligned}$$

Since the distribution of utilities is ordered and s is strictly increasing in the first argument, it must be that $G_{\theta, f(\theta')} \not\succ_{FOSD} G_{\theta', f(\theta')}$ or $\theta \not\succ^f \theta'$. Hence, f is acyclic. ■

Remark. Though we required the security contract to be increasing in the first argument (the realized utility) for Proposition 1, we can weaken it to be weakly increasing under additional assumptions. To see this, suppose the support of the realized utilities for any type and any alternative is $[\underline{u}, \overline{u}]$, where $\underline{u} < \overline{u}$. The distributions of utilities are **strictly ordered** if they are ordered and for all $\theta, \theta' \in \Theta$ and for all $a \in A$, we have either $G_{\theta,a} = G_{\theta',a}$ or $\{w \in [\underline{u}, \overline{u}] : G_{\theta,a}(w) = G_{\theta',a}(w)\} = \{\underline{u}, \overline{u}\}$. We will say a security contract s is almost strictly increasing if it is weakly increasing in the first argument and there is some positive length interval $L \subseteq [\underline{u}, \overline{u}]$ over which s is strictly increasing in the first argument. With these two assumptions, it is not difficult to see that for any $\theta, \theta' \in \Theta$ and any $a \in A$, if $G(\theta', a) \succ_{FOSD} G(\theta, a)$, then $\mathbb{E}_u[s(u(\theta', a), \theta'')] > \mathbb{E}_u[s(u(\theta, a), \theta'')]$ for all $\theta'' \in \Theta$. Consequently, Proposition 1 will continue to hold under these assumptions. As we will show next, with finite type spaces, acyclicity is a sufficient condition for implementability using a *royalty* contract. Hence, the main result (Theorem 1) will continue to hold true with these modified assumptions.

Now, we establish that with finite type spaces, acyclicity implies that an scf can be implemented using a *royalty* contract.

PROPOSITION **2** *Suppose the type space is finite and the distributions of utilities are ordered. If an scf is acyclic, then it can be implemented using a royalty contract.*

Proof: Fix an scf f that is acyclic. Notice that to show f can be implemented by a royalty contract, we need to show that there exist maps $r : \Theta \to (0, 1]$ and $t : \Theta \to \mathbb{R}$ such that for every $\theta, \theta' \in \Theta$, we have

$$\begin{aligned}r(\theta)\mathbb{E}_u[u(\theta, f(\theta))] - t(\theta) &\geq r(\theta')\mathbb{E}_u[u(\theta, f(\theta'))] - t(\theta') \\ &= r(\theta')\mathbb{E}_u[u(\theta', f(\theta'))] - t(\theta') \\ &\quad + r(\theta')\big[\mathbb{E}_u[u(\theta, f(\theta'))] - \mathbb{E}_u[u(\theta', f(\theta'))]\big].\end{aligned}$$

Notice that if a royalty contract (r,t) implements f, then $r(\theta)\mathbb{E}_u[u(\theta,f(\theta))]-t(\theta)$ is the equilibrium net utility of telling the truth when his type is θ. If we denote the net utility of telling the truth when the type is θ as $W(\theta):=r(\theta)\mathbb{E}_u[u(\theta,f(\theta))]-t(\theta)$, then the royalty contract (r,t) implements f if for every $\theta,\theta'\in\Theta$, we have

$$W(\theta')-W(\theta)\leq r(\theta')\left[\mathbb{E}_u[u(\theta',f(\theta'))]-\mathbb{E}_u[u(\theta,f(\theta'))]\right] \tag{1}$$

Hence, we will construct maps $W:\Theta\to\mathbb{R}$ and $r:\Theta\to(0,1]$ such that for all θ,θ', Inequality (1) is satisfied - note that constructing the maps W and r also specifies the map t by setting $t(\theta):=r(\theta)\mathbb{E}_u[u(\theta,f(\theta))]-W(\theta)$ for all $\theta\in\Theta$. Further, note that if we can construct W and r maps satisfying Inequality (1) such that $r(\theta)>0$ for all $\theta\in\Theta$, we can always scale them, by dividing by $\max_{\theta\in\Theta}r(\theta)$, and get a new set of W and r maps that satisfy Inequality (1). Hence, without loss of generality, we will construct maps $W:\Theta\to\mathbb{R}$ and $r:\Theta\to\mathbb{R}_{++}$ such that for all θ,θ', Inequality (1) is satisfied.

We do this construction using induction on $|\Theta|$. The base case is trivial. Suppose, we can construct W and r for all type spaces with less than K types and let $|\Theta|=K$.

For every $\theta\in\Theta$, let $C(\theta):=\{\theta'\in\Theta:\theta'\succ^f\theta\}$. A type θ is maximal with respect to $\succeq^f$ if $C(\theta)=\emptyset$. Denote the set of types that are maximal with respect to $\succeq^f$ as $M^{\succeq^f}$. A consequence of acyclicity of $\succeq^f$ and finite Θ is that $M^{\succeq^f}$ is non-empty (Sen, 1970). Then, for every type $\theta'\in M^{\succeq^f}$ and for every $\theta\in\Theta$, we have $\theta\not\succ^f\theta'$, which implies that $G_{\theta',f(\theta'))}\succeq_{FOSD}G_{\theta,f(\theta')}$ because distributions of utilities are ordered. Hence, for every type $\theta'\in M^{\succeq^f}$ and for every $\theta\in\Theta$, we have

$$\mathbb{E}_u[u(\theta',f(\theta'))]-\mathbb{E}_u[u(\theta,f(\theta'))]\geq 0. \tag{2}$$

Hence, if $M^{\succeq^f}=\Theta$, then we can just set $W(\theta)=0$ and $r(\theta)$ any value in $(0,1]$ for all $\theta\in\Theta$, and this will satisfy Inequality 1. So, we assume that $M^{\succeq^f}\neq\Theta$. By our induction hypothesis, we can find $W(\theta)$ and $r(\theta)$ for all $\theta\in\Theta\setminus M^{\succeq^f}$ such that Inequality (1) is satisfied for each $\theta,\theta'\in\Theta\setminus M^{\succeq^f}$. We now proceed by defining for all $\theta\in M^{\succeq^f}$,

$$W(\theta):=\max_{\theta'\in\Theta\setminus M^{\succeq^f}}\left[W(\theta')+r(\theta')\left[\mathbb{E}_u[u(\theta,f(\theta'))]-\mathbb{E}_u[u(\theta',f(\theta'))]\right]\right].$$

Next, for all $\theta\in M^{\succeq^f}$, we set large enough $r(\theta)$ such that it is positive and

$$r(\theta)>\max_{\theta'\in\Theta\setminus M^{\succeq^f}}\left[\frac{W(\theta)-W(\theta')}{\mathbb{E}_u[u(\theta,f(\theta))]-\mathbb{E}_u[u(\theta',f(\theta))]}\right]. \tag{3}$$

Notice that if $\theta'\in\Theta\setminus M^{\succeq^f}$ and $\theta\in M^{\succeq^f}$, we have $\theta'\not\succeq^f\theta$, and hence, $\mathbb{E}_u[u(\theta,f(\theta))]-\mathbb{E}_u[u(\theta',f(\theta))]>0$. This ensures that the denominator of Inequality 3 is positive. By the definition of W, Inequality 1 is satisfied if $\theta\in M^{\succeq^f}$ and $\theta'\in\Theta\setminus M^{\succeq^f}$. By our induction hypothesis, it is also satisfied if $\theta, \theta' \in \Theta \setminus M^{\succeq^f}$. If $\theta, \theta' \in M^{\succeq^f}$, then, by construction $W(\theta') - W(\theta) = 0$ and Equation 2 ensures that Inequality (1) is satisfied. Finally, by the definition of $r(\theta')$ (Inequality 3), Inequality 1 is satisfied for the case, where $\theta \in \Theta \setminus M^{\succeq^f}$ and $\theta' \in M^{\succeq^f}$. This concludes the proof. ■

Proof of Theorem 1. The proof follows from Propositions 1 and 2, and the fact that a royalty contract is also a security contract.

4 Implementation Using Royalty Contracts

In this section, we provide conditions which characterize scfs implementable by royalty contracts. Due to Theorem 1, this will also characterize all scfs implementable by securities in finite type spaces. This characterization will also enable us to demonstrate via a counterexample that Theorem 1 doesn't hold in general for infinite type spaces.

It is well known that a condition called *cycle monotonicity* characterizes the set of scfs implementable by a quasi-linear mechanism (Rochet, 1987; Rockafellar, 1970). We adapt the cycle monotonicity condition below and show it to be necessary and sufficient for scfs to be implementable by a royalty contract.

Definition 6 *An scf f is* **multiplier K-cycle monotone**, *where $K \geq 2$ is a positive integer, if there exists $\lambda : \Theta \to (0, 1]$ such that for all finite sequence of types $(\theta^1, \dots, \theta^k)$ with $k \leq K$, we have*

$$\sum_{j=1}^{k} \lambda(\theta^j) \left[\mathbb{E}_u[u(\theta^j, f(\theta^j))] - \mathbb{E}_u[u(\theta^{j+1}, f(\theta^j))] \right] \geq 0,$$

where $\theta^{k+1} \equiv \theta^1$. An scf f is **multiplier cycle monotone** *if it is multiplier K-cycle monotone for all integers $K \geq 2$.*

The next proposition establishes that multiplier cycle monotonicity is necessary and sufficient for implementation by a royalty contract. The proof follows standard steps using cycle monotonicity results in convex analysis (Rochet, 1987; Rockafellar, 1970) and it is relegated to the Appendix.

Proposition 3 *An scf is implementable by a royalty contract if and only if it is multiplier cycle monotone.*

This leads to the following immediate characterization.

Theorem 2 *Suppose the type space is finite and the distributions of utilities are ordered. Then, the following conditions are equivalent for an scf f.*

1. *f is acyclic.*

2. *f is multiplier cycle monotone.*

3. *f is implementable by a royalty contract.*

4. *f is implementable by a security contract.*

Proof: The proof follows from Propositions 1, 2, 3, and the fact that a royalty contract is also a security contract. ■

We now present an example with countably infinite type space Θ where Theorem 1 breaks down. For simplicity, the example is deterministic – for every type $\theta \in \Theta$ and for every $a \in A$, the realized utility is $\bar{u}(\theta, a)$ with probability one or, in other words, the utility distribution $G_{\theta,a}$ is degenerate. Uncertainty can be introduced into this example if we assume that for each θ, a, there is a utility $\bar{u}(\theta, a)$ which occurs with probability almost one and all other utilities occur with probability sufficiently close to zero. This uncertainty can be chosen in a way so that the distributions remain ordered.[9]

We first show that acyclicity is a sufficient condition for implementability in this model. The proof of this lemma is given in the Appendix.

Lemma **1** *Suppose Θ is countable and the distributions $G_{\theta,a}$ are degenerate for all θ, a. If an scf is acyclic, it can be implemented using a security contract.*

We now give an example of an scf in this model that can be implemented using a security contract but cannot be implemented using a royalty contract.

Example **2**

Consider the following countably infinite type space

$$\Theta = \left\{\theta^2, \theta^3, \dots\right\} \cup \left\{\theta^\infty\right\}.$$

Take an outcome space of equal cardinality and consider an scf f which satisfies

$$f\left(\theta^k\right) \neq f\left(\theta^{k'}\right) \qquad \text{for all } k \neq k'.$$

Define a utility function u satisfying

$$\bar{u}(\theta^k, f(\theta^{k'})) = \begin{cases} \frac{2}{k'} & \text{if } k' < k \\ \frac{1}{k} & \text{if } k' = k \\ \frac{1}{2k'} & \text{if } k' > k \\ 0 & \text{if } k' = \infty \end{cases}$$

[9] A simple way would be to assume $u(\theta, a) = \bar{u}(\theta, a) + \varepsilon$, where ε has mean zero and its distribution does not depend on θ, a.

Finally, we define utility for type θ_∞ as

$$\bar{u}(\theta^\infty, f(\theta^{k'})) = \begin{cases} \frac{2}{k'} & \text{if } k' < \infty \\ 1 & \text{otherwise} \end{cases}$$

We first argue that f is acyclic. This is because $\theta^k \succ \theta^{k'}$ for all $k' < k \leq \infty$ as

$$\bar{u}(\theta^k, f(\theta^{k'})) = \frac{2}{k'} > \frac{1}{k'} = \bar{u}(\theta^{k'}, f(\theta^{k'})).$$

Moreover, when $k < k' < \infty$ then $\theta^k \not\succeq \theta^{k'}$ as

$$\bar{u}(\theta^k, f(\theta^{k'})) = \frac{1}{2k'} < \frac{1}{k'} = \bar{u}(\theta^{k'}, f(\theta^{k'})).$$

Finally, $\theta^k \not\succeq \theta^\infty$

$$\bar{u}(\theta^k, f(\theta^\infty)) = 0 < 1 = \bar{u}(\theta^\infty, f(\theta^\infty)).$$

Hence, f is acyclic. By Lemma 1, f can be implemented using a security mechanism.
We now show that f does not satisfy multiplier cycle monotonicity, and hence, by Proposition 3, cannot be implemented by a royalty contract. Let us assume to the contrary that it does. Then it must be true for the 2-cycle consisting of θ^k, θ^∞ that

$$\begin{aligned} &\lambda(\theta^k)[\bar{u}(\theta^k, f(\theta^k)) - \bar{u}(\theta^\infty, f(\theta^k))] + \lambda(\theta^\infty)[\bar{u}(\theta^\infty, f(\theta^\infty)) - \bar{u}(\theta^k, f(\theta^\infty))] \geq 0 \\ \Rightarrow &\frac{\lambda(\theta^\infty)}{\lambda(\theta^k)} \geq -\frac{\frac{1}{k} - \frac{2}{k}}{1 - 0} = \frac{1}{k} \end{aligned}$$

The last inequality follows from the fact that λ's are strictly positive and that $\theta^k \not\succeq \theta^\infty$. Similarly, it must be true for 2 cycles consisting of θ^k, θ^{k+1} that

$$\begin{aligned} &\lambda(\theta^k)[\bar{u}(\theta^k, f(\theta^k)) - \bar{u}(\theta^{k+1}, f(\theta^k))] + \lambda(\theta^{k+1})[\bar{u}(\theta^{k+1}, f(\theta^{k+1})) - \bar{u}(\theta^k, f(\theta^{k+1}))] \geq 0 \\ \Rightarrow &\frac{\lambda(\theta^{k+1})}{\lambda(\theta^k)} \geq -\frac{\frac{1}{k} - \frac{2}{k}}{\frac{1}{k+1} - \frac{1}{2(k+1)}} = \frac{2(k+1)}{k} \end{aligned}$$

Since both sides of the above inequality are positive, we can multiply inequalities for succeeding $k = 2, \ldots, K-1$ to get

$$\frac{\lambda(\theta^K)}{\lambda(\theta^2)} \geq 2^{K-3}K.$$

Combining inequalities we get

$$\lambda(\theta^\infty) \geq \frac{\lambda(\theta^K)}{K} \geq 2^{K-3}\lambda(\theta^2)$$

Taking the limit $K \to \infty$, we observe that the right side diverges, which implies that $\lambda(\theta^\infty)$ must be ∞ which is a contradiction. Hence, f does not satisfy multiplier cycle monotonicity, and cannot be implemented using a royalty contract.

5 The Max-min SCF

We now revisit the max-min scf we defined in Section 2.1. In order to define this scf formally for n agents and to present our results, we revert to the n player notation in this subsection.

Definition **7** *An scf f is a* **max-min scf** *if for every type profile $\theta \equiv (\theta_1, \ldots, \theta_n)$*

$$f(\theta) \in \arg\max_{a \in A} \min_{i \in N} \mathbb{E}_u[u_i(\theta_i, a)].$$

Further, f is a max-min scf satisfying **consistent tie-breaking** *if there exists a linear ordering P on alternatives A such that at any type profile θ, $f(\theta)$ is the maximal alternative in the set $\{a \in A : \min_{i \in N} \mathbb{E}_u[u_i(\theta_i, a)] \geq \min_{i \in N} \mathbb{E}_u[u_i(\theta_i, b)] \ \forall \ b \neq a\}$ according to P.*

We had earlier discussed how max-min scfs cannot be implemented using quasi-linear transfers. By expanding the set of mechanisms to include royalty contracts, the max-min scfs can now be implemented. The consistent tie-breaking condition is required since otherwise an agent may be able to manipulate when more than one alternative maximizes the minimum expected utility of the agents. [10]

Theorem **3** *Suppose the type space is finite and the distribution of utilities is ordered. Then, a max-min scf with consistent tie-breaking can be implemented using a royalty contract.*

Proof: Suppose f is a max-min scf that satisfies consistent tie-breaking. We will show that f is acyclic, and by Theorem 2, it can be implemented using a royalty contract. Fix an agent i. Define for any alternative $x \in A$ and any type profile $(\bar{\theta}_i, \bar{\theta}_{-i})$

$$U(x, \bar{\theta}_i, \bar{\theta}_{-i}) = \min_{j \in N} \mathbb{E}_u[u_j(\bar{\theta}_j, x)].$$

Note that for any alternative $x \in A$ and any type profiles $(\bar{\theta}'_i, \bar{\theta}_{-i})$ and $(\bar{\theta}_i, \bar{\theta}_{-i})$,

$$\mathbb{E}_u[u_i(\bar{\theta}'_i, x)] \geq \mathbb{E}_u[u_i(\bar{\theta}_i, x)] \Rightarrow U(x, \bar{\theta}'_i, \bar{\theta}_{-i}) \geq U(x, \bar{\theta}_i, \bar{\theta}_{-i}).$$

Consider two types $\theta'_i, \theta_i \in \Theta_i$ and a type profile θ_{-i} of other agents. Let $\succeq^f_{\theta_{-i}} \equiv \succeq$. Suppose there is a sequence of types $(\theta^1_i, \ldots, \theta^k_i)$ such that $\theta'_i \succeq \theta^1_i$, $\theta^1_i \succeq \theta^2_i$, $\ldots$, $\theta^{k-1}_i \succeq \theta^k_i$, $\theta^k_i \succeq \theta_i$. Let $\theta^0_i = \theta'_i$ and $\theta^{k+1}_i \equiv \theta_i$. Pick any $j \in \{0, 1, \ldots, k\}$. Let $f(\theta^{j+1}_i, \theta_{-i}) = a$ and $f(\theta^j_i, \theta_{-i}) = b$. Since $\theta^j_i \succeq \theta^{j+1}_i$, $\mathbb{E}_u[u_i(\theta^j_i, a)] \geq \mathbb{E}_u[u_i(\theta^{j+1}_i, a)]$. Hence, we have

$$U(a, \theta^j_i, \theta_{-i}) \geq U(a, \theta^{j+1}_i, \theta_{-i}).$$

[10] It is known that careful tie-breaking may be necessary to ensure implementability of scfs even in the deterministic quasi-linear environments. For instance, Carbajal et al. (2013) show that some *affine maximizer* scfs may not be implementable in deterministic quasi-linear environments if ties are not broken consistently.

Since $f(\theta_i^j, \theta_{-i}) = b$, we have

$$U(b, \theta_i^j, \theta_{-i}) \geq U(a, \theta_i^j, \theta_{-i}).$$

Combining this with the previous inequality, we get

$$U(f(\theta_i^j, \theta_{-i}), \theta_i^j, \theta_{-i}) \geq U(f(\theta_i^{j+1}, \theta_{-i}), \theta^j, \theta_{-i}) \geq U(f(\theta_i^{j+1}, \theta_{-i}), \theta_i^{j+1}, \theta_{-i}).$$

Using it over all $j \in \{0, 1, \ldots, k\}$, we get

$$\begin{aligned} U(f(\theta_i', \theta_{-i}), \theta_i', \theta_{-i}) &\geq U(f(\theta_i^1, \theta_{-i}), \theta_i', \theta_{-i}) \\ &\geq U(f(\theta_i^1, \theta_{-i}), \theta_i^1, \theta_{-i}) \\ &\geq \ldots \\ &\geq U(f(\theta_i, \theta_{-i}), \theta_i, \theta_{-i}). \end{aligned}$$

By definition of f, $U(f(\theta_i, \theta_{-i}), \theta_i, \theta_{-i}) \geq U(f(\theta_i', \theta_{-i}), \theta_i, \theta_{-i})$. This implies that

$$U(f(\theta_i', \theta_{-i}), \theta_i', \theta_{-i}) \geq U(f(\theta_i', \theta_{-i}), \theta_i, \theta_{-i}). \tag{4}$$

Now, assume for contradiction that $\theta_i \succ \theta_i'$. Hence, $\mathbb{E}_u[u_i(\theta_i', f(\theta_i', \theta_{-i}))] < \mathbb{E}_u[u_i(\theta_i, f(\theta_i', \theta_{-i}))]$. Notice that since $\theta_i' \succeq \theta_i$, $f(\theta_i', \theta_{-i}) \neq f(\theta_i, \theta_{-i})$. Then, $U(f(\theta_i', \theta_{-i}), \theta_i, \theta_{-i}) \geq U(f(\theta_i', \theta_{-i}), \theta_i', \theta_{-i})$. Using Inequality 4, we get

$$\begin{aligned} U(f(\theta_i', \theta_{-i}), \theta_i', \theta_{-i}) &= U(f(\theta_i^1, \theta_{-i}), \theta_i', \theta_{-i}) \\ &= U(f(\theta_i^1, \theta_{-i}), \theta_i^1, \theta_{-i}) \\ &= \ldots \\ &= U(f(\theta_i, \theta_{-i}), \theta_i, \theta_{-i}) \\ &= U(f(\theta_i', \theta_{-i}), \theta_i, \theta_{-i}). \end{aligned}$$

Let the linear order used by f to consistently break ties be P. For any $j \in \{0, 1, \ldots, k\}$, we see that

$$U(f(\theta_i^j, \theta_{-i}), \theta_i^j, \theta_{-i}) = U(f(\theta_i^{j+1}, \theta_{-i}), \theta_i^j, \theta_{-i}).$$

Hence, it must be that either

$$f(\theta_i^j, \theta_{-i}) = f(\theta_i^{j+1}, \theta_{-i}) \text{ or } f(\theta_i^j, \theta_{-i}) P f(\theta_i^{j+1}, \theta_{-i}).$$

Hence, either $f(\theta_i', \theta_{-i}) = f(\theta_i, \theta_{-i})$ or $f(\theta_i', \theta_{-i}) P f(\theta_i, \theta_{-i})$. But

$$U(f(\theta_i', \theta_{-i}), \theta_i, \theta_{-i}) = U(f(\theta_i, \theta_{-i}), \theta_i, \theta_{-i})$$

implies, we must have $f(\theta_i, \theta_{-i}) = f(\theta_i', \theta_{-i})$ (because of consistent tie-breaking). This is a contradiction to the fact that $\theta_i' \succeq \theta_i$ and $\theta_i \succ \theta_i'$. ■

6 Discussion

6.1 Failure of Payoff Equivalence

Given the equivalence in terms of implementability between security and royalty mechanisms, a natural question to ask is whether the payoffs from a security mechanism can be achieved by a royalty mechanism. Put differently, given an scf f and a security contract s that implements f, we ask if there exists a royalty contract (s, t) that implements f such that

$$\mathbb{E}_u[s(u(\theta, f(\theta)), \theta)] = r(\theta)\mathbb{E}_u[u(\theta, f(\theta))] - t(\theta).$$

Note that this requirement is only on payoffs on the equilibrium path. This is the usual revenue/payoff equivalence formulation in the mechanism design literature in quasi-linear environments (Krishna, 2009). The following example shows that this does not hold.

Example **3**

Consider a type space $\Theta := \{\theta^1, \theta^2, \theta^3\}$. Let the set of alternatives be $A := \{a_1, a_2, a_3\}$. Assume that for every $a \in A$ and for every $\theta \in \Theta$, the set of values of $u(\theta, a)$ is a finite set with possible values $\{30, 20, 10\}$. The distribution of utilities is shown in Table 2, where $\epsilon > 0$ is sufficiently small – note that the distributions are ordered.

	(θ^1, a_1)	(θ^1, a_2)	(θ^1, a_3)	(θ^2, a_1)	(θ^2, a_2)	(θ^2, a_3)	(θ^3, a_1)	(θ^3, a_2)	(θ^3, a_3)
30	$1-2\epsilon$	$1-2\epsilon$	ϵ	ϵ	ϵ	ϵ	ϵ	ϵ	ϵ
20	ϵ	ϵ	$1-2\epsilon$	$1-2\epsilon$	$1-2\epsilon$	ϵ	ϵ	ϵ	ϵ
10	ϵ	ϵ	ϵ	ϵ	ϵ	$1-2\epsilon$	$1-2\epsilon$	$1-2\epsilon$	$1-2\epsilon$

Table 2: Distribution utilities

Consider the scf f defined as: $f(\theta^i) = a_i$ for $i \in \{1, 2, 3\}$. This can be implemented by a security contract s, which is as follows:

$$s(30, \theta^1) = 20, s(20, \theta^1) = 5, s(10, \theta^1) = 1$$
$$s(30, \theta^2) = 16, s(20, \theta^2) = 15, s(10, \theta^2) = 1$$
$$s(30, \theta^3) = 16, s(20, \theta^3) = 10, s(10, \theta^3) = 5.$$

As $\epsilon \to 0$, this security mechanism generates the expected utilities to the agent as given in Table 3. To verify that s implements f, note that the columns represent the true type of the agent, and the maximum in each column i appears at the i-th row.

Suppose there is a royalty contract (r, t) that implements f and provides the same utility for the agent upon truth telling. Denote by $W(\theta)$, the net utility of the agent in the royalty

	θ^1	θ^2	θ^3
$\mathbb{E}_u[s(u(\cdot, f(\theta^1)), \theta^1)]$	20	5	1
$\mathbb{E}_u[s(u(\cdot, f(\theta^2)), \theta^2)]$	16	15	1
$\mathbb{E}_u[s(u(\cdot, f(\theta^3)), \theta^3)]$	10	5	5

Table 3: Exepcted utilities of (f, s) as $\epsilon \to 0$.

	a_1	a_2	a_3
$\mathbb{E}_u[u(\theta^1, \cdot)]$	30	30	20
$\mathbb{E}_u[u(\theta^2, \cdot)]$	20	20	10
$\mathbb{E}_u[u(\theta^3, \cdot)]$	10	10	10

Table 4: Expected utilities as $\epsilon \to 0$.

mechanism (f, r, t) upon telling the truth when his type is θ. Incentive compatibility of the royalty mechanism implies that for all θ, θ', Inequality 1 is satisfied, i.e.,

$$W(\theta') - W(\theta) \leq r(\theta')\big[\mathbb{E}_u[u(\theta', f(\theta'))] - \mathbb{E}_u[u(\theta, f(\theta'))]\big] \tag{5}$$

But by our assumption $W(\theta) = \mathbb{E}_u[s(u(\theta, f(\theta)), \theta)]$ for all $\theta \in \Theta$. Substituting in Inequality 5, we get

$$\mathbb{E}_u[s(u(\theta', f(\theta')), \theta')] - \mathbb{E}_u[s(u(\theta, f(\theta)), \theta)] \leq r(\theta')\big[\mathbb{E}_u[u(\theta', f(\theta'))] - \mathbb{E}_u[u(\theta, f(\theta'))]\big] \tag{6}$$

Taking $\theta' = \theta^2$ and $\theta = \theta^3$ in Inequality 6 and letting $\epsilon \to 0$, we get $r(\theta^2) \leq \frac{1}{2}$. Taking $\theta' = \theta^2$ and $\theta = \theta^1$ in Inequality 6 and letting $\epsilon \to 0$, we get $r(\theta^2) \geq 1$, which gives us a contradiction. Thus, for small $\epsilon > 0$, there is no royalty contract that yields the same payoff as s .

6.2 A Model with a One Dimensional Uncountable Type Space

As Example 2 demonstrated, the finite type space assumption is crucial in Theorem 1 and the equivalence does not hold in general for infinite type spaces. In this section, we describe a simple model of a one dimensional type space, where Theorem 1 extends without the finite type space assumption.

We assume that the set of alternatives A is finite and $\Theta \subseteq \mathbb{R}$ (hence, Θ need not be a finite set or even countable). We also assume that $\mathbb{E}_u[u(\theta, a)]$ is **linear** in θ for every $a \in A$. In particular, for every alternative $a \in A$, there exists $\kappa_a \geq 0$ and γ_a such that

$$\mathbb{E}_u[u(\theta, a)] = \kappa_a \theta + \gamma_a \qquad \forall\, \theta \in \Theta.$$

We call this the **linear expected utility (LEU)** assumption.[11] The main result of this section is a generalization of Theorem 1 and Theorem 2 under these assumptions. In this model, implementability can be characterized by simpler conditions than those required for Theorem 2.

Definition 8 *An scf f is* **2-acyclic** *if for every $\theta, \theta' \in \Theta$ such that $\theta' \succeq^f \theta$, we have $\theta \not\succ^f \theta'$.*

Theorem 4 *Suppose A is finite, $\Theta \subseteq \mathbb{R}$, distributions of utilities is ordered, and LEU assumption holds. Then, the following conditions on an scf f are equivalent.*

1. *f satisfies 2-acyclicity.*

2. *f is multiplier 2-cycle monotone.*

3. *f is implementable by a royalty contract.*

4. *f is implementable by a security contract.*

Proof: $1 \Rightarrow 2$. Define the map $\nu : A \to \mathbb{R}_+$ as follows. For every $a \in A$,

$$\nu(a) = \begin{cases} \frac{1}{\kappa_a} & \text{if } \kappa_a \neq 0 \\ 0 & \text{if } \kappa_a = 0 \end{cases}$$

Further, define $\nu^* := \max_{a \in A} \nu(a)$ and $\Theta_0 := \{\theta \in \Theta : \kappa_{f(\theta)} = 0\}$. Now, define $\lambda : \Theta \to (0, 1]$ as follows. Fix an $\epsilon \in (0, 1]$. For every $\theta \in \Theta$,

$$\lambda(\theta) = \begin{cases} \epsilon & \forall\ \theta \in \Theta_0 \\ \frac{\nu(f(\theta))}{\nu^*} & \forall\ \theta \in \Theta \setminus \Theta_0. \end{cases}$$

Now, note that if $\theta \in \Theta_0$, then $\lambda(\theta)\kappa_{f(\theta)} = 0$ and if $\theta \in \Theta \setminus \Theta_0$, then $\lambda(\theta)\kappa_{f(\theta)} = \frac{1}{\nu^*}$. Hence, for every $\theta \in \Theta_0$ and $\theta' \in \Theta \setminus \Theta_0$, we have

$$\lambda(\theta')\kappa_{f(\theta')} > \lambda(\theta)\kappa_{f(\theta)}. \tag{7}$$

Now, consider any $\theta, \theta' \in \Theta$. Since f is 2-acyclic, it means $\theta' \succeq^f \theta$ implies $\theta \not\succ^f \theta'$. Equivalently, $(\theta' - \theta)\kappa_{f(\theta)} \geq 0$ implies $(\theta' - \theta)\kappa_{f(\theta')} \geq 0$. Equivalently, if $\theta > \theta'$ and $\kappa_{f(\theta)} = 0$ then $\kappa_{f(\theta')} = 0$. This further means, if $\theta \in \Theta_0$ and $\theta' < \theta$, then $\theta' \in \Theta_0$. Hence, using Inequality 7, we get that if $\theta' > \theta$, then

$$\lambda(\theta')\kappa_{f(\theta')} \geq \lambda(\theta)\kappa_{f(\theta)}. \tag{8}$$

[11] Most recently, Gershkov et al. (2013) study the equivalence of Bayesian and dominant strategy implementation with quasilinear transfers in such environments.

Now, for any $\theta, \theta' \in \Theta$ with $\theta' > \theta$, multiplier 2-cycle monotonicity requires that

$$(\theta' - \theta)\big(\lambda(\theta')\kappa_{f(\theta')} - \lambda(\theta)\kappa_{f(\theta)}\big) \geq 0. \tag{9}$$

This is true because of Inequality 8.

2 ⇒ 3. Using Proposition 3, it is enough to show that if f is multiplier 2-cycle monotone, then it is multiplier cycle monotone. Because f satisfies multiplier 2-cycle monotonicity, for any $\theta' > \theta$, Inequality 9 is satisfied. But, this implies that Inequality 8 is satisfied.

Assume for contradiction that f fails multiplier cycle monotonicity. Let k be the smallest integer such that f fails multiplier k-cycle monotonicity. Since f satisfies multiplier 2-cycle monotonicity, $k \geq 3$. This means for every $\lambda : \Theta \to (0,1]$ and for some finite sequence of types $(\theta^1, \ldots, \theta^k)$, we have

$$\sum_{j=1}^{k} \ell^{f,\lambda}(\theta^j, \theta^{j+1}) < 0,$$

where $\theta^{k+1} \equiv \theta^1$ and $\ell^{f,\lambda}(\cdot,\cdot)$ is as defined in Equation 10. Consider a $\lambda : \Theta \to (0,1]$. Let $\theta^j > \theta^p$ for all $p \in \{1,\ldots,k\} \setminus \{j\}$. We will show that $\ell^{f,\lambda}(\theta^{j-1},\theta^j) + \ell^{f,\lambda}(\theta^j,\theta^{j+1}) - \ell^{f,\lambda}(\theta^{j-1},\theta^{j+1}) \geq 0$. To see this,

$$\begin{aligned}
\ell^{f,\lambda}(\theta^{j-1},\theta^j) + \ell^{f,\lambda}(\theta^j,\theta^{j+1}) - \ell^{f,\lambda}(\theta^{j-1},\theta^{j+1}) &= \theta^j[\lambda(\theta^j)\kappa_{f(\theta^j)} - \lambda(\theta^{j-1})\kappa_{f(\theta^{j-1})}] \\
&\quad + \theta^{j+1}[\lambda(\theta^{j+1})\kappa_{f(\theta^{j+1})} - \lambda(\theta^j)\kappa_{f(\theta^j)}] \\
&\quad - \theta^{j+1}[\lambda(\theta^{j+1})\kappa_{f(\theta^{j+1})} - \lambda(\theta^{j-1})\kappa_{f(\theta^{j-1})}] \\
&= (\theta^j - \theta^{j+1})[\lambda(\theta^j)\kappa_{f(\theta^j)} - \lambda(\theta^{j-1})\kappa_{f(\theta^{j-1})}] \\
&\geq 0,
\end{aligned}$$

where the last inequality follows from the fact that $\theta^j > \theta^{j+1}$ and applying Inequality 8. Since f satisfies multiplier $(k-1)$-cycle monotonicity, we know that $\ell^{f,\lambda}(\theta^1,\theta^2) + \ldots + \ell^{f,\lambda}(\theta^{j-2},\theta^{j-1}) + \ell^{f,\lambda}(\theta^{j-1},\theta^{j+1}) + \ell^{f,\lambda}(\theta^{j+1},\theta^{j+2}) + \ldots + \ell^{f,\lambda}(\theta^k,\theta^1) \geq 0$. But, because of the last inequality, we must have

$$\sum_{j=1}^{k} \ell^{f,\lambda}(\theta^j, \theta^{j+1}) \geq 0,$$

which gives us a contradiction.

Of course, 3 ⇒ 4 and Proposition 1 establishes that 4 ⇒ 1. This concludes the proof. ■

Remark. A closer look at the proof of Theorem 4 reveals that if $\kappa_a > 0$ for all $a \in A$, then for every scf f, $\Theta_0 = \emptyset$, and hence, every scf f satisfies 2-acyclicity vacuously. Thus, *every* scf can be implemented using a royalty contract.

7 Related Literature

Mechanism design with securities originated with the literature on security auctions (Hansen, 1985; Riley, 1988). This paper has been partly inspired by the recent work which discuss the revenue ranking of auctions conducted with different securities (DeMarzo et al., 2005; Che and Kim, 2010; Abhishek et al., 2012). These papers study how a seller's revenue is affected by the "steepness" of securities that are admissible as bids. In contrast to the work on security auctions, our focus is on a general mechanism design environment and our goal is to characterize implementable scfs. Additionally, since we do not focus on auctions, we do not need the space of admissible security contracts to be ordered - securities are completely ranked and better securities provide a higher expected payoff to the seller irrespective of bidder type. This restriction is required in security auctions to ensure that a winner can be declared based on the bids but before the utility is realized. In other words, the royalty mechanisms we consider (which are not ordered) are explicitly prohibited in the security auctions literature. For a recent survey of work on auctions with contingent payments, see Skrzypacz (2013).

This paper is related to the dominant strategy implementation literature in quasi-linear environments, which originated with Rochet (1987). As we have mentioned earlier, he derived a condition called cycle monotonicity which characterized implementability with quasilinear transfers. Recent contributions to this literature (Bikhchandani et al., 2006; Saks and Yu, 2005; Ashlagi et al., 2010; Mishra and Roy, 2013) investigate conditions that are weaker than cycle monotonicity which characterize implementability in such environments with finitely many allocations. In the context of single object auctions, Myerson (1981) showed that a monotonicity condition is equivalent to implementation in Bayes-Nash equilibrium. Myerson's result can be straightforwardly adapted to give a characterization of dominant strategy implementable allocation rules in single object auction framework (see also Laffont and Maskin (1980)). The monotonicity condition in Myerson (1981) is a simplification to Rochet's cycle monotonicity condition in the context of single object auction. A comprehensive review of this literature appears in Vohra (2011).

Perhaps the closest related paper to this work is Rahman (2011) which characterizes implementation in an environment where the principal can observe and condition the mechanism on a noisy signal which is correlated with the agent's type. The environment he considers differs fundamentally from ours in at least two important respects. Firstly, in his model, both the scf and the payments are functions of the signal and the agent's report. Hence, the signals in his model depend only on the agent's type and not on the allocation. By contrast, in our setting, the scf depends only on the reports whereas the contracts depend additionally on the realized utility. Secondly, while we consider general securities, he restricts attention to quasilinear transfers. That said, it should be noted that he considers a signal structure which is more general than ours as he does not impose the ordering condition. A challenging and fruitful problem for future research would be to characterize implementation by securities in an environment where utility distributions depend on allocations but are otherwise unrestricted as in Rahman (2011).

The task scheduling problem in algorithmic mechanism design is related to the implementation of the Rawlsian scf. In the task scheduling problem, a principal is trying to minimize the time taken to complete the entire set of given tasks. These tasks must be allocated to a set of agents whose private information is the time they take to complete the different tasks. Nisan and Ronen (2001) consider a linear environment and argue that no quasilinear transfers can achieve the optimal time. Instead they show that the optimum can be achieved if the principal can condition payments on the realized times of completion. Our characterization could potentially be useful in extending the results of Nisan and Ronen (2001) to general nonlinear environments. Here, the agents may have synergies in production– groups of tasks may be completed in less than the sum of time they would take to complete each task in the group individually. Additionally, the principal may have more complicated preferences in which certain tasks take precedence over others. We leave this interesting problem for future research.

8 Concluding Remarks

In this paper, we study a general version of the classic dominant strategy implementation problem introduced by Rochet (1987). We consider environments where the principal can offer contracts which depend on the random realized utilities of the agents, the distributions of which depend on the private types of the agents and the outcome. This model nests the deterministic quasilinear setting. Our main result shows that any implementable scf is implementable using a simple royalty contract which consists of an upfront payment and a percentage of future profits. The result is similar in spirit to Afriat's theorem of revealed preference. Following Rochet (1987), there has been a large and insightful body of work in dominant strategy quasilinear implementation. We hope that this paper spurs an interest in studying implementation with securities in random environments. To this end, we suggest a few avenues for future research.

In Section 6.2 we examined a linear one dimensional setting where our main result continues to hold even when the type space may be uncountable. A natural extension is then a characterization of infinite type spaces where there is an equivalence between the space of scfs implementable by security and royalty contracts respectively. Moreover, when this equivalence fails, one can ask whether there is a different class of simple contracts which can implement all implementable scfs. As we mentioned earlier, another possible extension is to extend our analysis to type spaces which do not satisfy our ordering condition. For this, the techniques developed in Rahman (2011) may be useful.

Another interesting generalization would be to consider interdependent value settings. Even the utilitarian efficient outcome is difficult to implement in this setting using quasi-linear mechanisms - Maskin (1992) shows that if the utility function of each agent satisfies a single crossing condition then the utilitarian efficient outcome can be implemented. However, Mezzetti (2004) has shown that using two-stage mechanisms that depend on the realized utilities of agents, the utilitarian efficient outcome can always be implemented even in the interdependent values model. We are not aware of work analyzing the implementability of Rawlsian scfs in an interdependent value setting. [12]

More generally, mechanism design with securities is necessary for the design of contracts in environments where the distribution of the agents' utilities depend on their private information and additionally on an action they may undertake which is unobservable to the principal. Here, contracts which depend on the realized utility are necessary to provide the appropriate incentives for the agents to undertake desirable actions. A classic example of such an environment is Laffont and Tirole (1986) where a principal is trying to regulate the cost of an agent who has a private efficiency parameter and can reduce his cost by conducting costly unobservable effort. In their setting, the principal is permitted to use very general contracts and they surprisingly show that the second best solution can be achieved using simple linear contracts (akin to the royalty contracts in our setting). An important question is to provide conditions characterizing implementable scfs in such environments. Additionally, it would be interesting to characterize the subset of implementable scfs which can be implemented using royalty contracts. We hope to answer such questions in future research.

[12] The general issue of incentive constraints becoming too restrictive to implement desirable social choice functions have been considered by many. For instance, in the standard mechanism design framework, Jackson and Sonnenschein (2007) show how to overcome incentive constraints by *linking* decisions of sufficiently many replicas of the same decision problem.

Appendix

Omitted Proofs

Proof of Proposition 3

Proof: Let f be an scf. Define for every $\theta, \theta' \in \Theta$ and for every $\lambda : \Theta \to (0, 1]$,

$$\ell^{\lambda,f}(\theta', \theta) = \lambda(\theta)\mathbb{E}_u[u(\theta, f(\theta))] - \lambda(\theta')\mathbb{E}_u[u(\theta, f(\theta'))]. \tag{10}$$

By definition, f is implementable by a royalty mechanism if and only if there exist $t : \Theta \to \mathbb{R}$ and $\lambda : \Theta \to (0, 1]$ such that for all θ, θ', we have

$$t(\theta) - t(\theta') \le \ell^{\lambda,f}(\theta', \theta). \tag{11}$$

We will denote Inequality (11) as incentive constraint $(\theta \to \theta')$. Suppose f is an scf implementable by a royalty mechanism (r, t). Consider a sequence of types $(\theta^1, \ldots, \theta^K)$ for some integer $K \ge 2$ and denote $\theta^{K+1} \equiv \theta^1$. By adding the incentive constraints (11) for $(\theta^1 \to \theta^2), \ldots, (\theta^K \to \theta^{K+1})$, we get that $\sum_{j=1}^{K} \ell^{r,f}(\theta^j, \theta^{j+1}) \ge 0$. Hence, f is multiplier cycle monotone.

For the converse, suppose f is multiplier cycle monotone. Then, there is a $\lambda : \Theta \to \mathbb{R}$ such that for every integer $K \ge 2$ and $(\theta^1, \ldots, \theta^K, \theta^{K+1})$, where $\theta^{K+1} \equiv \theta^1$, we have $\sum_{j=1}^{K} \ell^{r,f}(\theta^j, \theta^{j+1}) \ge 0$. Then, by the Rochet-Rockefellar theorem (Rochet, 1987; Rockafellar, 1970), there exists $t : \Theta \to \mathbb{R}$ such that for all θ, θ', the incentive constraint $(\theta \to \theta')$ is satisfied. Hence, f is implementable by the royalty contract (λ, t). ■

Proof of Lemma 1

Proof: Consider an scf f. Notice that because of the assumptions in the model, we need to show that there exists a security contract s such that for every $\theta, \theta' \in \Theta$, we have

$$s(\bar{u}(\theta, f(\theta)), \theta) \ge s(\bar{u}(\theta, f(\theta')), \theta'). \tag{12}$$

We will define an incomplete binary relation $\succ_s$, $\sim_s$ over tuples $\{\bar{u}(\theta, f(\theta')), \theta'\}$ for all $\theta, \theta' \in \Theta$. These tuples correspond to a type θ making a report of θ'. We first define the relation $\succ_{s_0}$ and $\sim_s$ now.

$$\begin{aligned}
&\{\bar{u}(\theta, f(\theta')), \theta'\} \succ_{s_0} \{\bar{u}(\theta', f(\theta')), \theta'\} && \text{if } \bar{u}(\theta, f(\theta')) > \bar{u}(\theta', f(\theta')) \\
&\{\bar{u}(\theta', f(\theta')), \theta'\} \succ_{s_0} \{\bar{u}(\theta, f(\theta')), \theta'\} && \text{if } \bar{u}(\theta, f(\theta')) < \bar{u}(\theta', f(\theta')) \\
&\{\bar{u}(\theta, f(\theta')), \theta'\} \sim_s \{\bar{u}(\theta', f(\theta')), \theta'\} && \text{if } \bar{u}(\theta, f(\theta')) = \bar{u}(\theta', f(\theta')) \\
&\{\bar{u}(\theta', f(\theta')), \theta'\} \sim_s \{\bar{u}(\theta, f(\theta')), \theta'\} && \text{if } \bar{u}(\theta, f(\theta')) = \bar{u}(\theta', f(\theta')) \\
&\{\bar{u}(\theta, f(\theta)), \theta\} \succ_{s_0} \{\bar{u}(\theta, f(\theta')), \theta'\} && \text{for all } \theta' \ne \theta
\end{aligned}$$

We define $\succ_s$ as the transitive closure of $\succ_{s_0}$. Formally, we say $\{\bar{u}(\theta, f(\theta')), \theta'\} \succ_s \{\bar{u}(\hat{\theta}, f(\hat{\theta}')), \hat{\theta}'\}$ if there exists a finite sequence $\{\{\bar{u}(\theta^1, f(\theta'^1)), \theta'^1\}, \ldots, \{\bar{u}(\theta^K, f(\theta'^K)), \theta'^K\}\}$ such that

$$\{\bar{u}(\theta, f(\theta')), \theta'\} R_1 \{\bar{u}(\theta^1, f(\theta'^1)), \theta'^1\} R_2 \cdots R_K \{\bar{u}(\theta^K, f(\theta'^K)), \theta''^K\} R_{K+1} \{\bar{u}(\hat{\theta}, f(\hat{\theta}')), \hat{\theta}'\}$$

where $R_k \in \{\succ_{s_0}, \sim_s\}$ and at least one $R_k \equiv \succ_{s_0}$. It is easy to argue that acyclicality of f implies that the relation $\succ_s$ is irreflexive.

Since $\succ_s$ is irreflexive and transitive and Θ is countable, we can then use a standard representation theorem (Fishburn, 1970) which guarantees the existence of a function s which respects $\succ_s$. But this only defines $s(\bar{u}(\theta, f(\theta')), \theta')$ for all $\theta, \theta' \in \Theta$ - note that this s is increasing in its first argument. This can be trivially extended to $s(x, \theta)$ for all $x \in \mathbb{R}$ such that it is increasing in the first argument. By construction of the relation $\succ_s$, this function s satisfies Inequality 12. ■

References

Abhishek, V., B. Hajek, and S. R. Williams (2012): "On Bidding with Securities: Risk aversion and Positive Dependence," *arXiv preprint arXiv:1111.1453.*

Afriat, S. N. (1967): "The Construction of Utility Functions from Expenditure Data," *International Economic Review*, 8, 67–77.

Arrow, K. J. and A. K. Sen (2002): *Handbook of Social Choice and Welfare*, vol. 1, North Holland.

Ashlagi, I., M. Braverman, A. Hassidim, and D. Monderer (2010): "Monotonicity and Implementability," *Econometrica*, 78, 1749–1772.

Bikhchandani, S., S. Chatterji, R. Lavi, A. Mualem, N. Nisan, and A. Sen (2006): "Weak Monotonicity Characterizes Deterministic Dominant Strategy Implementation," *Econometrica*, 74, 1109–1132.

Brown, D. J. and C. Calsamiglia (2007): "The Nonparametric Approach to Applied Welfare Analysis," *Economic Theory*, 31, 183–188.

Carbajal, J. C., A. McLennan, and R. Tourky (2013): "Truthful Implementation and Preference Aggregation in Restricted Domains," *Forthcoming, Journal of Economic Theory.*

Che, Y.-K. and J. Kim (2010): "Bidding with Securities: Comment," *American Economic Review*, 100, 1929–1935.

DeMarzo, P. M., I. Kremer, and A. Skrzypacz (2005): "Bidding with Securities: Auctions and Security Design," *American Economic Review*, 95, 936–959.

Fishburn, P. C. (1970): *Utility Theory for Decision Making*, John Wiley and Sons, New York.

Gershkov, A., J. K. Goeree, A. Kushnir, B. Moldovanu, and X. Shi (2013): "On the Equivalence of Bayesian and Dominant Strategy Implementation," *Econometrica*, 81, 197–220.

Gorbenko, A. and A. Malenko (2010): "Competition Among Sellers in Securities Auctions," *American Economic Review*, 101, 1806–1841.

Hansen, R. G. (1985): "Auctions with Contingent Payments," *American Economic Review*, 75, 862–865.

Jackson, M. O. and H. F. Sonnenschein (2007): "Overcoming Incentive Constraints by Linking Decisions," *Econometrica*, 75, 241–257.

Krishna, V. (2009): *Auction Theory*, Academic Press.

Laffont, J.-J. and E. Maskin (1980): "A Differential Approach to Dominant Strategy Mechanisms," *Econometrica*, 1507–1520.

Laffont, J.-J. and J. Tirole (1986): "Using Cost Observation to Regulate Firms," *Journal of Political Economy*, 614–641.

Maskin, E. (1992): *Privatization*, Mohr, Tubingen, chap. Auctions and Privatization, 115–136, editor: Horst Siebert.

Mezzetti, C. (2004): "Mechanism Design with Interdependent Valuations: Efficiency," *Econometrica*, 72, 1617–1626.

Milgrom, P. R. and R. J. Weber (1982): "A Theory of Auctions and Competitive Bidding," *Econometrica*, 50, 1089–1122.

Mishra, D. and S. Roy (2013): "Implementation in Multidimensional Dichotomous Domains," *Theoretical Economics*, 8.

Myerson, R. B. (1981): "Optimal Auction Design," *Mathematics of Operations Research*, 6, 58–73.

Nisan, N. and A. Ronen (2001): "Algorithmic Mechanism Design," *Games and Economic Behavior*, 35, 166–196.

Rahman, D. (2011): "Detecting Profitable Deviations," Working Paper, University of Minnesota.

Riley, J. G. (1988): "Ex Post Information in Auctions," *Review of Economic Studies*, 55, 409–429.

Rochet, J. C. (1987): "A Necessary and Sufficient Condition for Rationalizability in a Quasi-linear Context," *Journal of Mathematical Economics*, 16, 191–200.

Rockafellar, R. T. (1970): *Convex Analysis*, Princeton University Press.

Saks, M. E. and L. Yu (2005): "Weak Monotonicity Suffices for Truthfulness on Convex Domains," in *Proceedings of* 7^{th} *ACM Conference on Electronic Commerce*, ACM Press, 286–293.

Sen, A. K. (1970): *Collective Choice and Social Welfare*, vol. 5, Holden-Day.

Skrzypacz, A. (2013): "Auctions with Contingent Payments - An Overview," *International Journal of Industrial Organization.*

Varian, H. R. (1982): "The Nonparametric Approach to Demand Analysis," *Econometrica*, 50, 945–973.

Vohra, R. V. (2011): *Mechanism Design: A Linear Programming Approach*, Cambridge University Press.