Definable and Contractible Contracts*

Michael Peters and Balázs Szentes

June 28, 2011

Abstract

This paper analyzes Bayesian normal form games in which players write contracts that condition their actions on the contracts of other players. These contracts are required to be representable in a formal language. This is accomplished by constructing contracts which are definable functions of the Godel code of every other player's contract. We provide a complete characterization of the set of allocations supportable as pure strategy Bayesian equilibria of this contracting game. When information is complete, this characterization provides a folk theorem. In general, the set of supportable allocations is smaller than the set supportable by a centralized mechanism designer.

1 Self Referential Strategies and Reciprocity in Static Games

In this paper we characterize the allocation rules attainable by players in a Bayesian game when they have the ability to commit themselves by writing contracts that condition on other players' contracts.

The idea that contracts might condition on other contracts is not new in economics. The most commonly known expression of this idea is well known in the industrial organization literature (e.g. (Salop 1986)) as the 'meet the competition' clause in which one firm commits itself to lower its price when any of its competitors does. A similar idea appears in trade theory as the principle of *reciprocity* ((Bagwell and Staiger 2001)). Countries enact trade legislation in which they agree to abide by a trade agreement like GATT. Such legislation commits the country to lowering tariffs in response to trade legislation by another country that lowers tariffs, provided this other country's legislation agrees to abide by GATT. Finally, tax treaties sometimes have this flavor - for example, out of state residents who work in Pennsylvania are exempt from Pennsylvania tax as long as they live in a state that has a 'reciprocal' agreement that exempts out of state residents (presumably from Pennsylvania) from state taxes.[1]

*We would like to thank the editor and three referees for many useful suggestions that we ended up incorporating into the paper.

[1] http://www.revenue.state.pa.us/revenue/cwp/view.asp?A=238&Q=244681

In all of these examples, commitments are made that are conditional on commitments of others, and are used to support cooperative outcomes. The literature treats these situations as static games of complete information. Additionally, the contracts that are used to support equilibrium are idiosyncratic, so that only the simplest most stylized problems are amenable to analysis. For example, in the 'meet the competition argument', firm A offers to sell at a high price provided its competitor, firm B, also sets a high price. If instead, B offers any price below the highest price, A commits itself to sell at its marginal cost. If B believes this commitment, then one best reply for B is to set the highest price. In the trade and taxation treaties mentioned above, a state cooperates by offering a 'reciprocal' contract that cooperates if and only if the other state does the same.

(Tennenholtz 2004) suggests a way to model mutually dependent commitment devices. His players compete using computer programs that condition their actions on other programs. He shows that all individually rational outcomes in complete information games can be supported as equilibria.[2] Basically, two programs implement cooperative actions if they know that they have the same syntax, otherwise, they punish each other. As he is only concerned with showing that individually rational outcomes can be supported as program equilibrium, he does not give a complete description of what the set of possible programs looks like. (A.T. Kalai and Samet 2010), whose method we illustrate below, specifically construct a set of commitment devices for a two player game of complete information with the property that outcome functions are supportable as equilibria in these commitment devices if and only if they are individually rational.

This paper considers a two-stage contracting game in a Bayesian environment. At the first stage, players offer contracts. A contract restricts the action spaces of a player as a function of the other contracts. At the second stage, players take actions from their restricted action spaces. Our objective in this paper is twofold. First, we identify two properties of any abstract contract space that lead to a complete characterization of supportable outcomes. We refer to these properties as 'cross-referentiality' and 'invariant punishment'. Cross referentiality is a generalization of the Tennenholz idea that contracts can recognize each other. Invariant punishment means that if a player wants to, he can write contracts that commit him to any of his pure actions, while inducing the same reaction from the other players each time. It is this property that allows us to show that players' payoffs cannot be held below their individually rational level in games of complete information.[3] Second, we show that there exists a contract space which satisfy these properties. We assume that the set of feasible contracts is a set consisting of finite texts written in some language. Conditional on the language, this description of contracts is natural and has the advantage that the set of contracts is universal in the sense that the same set of feasible contracts can be used to model competition in every environment. This set is at once rich, descriptive and must, within

[2] By 'individually rational outcome' in a game of complete information, we mean an outcome in which each player receives at least his minmax payoff.

[3] More specifically, if players can react to the outcomes of a deviation by appropriately crafting their contract, then they could conceivable respond to the action that a deviating player takes. This would make it possible to keep some players' payoffs at their maxmin level, instead of just their minmax.

limits set by the language in which contracts are written, be robust to the introduction of new contracts. We explain how these contracts can be written so that they specify actions that are conditional on the contracts of other players.

We use our methods to show how to extend the static complete information folk theorem results of papers like (Tennenholtz 2004) or (A.T. Kalai and Samet 2010) to games of incomplete information. We provide a complete characterization of outcomes that can be supported as (pure strategy) equilibria in finite contracting games of incomplete information. Though there is no agreed definition in the literature of what a folk theorem would say in games with incomplete information, our results do indicate that contracts are more restrictive in such games. In the complete information case, it is known that outcomes in which every player receives more than his or her minmax payoff can be supported as an equilibrium in contracts with the appropriate choice of commitment devices. One way to think of such outcomes is that they are the outcomes that could be supported by a centralized mechanism designer, who can enforce actions of all players who agree to participate in his mechanism, and who minmaxes players who unilaterally decide not to participate. In the incomplete information case, we show that there are outcome functions that a centralized mechanism designer like this can support, but which cannot be supported with contracts.

The reason that a 'folk theorem' like result doesn't hold with incomplete information has to do with 'participation'. A player who doesn't play along with some cooperative agreement enforced by contracts will still observe any information that the contracts themselves convey about the types of the other players. In the play of the ensuing game, the deviator can make use of this information when choosing his actions. To capture this, we show that outcome functions are supportable if and only if they are supportable by a mechanism designer who can condition actions only on publicly observable messages.

1.1 Contractible Contracts

When trying to describe a broad set of feasible contracts, it is easy to get lost in complexities associated with the infinite regress that arises when a contract specifies a commitment that depends on whether another contract specifies a commitment that depends on whether the first contract specifies... One way to get around this infinite regress is simply to impose ad hoc restrictions on the set of feasible contracts to ensure that the infinite regress doesn't arise. This is the approach developed in (A.T. Kalai and Samet 2010). We explain this approach briefly in order to motivate the broad approach that we adopt.

We can apply the argument in (A.T. Kalai and Samet 2010) to a simple two player prisoner's dilemma. We want to construct a set of commitment devices that will support cooperation. Define a contract called 'this contract' which works as follows:

$$\text{this contract} = \begin{cases} C \text{ if other player's contract} = \text{this contract} \\ D \text{ otherwise.} \end{cases}$$

If both players in the prisoners dilemma offer 'this contract', then they are unambiguously obliged to cooperate. If one of the players offers something else, then the other is unambiguously required to defect. As long as the deviator's contract specifies some unambiguous outcome against 'this contract', cooperation is a Nash equilibrium. Depending on what other contracts are described by the word 'otherwise' in the sentence above, there may be many other possible equilibrium outcomes as well.

To complete the description of the contracting game, let Θ be a set of feasible contracts defined in such a way that $\theta \in \Theta \implies \theta : \Theta \to A$. We are about to judiciously construct Θ. We have just described a reciprocal contract θ^* such that

$$\theta^* \left(\theta'\right) = \begin{cases} C \text{ if } \theta' = \theta^* \\ D \text{ otherwise.} \end{cases}$$

Suppose we simply add to this a pair of 'constant' contracts $\theta_c \in \Theta$ such that $\theta_c \left(\theta'\right) \equiv C$ and a contract $\theta_d \in \Theta$ such that $\theta_d \left(\theta'\right) \equiv d$. The constant contracts could be used to support the Nash equilibrium in the prisoners dilemma in the obvious way. Since the outcomes (C, C) and (D, D) are the only two outcomes for which all players receive at least their minmax payoffs, the collection consisting of the contract θ^* and the two constant contracts already supports all outcomes for which each player receives at least her minmax payoff. Furthermore, if $\Theta = \{\theta^*, \theta_c, \theta_d\}$, it is pretty clear that no other (pure) outcomes can be supported as equilibrium in contracts. In other words, now that we have the set Θ, we have a complete characterization of the set of equilibrium outcomes. This characterization amounts to a folk theorem.

One desirable feature of contracts that this approach lacks is that they be robust to contractual innovation. Absent such a property, one is never sure whether economic properties that emerge from a contractual model aren't just artifacts of the way the contracts are modelled. Ideally, if we want commitment devices that can condition on the devices used by others, it would be desirable that the set of feasible contracts or commitment devices would include *all* functions from itself into the set of feasible actions. This would ensure that any new contract we could dream up would already be a feasible contract. Unfortunately, this is impossible because the cardinality of the set of functions with a given domain is larger than the cardinality of that domain (Cantor's Theorem). Our approach is instead to describe the largest set of contracts that can be written in a finite set of characters using first order logic.

This isn't simply a theoretical issue. Observe that in the formulation above, the set of contracts made available seems much too restrictive. Contracts depend on other contracts, but in a very limited way. If a player offers the contract that he is 'supposed' to offer, then things go well.

Otherwise something bad happens. Yet this bad outcome can't depend in any way on what the deviator actually does. In other words, the punishment imposed on a deviator only depends on the deviator's identity and not on the actual deviation. This is not at all a natural property of contracts that depend on other contracts. In (A.T. Kalai and Samet 2010) and (Tennenholtz 2004), this is true by assumption.

Relaxing this constraint on contracts creates a difficulty when attempting to provide a full characterization of supportable outcomes. The reason is that natural bounds on outcomes like the minmax payoff themselves rely on this property. When players minmax a deviating player, their actions don't vary with the action the deviator takes. If contracts allow reactions to depend on deviations, then in principle, it might be possible to support outcomes in which some players' payoffs are below their minmax payoffs. In the complete information environment, there is an easy fix to this problem, which is to allow players to offer contracts which do not specify a single action, but rather specify a subset of their action spaces. With such a contract space, a deviator can always offer a contract which does not restrict his action space at all, and then best-respond to the action profile of the others. Then the worst possible punishment players can impose on a deviator is indeed an action profile which minmaxes him. Therefore, restricting attention to punishments which are independent of the actual deviation is without loss of generality. Unfortunately, this same approach fails in games with incomplete information. The reason is that players who punish a deviator might not restrict their action spaces to a single action precisely because they might not want to reveal too much information to a deviator. Therefore, following a deviation, even the non-deviators will be forced to choose their actions strategically. Hence, a deviator might actually benefit from restricting his action space.

Furthermore, the set of feasible contracts described above is specially tailored to the economic problem it is applied to. The commitment devices we just described obviously won't be much use in a game with more than two players or more than two actions. Indeed, if we simply relabel the actions so that C stands for defection and D stands for cooperation, then these commitment devices will support only the non-cooperative outcome. We can create new sets of commitment devices to handle these changes without much problem. What we would like to do instead, is to provide a set of commitment devices that always works.

Perhaps more important, the contracts are obviously chosen because we know we want to support a specific pair of actions. If we perturb the payoffs in a way that changes the essential economics of the problem (for example, by making the socially desirable outcome (C, D)) then we have to reconstruct the set of contracts to get the result we want. We provide a set of contracts that can be described independently of the (payoffs in the) game to which they are applied. So there are two ways in which our contracts are universal. First, any particular set of contracts, like the contracts we described above, can be rewritten in our language so that they are embedded in the larger set of contracts we describe. Second, the set of contracts we describe can be used to model contractual competition no matter what the actual payoffs in the game are.

Perhaps the main contribution of our approach is to show that even when contracts are universal in the sense that we have described, we can still understand the economic logic of contractual situations using a modified version of the 'minmax a deviator' logic that is described above.

1.2 How Definability Works

We accomplish this universality by allowing players to write finite texts in a first order arithmetic language that describes their commitments. This allows players to punish deviators in potentially very complex ways and involves no ad hoc restrictions like those above. The way we do this is to observe that the texts associated with contracts can be reinterpreted as definable functions of the Gödel codes of texts written by other players. Since *Gödel* codes and definable functions are unfamiliar to most economists, we give an informal description of the method below.

We start by endowing each player with a formal language and require each contract to be a text written in this language. A text is a finite string of symbols. It is well known that there are bijections from the set of texts into the set of integers. One such mapping is called the *Gödel Coding.* This implies that any contract uniquely corresponds to an integer. A contract of a player is a mapping from contract profiles to subsets of his action space. Since the contracts correspond to integers, one can think of such a contract as a description of an arithmetic correspondence from the codes of contracts to the codes of the names of the actions. There is a well-known set of arithmetic correspondences, called the *definable correspondences,* which can be precisely described in the formal language by using finitely many characters. (We shall formally define this set later.) Hence, one can think of the contract space as the set of definable functions from $\mathbb{N}^m \rightarrow 2^{\mathbb{N}}$, where m is the number of players. The domain of these functions are the vectors of the codes of the players' contracts and the range of these functions are the subsets of the codes of the names of the actions. We identify the contract space of a player with the set of definable correspondences.

To see how our approach works, return to the simple prisoner's dilemma game. Let $[c]$ denote the Gödel code of the contract c and refer to $[c]$ as the 'encoding' of c. For any pair of contracts c_1 and c_2, the action (C or D) taken by Player 1 is $c_1\left([c_2]\right)$ and similarly for Player 2. Since every pair of actions determines a payoff, this procedure associates a unique payoff with every pair of contracts.

There are many things that aren't definable contracts that also have Gödel codes. We want to make use of some of these other things. In particular, we want to use definable functions with *free variables.* Interpreting n as the encoding of the other player's contract, here is a definable contract with a free variable

$$\gamma_x\left(n\right) = \begin{cases} C & n = x, \\ D & \text{otherwise.} \end{cases}$$

A free variable has the natural interpretation that x can take on any integer value. Definable contracts with free variables also have Gödel codes. The contract with free variable that we want is a slight modification of the one above, in particular

$$c_x(n) = \begin{cases} C & n = \left[< x >^{(x)}\right], \\ D & \text{otherwise.} \end{cases} \tag{1}$$

is also a contract with a free variable. The mapping $< x >^{(x)}$is the composition of two functions. First, the function $< x >$ is the inverse operation to the Gödel coding. That is, $< n >$ is the text whose Gödel code is n. Second, if ϕ is a text with one free variable, then $\phi^{(n)}$ is the same text where the value of the free variable is set to be n. Hence, if n is a Gödel code of a definable contract with one free variable, then $< n >^{(n)}$ is itself a definable contract (without a free variable). $\left[< n >^{(n)}\right]$ is just the Gödel code of whatever this definable contract happens to be.

We want to create a contract by fixing a very specific value for x in (1). In particular, the value of x we are interested in is $[c_x]$. Since $[c_x]$ is the Gödel code of a contract with a free variable, the right hand side of (1) requires that we decode $[c_x]$ to get c_x, then fix x at $[c_x]$ to get the contract $c_{[c_x]}$. Putting all this together gives

$$c_{[c_x]}(n) = \begin{cases} C & n = \left[c_{[c_x]}\right] \\ D & \text{otherwise} \end{cases}$$

So

$$c_{[c_x]}([c_2]) = \begin{cases} C & [c_2] = \left[c_{[c_x]}\right] \\ D & \text{otherwise} \end{cases}$$

This is the contract which corresponds to the one we called 'this' contract or θ^* in our discussion above. The difference is that this contract now reacts to a much broader set of contracts than to what θ^* did. The contract θ^* could only respond to itself and to the two constant contracts. The contract $c_{[c_x]}$ responds to any definable function (in fact it specifies an action for every finite text).

To press the analogy with θ^* in the problem above, if Player 2 also uses strategy $c_{[c_x]}$, then $[c_2] = \left[c_{[c_x]}\right]$, which evidently triggers the cooperative action by Player 1. The same argument applies for Player 2. Player 2 can deviate to any alternative definable strategy c' that she likes. Since every definable strategy has a Gödel code, the reaction of Player 1, and consequently both players' payoffs are well defined. As the Gödel coding is injective, $c' \neq c_{[c_x]}$ implies the Gödel code of c' is not equal to $\left[c_{[c_x]}\right]$, and the deviation by Player 2 induces Player 1 to respond by switching from C to D.

What this argument illustrates is that our contract space is large enough that we can always find a contract corresponding to θ^* in our existing set of contracts, without having to construct it explicitly from the details of the game. This is the property that makes our contracts universal in the sense that exactly the same set of contracts can be used to characterize equilibrium outcomes in all games.

The introduction of Gödel coding into our model requires some explanation. Take it as given that contracts must be expressed in a formal language. Then a contract of a player must give *precise instructions* on how to restrict his action space as a function of the texts submitted by the other players. In order to describe the contracting game, one must carefully define the notion of "precise instructions" and the set of those texts which give these instructions. Any such definition would lead to a definition of a set of arithmetic correspondences which can be described as finite texts. To see this, suppose that there is a text which gives instructions on how to pin down a subset of a player's action space as a function of the other texts. Then, there is also a text that gives the same instructions as a function of the Gödel Codes of texts of other players instead of their texts. This is because the Gödel Coding and its inverse are definable functions, that is, they can be described as texts.[4] This implies that this new text describes an arithmetic correspondence. In the paper we adopt the definition of definable functions from Number Theory instead of introducing a new definition.

Since, the set of definable functions is the largest set of arithmetic functions which can be described in a first-order language, our contract space is the largest given the restriction to contracts which can be expressed as texts. Implicitly, our approach makes it possible for players to offer any finite text as a contract. We simply identify the original text with the corresponding definable mapping.

In the Prisoner's Dilemma example above, we assume that a contract of Player 1 is a definable function, say $c(x)$, and the Player 1 can take action $a\,(\in \{C, D\})$ if and only if the code of a is in the set $c([c_2])$, where c_2 is the code of Player 2's contract. If we allow players to write any text as contracts, then they could write down the following text:

"My contract can be described by the following definable function: $c(x)$, where the interpretation of $c(x)$ is the following. If the code of the text of the contract of Player 2 is x, then I can only take action a if the Gödel Code of a is in $c(x)$. Finally, the Gödel Code of a text is defined as follows: ..."

Since the Gödel Coding is a recursive function, the '...' can be replaced by a precise description of this coding. All we do is to identify the text above with $c(x)$.

2 Literature

As we mentioned in the introduction, our paper is not the first to show how contractual devices can be used to support cooperative play. Much of the literature in this area follows an idea developed by (Fershtman and Judd 1987) in which actions are delegated to an agent who is given the appropriate incentives to carry out actions that might not otherwise be a part of a non-cooperative equilibrium. This idea was developed by (Katz 2006) who used it to prove a 'folk theorem' for a very specialized environment.

[4] In particular, this implies that players don't need to agree to use the Godel code of other contracts. They can use the Godel code unilaterally, and the implications of the contract will be understood by the others provide they agree on the underlying language in which contracts are written.

The idea that agents might report deviations provides the basis for the menu theorems in common agency, like (Martimort and Stole 1998), (Peters 2001) and (Han 2006). (Martimort and Moreira 2007). Recently (Yamashita 2010) suggests a method that can be used to extend the common agency approach to games in which there are many agents.[5]

The idea that principals could learn about deviations by communicating with agents is developed in (Epstein and Peters 1999). They show that for every environment, there is a universal set of mechanisms for that environment such that any set of indirect mechanisms used to model competition between principals can be embedded in that universal set. Each element of this universal set is described by a sequence of payoffs. Since each indirect mechanism that a principal can offer corresponds to some sequence of payoffs, agents can report the mechanisms being used by other principals by reporting the sequences that correspond to each of the other principals' mechanisms. In this sense, the agent's type corresponds with his usual payoff type along with a sequence representing the mechanisms of the other principals. Every equilibrium in a competing mechanism game can then be represented as an equilibrium in a game where principals offer universal mechanisms and agents truthfully report both their payoff type and their market information.

There are a couple of important differences between our paper and theirs. First of all, our formulation makes it possible to provide a characterization of the outcome functions that are supportable as equilibria. This is a major advantage over (Epstein and Peters 1999) which only provides a set of contracts that might be used to support equilibrium. Secondly, players offer contracts that condition directly on contracts of other players instead of asking agents to describe these contracts. In our model there is no communication at all between players after contracts have been announced, so there are really no agents at all. Indeed we illustrate that despite this limit on players' ability to communicate, contracts support a rich set of type contingent outcome functions. Our main theorem shows that contracts are equivalent to a mechanism in which players communicate their type information publicly. In the private value case, contracts support all the outcome functions that can be supported by a centralized mechanism designer, illustrating the flexibility of this approach.

However the primary objective of (Epstein and Peters 1999) and our paper is the same - to find a language that makes it possible to describe contracts that depend on other contracts. The essential conceptual difference is that (Epstein and Peters 1999) create a language to describe contracts that uses sequences of payoffs from the game to describe contracts. In this sense, the approach in (Epstein and Peters 1999) resembles the approach in (A.T. Kalai and Samet 2010) that we described above, except that the contracts in (Epstein and Peters 1999) are immune to contractual innovation. So the contracts that 'work' in one game won't work in a new game with a different payoff structure. As we mentioned, the contracts that we describe work for all games. The cost of our 'universality' is that the set of contracts we describe is countable for the finite environments we are interested in. The universal set of contracts in (Epstein and Peters 1999) is

[5] See (Peters and Troncoso-Valverde 2009) for a full characterization of supportable outcomes using his method.

finite in finite environments because the set of payoffs is finite.

Our approach is closely related to ideas in the computer science literature. One paper we have already mentioned that uses this approach is (Tennenholtz 2004). He has players writing programs that determine their actions. Using an idea due to von Neumann, he allows these programs to use other programs as data, which has the effect of making the output of each player's program depend on the other players' programs. We illustrated the idea with our description of 'this contract' in the introduction. Tennenholz doesn't give a complete description of the set of feasible contracts. We explain below how our approach differs from the assumption that players choose Turing machines to play against one another.

The paper by (A.T. Kalai and Samet 2010) gives a complete characterization of equilibrium outcomes in two-player complete information games. They construct a set of (game specific) commitment devices which can be used to support *correlated* strategies in which all players' payoffs exceed their minmax payoffs. Specifically, in some games their devices support outcomes in which all players receive payoffs that exceed their best payoffs with Tennenholz's programs. This is accomplished by constructing commitment devices that allow players to correlate their actions while using independent randomizing devices. We extend part of their argument to games of incomplete information. We do not deal with correlation, or any other form of randomization simply because we are trying to keep a notationally and technically demanding problem relatively simple.

Finally, the problem we model is one in which privately informed players offer contracts which will depend on their types in most Bayesian equilibria of the contracting game. Since our interest is in contracts rather than mechanism design, we do not allow players to communicate privately after agreeing to the contract like it is done in an informed principal problem (Myerson 1983). Despite this, we are able to show that contract equilibria support a rich set of type contingent outcomes. Indeed, this is a major advantage of our approach over the simple 'cooperate or be minmaxed' approach in the computer science literature since contracts have to respond to other players' contracts in a much more sophisticated way to make it possible for one player's action to depend on another player's type without any explicit communication. We could not have shown this as effectively if we had allowed private communication. Of course, the results we have about the limits of contract equilibrium arise from this restriction. We could support a larger class of outcome functions if we allowed private communication.

3 The Model

There are m players indexed by $i \in \{1, ..., m\}$. Player i has a finite action space A_i, and let A denote $\times_{i=1}^{m} A_i$. Player i has a type t_i drawn from a finite set T_i, and let T denote $\times_{i=1}^{m} T_i$. Each type profile has a strictly positive probability and the joint distribution of types is common knowledge. The payoff of Player i is $u_i : A \times T \to \mathbb{R}$.

Each Player i is endowed with a contract space C_i and let C denote $\times_{i=1}^{m} C_i$. Each element of C_i defines a mapping from C to $2^{A_i} \backslash \{\emptyset\}$. Let $\widetilde{c}_i$ denote the mapping induced by c_i. That is, a contract of Player i specifies a nonempty subset of his action space for each contract profile. It is important to note that we do not identify a contract with the mapping it defines. This allows for the possibility that two different contracts induce the same mapping, that is, $\widetilde{c}_i(c) = \widetilde{c}'_i(c)$ for all $c \in C$, but $c_i \neq c'_i$.

The contracting game takes place in two stages. In the first stage, each player submits a contract from his contract space simultaneously. Let $c_i \in C_i$ denote the contract submitted by Player i. For each Player i, consider the following subset of A_i determined by the contract profile $c = (c_1, ..., c_n)$:

$$S_i(c) = \{a_i : a_i \in \widetilde{c}_i(c)\}.$$

In the second stage, Player i takes action from $S_i(c)$ simultaneously. As always $S(c) = \times_i S_i(c)$.

In what follows we restrict attention to pure strategies. A strategy of Player i consists of a mapping from his type space to his contract space and a mapping from his types and first-stage contract profiles to his action space. Let Γ_i denote the first stage strategies of Player i, that is,

$$\Gamma_i = \left\{\gamma_i : \ \gamma_i \in C_i^{T_i}\right\},$$

where $C_i^{T_i}$ denotes the set of functions with domain T_i and range C_i. Similarly let $\mathcal{A}_i$ denote the set of second-stage strategies of Player i, that is,

$$\mathcal{A}_i = \left\{\alpha_i : \ \alpha_i \in A_i^{T_i \times C} \text{ and } \forall t_i \in T_i, \forall c \in C, \alpha_i(t_i, c) \in S_i(c)\right\}.$$

Let $\gamma(t)$ denote $(\gamma_1(t_1), ..., \gamma_m(t_m))$ and $\alpha(t, c) = (\alpha_1(t_1, c), ..., \alpha_m(t_m, c))$ for all $t \in T$ and $c \in C$. The strategy profile $(\gamma^*, \alpha^*) \in (\times_i \Gamma_i) \times (\times_i \mathcal{A}_i)$ constitute a Bayesian equilibrium if and only if for all $i \in \{1, ..., m\}$, $t_i \in T_i$, $\gamma_i \in \Gamma_i$ and $\alpha_i \in \mathcal{A}_i$:

$$\begin{aligned} & E_{t_{-i}}\left(u_i\left(\alpha^*\left((t_i, t_{-i}), \gamma^*(t_i, t_{-i})\right), (t_i, t_{-i})\right) \mid t_i\right) \\ \geq \quad & E_{t_{-i}}\left(u_i\left(\overline{\alpha}\left((t_i, t_{-i}), \overline{\gamma}(t_i, t_{-i})\right), (t_i, t_{-i})\right) \mid t_i\right), \end{aligned} \tag{2}$$

where $\overline{\alpha} = \left(\alpha_i, \alpha^*_{-i}\right)$ and $\overline{\gamma} = \left(\gamma_i, \gamma^*_{-i}\right)$.

A deterministic outcome function in our model is a mapping from T to A. We say that an outcome function $s : T \to A$ is supportable as a Bayesian equilibrium in the contracting game if there is a Bayesian equilibrium (γ^*, α^*) such that $\alpha^*(t, \gamma^*(t)) = s(t)$ for all t.

Bayesian equilibrium imposes no restriction at all on the second stage actions $\alpha^*_{-i}\big(t_{-i}, \left(\gamma_i, \gamma^*_{-i}(t_{-i})\right)\big)$ that Player i anticipates when he deviates, apart from $\alpha^*_{-i}\left(t_{-i}, \left(\gamma_i, \gamma^*_{-i}(t_{-i})\right)\right) \in S_{-i}\left(\gamma_i, \gamma^*_{-i}(t_{-i})\right)$. For example, $\alpha^*_j\left(t_j, \left(\gamma_i, \gamma^*_{-i}(t_{-i})\right)\right)$ could be strictly dominated for Player j with type t_j by some other action in $S_j\left(\gamma_i, \gamma^*_{-i}(t_{-i})\right)$. For this reason, it may be that refinements of equilibrium are necessary in applications to rule out this kind of second stage behavior.

Refinements are completely incidental to our formalism, although they are obviously going to make a difference to the precise set of outcome functions that are supportable as equilibria. The less controversial refinements, like sequential equilibrium and perfect Bayesian equilibrium do not fit easily into our environment since our contract spaces aren't necessarily finite and player types can be correlated. The detailed formalism we need to modify these concepts takes us well beyond our main purpose in this paper. For these reasons, we restrict attention to equilibria where players do not take strictly dominated actions in the continuation games generated by a contract profile. In the appendix, we provide a more abstract description that describes refinements in a manner that is independent of the particular game that is being played (see Section 8.1).

For each $t_{-i} \in T_{-i}$ and $\overline{A} = \times_{i=1}^{m} \overline{A}_i \subset A$, define $\mathcal{R}_i \left(\overline{A}, t_{-i} \right)$ as the set of action profiles $a_{-i} \in A_{-i}$ such that for each $j \neq i$, a_j is not strictly dominated[6] for Player j when his type is t_j, given that the players are constrained to choose actions in $\overline{A}$. We say that (γ^*, α^*) is an $\mathcal{R}$-equilibrium of the contracting game if (2) holds, and in addition for every i, $\gamma_i \in C_i$, and $t_{-i} \in T_{-i}$

$$\alpha^*_{-i} \left(t_{-i}, \overline{\gamma} \left(t_{-i} \right) \right) \in \mathcal{R}_i \left(S \left(\overline{\gamma} \left(t_{-i} \right) \right), t_{-i} \right), \tag{3}$$

where $\overline{\gamma} \left(t_{-i} \right) = \left(\gamma_i, \gamma^*_{-i} \left(t_{-i} \right) \right)$.

3.1 Properties of the Contract Space

Our main characterization theorem relies on two properties, which we describe formally in this section. We have already mentioned these properties - cross referentiality, and invariant punishment. Cross-referentiality is intended to generalize the term 'this contract' that we used to support cooperation in the prisoner's dilemma game in the introduction. The complication is that, generally, a contract has to respond to many players of who have many different types. For each such player and type, a contract needs to recognize the particular contract for that player and the type which reciprocally recognizes the player's own contract. Formally

Cross-referential Property. For all $(N_1, ..., N_m) \in \mathbb{N}^m$, $N_i \geq 1$, for all functions $r_i : \mathbb{N}^m \to 2^{A_i} \backslash \{\emptyset\}$, $p_i^j : \mathbb{N}^{m-1} \to 2^{A_i} \backslash \{\emptyset\}$ $(i, j \in \{1, ..., m\}$, $i \neq j)$, there exists a set of contracts for all $i \in \{1, ..., m\}$, $\{c_i^{n_i}\}_{n_i \in \{1, ... N_i\}} \subset C_i$, such that $c_i^{n_i} \neq c_i^{n'_i}$ if $n_i \neq n'_i$ and:

$$\widetilde{c}_i^{n_i} \left(c_1, ..., c_m \right) = \begin{cases} r_i \left(n_i, n_{-i} \right) & \text{if } \forall k \neq i \ c_k^{n_k} = c_k, \\ p_i^j \left(n_i, n_{-ij} \right) & \text{if } \forall k \neq i, j \ c_k^{n_k} = c_k \text{ and } \nexists n_j \text{ s.t. } c_j^{n_j} = c_j, \\ A_i & \text{otherwise.} \end{cases} \tag{4}$$

Notice that the conditions on the right-hand-side of (4) explicitly depend on (cross referential contracts) $c_j^{n_j}$, and vice versa, the mapping determined by $c_j^{n_j}$ explicitly depends on $c_i^{n_i}$. If each

[6] Here, strictly dominated means "no matter what he believes about the types of his opponents". This is intended to be a very weak refinement. In particular, we ignore here any information j might have learned from the contracting phase when evaluating whether or not a particular action is strictly dominated.

Player j offers a contract $c_j^{n_j}$ for some $n_j \leq N_j$, then $c_i^{n_i}$ restricts the action space of Player i to be $r_i(n)$ where $n = (n_1, ..., n_m)$. If each player offers such a contract except Player j, then $c_i^{n_i}$ restricts the action space of Player i to be $p_i^j(n_{-j})$. In any other cases, $c_i^{n_i}$ imposes no restrictions on the actions of Player i. Intuitively, $\{c_i^{n_i}\}_{n_i}$ can be thought of as the set of contracts from which Player i is supposed to choose in an equilibrium. If each player chooses from these sets, then the second-stage restrictions are defined by the functions $\{r_i\}_i$. If a single player, say Player j, offers a contract which is not in $\{c_j^{n_j}\}_{n_j}$, then the contracts of the other players restrict their action spaces according to the functions $\left\{p_i^j\right\}_i$. One can think about these functions as the *contractual punishments* imposed on deviators.

In the construction given above, the reaction of each player when one of the others fails to use one of the cross referential contracts is independent of what contract this player actually offers. This is similar to the approach we used in the prisoner's dilemma problem. The next property is used to ensure that the logic associated with fixed punishments like this will still be valid in richer contract spaces.

This property requires, that no matter what contracts the other players are offering, the contract space is rich enough that the remaining player is able to write contracts that will commit him to any subset of his actions that he wants while inducing the same commitments from his opponents in response to each different subset. Of course, any sensible contract space will let a player commit himself to an arbitrary subset of his actions. Yet the player must make these commitments by announcing different contracts. The other players' contracts make commitments that depend on the contract this player offers. So the other players' commitments will generally change as the remaining player's commitment changes. We want the contract space to be rich enough that the player can write the contracts in such a way that the different commitments he makes induce exactly the same commitments from his opponents.

We don't need to know what these contracts are, or what commitments they elicit from the other players. All we require is that these contracts exist so that there is *some* fixed commitment a player can elicit from the others. What this property delivers is the fact that in any equilibrium, a player must do at least as well as he does against this fixed commitment. This ensures that all equilibrium outcomes can be implemented with the cross referential contracts we described above. Formally

Invariant Punishment Property. For all $(N_1, ..., N_m) \in \mathbb{N}^m$, $N_i \geq 1$, for all set of contracts $\left\{c_j^{n_j}\right\}_{j, n_j \in \{1, ... N_j\}}$ ($c_j^{n_j} \in C_j$, $c_j^{n_j'} \neq c_j^{n_j}$ if $n_j' \neq n_j$), and for every i, there are functions $p_k^i : \mathbb{N}^{m-1} \to 2^{A_k} \setminus \{\emptyset\}$ ($k \neq i$), such that for any function $f_i : \mathbb{N}^{m-1} \to 2^{A_i} \setminus \{\emptyset\}$, there is a contract $c_i^* \in C_i$ such that for all $n_{-i} \in \times_{j \neq i} \{1, ..., N_j\}$:

$$\widetilde{c}_i^* \left(c_i^*, \left(c_j^{n_j} \right)_{j \neq i} \right) = f_i(n_{-i}) \tag{5}$$

and for all $k \neq i$

$$\widetilde{c}_k^{n_k} \left(c_i^*, \left(c_j^{n_j} \right)_{j \neq i} \right) = p_k^i(n_{-i}). \tag{6}$$

Again, the set $\{c_i^{n_i}\}_{n_i}$ can be thought of the collection of those contracts from which Player i chooses depending on his type in an equilibrium. Given the strategies of the others, each alternative contract that he offers induces a commitment correspondence $f_i\left(n_{-i}\right)$ and elicits some kind of response by the others. The Invariant Punishment Property guarantees that there must exist some collection of punishment correspondences, $\left\{p_j^i\right\}_{j \neq i}$, such that for *any* commitment correspondence f_i that Player i wants, he can write his own contract in such a way that the others respond with exactly the same punishment $\left\{p_j^i\right\}_{j \neq i}$.

In Section 5 we specify a contract space which satisfies both of these properties. This space is going to be the set of definable functions. There are other spaces which contain cross-referential objects. One such a space is the set of Turing machines which is often used in game theoretic analysis. One can even think about the programs in Tenneholz (2006) as Turing machines, who used this space to model contracts in a complete information environment. If we modeled the contracts by Turing machines, then the input of a player's machine would be the descriptions of the machines submitted by the other players, and the output would be a subset of the player's action space.

This space would satisfy the Cross-referential Property, but not the Invariant Punishment Property. This is because players could submit *universal machines* that would simulate the machine of a deviator. Once the simulation is completed, these machines could recommend an action profile which is worse for the deviator. This action profile can depend on the result of the simulation, that is, on the actual deviation. This suggests that players can push the payoff of a deviator below his minmax value. Of course, a deviator could also submit a universal machine that simulates the machines of the others, and then best-responds to their outputs. The problem is that, in general, these universal machines will not halt on each other. Indeed, it is not clear how one can properly define the game because of this halting problem.

4 The Characterization Theorem

There are several ways of stating our characterization theorem. One of our objectives is to compare the set of equilibrium outcomes of the contracting game to the set of outcome functions implementable by a centralized mechanism designer who can control the actions of all the players who agree to participate in his mechanism. To help illustrate the relationship, we define a class of mechanisms, called *Public Message Mechanisms* (PMM) and show that the set of equilibrium outcomes relative to these mechanisms is identical to the set of equilibrium outcomes in the contracting game.

The following class of two-stage mechanisms are called Public Message Mechanisms. In the first stage, players simultaneously decide whether or not to participate in the mechanism. Players who participate send public messages from a countable message space. At the same time, a player who does not participate publicly submits a commitment device, which imposes a restriction on his action space as a function of the messages sent by the participants.[7] In the second stage, the mechanism restricts the action spaces of the participating players, as a function of the messages of the participants. These restrictions, however, cannot depend on the functions submitted by the non-participants. Finally, players take actions from their restricted actions spaces simultaneously.

Next, we define the Public Message Mechanism formally.

Definition 1 *Let N_i be a countable message space for each i, and let $N = \times_{i=1}^{m} N_i$. Suppose that $\varnothing \notin \cup_{i=1}^{m} N_i$ and let $\overline{N}_i = N_i \cup \{\varnothing\}$ for each i.*[8] *In addition, let*

$$\varrho_i = \left\{ \rho_i : \rho_i \in \left(2^{A_i} \backslash \{\emptyset\}\right)^{\times_{i=1}^{m} \overline{N}_i} \right\}.$$

For each $\rho^* = (\rho_1^*, ..., \rho_m^*)$, $\rho_i^* \in \varrho_i$, consider the following two-stage game. In the first stage, players take actions simultaneously. The first-stage action space of Player i is $N_i \cup \varrho_i$. In the second stage, after observing the first-stage action profile, players take actions simultaneously. Suppose that the first stage action of Player j is δ_j. Let $n_j = \delta_j$ if $\delta_j \in N_j$ and $\varnothing$ otherwise. Then the second stage action space of Player i is $\rho_i^* (n_1, ..., n_m)$ if $\delta_i \in N_i$ and $\rho_i (n_1, ..., n_m)$ if $\delta_i = \rho_i \in \varrho_i$. We call this game the Public Message Mechanism defined by (N, ρ^*).

The main difference between a PMM and a standard direct mechanism is that the reports of the players are publicly observable. This has several consequences. First, a non-participant player can learn about the types of the participants through their messages and can make his action contingent on these messages. As a result, non-participation is more profitable in a PMM than it would be in a comparable mechanism when messages are privately conveyed to the mechanism designer. Second, to prevent players from refusing to participate so that they can make use of this information, the mechanism might not induce participants to fully reveal their types in the first stage. Finally, since the messages do not necessarily coincide with the types, the mechanism designer might not want to specify a single action for every player in response to some public messages. If, instead, he allows the player to choose from a subset of his actions, then he makes it possible for the player's action to depend on his private information which was not revealed through his message. Furthermore, he can exploit a non-participant's uncertainty about the types of the others when implementing a punishment.

Intuitively, the relationship between a PMM and the contracting game can be explained as follows. In a contracting equilibrium, a player with different types offers different contracts. Since the contracts are publicly observable, players learn about each other's types from the contracts. The contracts' information content about the types corresponds to the public messages in a PMM. An equilibrium contract profile specifies restrictions on the action spaces of the players. These restrictions correspond to the second-stage restrictions of a PMM if each player participates. The

[7] A restriction is a nonempty subset of the action space.

[8] The symbol $\varnothing$ can be interpreted as the message of a player who does not participate in the mechanism.

Invariant Punishment Property corresponds to the property of a PMM which says that the restrictions imposed on participants do not depend on the commitment devices submitted by the non-participants. The reader should think about a non-participant in a PMM as a deviator in the contracting game, and the commitment device of a non-participant as the deviator's contract. The Invariant Punishment Property implies that it is without loss of generality to assume that uncooperative behavior by one player in the contracting game provokes a punishing contractual response from the others that doesn't depend on how the deviator goes about being uncooperative.

We restrict attention to deterministic mechanisms and pure strategies. Our main theorem can be stated as follows.

Theorem 1 *An outcome function is implementable as an $\mathcal{R}$-equilibrium in the contracting game if and only if it is implementable as an $\mathcal{R}$-equilibrium by a public message mechanism.*

A *simple public message mechanism* is a public message mechanism in which each player's message space is a partition of his type space. The mechanism is incentive compatible if each player prefers to report the partition element that contains his true type. It is individually rational if every player, irrespective of his type, would prefer to participate in the mechanism than unilaterally commit himself to a subset of his actions that depends on the partition elements reported by the other players. By standard arguments in mechanism design, an outcome function can be supported as an $\mathcal{R}-$equilibrium in a public message mechanism if and only if there is an incentive compatible and individually rational simple public message mechanism that implements the same outcome function as an $\mathcal{R}-$equilibrium. For this reason we will restrict attention to simple public message mechanisms which are incentive compatible and individually rational.

Next we characterize the set of implementable outcome functions with constraints. Let $\tau_i : T_i \rightarrow 2^{T_i} \backslash \{\emptyset\}$ be the partition from which Player i must choose his report. Let τ, τ_{-i}, and τ_{-ij} denote $\times_{i=1}^{n} \tau_i$, $\times_{j \neq i} \tau_j$, and $\times_{k \neq i,j} \tau_k$ respectively. Let $r_i(t) \in 2^{A_i} \backslash \{\emptyset\}$ denote the restricted action space of Player i if each player participates, and the message sent by Player j is $\tau_j(t_j)$. Since the restrictions can depend only on the partition elements that each player reports, r_i must be measurable with respect to τ, that is, $r_i(t) = r_i(t')$ whenever $\tau(t) = \tau(t')$. Furthermore, let $p_i^j(t_{-j}) \in 2^{A_i} \backslash \{\emptyset\}$ denote the restriction on the action space of Player i if all players but Player j participate and the message sent by Player q is $\tau_q(t_q)$ for all $q \neq j$. The function $p_i^j(t_{-j})$ is measurable with respect to τ_{-ij}. A simple public message mechanism is given by $(\tau, r, p) = \left(\{\tau_i\}_{i=1}^m, \{r_i\}_{i=1}^m, \left\{p_i^j\right\}_{i,j=1}^m \right)$.

A public message mechanism only constrains players to subsets of their action spaces, so we need to describe what happens at the second stage. We start with the equilibrium path. Let s_i denote the strategy of Player i at the second stage if each player participates. That is, $s_i : T \rightarrow A_i$, such that $s_i(t) \in r_i(t)$ for all t, and s_i is measurable with respect to τ_{-i}. Note that since Player i knows his own type, s_i does not have to be measurable with respect to τ_i. Next, we describe the strategies of the players following a deviation. Let

$$F_i^{\tau}=\left\{f_i:\ f_i\in\left(2^{A_i}\setminus\{\emptyset\}\right)^{T_{-i}},\ f_i \text{ is } \tau_{-i} \text{ measurable}\right\}.$$

The set F_i^{τ} is the action space of Player i in the first stage if he does not participate in the mechanism. If Player i submits $f_i\left(\in F_i^{\tau}\right)$ and Player j reports $\tau_j\left(t_j\right)$ for $j\neq i$, then Player i's restricted action space is $f_i\left(t_{-i}\right)\left(\subset A_i\right)$ in the second stage . Let s_i^j denote the second-stage strategy of Player i if all players but Player j participate. That is, $s_i^j: T_{-j}\times F_j^{\tau}\rightarrow A_i$ such that $s_i^j\left(t_{-j},f_j\right)\in p_i^j\left(t_{-j}\right)$, and s_i^j is measurable with respect to τ_{-ij}.

An outcome function $s=(s_1,...,s_m)$ is supportable as an equilibrium in the public message game (or alternatively, is implementable by a simple public message mechanism) if there is a simple public message mechanism (τ,r,p) such that the following inequalities hold. The first one guarantees that each player sends a truthful message in the first stage of the game. For each $i=1,\ldots m$, and for all t_i,$t_i'\in T_i$

$$\begin{gathered}E_{t_{-i}}\left(u_i\left(s\left(t\right),t\right)\ \mid t_i\right)\\ \geq E_{t_{-i}}\left(\max_{a_i\in r_i\left(t_i',t_{-i}\right)}E_{t_{-i}'}\left(u_i\left(a_i,s_{-i}\left(t_i',t_{-i}'\right),\left(t_i,t_{-i}'\right)\right)\ \mid t_{-i}'\in\tau_{-i}\left(t_{-i}\right)\right)\ \mid t_i\right).\end{gathered}\tag{7}$$

The max operator on the right hand side implies that the player has to choose a best reply from his restricted action space given his posterior belief. Taken together, these constraints for all the players require that the play in the second stage constitutes a Bayesian equilibrium of the game in which each player chooses an action from his restricted set of actions, given posterior beliefs about other players' types.

To deal with deviations at the first stage of the PMM, we require that for each $t_i\in T_i$

$$\begin{gathered}E_{t_{-i}}\left(u_i\left(s\left(t\right),t\right)\ \mid t_i\right)\geq\\ \max_{f_i\in F_i^{\tau}}E_{t_{-i}}\left(\max_{a_i\in f_i\left(t_{-i}\right)}E_{t_{-i}'}\left(u_i\left(a_i,s_{-i}^i\left(t_{-i}',f_i\right),\left(t_i,t_{-i}'\right),t\right)\ \mid t_{-i}'\in\tau_{-i}\left(t_{-i}\right)\right)\ \mid t_i\right).\end{gathered}\tag{8}$$

The inequality says that even if the deviator chooses a best reply from the set of actions to which he is restricted, he cannot gain by deviating.

We say that an outcome function is supported as an $\mathcal{R}$-equilibrium of the public message mechanisms (alternatively, is $\mathcal{R}$-implementable by a simple public message mechanism) if (7) and (8) hold, and in addition, for every i and $f_i\in F_i^{\tau}$

$$s_{-i}^i\left(t_{-i},f_i\right)\in\mathcal{R}_i\left(f_i\times p_{-i}^i,t_{-i}\right).\tag{9}$$

With this formalism, we can restate our theorem as follows:

Corollary 1 *The outcome function s is implementable as an $\mathcal{R}$-equilibrium in the contracting game if and only if there is a simple public message mechanism (τ,r,p) for which (7), (8) and (9) hold.*

4.1 The Proof of Theorem 1

Proof of the "if" part of Theorem 1. Suppose that there exists a PMM which $\mathcal{R}$-implements the outcome function $s = (s_1, ..., s_m) : T \to A$. According to the arguments in the previous section, this implies that for all i and j there exists a partition of the type space, $\tau_i : T_i \to 2^{T_i} \backslash \{\emptyset\}$, on-path restriction $r_i : T \to 2^{A_i} \backslash \{\emptyset\}$, off-path restrictions, $p_i^j : T_{-j} \to 2^{A_i} \backslash \{\emptyset\}$, and off-path strategies $s_i^j : T_{-j} \times F_j^\tau \to A_i$ such that s_i and r_i are measurable with respect to τ_{-i}, p_i^j and s_i^j are measurable with respect to τ_{-ij}, $s_i(t) \in r_i(t)$, $s_i^j(t_{-j}, f_j) \in p_i^j(t_{-j})$, and (7), (8), and (9) are satisfied. In what follows we construct an $\mathcal{R}$-equilibrium in the contracting game which implements the outcome function s.

For each i let N_i denote the number of elements in the partition generated by τ_i. Then there exists a τ_i-measurable surjection $\sigma_i : T_i \to \{1, ..., N_i\}$. Define $\overline{r}_i : \mathbb{N}^m \to 2^{A_i} \backslash \{\emptyset\}$ such that $\overline{r}_i(\sigma_1(t_1), ..., \sigma_m(t_m)) = r_i(t_1, ..., t_m)$, and $\overline{p}_i^j : \mathbb{N}^{m-1} \to 2^{A_i} \backslash \{\emptyset\}$, such that $\overline{p}_i^j\left((\sigma_k(t_k))_{k \neq j}\right) = p_i^j\left((t_k)_{k \neq j}\right)$. This is possible because σ_i and r_i are τ_i measurable and p_i^j is τ_{-ij} measurable. The Cross-referential Property guarantees that there exists a set of contracts for all i, $\{c_i^{n_i}\}_{n_i \in \{1, ...N_i\}} \subset C_i$, such that $c_i^{n_i} \neq c_i^{n_i'}$ for $n_i \neq n_i'$ and:

$$\widetilde{c}_i^{n_i}(c_1, ..., c_m) = \begin{cases} \overline{r}_i(n_i, n_{-i}) & \text{if } \forall k \neq i \ c_k^{n_k} = c_k, \\ \overline{p}_i^j(n_{-j}) & \text{if } \forall k \neq i, j \ c_k^{n_k} = c_k \text{ and } \nexists n_j \text{ s.t. } c_j^{n_j} = c_j, \\ A_i & \text{otherwise.} \end{cases}$$

Now define $c_i^{t_i}$ to be $c_i^{\sigma_i(t_i)}$. The first-stage strategy of Player i, $\gamma_i : T_i \to C_i$ is given by $\gamma_i(t_i) = c_i^{t_i}$. Since σ_i is τ_i measurable, γ_i is also τ_i measurable. Using the definitions of $\overline{r}_i$ and $\overline{p}_i^j$, $\widetilde{c}_i^{t_i}$ can be written as

$$\widetilde{c}_i^{t_i}(c_1, ..., c_m) = \begin{cases} r_i(t_i, t_{-i}) & \text{if } \forall k \neq i \ c_k^{t_k} = c_k, \\ p_i^j(t_{-j}) & \text{if } \forall k \neq i, j \ c_k^{t_k} = c_k \text{ and } \nexists t_j \text{ s.t. } c_j^{t_j} = c_j, \\ A_i & \text{otherwise.} \end{cases} \tag{10}$$

It remains to specify the second-stage strategy of Player i, $\alpha_i : T_i \times C \to A_i$ for each i. If, for all j, there is a $t_j \in T_j$ such that Player j offered a contract $\gamma_j(t_j) = c_j^{t_j}$, then $\alpha_i(t_i, \gamma(t)) = s_i(t)$. This strategy is well-defined because $s_i(t_i, t_{-i}) = s_i(t_i, t'_{-i})$ whenever $\alpha_i(t_i, \gamma(t_i, t_{-i})) = \alpha_i(t_i, \gamma(t_i, t'_{-i}))$.[9] Suppose now that one player deviated, say Player j, and offered a contract c_j, and Player k offered $c_k^{t_k}$, for some $t_k \in T_k$, for all $k \neq j$. Define $f_j^{c_j} : T_{-j} \to 2^{A_j}$ as follows:

$$f_j^{c_j}(t_{-j}) = \widetilde{c}_j\left(c_j, (c_k^{t_k})_{k \neq j}\right) = \widetilde{c}_j(c_j, \gamma_{-j}(t_{-j})). \tag{11}$$

Notice that $f_j^{c_j} \in F_j^\tau$ because γ_{-j} is τ_{-j} measurable. Define $\alpha_i\left(t_i, \left(c_j, (c_k^{t_k})_{k \neq j}\right)\right)$ to be $s_i^j\left(t_{-j}, f_j^{c_j}\right)$. These strategies are well-defined because s_i and γ_{-i} are τ_{-i} measurable, s_i^j is τ_{-ij} measurable, and

[9] This follows from σ_j being a surjection for each j.

σ_j is a surjection. In addition, they are consistent with the restrictions imposed by the contracts defined by (10), that is, $s_i(t) \in r_i(t)$ and $s_i^j(t_{-j}, f_j) \in p_i^j(t_{-j})$.

Next, we shall argue that the strategies described above constitute an $\mathcal{R}-$equilibrium in the contracting game. First, we show that the strategies $\{s_i\}_{i=1}^m$ are optimal in the second stage. The constraint (7) with $t_i = t_i'$ requires $s_i(t_i, t_{-i})$ to be a best response of Player i with type t_i to the strategies of the other players. It remains to show that players do not have an incentive to deviate at the contracting stage. Suppose that Player j with type t_j offers a contract $c_j \neq c_j^{t_j}$. We shall consider two cases. Case 1: there exists a $t_j' \in T_j$ such that $c_j = c_j^{t_j'}$. Then, by (7), this deviation is not profitable no matter what the second-stage strategy of Player j is. Case 2: $c_j \neq c_j^{t_j'}$ for all $t_j' \in T_j$. Such a deviation induces Player i $(i \neq j)$ with type t_i to take action $s_i^j\left(t_{-j}, f_j^{c_j}\right)$, where $f_j^{c_j}$ is defined by (11). Hence, by (8), such a deviation cannot be profitable.

Moreover, these strategies satisfy the constraints imposed by the refinement concept (3). This is because

$$\alpha_i\left(t_i, \left(c_j, \gamma_{-j}\left(t_{-j}\right)\right)\right) = s_{-j}^j\left(t_{-j}, f_j^{c_j}\right) \in \mathcal{R}_j\left(f_j^{c_j} \times p_{-j}^j, t_{-j}\right) = \mathcal{R}_j\left(S\left(c_j, \gamma_{-j}\left(t_{-j}\right)\right), t_{-j}\right),$$

where the equalities are satisfied by the definitions of α_i, $f_j^{c_j}$ and γ and the middle of the chain is just (9).

Finally, since $\alpha_i(t_i, \gamma(t)) = s_i(t)$ for all i and t, these strategies indeed implement the outcome function s. ■

Proof of the "only if" part of Theorem 1. Fix an $\mathcal{R}$-equilibrium in the contracting game which implements the outcome function $s = (s_1, ..., s_m) : T \to A$. For all i and j we shall construct the following objects: $\tau_i : T_i \to 2^{T_i}$, $r_i : T \to 2^{A_i} \backslash \{\emptyset\}$, $p_i^j : T_{-j} \to 2^{A_i} \backslash \{\emptyset\}$ and $s_i^j : T_{-j} \times F_j^\tau$ such that s_i and r_i are measurable with respect to τ_{-i}, p_i^j and s_i^j are measurable with respect to τ_{-ij}, $s_i(t) \in r_i(t)$, $s_i^j(t_{-j}, f_j) \in p_i^j(t_{-j})$. Then we show that (7), (8) and (9) are satisfied.

Denote the equilibrium contract of Player i with type t_i by $c_i^{t_i}$. Define the partition, τ, as follows:

$$\tau_i(t_i) = \left\{t_i' \in T_i : c_i^{t_i} = c_i^{t_i'}\right\}.$$

For all $i \in \{1, ..., m\}$, let

$$r_i(t) = \widetilde{c}_i^{t_i}\left(c_i^{t_i}, \left(c_j^{t_j}\right)_{j \neq i}\right). \tag{12}$$

Notice that r_i is measurable with respect to τ_{-i} by the definition of $\tau = \times_{i=1}^m \tau_i$.

Let $\alpha_i : T_i \times C \to A_i$ denote the second stage strategy of Player i. Observe that

$$\alpha_i\left(t_i, \left(c_j^{t_j}\right)_j\right) \in \widetilde{c}_i^{t_i}\left(c_i^{t_i}, \left(c_j^{t_j}\right)_{j \neq i}\right) \tag{13}$$

by the rules of the contracting game. Since the contracting equilibrium implements $s = (s_1, ..., s_m)$, it follows that $\alpha_i\left(t_i, \left(c_j^{t_j}\right)_j\right) = s_i(t_1, ..., t_m)$. Notice that $s_i(t)$ is measurable with respect to τ_{-i}. In addition, $s_i(t) \in r_i(t)$ by (12) and (13).

We are ready to show that the triple $(\{\tau_i\}_{i=1}^m, \{r_i\}_{i=1}^m, s)$ satisfies (7). First, consider this constraint with $t_i' = t_i$. This constraint requires $\alpha_i\left(t_i, \left(c_j^{t_j}\right)_j\right)$ to be a best-response of Player i with type t_i to the strategies of the other players on the equilibrium path. Since α_i was an equilibrium strategy, it has to be a best response and hence (7) must be satisfied. Second, consider (7) with $t_i' \neq t_i$. Then, this constraint says that Player i with type t_i is better off offering the contract $c_i^{t_i}$ than offering $c_i^{t_i'}$. Indeed, the left-hand-side is just his expected equilibrium payoff and the right-hand-side is the maximum payoff of Player i with type t_i if he offered $c_i^{t_i'}$. Since, $c_i^{t_i}$ was an equilibrium contract, Player i is better off offering $c_i^{t_i}$ than any other contract, hence, (7) is satisfied.

It remains to construct p_i^j and s_i^j for all i, j $(i \neq j)$ and show that (8) is also satisfied. For each i let N_i denote the number of elements in the partition of T_i generated by τ_i. For each i, fix a τ_i-measurable surjection $\sigma_i : T_i \to \{1, ..., N_i\}$. For each $f_j \in F_j^\tau$ and for all $n_{-j} \in \times_{k\neq j}\{1, ..., N_k\}$, define $\overline{f}_j(n_{-j})$ to be $f_j(t_{-j})$ if $n_{-j} = \sigma_{-j}(t_{-j})$. The function $\overline{f}_j$ is well defined because f_j is τ_{-j} measurable and σ_{-j} is a τ_{-j} measurable surjection. For all $n_i \in \{1, ..., N_i\}$, define $c_i^{n_i}$ to be $c_i^{t_i}$ if $\sigma_i(t_i) = n_i$. The Invariant Punishment Property guarantees that there are functions $\overline{p}_k^j : \mathbb{N}^{m-1} \to 2^{A_k}\backslash\{\emptyset\}$ $(k \neq j)$ such that for each $f_j \in F_j^\tau$ there is a contract $c_j^{f_j} \in C_j$ which satisfies

$$\widetilde{c}_j^{f_j}\left(c_j^{f_j}, (c_i^{n_i})_{i\neq j}\right) = \overline{f}_j(n_{-j}) \text{ and } \widetilde{c}_k^{n_k}\left(c_j^{f_j}, (c_i^{n_i})_{i\neq j}\right) = \overline{p}_k^j(n_{-j}), \tag{14}$$

for all $n_{-j} \in \times_{i\neq j}\{1, ..., N_i\}$ and $k \neq j$. For all $t_{-j} \in T_{-j}$, define $p_k^j(t_{-j})$ to be $\overline{p}_k^j(\sigma_{-j}(t_{-j}))$. Notice that p_k^j is τ_{-j} measurable. Using this notation and the definition of $\overline{f}_j$, (14) can be rewritten as

$$\widetilde{c}_j^{f_j}\left(c_j^{f_j}, (c_i^{t_i})_{i\neq j}\right) = f_j(t_{-j}) \text{ and } \widetilde{c}_k^{t_k}\left(c_j^{f_j}, (c_i^{t_i})_{i\neq j}\right) = p_k^j(t_{-j}),$$

for all $t_{-j} \in T_{-j}$ and $k \neq j$. For each $f_j \in F_j^\tau$ and $k \neq j$, define $s_k^j(t_{-j}, f_j)$ to be $\alpha_k\left(t_k, \left(c_j^{f_j}, (c_i^{t_i})_{i\neq j}\right)\right)$. The function s_i^j is obviously measurable with respect to τ_{-ij}. Given these notations, (8) requires that Player j cannot profitably deviate by offering $c_j^{f_j}$. Therefore, this constraint is satisfied. Finally, since the strategies in the contracting game satisfies the refinement,

$$s_{-j}^j(t_{-j}, f_j) = \alpha_{-j}\left(t_{-j}, \left(c_j^{f_j}, (c_k^{t_k})_{k\neq j}\right)\right) \in$$

$$\mathcal{R}_j\left(\widetilde{c}_j^{f_j}\left(c_j^{f_j}, (c_k^{t_k})_{k\neq j}\right), \left(\widetilde{c}_i^{t_i}\left(c_j^{f_j}, (c_k^{t_k})_{k\neq j}\right)\right)_{i\neq j}, t_{-j}\right) =$$

$$\mathcal{R}_j\left(f_j(t_{-j}) \times p_{-j}^j(t_{-j}), t_{-j}\right)$$

and hence, (9) is satisfied. ■

5 Definable Contracts

5.1 The Language and the Gödel Coding

We consider a formal language that is sufficiently rich to allow its user to state propositions in arithmetic. (The Appendix provides a formal definition of a first-order language, see Section 8.2.) Furthermore, the set of statements in this language is closed under the finite applications of the Boolean operations: $\urcorner$, $\vee$, and $\wedge$. In addition, the language contains variable symbols, such as x, y, which enables one to express, for example, Fermat's Last Theorem:

$$\forall n, x, y, z \left\{ \left[(n \geq 3) \wedge (x \neq 0) \wedge (y \neq 0) \wedge (z \neq 0) \right] \rightarrow (x^n + y^n \neq z^n) \right] \right\}.$$

In fact, one can also express statements in the language that involves any finite number of free variables. For example,"x is a prime number" is a statement in the language. The symbol x is a free variable in the statement. Another example for a predicate that has one free variable is"$x < 4$."One can substitute any integer into x and then the predicate is either true or false. This particular one is true if $x = 0, 1, 2, 3$ and false otherwise.

A *text* is a finite string of symbols. Let $\mathfrak{L}$ be the set of all texts of the formal language. It is well known that one can *construct* a one-to-one function $\mathfrak{L} \rightarrow \mathbb{N}$. Let $[\varphi]$ be the value of this function at $\varphi \in \mathfrak{L}$; call it the Gödel Code of the text φ.

In what follows, we define a class of functions which can be represented by finitely many characters in our formal language.

Definition 2 *The function $f : \mathbb{N}^k \rightarrow 2^{\mathbb{N}}$ is said to be* definable *if there exists a first-order arithmetic statement ϕ in $k+1$ free variables such that for all $a \in \mathbb{N}^k$: $b \in f(a)$ if and only if $\phi(a, b)$ is true.*

We provide a formal definition of first-order arithmetic statement in the appendix. We illustrate the previous definition with an example.

Example. Consider the following function defined on $\mathbb{N}$:

$$f(a) = \begin{cases} 0 & \text{if } a \text{ is an even number,} \\ 1 & \text{if } a \text{ is an odd number.} \end{cases}$$

We show that this function is definable by constructing the corresponding predicate ϕ.

$$\phi(x, y) \equiv \{\{y = 1\} \vee \{y = 0\}\} \wedge \{\exists z : 2z = y + x\}.$$

Notice that ϕ indeed has two free variables. (The variable z is not free because there is a quantifier in front of it.) The first part of ϕ states that y is either one or zero. The second part says that $x + y$ is divisible by two. Clearly, $f(a) = 0$ if and only if $\phi(a, 0)$ is true and $f(a) = 1$ if and only if $\phi(a, 1)$ is true.

If the statement ϕ defines the function f and θ is true, then $\phi \wedge \theta$ also defines f. We make use of this observation to construct different but computationally equivalent contracts in the next section.

Definition 3 *Suppose that f_n is a function mapping from $\mathbb{N}^k$ to $2^{\mathbb{N}}$ for all $n \in \mathbb{N}^q$. Suppose that there exists a first-order arithmetic statement ϕ in $q+k+1$ free variables such that for all $n \in \mathbb{N}^q$ and $a \in \mathbb{N}^k$: $b \in f_n(a)$ if and only if $\phi(n, a, b)$ is true. Then we call the expression f_x a* definable function with q free variables, *where $x = (x_1, \ldots, x_q)$ is a vector of variable symbols.*

A definable function from $\mathbb{N}^k$ to $2^{\mathbb{N}}$ with q free variables is essentially a definable function from $\mathbb{N}^{k+q}$ into $2^{\mathbb{N}}$. Therefore, many properties of definable functions are also properties of definable functions with free variables.

We can now describe some properties of definable functions that we need in our proofs. We need two pieces of notations. First, recall from the introduction that if $n \in \mathbb{N}$, then $< n >$ denotes the text whose Gödel code is n, that is, $[< n >] = n$. Let g be a function from $\{1, ..., q\}$ to the set of variable symbols such that $g(i) \neq g(j)$ if $i \neq j$. That is $(g(1), ..., g(q))$ is a finite vector of variable symbols. Then, for any text φ and $(n_1, ..., n_q) \in \mathbb{N}^q$ let $\varphi^{(n_1, \ldots, n_q)}$ denote the text where if the symbol $g(k)$ stands for a free variable in φ then $g(k)$ is replaced by n_k in φ for $k = 1, ..., q$.[10] For example, if $g(1) = x_1$ and $g(2) = x_2$, φ is"$x_1 < x_2$", $n_1 = 1$ and $n_2 = 2$ then $\varphi^{(n_1, n_2)}$ is $1 < 2$.[11]

Suppose that $g(i) = x_i$ for $i = 1, ..., q$. Consider the following text in n free variable: $< x_k >^{(x_1, \ldots, x_q)}$, where $k \leq q$. Since the Gödel coding is a bijection, $< n_k >$ is a text for each $n_k \in \mathbb{N}$. Since $\varphi^{(n_1, \ldots, n_q)}$ is defined for all φ and $(n_1, ..., n_q) \in \mathbb{N}^q$, $< n_k >^{(n_1, \ldots, n_q)}$ is a text for all $(n_1, ..., n_q) \in \mathbb{N}^q$. The following is a well-known result in Mathematical Logic:

Lemma 1 *Let $(g(1), ..., g(q))$ be a vector of distinct free variables, and for all $k \in \{1, ..., q\}$, let $f_k(n_1, ..., n_q) = \left[< n_k >^{(n_1, \ldots, n_q)}\right]$. Then f_k is a definable function for all $k \in \{1, ..., q\}$.*

This basic result is central to the construction of cross-referential contracts. The next result is used to show that the contracts we construct to support various kinds of equilibrium are definable.

Lemma 2 *For any set A, let χ_A denote the characteristic function of A.*
(i) If $A \subset \mathbb{N}^k$ is finite, then χ_A is definable.
(ii) Let $A, B \subset \mathbb{N}^k$. If χ_A and χ_B are definable, then $\chi_{A \cap B}$, $\chi_{A \cup B}$, and $\chi_{A \backslash B}$ are definable.
(iii) Let $A_1, ..., A_m \subset \mathbb{N}^k$, $B_1, ..., B_{m+1} \subset \mathbb{N}$. If $\cap_{i=1}^m A_i = \{\emptyset\}$, $\chi_{A_1}, ..., \chi_{A_m}$ are definable, and

[10] The text $\varphi^{(n_1, \ldots, n_q)}$ depends on g which specifies the list of free variables that are to be replaced. We suppress this dependence in the notation just to make it simpler.

[11] Of course, it is possible that the text φ does not contain some of the symbols $\{g(1), ..., g(q)\}$. In that case, there is no substitution for the missing symbols in $\varphi^{(n_1, \ldots, n_q)}$. For example, $g(1) = x_1$ and $g(2) = x_2$ and φ is "$x_2 > 2$", then $\varphi^{(3,4)}$ is "$4 > 2$", because x_1 does not appear in φ.

$B_1, ..., B_m$ *are finite, then the following correspondence,* $f : \mathbb{N}^k \to \mathbb{N}$*, is definable:*

$$f(n) = \begin{cases} B_1 & \textit{if } n \in A_1, \\ ... & ... \\ B_m & \textit{if } n \in A_m, \\ B_{m+1} & \textit{otherwise.} \end{cases}$$

(iv) Suppose that $D_1, ..., D_k \subset \mathbb{N}$ *are finite sets,* $g : \times_{i=1}^k D_i \to 2^{\mathbb{N}}$*, and* $B_1, ..., B_{k+1} \subset \mathbb{N}$ *are finite sets. For all* $n \in \mathbb{N}^k$ *let* n_i *denote the ith coordinate of* n*. Then the following correspondence is definable:*

$$f(n) = \begin{cases} g(n) & \textit{if } n \in \times_{i=1}^k D_i, \\ B_1 & \textit{if } \{i : n_i \notin D_i\} = \{1\}, \\ ... & \\ B_k & \textit{if } \{i : n_i \notin D_i\} = \{k\}, \\ B_{k+1} & \textit{otherwise.} \end{cases}$$

We point out that all the contracts we use to construct equilibrium in the paper are in the form of (iv) of Lemma 2.

Proof. See the Appendix. ■

5.2 The Contract Space

With a slight abuse of notation we shall refer to $[a_i]$ as the Gödel code of the text describing action a_i.[12] In addition, for each $\overline{A_i} \subset A_i$, let $\left[\overline{A_i}\right] = \{[a_i] : a_i \in A_i\}$. We define the contract space of Player i, C_i, as the set of all arithmetic statements defining functions from $\mathbb{N}^m$ to $2^{[A_i]} \backslash \{\emptyset\}$ in the sense of Definition 2.

For all $c_i \in C_i$, define the correspondence $\widetilde{c_i} : C \to 2^{A_i} \backslash \{\emptyset\}$ induced by c_i as follows:

$$a_i \in \widetilde{c_i}(c_1, ..., c_m) \Leftrightarrow c_i([c_1], ..., [c_m], [a_i]) \text{ is true.} \tag{15}$$

That is, given the contract profile $(c_1, ..., c_m)$, Player i can take action a_i if and only if the Gödel code of this action is an element of the function defined by c_i evaluated at the vector $([c_1], ..., [c_m])$.

In order to show that the statement of Theorem 1 is valid with this contract space we have to show that both the Cross-referential Property and the Invariant Punishment Property hold.

5.2.1 The Cross-referential Property

Let $N_i \geq 1$, $r_i : \mathbb{N}^m \to 2^{A_i} \backslash \{\emptyset\}$ and $p_i^j : \mathbb{N}^{m-1} \to 2^{A_i} \backslash \{\emptyset\}$ for $i, j \in \{1, ..., m\}$, $i \neq j$. Let x_j denote the vector $\left(x_j^1, ..., x_j^{N_j}\right)$ and $x = (x_1, ..., x_m)$ where $x_j^{n_j}$ is a variable symbol for all j and n_j.

[12] In other word, we identify an action with a text which describes it.

For each i and $n_i \in \{1, ..., N_i\}$ let $h_i^{n_i}(x)$ denote $\left[< x_i^{n_i} >^{(x)}\right]$ and consider the following function in $\sum_{i=1}^m N_i$ free variables:

$$f_x^{i,n_i}([c_1], ..., [c_m]) \tag{16}$$

$$= \begin{cases} [r_i(n_i, n_{-i})] & \text{if } \forall k \neq i \ h_k^{n_k}(x) = [c_k] \text{ and } h_k^{n_k}(x) \neq h_k^{n'_k}(x) \text{ if } n_k \neq n'_k \\ \left[p_i^j(n_i, n_{-ij})\right] & \begin{array}{c}\text{if } \forall k \neq i, j \ h_k^{n_k}(x) = [c_k] \text{ and } h_k^{n_k}(x) \neq h_k^{n'_k}(x) \text{ if } n_k \neq n'_k \\ \text{and } \nexists n_j \text{ s.t.} h_j^{n_j}(x) = [c_j], \end{array} \\ [A_i] & \text{otherwise.} \end{cases}$$

We prove in the Appendix that this function is definable (see Lemma 6). Let $\varphi^{i,n_i}(x, y, z)$ be a statement which defines the function f_x^{i,n_i} where y is an m dimensional vector of variable symbols and z is a variable symbol. Define $c_x^{i,n_i}(y, z)$ to be $\varphi^{i,n_i}(x, y, z) \wedge (n_i + 1 > n_i)$ and let $\gamma_i^{n_i}$ denote its Gödel Code.[13] In addition, let $\gamma_i = \left(\gamma_i^1, ..., \gamma_i^{N_i}\right)$ and $\gamma = (\gamma_1, ..., \gamma_m)$. Observe that the contract c_γ^{i,n_i} defines the following function:

$$f_\gamma^{i,n_i}([c_1], ..., [c_m]) = \begin{cases} [r_i(n_i, n_{-i})] & \text{if } \forall k \neq i \ h_k^{n_k}(\gamma) = [c_k] \text{ and } h_k^{n_k}(\gamma) \neq h_k^{n'_k}(\gamma) \text{ if } n_k \neq n'_k \\ \left[p_i^j(n_{-j})\right] & \begin{array}{c}\text{if } \forall k \neq i, j \ h_k^{n_k}(\gamma) = [c_k] \text{ and } h_k^{n_k}(\gamma) \neq h_k^{n'_k}(\gamma) \text{ if } n_k \neq n'_k \\ \text{and } \nexists n_j \text{ s.t.} h_j^{n_j}(\gamma) = [c_j], \end{array} \\ [A_i] & \text{otherwise.} \end{cases}$$

Recall that $h_q^{n_q}(x) = \left[< x_q^{n_q} >^{(x)}\right]$ and $\gamma_q^{n_q} = \left[c_x^{q,n_q}\right]$. Hence $h_q^{n_q}(\gamma) = \left(c_x^{q,n_q}\right)^{(\gamma)} = c_\gamma^{q,n_q}$. Therefore, the previous function can be rewritten as[14]

$$f_\gamma^{i,n_i}([c_1], ..., [c_m]) = \begin{cases} [r_i(n_i, n_{-i})] & \text{if } \forall k \neq i \ \left[c_\gamma^{k,n_k}\right] = [c_k], \\ \left[p_i^j(n_{-j})\right] & \text{if } \forall k \neq i, j \ \left[c_\gamma^{k,n_k}\right] = [c_k] \text{ and } \nexists n_j \text{ s.t. } \left[c_\gamma^{j,n_j}\right] = [c_j], \\ [A_i] & \text{otherwise.} \end{cases} \tag{17}$$

Define $c_i^{n_i}$ to be c_γ^{i,n_i}. Therefore, by (15),

$$\widetilde{c}_i^{n_i}(c_1, ..., c_m) = \begin{cases} r_i(n_i, n_{-i}) & \text{if } \forall k \neq i \ c_k^{n_k} = c_k, \\ p_i^j(n_{-j}) & \text{if } \forall k \neq i, j \ c_k^{n_k} = c_k \text{ and } \nexists n_j \text{ s.t. } c_j^{n_j} = c_j, \\ A_i & \text{otherwise,} \end{cases}$$

which is just (4). In addition, since $c_i^{n_i} = \varphi^{i,n_i}(\gamma, y, z) \wedge (n_i + 1 > n_i)$, $c_i^{n_i} \neq c_i^{n'_i}$ if $n_i \neq n'_i$, and hence the Cross-referential Property is satisfied.

[13] Notice that $n_i + 1 > n_i$ is always true, and hence, $c_x^{i,n_i}(y, z)$ and $\varphi^{i,n_i}(x, y, z)$ define the same function. Such a statement, however, makes it possible that a player with two different types offers two different but computationally equivalent contracts.

[14] Notice that $c_\gamma^{k,n_k} \neq c_\gamma^{k,n'_k}$ if $n_k \neq n'_k$, hence, these conditions became irrelevant in the definition of f_γ^{i,n_i} (see the previous footnote).

5.2.2 The Invariant-punishment Property

In order to prove that our contract space satisfies the Invariant Punishment Property it is enough to prove the following

Lemma 3 *For all* $(N_1,...,N_m) \in \mathbb{N}^m$, $N_i \geq 1$, *for all set of contracts* $\left\{c_j^{n_j}\right\}_{j,n_j \in \{1,...N_j\}}$ *(*$c_j^{n_j} \in C_j$, $c_j^{n_j} \neq c_j^{n_j'}$ *if* $n_j \neq n_j'$*) and for every* i*, there are functions* $p_k^i : N^{m-1} \rightarrow 2^{A_k} \backslash \{\emptyset\}$ *(*$k \neq i$*), such that for any function* $f_i : \mathbb{N}^{m-1} \rightarrow 2^{A_i} \backslash \{\emptyset\}$*, there is a contract* $c_i^* \in C_i$ *such that for all* $n_{-i} \in \times_{j \neq i} \{1,...,N_j\}$ *:*

$$\widetilde{c}_i^* \left(c_i^*, c_{-i}^{n_{-i}}\right) = f\left(n_{-i}\right), \tag{18}$$

and for all $k \neq i$

$$\widetilde{c}_k^{n_k} \left(c_i^*, c_{-i}^{n_{-i}}\right) = p_k^i\left(n_{-i}\right), \tag{19}$$

where $\widetilde{c}_j$ *is defined by (15) for all* j *and* $c_j \in C_j$*.*

First, we reformulate the statement of the lemma. Let $\mathcal{N}_i$ denote $\{1,...,N_i\}$ and $\mathcal{N} = \times_{i=1}^m \mathcal{N}_i$. Let $\left(2^{A_i}\right)^{|\mathcal{N}_{-i}|}$ denote the set of $\times_{j \neq i} N_j$ dimensional vector of nonempty subsets of A_i. For all $\left(A_i^{n_{-i}}\right)_{n_{-i} \in \mathcal{N}_{-i}} \in \left(2^{A_i}\right)^{|\mathcal{N}_{-i}|}$ define $S\left(\left(A_i^{n_{-i}}\right)_{n_{-i} \in \mathcal{N}_{-i}}\right)$ as follows:

$$\left\{\left(A_{-i}^{n_{-i}}\right)_{n_{-i} \in \mathcal{N}_{-i}} : \forall n_{-i} \in \mathcal{N}_{-i},\ \exists c_i \text{ s. t. } \widetilde{c}_i\left(c_i, c_{-i}^{n_{-i}}\right) = A_i^{n_{-i}},\ \widetilde{c}_{-i}^{n_{-i}}\left(c_i, c_{-i}^{n_{-i}}\right) = A_{-i}^{n_{-i}}\right\}.$$

$\left(A_{-i}^{n_{-i}}\right)_{n_{-i} \in \mathcal{N}_{-i}} \in S\left(\left(A_i^{n_{-i}}\right)_{n_{-i} \in \mathcal{N}_j}\right)$ implies that there exists a contract, c_i, available for Player i such that if he offers c_i and the contract profile of the other players is $c_{-i}^{n_{-i}}$, then the contract profile $\left(c_i, c_{-i}^{n_{-i}}\right)$ restricts the action space of Player i to be $A_i^{n_{-i}}$ and the action spaces of the other players to be $A_{-i}^{n_{-i}}$. We claim the following

Lemma 4 *The statement of Lemma 3 is equivalent to the following statement. For all* $(N_1,...,N_m) \in \mathbb{N}^m$, $N_i \geq 1$, *for all set of contracts* $\left\{c_j^{n_j}\right\}_{j,n_j \in \{1,...N_j\}}$ *(*$c_j^{n_j} \in C_j$, $c_j^{n_j} \neq c_j^{n_j'}$ *if* $n_j \neq n_j'$*), and for every* i*:*

$$\cap_{\left(A_i^{n_{-i}}\right)_{n_{-i} \in \mathcal{N}_{-i}}} S\left(\left(A_i^{n_{-i}}\right)_{n_{-i} \in \mathcal{N}_{-i}}\right) \neq \{\emptyset\}. \tag{20}$$

Proof. Suppose first that (20) is true and $\left(A_{-i}^{n_{-i}}\right)_{n_{-i} \in \mathcal{N}_{-i}}$ is an element of the intersection. Define $p_{-i}^i\left(n_{-i}\right) = \left(p_k^i\left(n_{-i}\right)\right)_{k \neq i}$ to be $A_{-i}^{n_{-i}}$ for all n_{-i}. Fix a function $f_i : \mathcal{N}_{-i} \rightarrow 2^{A_i} \backslash \{\emptyset\}$ and consider $S\left(\left(f_i\left(n_{-i}\right)\right)_{n_{-i} \in \mathcal{N}_{-i}}\right)$. Then, there exists a c_i^* such that equations (18) and (19) are satisfied because $\left(p_{-i}^i\left(n_{-i}\right)\right)_{n_{-i}} \in S\left(\left(f_i\left(n_{-i}\right)\right)_{n_{-i}}\right)$.

Conversely, suppose that (20) is false. Then for all $p_{-i}^i\left(n_{-i}\right) = \left(p_k^i\left(n_{-i}\right)\right)_{k \neq i} : \mathcal{N}_{-i} \rightarrow A_{-i}$ there exists $\left(A_i^{n_{-i}}\right)_{n_{-i} \in \mathcal{N}_{-i}} \in \left(2^{A_i}\right)^{|\mathcal{N}_{-i}|}$ such that

$$\left(p_{-i}^i\left(n_{-i}\right)\right)_{n_{-i} \in \mathcal{N}_{-i}} \notin S\left(\left(A_i^{n_{-i}}\right)_{n_{-i} \in \mathcal{N}_{-i}}\right).$$

Define $f_i\left(n_{-i}\right)$ to be $A_i^{n_{-i}}$ for all $n_{-i} \in \mathcal{N}_{-i}$. Then, by the definition of S, there does not exist a contract c_i^* such that (18) and (19) are satisfied. ■

By the previous lemma, in order to show Lemma 3, we only have to prove (20). We have relegated this proof to the Appendix. Here, we sketch the proof for the case of two players, and where $N_1 = N_2 = 1$. Let c_2 denote the contract of Player 2. For all $(\{\emptyset\} \neq) B_1 \subset A_1$:

$$S(B_1) = \{B_2 : \exists c_1 \text{ s. t. } \widetilde{c}_1(c_1, c_2) = B_1,\ \widetilde{c}_2(c_1, c_2) = B_2\}.$$

We have to show that $\cap_{\{\emptyset\} \neq B_1 \subset A_1} S(B_1) \neq \{\emptyset\}$. Suppose that, by contradiction, $\cap_{B_1} S(B_1) = \{\emptyset\}$. Then for all $B_2 \subset A_2$ there exists a $(\{\emptyset\} \neq) B_1 \subset A_1$ such that $B_2 \notin S(B_1)$. Therefore, one can construct a function, $g : 2^{A_2} \backslash \{\emptyset\} \to 2^{A_1} \backslash \{\emptyset\}$, such that

$$\forall B_2 \subset A_2 : B_2 \notin S(g(B_2)).$$

Let f^{c_2} denote the function defined by c_2. Define the function in one free variable, f_x, as follows:

$$f_x([c_1], [c_2]) = \left[g\left(< f^{c_2}\left(\left[< x >^{(x)}\right], [c_2]\right) >\right)\right].$$

We shall show (see Lemma 7 in the Appendix) that since g has a finite domain, f_x is a definable function in one free variable. Let c_x denote a statement which defines f_x, and let γ denote its Gödel code. Notice that $< \gamma >^{(\gamma)} = c_\gamma$. Hence,

$$f_\gamma([c_1], [c_2]) = [g(< f^{c_2}([c_\gamma], [c_2]) >)]. \tag{21}$$

Notice that

$$\widetilde{c}_2(c_\gamma, c_2) \in S(\widetilde{c}_\gamma(c_1, c_2))$$

by the definition of S. On the other hand,

$$\widetilde{c}_2(c_\gamma, c_2) \notin S(g(\widetilde{c}_2(c_\gamma, c_2))) = S(\widetilde{c}_\gamma(c_\gamma, c_2)),$$

where the exclusion follows from the construction of g and the equality follows from (15) and (21). The previous two displayed statements contradict each other and hence, $\cap_{B_1 \subset A_1} S(B_1) \neq \{\emptyset\}$.

6 Applications and Examples

This section accomplishes two goals. First, we illustrate some properties of the contracting equilibrium by two examples. Second, we compare the set of outcome functions implementable by a centralized mechanism with those implementable by our contracting game.

6.1 Example 1

Suppose that there are two players. The action space of Player 1 is $\{a, b\}$ and the action space of Player 2 is $\{l, m, r\}$. The type space of Player 1 is $D = \{-1, 1\}$ and the type space of Player 2 is $T \times D = \{3, -3\} \times \{-1, 1\}$. Any realization of the type of each player is equally likely and the types of the players are independently distributed. If the type of Player 1 is d_1 and the type of Player 2 is (t, d_2), then the payoffs to the players are defined by the following matrix:

$$\begin{array}{c|c|c|c|}
 & l & m & r \\ \hline
a & 6, t & 2 + d_1, 2 + d_2 & 0, 0 \\ \hline
b & 0, -t & 0, 0 & 2, 2 \\ \hline
\end{array}.$$

We want to show that the following outcome function can be supported as an $\mathcal{R}-$equilibrium in the contracting game:

$$s\left(d_1, t, d_2\right) = \begin{cases} (a, l) & \text{if } t = 3, \\ (b, l) & \text{if } t = -3. \end{cases}$$

In this outcome function the action of Player 1 varies with the type of Player 2. Therefore, if the players played this game without the contracting stage, the outcome function s could not be implemented as a Bayesian equilibrium. Moreover, Player 2 always takes action l in the outcome function s. Given l, Player 1 would like to deviate and take action a no matter what the types are. Below we show how to construct a contract of Player 2 which prevents such a deviation.

In what follows c will denote the equilibrium contract of Player 1 and c^t denotes the equilibrium contract of Player 2, if the second coordinate of his type is t. Define the mappings determined by these contracts as follows:

$$\widetilde{c}\left(c_1, c_2\right) = \begin{cases} a & \text{if } c_2 = c^3, \\ b & \text{if } c_2 = c^{-3}, \\ \{a, b\} & \text{otherwise,} \end{cases}$$

and for $t = 3, -3$:

$$\widetilde{c}^t\left(c_1, c_2\right) = \begin{cases} l & \text{if } c_1 = c, \\ r & \text{otherwise,} \end{cases} \tag{22}$$

such that $c^3 \neq c^{-3}$. The Cross-referential Property guarantees that these contracts lie in the contract space.

Notice that these contracts refer to each other and they implement the outcome function s. Indeed, Player 1's contract, c, prescribes taking action a if the contract of Player 2 is c^3 and action b if Player 2's contract is c^{-3}. Similarly, Player 2's contracts, c^3 and c^{-3}, prescribe taking action l if the contract of Player 1 is c. We only have to show that players cannot profitably deviate.

The contracts always constrain the players to a specific action so they cannot deviate in the second stage of the game. We only need to check deviations in contracts. The payoff of Player 2 is maximized by the outcome function, so we only have to show it for Player 1. Any deviation of Player 1 triggers action r by Player 2. The best response of Player 1 to this action is b and it provides him with a payoff of 2, which is smaller than his expected equilibrium payoff, 3.

Next, we discuss a number of features of this contract equilibrium that will be of interest in the general model.

Information Content of the Contracts.— The contracts offered by Player 2 for each of the realizations of the first coordinate of his types is different. In this sense equilibrium play reveals something about Player 2's type. In fact, it is precisely this feature that makes it possible for Player 1 to take an action on the equilibrium path that depends on Player 2's type. However, there is a limit to this in the sense that Player 1's action can only vary with Player 2's type to the extent that type information is revealed through the equilibrium contract. In this example, the action of Player 1 cannot depend on d_2 because d_2 is not revealed through the equilibrium contracts.

The two contracts of Player 2, c^3 and c^{-3}, are computationally equivalent, that is, they determine the same actions as a function of the contract profile. However, $c^3 \neq c^{-3}$, and Player 2 uses his contract to communicate his type to Player 1. This communication is not cheap talk because the contract c^t is also used by Player 2 as a commitment to punish Player 1 unless he credibly promises to make his action contingent on t by offering c.[15]

On the other hand, since contracts reveal information about types, a deviating player can make his second-stage action contingent on types of the other players. This limits the set of outcome functions which are implementable by contractible contracts. We will show that a centralized mechanism designer can implement more outcome functions than our contracting game because he can prevent a non-participant player from learning something about others' types.

Invariant Punishments.— Any deviation of a player at the contracting stage triggers a restriction on the action space of the other player. These restrictions can be viewed as a punishment for a deviation. Observe that our equilibrium is supported by 'punishments' that don't vary with the transgression. For example, Player 2 simply commits himself to choose r whenever Player 1 offers a contract that is different from his equilibrium contract. The Invariant Punishment property implies that assuming that contractual punishment is invariant to the deviation is without loss of generality. (Of course, the punishment usually depends on the type of the punisher.)

We further explain the significance of this property. Since the punishment for any contractual deviation of Player 1 is the same, he can best-respond to the equilibrium contract of Player 2. That is, the most profitable deviation of Player 1 specifies a restriction which is a best response to the punishment of Player 2 given the second-stage strategies. This allows us to use a logic that is similar to the minmax logic in games of complete information. The following table summarizes the best responses of Player 1 as a function of the restrictions of Player 2:[16]

	restrictions of Player 2		
	$\{m\}$	$\{r\}$	$\{m, r\}$
best response of Player 1	$\{a\}$	$\{b\}$	$\{a\}$ if $d_1 = 1$ $\{b\}$ otherwise
expected payoff of Player 1	2	2	2.5

[15] In fact, the allocation s cannot be implemented by introducing a cheap talk stage instead of the contracting stage because Player 1 would not take action b if Player 2 takes action l.

[16] It is never optimal to punish Player 1 by taking action l. Therefore, we left this possibility out from the following table.

Consider for example the last column of the table. The punishment of Player 2 is $\{m, r\}$. What is the best response of Player 1? If $d_1 = 1$, he can restrict his action space to be $\{a\}$. Player 2 observes this restriction and is forced to take his strictly dominant action m in the second stage. Similarly, Player 1 can restrict his action space to be $\{b\}$ if $d_1 = -1$ and the best response of Player 2 is r. Player 1's expected payoff is 2.5. It is clear from the table that Player 1 can always achieve a payoff of 2 no matter what the punishment of Player 2 is.

Consider now a modification of the original game such that the action profile (a, l) generates a payoff of 3 to Player 1 instead of 6, but the payoffs are the same otherwise. In this case, our target outcome function generates a payoff of 1.5 to Player 1. This payoff is lower than 2 and hence, the outcome function cannot be implemented as an $\mathcal{R}$-equilibrium in the contracting game. Next, we argue that our outcome function could be implemented even in the modified game if the restriction of Player 2 could depend on the restriction of Player 1.

Suppose now, that the restriction of Player 2 can depend on the restriction of Player 1 generated by a deviation. Next, we present another table which summarizes the most effective punishment of Player 2, if the punishment can be contingent on the restriction implied by the deviation.

	restrictions of Player 1		
	$\{a\}$	$\{b\}$	$\{a, b\}$
best punishment of Player 2	$\{r\}$	$\{m\}$	$\{m, r\}$
expected payoff of Player 1	0	0	1.25

Therefore, if the restriction of Player 2 could depend on the restriction of Player 1, Player 1 can only achieve a payoff of 1.25. Hence, the target outcome function could be implemented even in the modified game.

Equilibrium Refinement.— The outcome functions that are supportable as equilibria will generally depend on the equilibrium refinement concept. As we explained above, an important feature of the contract equilibrium is the restriction on the action space triggered by a deviation. Since deviations are off the equilibrium path, players never have to choose from these restricted sets of actions on the equilibrium path. Different refinement concepts impose different restrictions on these off-equilibrium choices.

To see how refinement matters consider again the modified version of our example and suppose that Player 2 restricts his action space to $\{m, r\}$ whenever Player 1 deviates. Suppose that one is interested in every Bayesian equilibrium. This concept does not impose any restriction on off the equilibrium play. Hence, if Player 1 restricts his action space to be $\{a\}$, Player 2 can still take action r although it is strictly dominated. Similarly, if Player 1 restricts his action space to be $\{b\}$, Player 2 can take action m. In both of these cases, the payoff of Player 1 is zero. If Player 1 restricts his action space to be $\{a, b\}$, then Player 2 can play m if $d_2 = 1$ and r otherwise. This provides Player 1 with an expected payoff of at most 1.25. Hence, the outcome function s can be implemented as a Bayesian equilibrium even in the modified example.

6.2 Example 2

The next example illustrates two more properties of the contracting equilibria. First, we show that the informational partition is non-trivial in general. That is, players reveal some information about their types but do not reveal them fully. Second, we show that some outcome functions can only be implemented by a contract profile that does not restrict the action space of a player to a single action. That is, a player must still have some flexibility in choosing his action in the last stage of the game. In this sense, contracts are generally *incomplete*.

Suppose that $m = 2$, $T_1 = A_1 = \{-1, 1\}^2$, and $A_2 = \{-1, 1\}^2 \times \{\alpha, \beta\}$. (The type space of Player 2 is degenerate.) If the type of Player 1 is $t_1 = \left(t_1^1, t_1^2\right)$, the action of Player 1 is $a_1 = \left(a_1^1, a_2^1\right)$, and the action of Player 2 is $\left(a_2^1, a_2^2, \alpha\right)$, then the payoff of each player is

$$t_1^1 \left(a_1^1 + a_2^1\right) + t_1^2 \left(a_1^2 + a_2^2\right).$$

If the type of Player 1 is $\left(t_1^1, t_1^2\right)$ and the action of Player 2 is $\left(a_2^1, a_2^2, \beta\right)$ then the payoff of Player 2 is $5t_1^2 a_2^2$ and the payoff of Player 1 is zero.

Notice that if the third coordinate of the action of Player 2 is α, this game is a kind of coordination game where both players want to match the type of Player 1 with their actions coordinate wise. If both players do so, then each player receives a payoff of four conditional on Player 2 taking α. The problem is that if Player 2 knows the type of Player 1, he prefers to take action β and match the second coordinate of Player 1's type with the second coordinate of his action. This would provide him with a payoff of five.

Consider the following outcome function: $s_1 \left(t_1^1, t_1^2\right) = \left(t_1^1, t_1^2\right)$ and $s_2 \left(t_1^1, t_1^2\right) = \left(t_1^1, 1, \alpha\right)$. In this outcome function each player matches the first coordinate of the type of Player 1 but only Player 1 matches the second coordinate. This outcome function generates an expected payoff of three to each player. Next, we argue that this outcome function can be implemented by contractible contracts as an equilibrium. To see this, consider a PMM where Player 1 reports t_1^1 at the first stage. The mechanism imposes no restriction on the action space of the players. In the second stage, Player 1 takes action $\left(t_1^1, t_1^2\right)$ and Player 2 takes action $\left(t_1^1, 1, \alpha\right)$. Obviously, no player has incentive to deviate.

Notice that the information partition is described be $\tau_1 \left(t_1\right) = t_1^1$ for all t_1. Observe that (s_1, s_2) cannot be implemented such that all the information is revealed about Player 1's type. This is because if Player 1 would fully reveal his type, Player 2 would not participate in the mechanism and would take action $\left(1, t_1^2, \beta\right)$.

Also notice that (s_1, s_2) cannot be implemented by complete contracts. This is because Player 1 has to match the second coordinate of his own type by the second coordinate of his action. Recall that he can only reveal the first coordinate of his type. Therefore, it must be the case that $r_1 \left(t_1^1, t_1^2\right)$ contains both $\left(t_1^1, -1\right)$ and $\left(t_1^1, 1\right)$.

6.3 Comparison with Centralized Mechanisms

The set of implementable outcome functions in the contracting game is fairly large. However, as we mentioned before, contract equilibrium imposes a restriction on feasible outcome functions. When a player decides to deviate at the contracting stage, he knows that he will learn something about the types of the other players when he observes their contracts. Therefore, a deviator's action in the last stage of the game can depend on the information about the types of the other players that are revealed by their contracts. This suggests that there are outcome functions which are implementable by centralized mechanisms (where the messages are private) but not by contracts.

By the standard Revelation Principle, a centralized mechanism asks the players to report their types *privately.* Then the mechanism requires each participant to take an action as a function of the reported type profile of the participants. If a player does not participate in the mechanism, he can take any action he wants. It is without loss of generality to restrict attention to mechanism-equilibrium pairs where each player participates and truth-telling constitutes a Bayesian equilibrium.[17]

A centralized mechanism has to specify the target outcome function, $s : T \rightarrow A$, and what actions the others take if Player i does not participate: $\underline{s}_{-i}^{i} : T_{-i} \rightarrow A_{-i}$ for each i. Then the outcome function s is implementable by a centralized mechanism if and only if the following two sets of constraints are satisfied. For all $i \in \{1, ..., m\}$ and $t_i, t_i' \in T_i$:

$$E_{t_{-i}} \left(u_i \left(s \left(t \right), t \right) \ \mid t_i \right) \geq E_{t_{-i}} \left(u_i \left(s \left(t_i', t_{-i} \right), t \right) \ \mid t_i \right), \tag{23}$$

and

$$E_{t_{-i}} \left(u_i \left(s \left(t \right), t \right) \ \mid t_i \right) \geq \max_{a_i \in A_i} E_{t_{-i}} \left(u_i \left(a_i, \underline{s}_{-i}^{i} \left(t_{-i} \right), t \right) \ \mid t_i \right). \tag{24}$$

The inequality (23) is the incentive compatibility constraint guaranteeing that a participant player reports his type truthfully. The inequality (24) is the participation constraint guaranteeing that each player prefers to participate irrespective of his type.

To show that one can implement more outcome functions with centralized mechanisms than with contracts, we revisit the example of the previous subsection. Consider the outcome function $s^* = (s_1^*, s_2^*) : T \rightarrow A$, such that $s_1^* \left(t_1^1, t_1^2 \right) = \left(t_1^1, t_1^2 \right)$ and $s_2^* \left(t_1^1, t_1^2 \right) = \left(t_1^1, t_1^2, \alpha \right)$. Notice that this outcome function provides each player with a payoff of four, and it maximizes the sum of the players' payoff among all outcome functions. We show that this outcome function can be implemented by a centralized mechanism but cannot be implemented by a PMM as an $\mathcal{R}$–equilibrium.

In order to show that s^* can be implemented by a centralized mechanism, it is enough to construct $\underline{s}_1^2 : T_1 \rightarrow A_1$ and $\underline{s}_2^1 : T_2 \rightarrow A_2$ such that (23) and (24) are satisfied for $i = 1, 2$ with $s = s^*$. Define $\underline{s}_1^2 \left(t_1^1, t_1^2 \right) \equiv (1, 1)$, that is if Player 2 does not participate, the mechanism requires

[17] Notice that the mechanism restricts the action space of the participants to singletons, that is, participating players do not choose actions strategically at the second stage. Therefore, each Bayesian equilibrium is also an $\mathcal{R}-$equilibrium.

Player 1 to take action $(1,1)$. Similarly, define $\underline{s}_2^1 \equiv (1,1,\beta)$. Notice that s^* maximizes the payoff of Player 1 among all outcome functions. Hence, (23) and (24) are satisfied for $i=1$. Since, the type space of Player 2 is degenerate, we only have to show that Player 2 prefers to participate, that is, (24) holds for $i=2$. Notice that the right-hand side of (24) is zero and the left-hand side is four.

Now, we argue that s^* cannot be implemented by a PMM. Notice that in order to implement s^*, Player 2 must know the type of Player 1. Therefore Player 1 has to fully reveal his type at the first stage of the mechanism. If Player 2 decides not to participate and Player 1 reveals that his type is $\left(t_1^1, t_1^2\right)$, Player 2 can take action action $\left(1, t_1^2, \beta\right)$ in the last stage of the mechanism. This would generate a payoff of five, which is larger than the payoff generated by s^*.

Next, we identify some environments where the set of outcome functions implementable by centralized mechanisms is the same as the set of outcome functions implementable by contracts.

Assumption 1. For all i and $t_i \in T_i$, there exist $\overline{a}_i(t_i) \in A_i$, $\overline{a}_{-i}^i \in A_{-i}$, and $U_i : T \to \mathbb{R}$ such that for all $(t_i, t_{-i}) \in T$:

(i) $u_i\left(\overline{a}_i(t_i), a_{-i}, t\right) \geq U_i(t)$ for all $a_{-i} \in A_{-i}$, and

(ii) $u_i\left(a_i, \overline{a}_{-i}^i, t\right) \leq U_i(t)$ for all $a_i \in A_i$.

Part (i) of Assumption 1 says that Player i has an action for each of his type which provides him a payoff of at least $U_i(t)$ no matter what the action profile of the other players is. Part (ii) says that players other than Player i can take an action profile which holds Player i down to at most $U_i(t)$, no matter what action Player i takes.

Assumption 1 is arguably a strong assumption but is satisfied in many economic environments. One way to interpret part (i) is that Player i can choose not to interact with the other players and take his outside option. The value $U_i(t)$ can be thought of as the value of the outside option. Similarly, the action profile $\overline{a}_{-i}^i$ can be thought of as a profile of the other players which forces Player i to exit.

Consider, for example, an auction environment where a single seller is selling a single object to many bidders. The players in this environment are the bidders and the seller. The type of a player is his signal about the value of the object. The action space of a bidder is the amount of transfers to the seller, and the action space of the seller is set of players (to whom he can sell the object). In this environment Assumption 1 is obviously satisfied because a bidder can choose not to pay and the seller can decide to keep the object.

Proposition 1 *Suppose that Assumption 1 is satisfied, and the allocation $s : T \to A$ can be implemented by a centralized mechanism. Then the allocation s can be implemented as an $\mathcal{R}$-equilibrium in the contractible contracting game.*

Proof. Suppose that a centralized mechanism implements s, that is, there exists $\left\{\underline{s}_{-i}^i\right\}_i$ such that (23) and (24) are satisfied. Notice that by part (i) of Assumption 1:

$$\max_{a_i \in A_i} E_{t_{-i}}\left(u_i\left(a_i, \underline{s}_{-i}^i(t_{-i}), t\right) : t_i\right) \geq E_{t_{-i}}\left(U_i(t) : t_i\right).$$

Therefore, (24) and the previous inequality implies:

$$E_{t_{-i}}\left(u_i\left(s\left(t\right),t\right):t_i\right)\geq E_{t_{-i}}\left(U_i\left(t\right):t_i\right). \tag{25}$$

Next, we construct a PMM which implements the outcome function s. Consider the following PMM defined by $\left(s,\left\{\tau_i\right\}_{i=1}^m,\left\{r_i\right\}_{i=1}^m,\left\{p_i^j\right\}_{i,j=1}^m\right)$, where $\tau_i\left(t_i\right)=\left\{t_i\right\}$, $r_i\left(t\right)=\left\{s_i\left(t\right)\right\}$, $p_i^j\left(t_{-j}\right)=\left\{\overline{a}_i^j\right\}$. That is, the information partition is the full information partition, the equilibrium restrictions are singletons corresponding to s, and if Player j does not participate, Player i's action space is restricted to be the singleton $\left\{\overline{a}_i^j\right\}$. Since, $p_i^j\left(t_{-j}\right)=\left\{\overline{a}_i^j\right\}$, $s_i^j\left(t_{-j}\right)$ must be $\overline{a}_i^j$. We have to show that (7), (8), and (9) are satisfied. Since $p_i^j\left(t_{-j}\right)$ is singleton for all i, j, and t_{-j}, (9) is satisfied. Since $r_i\left(t_i',t_{-i}\right)=\left\{s_i\left(t_i',t_{-i}\right)\right\}$, (7) coincides with (23). Since $\tau_i\left(t_i\right)=\left\{t_i\right\}$ and $p_i^j\left(t_{-j}\right)=\left\{\overline{a}_i^j\right\}$, (8) can be rewritten as

$$E_{t_{-i}}\left(u_i\left(s\left(t\right),t\right)\ \mid t_i\right)\geq E_{t_{-i}}\left(\max_{a_i\in A_i}u_i\left(a_i,\overline{a}_{-i}^i,t\right)\ \mid t_i\right).$$

By parts (i) and (ii) of Assumption 1, the right hand side of the previous inequality is $E_{t_{-i}}\left(U_i\left(t\right)\ \mid t_i\right)$. Hence, this inequality is just (25), which is indeed satisfied. ■

Notice that the statement of the proposition holds for any refinement. This is because the outcome function s can be implemented by a PMM in which all the restrictions on the action spaces of the players are singletons. That is, even off the equilibrium path, players do not make strategic choices at the second stage.

6.4 Complete Information Environment

In this section, we characterize the set of pure-strategy Subgame Perfect Nash Equilibrium (SPNE) in our model if the players do not possess private information. We prove a pure-strategy Folk Theorem for this environment. That is, we show that for each player there exists a value, such that an outcome function is implementable as a SPNE if and only if the payoff of each player is larger than his value.

Define the value for Player i as;

$$\underline{u}_i=\min_{a_{-i}\in A_{-i}}\max_{a_i\in A_i}u_i\left(a_i,a_{-i}\right),$$

We refer to $\underline{u}_i$ as the pure minmax value of Player i.

Theorem 2 *The action profile $a^*=\left(a_1^*,...,a_m^*\right)\in A$ is supportable as a pure-strategy SPNE outcome in the contracting game if and only if $u_i\left(a^*\right)\geq\underline{u}_i$ for each i.*

Proof. Suppose first that $u_i\left(a^*\right)\geq\underline{u}_i$ for all $i\in\{1,...,m\}$. We construct a PMM which implements a^* as a SPNE. For each j, let us fix an action a_i^j for Player i $(i\neq j)$, such that,

$$\left(a_1^j,...,a_m^j\right)\in\arg\min_{a_{-j}\in A_{-j}}\max_{a_j\in A_j}u_j\left(a_j,a_{-j}\right),$$

and let

$$\underline{a}_j \in \arg\max u_j \left(a_j, a_{-j}^j\right).$$

For all j and i $(i \neq j)$, define $r_i = \{a_i^*\}$, $p_i^j = \left\{a_i^j\right\}$, and $s_i^j(f_j) = a_i^j$ for all $f_j \in F_j$. Notice that $s_i^j(f_j) \in p_i^j$ and the restriction imposed by subgame perfection is irrelevant because the action space of each player is restricted to a single action followed by any deviation. By Theorem, 1, in order to prove that these strategies constitute a Nash equilibrium, we have to show that both (7) and (8) are satisfied. Notice that since players have no private information, (7) boils down to:

$$u_i(a^*) \geq \max_{a_i \in r_i} u_i\left(a_i, a_{-i}^*\right).$$

Since $r_i = \{a_i^*\}$, this inequality is obviously satisfied. Again, since players have no types, (8) can be rewritten as:

$$u_i(a^*) \geq \max_{f_i \in F_i} \max_{a_i \in f_i} u_i\left(a_i, s_{-i}^i(f_i)\right) = \max_{f_i \in F_i} \max_{a_i \in f_i} u_i\left(a_i, a_{-i}^i\right),$$

where the equality follows from $s_{-i}^i(f_i) = a_{-i}^i$. Notice that the right-hand-side is just $\underline{u}_i$, and hence, this inequality is indeed satisfied.

Suppose now that a PMM implements a^*. Then, by Theorem 1, there exist on-equilibrium restrictions $r_i \in 2^{A_i} \setminus \{\emptyset\}$, off-equilibrium restrictions, $p_i^j \in 2^{A_i} \setminus \{\emptyset\}$, and off-equilibrium strategies $s_i^j : F_j \to A_i$, $s_i^j(f_j) \in p_i^j$, such that (7) and (8) are satisfied. Hence:

$$u_i(a^*) \geq \max_{f_i \in F_i} \max_{a_i \in f_i} u_i\left(a, s_{-i}^i(f_i)\right) \geq \max_{a_i \in A_i} u_i\left(a_i, s_{-i}^i(A_i)\right) \geq \underline{u}_i.$$

The first inequality is just (8), the second one follows from setting $f_i = A_i$, and the third one follows from the definition of $\underline{u}_i$. We can conclude that (8) can only be satisfied if $u_i(a^*) \geq \underline{u}_i$ for all i. ■

One of the implications of Theorem 2 is that for the complete information case, the set of outcome functions that are implementable by a centralized mechanism designer is identical to the set of outcome functions that can be supported by equilibrium in the contracting game. As we have shown (by example) that the same is not generally true for games of incomplete information, this result serves to highlight one of the uses of our characterization theorem.

7 Conclusion

This paper shows how the contracts on contracts approach can be extended to environments with incomplete information. Definable contracts constitute the largest class of arithmetic contracts which can be written as a finite text in a first order language. In this sense definable contracts embed most other interesting classes of feasible contracts as subsets.

In contrast to the complete information case, we show that the 'folk theorem' doesn't generally hold in the following sense. A centralized mechanism designer can implement outcome functions that can't be supported as equilibrium with contractible contracts. This limitation is not a consequence of the set of feasible contracts, but rather of the fact that public contracts reveal information about non-deviators' type. The restriction to definable contracts allows us to provide a complete characterization of equilibrium and to prove this result. One of the results we provide as a part of our main theorem illustrates the role that punishments play in a static contracting environment.

We emphasize that this paper does not intend to do mechanism design, and restricting attention to our two-stage game is with loss of generality. For example, if we allow further communication after the contracting stage, the set of implementable outcome functions becomes larger. Similarly, allowing players to offer contracts and take actions sequentially leads to a different characterization theorem.

8 Appendix

8.1 Refinement

This section shows how to extend our characterization theorems to more general equilibrium refinements. A refinement is a restriction on the strategy rules of the different types of non-deviating players in the second stage following a deviation by some Player i in the first stage. In the main text, we only ruled out the possibility that players choose strictly dominated actions. Strict dominance is a notion that depends on feasible sets of actions and payoff functions, but not on the game in which these are embedded. Generally, refinements impose restrictions that can depend on the sets to which the players are constrained in the second stage, the information that has been revealed by the non-deviators' first period contracts and the outcome that would have prevailed had there not been any deviation. Informally, the sets to which players are constrained when choosing their second period actions are used to determine whether or not some actions are dominated for certain player types. The information conveyed by first period play is used for refinements like perfect Bayesian equilibrium that require the use of Bayes rule for making inferences about non-deviating players. The original equilibrium outcome is used in refinements like the 'intuitive criterion', which restrict beliefs that non-deviators can have about the deviating player based on what he might have expected to gain by deviating.

Formally, let $s : T \to A$ denote the equilibrium outcome function, that is, $s(t) = \alpha^*(t, \gamma^*(t))$ for each t. Let τ_i be the partition of T_i generated by i's equilibrium strategy, or in the context of our contract game, $\tau_i(t_i) = \{t_i' : \gamma_i^*(t_i') = \gamma_i^*(t_i)\}$. Let $\mathcal{F}^i$ be a τ_{-i} measurable correspondence from T_{-i} into A representing the sets to which the players are constrained when choosing their actions following a deviation by Player i. If Player i deviates to contract γ_i, then this correspondence is $\mathcal{F}^i(t_{-i}) = S\left(\gamma_i, \gamma_{-i}^*(t_{-i})\right)$.

A *refinement* specifies for every deviator i, every $\mathcal{F}^i$ measurable with respect to some information partition τ_{-i}, and every status quo outcome function, s, a non-empty set of action profiles for the non-deviators that the refinement allows for each profile of their types. Let

$\mathcal{R}_i\left(s,\mathcal{F}^i,\tau,t_{-i}\right) \in 2^{A_{-i}}\backslash\{\emptyset\}$ describe this correspondence.[18] If an equilibrium has the non-deviators using strategy rules $\alpha^*_{-i}\left(t_{-i},\left(\gamma_i,\gamma^*_{-i}\left(t_{-i}\right)\right)\right)$ in response to a deviation to contract γ_i by Player i, then the equilibrium satisfies the refinement if

$$\alpha^*_{-i}\left(t_{-i},\left(\gamma_i,\gamma^*_{-i}\left(t_{-i}\right)\right)\right) \in \mathcal{R}_i\left(s,S\left(\gamma_i,\gamma^*_{-i}\left(t_{-i}\right)\right),\tau,t_{-i}\right)$$

for each t_{-i}.

The properties to be built into the refinement correspondence $\mathcal{R}_i$ are going to depend on the particular application. For example, if types are independently distributed, then perfect Bayesian equilibrium is well defined and $\mathcal{R}_i\left(s,S\left(\gamma_i,\gamma^*_{-i}\left(t_{-i}\right)\right),\tau,t_{-i}\right)$ would, for each t_{-i} consist of all action profiles for the non-deviators that constitute actions these types would jointly take in some Bayesian equilibrium of the game with action spaces $S\left(\gamma_i,\gamma^*_{-i}\left(t_{-i}\right)\right)$ and beliefs given by posterior beliefs conditional on non-deviators types lying in $\tau_{-i}\left(t_{-i}\right)$.[19] In a game of complete information with equilibrium contracts $\left(c^*_i,c^*_{-i}\right)$ and equilibrium outcome a^*, a deviation to c' by Player i supports the collection of action profiles $\mathcal{F}^i = S\left(c',c^*_{-i}\right)$, which are just the actions to which players are constrained in the second stage by their first period contracts. The subgame perfection refinement of Nash equilibrium would specify $\mathcal{R}_i\left(s,\mathcal{F}^i\right) \subset \mathcal{N}\left(\mathcal{F}^i\right)$ where $\mathcal{N}\left(\mathcal{F}^i\right)$ is just the set of action profiles $a_{-i} \in A_{-i}$ for which there exists an action $a_i \in S_{-i}\left(c',c^*_{-i}\right)$ such that $\left(a_i,a_{-i}\right)$ constitutes a Nash equilibrium of the game with action spaces $S\left(c',c^*_{-i}\right)$.

Finally, if we simply want to describe Bayesian equilibrium, then we could do so by having $\mathcal{R}_i\left(s,\mathcal{F}^i,\tau,t_{-i}\right) = A_{-i}$. We refer to the collection of restrictions $\mathcal{R} = \{\mathcal{R}_i\}_{i=1,\ldots,m}$ as a *refinement.*

Fix a refinement $\mathcal{R}$. Let τ be the information partition induced by the equilibrium strategies γ^* (that is, $\tau\left(t\right) = \{t' \in T : \gamma^*\left(t'\right) = \gamma^*\left(t\right)\}$). We say that $\left(\gamma^*,\alpha^*\right)$ is an $\mathcal{R}$-equilibrium of the contracting game if (2) holds, and in addition

$$\alpha^*_{-i}\left(t_{-i},\overline{\alpha}\right) \in \mathcal{R}_i\left(\alpha^*\left(\cdot,\gamma^*\left(\cdot\right)\right),S\left(\gamma_i,\gamma^*_{-i}\left(t_{-i}\right)\right),\tau,t_{-i}\right),$$

for every i and $\gamma_i \in C_i$ (recall the notation $\overline{\alpha} = \left(\alpha_i,\alpha^*_{-i}\right)$).

Recall from Section 4 that an outcome function is implementable by a PMM if and only if (7), (8) and (9) hold. Using the more general refinement described above, we only have to modify (9) as follows. For every i, $t_{-i} \in T_{-i}$, and $f_i \in F_i$

$$s^i\left(t_{-i},f_i\right) \in \mathcal{R}_i\left(s,f_i \times p^i_{-i},\tau,t_{-i}\right). \tag{26}$$

Then an outcome function is $\mathcal{R}$-implementable by a public message mechanism if and only if (7), (8), and (9) hold.

[18] It is reasonable to require $\mathcal{R}_i$ to be measurable with respect to τ_{-i} but we do not need this property for our formalism.

[19] Perfect Bayesian equilibrium does not have a well accepted definition when types are correlated. To see why, observe that when Player i deviates, non-deviators have to make *some* inference about his type. The on path choices of the non-deviators reveal their types to be in some subset. The distribution of non-deviators' types within this subset depends on the deviating player's type. So either inferences about non-deviating players depend on actions of the deviator, or the common prior assumption has to be abandoned.

Finally, the statement of Theorem 1 is valid even if $\mathcal{R}$ denotes restrictions imposed by a more general refinement concept. The proof is essentially identical to the one in Section 4.

8.2 Definability

Our goal here is to provide formal definitions for *arithmetic statement* and arithmetic statements with free variables. We shall define statements for any first-order logic and explain what is specific about Number Theory.

Each formal language has a set of symbols. The symbols of a first-order language are divided into two disjoint sets: the logical-symbols, and the non-logic symbols. The logical-symbols include: (,), $\forall$, $\exists$, $\neg$, $=$, and infinitely many *variable symbols*, $x_0, x_1, ...$ The non-logic symbols include function-symbols and relation-symbols.

Definition 4 *$t = \langle F, R, \tau \rangle$ is a similarity type, where F is a set of function-symbols, R is a set of relation-symbols, and $\tau : F \cup R \to \mathbb{N}$ such that $\tau(r) > 0$ if $r \in R$.*

The function τ tells how many variables do the functions and relations have. If $\tau(f) = 0$, then f is referred to as a *constant-symbol*.

Example. One of the similarity type corresponding to the Peano Arithmetics, denoted by $q = \langle F, R, \tau \rangle$, is: $F = \{0, 1, +, *\}$, $R = \{<\}$, $\tau(0) = \tau(1) = 0$, $\tau(+) = \tau(*) = \tau(<) = 2$. Notice that the zero and the one are considered as functions with zero variables, that is, constant symbols. (We point out that the similarity type of arithmetics can be defined without the relation"$<$". This relation can be then defined recursively.)

Definition 5 *Let $t = \langle F, R, \tau \rangle$ be a similarity type. Then the set of* expressions *of type t, denoted by $K(t)$, is the smallest set for which:*

(i) $x \in K(t)$ for all variable symbols,

(ii) For all $f \in F$, if $\tau(f) = 0$ then $f \in K(t)$,

(iii) For all $f \in F$, if $\tau(f) = n$, and $k_1, ..., k_n \in K(t)$ then $f(k_1, ..., k_n) \in K(t)$.

Suppose that $t = q$. Then the following string of symbols are expressions in arithmetics: x, 0, 1, $x + 1$, $((x + 1) * (y + 1) + 1)$ etc.

Finally we are ready to define the set of statements corresponding to a similarity type.

Definition 6 *Let $t = \langle F, R, \tau \rangle$ be a similarity type. Then the set of* statements *of type t, denoted by $F(t)$, is the smallest set for which:*

(i) if $r \in R$, $\tau(r) = n$, and $k_1, ..., k_n \in K(t)$ then $r(k_1, ..., k_n) \in F(t)$,

(ii) if $k_1, k_2 \in K(t)$ then $k_1 = k_2 \in F(t)$

(iii) if $\phi, \eta \in F(t)$, then $(\phi) \vee (\eta) \in F(t)$, $\neg(\phi) \in F(t)$, and $\exists x(\phi) \in F(t)$.

The set of arithmetic statements are defined according to the previous definition with $t = q$. Then the following string of symbols are statements in arithmetics: $x = y$, $\neg\exists x\exists y\ (y = x + 1)$, etc.

For each statement, one can enumerate the number of different variable symbols appearing in the statement. A variable is called *free variable* in a statement if it does not appear right behind a quantifier. For example, the statement $\neg\exists x\exists y\ ((y = x + 1) \vee (z = 1))$ has three variable symbols: x, y, and z. However, both the x and the y appears behind a quantifier. Hence, the only free variable of this statement is z.

8.3 Proofs

Proof of Lemma 2. (i) Suppose that $A = \left\{n^1, ..., n^q\right\}$ where $n^i = \left(n_1^i, ..., n_k^i\right) \in \mathbb{N}^k$ for $i = 1, ..., q$. The characteristic function of the set A can be defined by the following statement in $k+1$ free variables:

$$\begin{aligned}\phi\left(x_1, ..., x_k, y\right) \quad \equiv \quad & \left(\vee_{i=1}^{q}\left(\wedge_{j=1}^{k}\left(x_j = n_j^i\right)\right) \wedge (y = 1)\right) \\ & \vee\left(\neg\left(\vee_{i=1}^{q}\left(\wedge_{j=1}^{k}\left(x_j = n_j^i\right)\right)\right) \wedge (y = 0)\right).\end{aligned}$$

(ii) Suppose that φ_A defines χ_A, φ_B defines χ_B, and let x denote $(x_1, ..., x_k)$. Then $\chi_{A\cap B}$, $\chi_{A\cup B}$ and $\chi_{A\smallsetminus B}$ are defined by

$$\begin{aligned}&\left[\varphi_A(x,1) \wedge \varphi_B(x,1) \wedge (y = 1)\right] \vee \left[\neg\left(\varphi_A(x,1) \wedge \varphi_B(x,1)\right) \wedge (y = 0)\right], \\ &\left[\left(\varphi_A(x,1) \vee \varphi_B(x,1)\right) \wedge (y = 1)\right] \vee \left[\neg\left(\varphi_A(x,1) \vee \varphi_B(x,1)\right) \wedge (y = 0)\right], \\ &\left[\varphi_A(x,1) \wedge \varphi_B(x,0) \wedge (y = 1)\right] \vee \left[\neg\left(\varphi_A(x,1) \wedge \varphi_B(x,0)\right) \wedge (y = 0)\right],\end{aligned}$$

respectively.

(iii) Let ϕ_{A_i} denote a statement defining χ_{A_i} for each i. Furthermore, let $B_i = \left\{b^{i1}, ..., b^{iq_i}\right\}$. Then the following statement obviously defines the function f:

$$\begin{aligned}&\left(\vee_{i=1}^{m}\left(\phi_{A_i}\left(x_1, ..., x_k, 1\right) \wedge \left(\vee_{l=1}^{q_i}\left(y = b^{il}\right)\right)\right)\right) \\ &\vee\left(\wedge_{i=1}^{m}\phi_{A_i}\left(x_1, ..., x_k, 0\right) \wedge \left(\vee_{l=1}^{q_{m+1}}\left(y = b^{(m+1)l}\right)\right)\right).\end{aligned}$$

(iv) Let ϕ_D define the characteristic function of $D = \times_{i=1}^{k} D_i$, let ϕ_{D_j} define the characteristic function of D_j, and let φ_n define the characteristic function of $g(n)$ for $n \in D$. The statements ϕ_D, ϕ_{D_j}, and φ_n exist because of part (i) of this lemma. Furthermore, let $A_i = \left\{n \in \mathbb{N}^k : \{i : n_i \notin D_i\} = i\right\}$ for $i \in \{1, ..., k\}$. The following statement defines the characteristic function of A_i for $i \in \{1, ..., k\}$:

$$\begin{aligned}\psi_i\left(x_1, ..., x_k, z\right) \quad = \quad & \left(\left(\wedge_{\substack{j=1 \\ j\neq i}}^{k}\phi_{D_j}\left(x_j, 1\right)\right) \wedge \phi_{D_i}\left(x_i, 0\right) \wedge (z = 1)\right) \\ & \vee\left(\neg\left(\left(\wedge_{\substack{j=1 \\ j\neq i}}^{k}\phi_{D_j}\left(x_j, 1\right)\right) \wedge \phi_{D_i}\left(x_i, 0\right)\right) \wedge (z = 0)\right).\end{aligned}$$

Finally, let ξ_j define the characteristic function of B_j for $j = 1, ..., k+1$. Then the following statement defines the function f:

$$\left(\phi_D(x,1) \wedge \left(\vee_{n \in D}\left(\varphi_n(y,1) \wedge (x=n)\right)\right)\right) \vee_{i=1}^{k} \left(\phi_D(x,0) \wedge \psi_i(x,1) \wedge \xi_i(y,1)\right)$$
$$\vee \left(\neg \left(\left(\phi_D(x,1) \wedge \left(\vee_{n \in D}\left(\varphi_n(y,1) \wedge (x=n)\right)\right)\right) \vee_{i=1}^{k} \left(\phi_D(x,0) \wedge \psi_i(x,1) \wedge \xi_i(y,1)\right)\right) \wedge \xi_{k+1}(y,1)\right).$$

■

Lemma 5 *Let $N_i \geq 1$ for all $i \in \{1, ..., m\}$. Suppose that $h_j^{n_j} : \mathbb{N}^{\times_j^m N_j} \to 2^{\mathbb{N}}$ is definable and $|h_j^{n_j}(q)| = 1$ for all $j \in \{1, ..., m\}$, $n_j \in \mathcal{N}_j$ and q. Suppose that $\overline{r} : \mathcal{N} \to 2^{\mathbb{N}} \backslash \{\emptyset\}$, and $\overline{p}_i^j : \mathcal{N}_{-j} \to 2^{\mathbb{N}} \backslash \{\emptyset\}$ for each $i, j \in \{1, ..., m\}$ $(i \neq j)$, such that $\overline{r}(n)$ and $\overline{p}_i^j(n_{-i})$ are finite for each $n \in \mathcal{N}$, $n_{-i} \in \mathcal{N}_{-i}$ and i. In addition, let $(\{\emptyset\} \neq) \overline{A}_i \subset \mathbb{N}$ be a finite set. For all $(q_1, ..., q_m) \in \mathbb{N}^{\times_i^m N_i}$ (where $q_i = \left(q_i^1, ..., q_i^{N_i}\right) \in \mathbb{N}^{N_i}$) and $l = (l_1, ..., l_m) \in \mathbb{N}^m$, define $f^{i,n_i}(q,l)$ as follows:*

$$f^{i,n_i}(q,l) \tag{27}$$
$$= \begin{cases} \overline{r}(n_i, n_{-i}) & \textit{if } \forall k \neq i \ h_k^{n_k}(q) = l_k \textit{ and } h_k^{n_k}(q) \neq h_k^{n'_k}(q) \ \textit{ if } n_k \neq n'_k \\ \overline{p}_i^j(n_i, n_{-ij}) & \begin{array}{c}\textit{if } \forall k \neq i,j \ h_k^{n_k}(q) = l_k, \textit{ and } h_k^{n_k}(q) \neq h_k^{n'_k}(q) \ \textit{ if } n_k \neq n'_k \\ \textit{and } \nexists n_j \ \textit{s.t.} h_j^{n_j}(q) = l_j, \end{array} \\ \overline{A}_i & \textit{otherwise.} \end{cases}$$

Then the function $f^{i,n_i} : \mathbb{N}^{\times_i^m N_i + m} \to 2^{\mathbb{N}}$ is definable.

Proof. Let $\mathcal{N}_i$ denote $\{1, ..., N_i\}$ and $\mathcal{N} = \times_{i=1}^m \mathcal{N}_i$. Let $\varphi_i^{n_i}(x, y_i)$ define $h_i^{n_i}$, where x is an $|\mathcal{N}|$-dimensional vector of variable symbols, and y_i is a variable symbol. Let $\theta_r^n(z,v)$ define the characteristic function of the set $\overline{r}(n)$, let $\theta_{p_i^j}^n(z,v)$ define the characteristic function of $\overline{p}_i^j$ and let $\theta_{A_i}(z,v)$ define the characteristic function of $\overline{A}_i$. The letters z and v are variable symbols.

For all $n = (n_1, ..., n_m) \in \mathbb{N}^m$, $n_k \in \{1, ..., N_k\}$ define

$$\psi^n(x,y) \equiv \wedge_{k \neq i}\left(\varphi_k^{n_k}(x, y_k) \wedge_{\substack{n'_k \in \{1,...,N_k\} \\ n'_k \neq n_k}} \left(\neg \varphi_k^{n'_k}(x, y_k)\right)\right).$$

Notice that $\psi^n(q,l)$ is true if and only if $h_k^{n_k}(q) = l_k$ for all $k \neq i$, and $h_k^{n_k}(q) \neq h_k^{n'_k}(q)$ whenever $n_k \neq n'_k$. That is, ψ^n corresponds to the condition in the first line of (27).

Similarly,

$$\psi_j^{n_{-j}}(x, y_{-j}) \equiv \wedge_{k \neq i,j}\left(\varphi_k^{n_k}(x, y_k) \wedge_{\substack{n'_k \in \mathcal{N}_k \\ n'_k \neq n_k}} \left(\neg \varphi_k^{n'_k}(x, y_k)\right)\right) \wedge_{n_j \in \mathcal{N}_j} \left(\neg \varphi_j^{n_j}(x, y_j)\right).$$

Notice that $\psi_j^{n_{-j}}(q, l_{-j})$ is true if and only if $h_k^{n_k}(q) = l_k$, $h_k^{n_k}(q) \neq h_k^{n'_k}(q)$ whenever $n_k \neq n'_k$ for all $k \neq i, j$, and there is no $n_j\,(\in \mathcal{N}_j)$ s.t. $h_j^{n_j}(q) = l_j$. That is, $\psi_j^{n_{-j}}(x, y_{-j})$ corresponds to the second line of (27).

We are ready to construct a statement which defines the f^{i,n_i} : ■

$$\vee_{n\in\mathcal{N}}\left[\psi^{n}\left(x,y\right)\wedge\theta_{r}^{n}\left(z,1\right)\right]\vee_{j,n_{-j}\in\mathcal{N}_{-j}}\left[\psi_{j}^{n}\left(x,y_{-j}\right)\wedge\theta_{p_{i}^{j}}^{n}\left(z,1\right)\right]\vee$$
$$\left(\neg\left(\vee_{n\in\mathcal{N}}\left[\psi^{n}\left(x,y\right)\wedge\theta_{r}^{n}\left(z,1\right)\right]\vee_{j,n_{-j}\in\mathcal{N}_{-j}}\left[\psi_{j}^{n}\left(x,y_{-j}\right)\wedge\theta_{p_{i}^{j}}^{n}\left(z,1\right)\right]\right)\wedge\theta_{A_{i}}\left(z,1\right)\right).$$

Lemma 6 *The function described by (16) is a definable function with* $|\times_{i=1}^{m}N_i|$ *free variables.*

Proof. By Definition 3, we have to show that $f^{i,n_i}:\mathbb{N}^{\times_1^m N_j}\times\mathbb{N}^m\rightarrow 2^{\mathbb{N}}$ is definable where $f^{i,n_i}\left(q,l\right)$ is defined by

$$\begin{cases}\left[r_i\left(n_i,n_{-i}\right)\right] & \text{if } \forall k\neq i\ h_k^{n_k}\left(q\right)=l_k \text{ and } h_k^{n_k}\left(q\right)\neq h_k^{n_k'}\left(q\right)\ \text{ if } n_k\neq n_k' \\ \left[p_i^j\left(n_i,n_{-ij}\right)\right] & \begin{array}{c}\text{if } \forall k\neq i,j\ h_k^{n_k}\left(q\right)=l_k,\ \text{ and } h_k^{n_k}\left(q\right)\neq h_k^{n_k'}\left(q\right)\ \text{ if } n_k\neq n_k' \\ \text{and } \nexists n_j \text{ s.t.} h_j^{n_j}\left(q\right)=l_j,\end{array} \\ \left[A_i\right] & \text{otherwise.}\end{cases}$$

for all $l=\left(l_1,...,l_m\right)\in\mathbb{N}^m$, $q_j=\left(q_j^1,...,q_j^{N_j}\right)\in\mathbb{N}^{N_j}$, and $q=\left(q_1,...,q_m\right)$.

Notice that $|h_k^{n_k}\left(q\right)|=1$ for all q and $h_k^{n_k}$ is definable by Lemma 1. Define $\overline{r}_i\left(n_i,n_{-i}\right)$ to be $\left[r_i\left(n_i,n_{-i}\right)\right]$ and $\overline{p}_i^j\left(n_i,n_{-ij}\right)$ to be $\left[p_i^j\left(n_i,n_{-ij}\right)\right]$. Finally let $\overline{A}_i=\left[A_i\right]$. Notice that Notice that $\overline{r}_i\left(n_i,n_{-i}\right)$, $\overline{p}_i^j\left(n_i,n_{-ij}\right)$ and $\overline{A}_i$ are finite sets of $\mathbb{N}$. Hence, the statement of the lemma follows from Lemma 5. ■

Lemma 7 *Suppose that* $g:\mathbb{N}^k\rightarrow 2^B\setminus\{\emptyset\}$ *is a definable function where* $B\left(\subset\mathbb{N}^q\right)$ *is finite. Let* $f:2^B\rightarrow 2^D$ *an arbitrary function where* $D\left(\subset\mathbb{N}^l\right)$ *is finite. Then* $f\circ g:\mathbb{N}^k\rightarrow 2^D$ *is a definable function.*

Proof. Suppose that the statement ϕ in $k+q$ free variables defines the function g. That is, $\phi\left(a_1,...,a_k,b_1,...,b_q\right)$ is true if and only if $\left(b_1,...,b_q\right)\in g\left(a_1,...,a_k\right)$. First, we construct a statement, φ in $k+q|B|$ free variables, such that,

$$\varphi\left(a,b_1,...,b_{|B|}\right)\text{ is true }\Leftrightarrow\ g\left(a\right)=\left\{b_1,...,b_{|B|}\right\}\tag{28}$$

where b_i is a q dimensional integer vector. (Notice that we do not assume that $b_i\neq b_j$ if $i\neq j$ in the previous equivalence.) To this end, let $x=\left(x^1,...,x^k\right)$, $y_i=\left(y_i^1,...,y_i^q\right)$ for all $i\in\{1,...,|B|\}$, and $z=\left(z^1,...,z^q\right)$. Define φ as follows:

$$\varphi\left(x,y_1,...,y_{|B|}\right)=\left[\wedge_{i=1}^{|B|}\phi\left(x,y_i\right)\right]\wedge\left[\nexists z\ \left(\wedge_{i=1}^{|B|}z\neq y_i\right)\wedge\phi\left(x,y_i\right)\right].$$

This φ obviously satisfies (28). Next, we construct a statement, ψ, in $ql|B||D|$ free variables such that

$$\psi\left(b_1,...,b_{|B|},d_1...,d_{|D|}\right)\text{ is true }\Leftrightarrow\ f\left(\{b_1,...,b_{|B|}\}\right)=\left\{d_1...,d_{|D|}\right\},\tag{29}$$

where b_i is a q dimensional integer vector for all i and d_i is an l dimensional integer vector for all i. Suppose that $B_i \subset B$, $B_i = \left\{b_i^1, ..., b_i^{|B|}\right\}$, and $f(B_i) = D_i = \left\{d_i^1, ..., d_i^{|D|}\right\}$. Consider

$$\begin{aligned}\psi_{B_i}\left(y_1, ..., y_{|B|}, z_1 ..., z_{|D|}\right) &= \left(\vee_{n=1}^{|B|}\left(y_j = b_i^n\right)\right) \wedge_{n=1}^{|B|}\left(\vee_{j=1}^{|B|}\left(y_j = b_i^n\right)\right) \wedge \\ &\quad \left(\vee_{n=1}^{|D|}\left(z_j = d_i^n\right)\right) \wedge_{n=1}^{|D|}\left(\vee_{j=1}^{|D|}\left(z_j = d_i^n\right)\right).\end{aligned}$$

The first part in the first line says that $y_j \in \left\{b_1, ..., b_{|B|}\right\}$ and the second part of the first line requires that for all b_i^n there exists a y_j such that $y_j = b_i^n$. Similarly, the first part in the second line says that $z_j \in \left\{d_1, ..., d_{|D|}\right\}$ and the second part of the second line requires that for all d_i^n there exists a z_j such that $z_j = d_i^n$. Obviously, $\psi_{B_i}\left(b^1, ..., b^{|B|}, d^1, ..., d^{|D|}\right)$ is true if and only if $\left\{b^1, ..., b^{|B|}\right\} = \left\{b_i^1, ..., b_i^{|B|}\right\}$ and $\left\{d^1, ..., d^{|D|}\right\} = \left\{d_i^1, ..., d_i^{|D|}\right\}$. Let $2^B \backslash \{\emptyset\} = \left\{B_1, ..., B_i, ...B_{2^{|B|}-1}\right\}$. We are ready to define ψ:

$$\psi\left(y_1, ..., y_{|B|}, z_1 ..., z_{|D|}\right) = \vee_{i=1}^{2^{|B|}-1} \psi_{B_i}\left(y_1, ..., y_{|B|}, z_1 ..., z_{|D|}\right).$$

This statement ψ obviously satisfies (29).

Finally, we are ready to construct the statement ζ which defines $g \circ f$.

$$\zeta(x, z) = \exists y_1, ..., y_{|B|}, z_1 ..., z_{|D|} \ \left(\varphi\left(x, y_1, ..., y_{|B|}\right) \wedge \psi\left(y_1, ..., y_{|B|}, z_1 ..., z_{|D|}\right) \wedge (z = z_1)\right).$$

■

Lemma 8 *Suppose that $f : \mathbb{N}^k \to 2^{\mathbb{N}}$ is definable, $g : \mathbb{N} \to 2^{\mathbb{N}} \backslash \{\emptyset\}$ is definable and $|g(n)| = 1$ for all n. For each $n \in \mathbb{N}$ define*

$$h(n_1, ..., n_k) = f(g(n_1), n_{-1}).$$

Then the function $h : \mathbb{N}^k \to 2^{\mathbb{N}}$ is definable.

Proof. Suppose that $\theta(x, y)$ defines f, where $x = (x_1, ..., x_k)$ is a vector of variable symbols and y is a variable symbol. Suppose that φ defines $g(z, v)$, where both z and v are variable symbols. Then the following statement, ϑ, obviously defines h:

$$\vartheta(x, y) \equiv \theta(z, x_{-1}, y) \wedge \varphi(z, x_1).$$

■

Lemma 9 *Let $(N_1, ..., N_m) \in \mathbb{N}^m$ $(N_i \geq 1)$ and $\mathcal{N}_i = \{1, ..., N_i\}$. Suppose that $h_{n_{-i}} : \mathbb{N} \to 2^{\mathbb{N}}$ is definable for all $n_{-i} \in \mathcal{N}_{-i}$. Let $\{\emptyset\} \neq \overline{A}_i \subset \mathbb{N}$ be a finite set. Let $g = (g_1, ..., g_{m-1}) : \mathcal{N}_{-i} \to \mathbb{N}^{m-1}$ be an injection. For each $l \in \mathbb{N}$ and $q = (q_1, ..., q_m) \in \mathbb{N}^m$ define $f(l, q)$ as follows*

$$f(l, q_i, q_{-i}) = \begin{cases} h_{n_{-i}}(l) & \textit{if } q_{-i} = g(n_{-i}), \\ \overline{A}_i & \textit{otherwise.} \end{cases}$$

Then the function f is definable.

Proof. Let $\theta_{n_{-i}}(v,y)$ be a statement which defines $h_{n_{-i}}$ and let $\theta_{A_i}(y,z)$ denote a statement which defines the characteristic function of $\overline{A}_i$. The letters y,z and v are variable symbols. Let $x=(x_1,...,x_m)$ a vector of variable symbols. Then the following statement, ϑ, obviously defines f:

$$\begin{aligned}\vartheta(v,x,y) = \quad & \vee_{n_{-i}\in\mathcal{N}_{-i}}\left(\theta_{n_{-i}}(v,y)\wedge(g(n_{-i})=x_{-i})\right) \\ & \vee\left(\neg\left(\vee_{n_{-i}\in\mathcal{N}_{-i}}\left(\theta_{n_{-i}}(v,y)\wedge(g(n_{-i})=x_{-i})\right)\right)\wedge\theta_{A_i}(y,1)\right).\end{aligned}$$

■

Proof of Lemma 3. By Lemma 4, we only have to show that (20) holds for all $i\in\{1,...,m\}$ and any contract profile $\left\{c_j^{n_j}\right\}_{j,\,n_j}$. Suppose by contradiction that there exists an $i\in\{1,...,m\}$, and a contract profile $\left\{c_j^{n_j}\right\}_{j,\,n_j}$ such that $\cap_{\left(A_i^{n_{-i}}\right)_{n_{-i}}} S\left(\left(A_i^{n_{-i}}\right)_{n_{-i}\in\mathcal{N}_{-i}}\right)=\{\emptyset\}$. Then, for all $\left(A_{-i}^{n_{-i}}\right)_{n_{-i}\in\mathcal{N}_{-i}}\in\left(2^{A_{-i}}\right)^{|\mathcal{N}_{-i}|}$ there exists an $\left(A_i^{n_{-i}}\right)_{n_{-i}\in\mathcal{N}_{-i}}\in\left(2^{A_i}\right)^{|\mathcal{N}_{-i}|}$ such that $\left(A_{-i}^{n_{-i}}\right)_{n_{-i}\in\mathcal{N}_{-i}}\notin S\left(\left(A_i^{n_{-i}}\right)_{n_{-i}\in\mathcal{N}_{-i}}\right)$. Let us fix a function $f=\left(f_{n_{-i}}\right)_{n_{-i}\in\mathcal{N}_{-i}}:\left(2^{A_{-i}}\right)^{|\mathcal{N}_{-i}|}\to\left(2^{A_i}\right)^{|\mathcal{N}_{-i}|}$ such that for all $\left(A_{-i}^{n_{-i}}\right)_{n_{-i}\in\mathcal{N}_{-i}}\in\left(2^{A_{-i}}\right)^{|\mathcal{N}_{-i}|}$

$$\left(A_{-i}^{n_{-i}}\right)_{n_{-i}\in\mathcal{N}_{-i}}\notin S\left(f\left(\left(A_{-i}^{n_{-i}}\right)_{n_{-i}\in\mathcal{N}_{-i}}\right)\right).$$

Let $f^{c_{-i}^{n_{-i}}}$ denote the function defined by $c_{-i}^{n_{-i}}$. Define a function with one free variable, f_x, as follows:

$$f_x([c_i],[c_{-i}])=\begin{cases}\left[f_{n'_{-i}}\left(\left(<\left(f^{c_{-i}^{n_{-i}}}\left([<x>^{(x)}],[c_{-i}^{n_{-i}}]\right)\right)>\right)_{n_{-i}\in\mathcal{N}_{-i}}\right)\right] & \text{if } c_{-i}=c_{-i}^{n'_{-i}}, \\ [A_i] & \text{otherwise.}\end{cases} \tag{30}$$

We shall prove that f_x is a definable function in one free variable, see Lemma 10. Let c_x be an arithmetic statement defining f_x and let γ denote its Gödel code. Then

$$f_\gamma([c_i],[c_{-i}])=\begin{cases}\left[f_{n'_{-i}}\left(\left(\widetilde{c}_{-i}^{n_{-i}}\left(c_\gamma,c_{-i}^{n_{-i}}\right)\right)_{n_{-i}\in\mathcal{N}_{-i}}\right)\right] & \text{if } c_{-i}=c_{-i}^{n'_{-i}}, \\ [A_i] & \text{otherwise.}\end{cases}$$

Notice that

$$\left(\widetilde{c}_{-i}^{n_{-i}}\left(c_\gamma,c_{-i}^{n_{-i}}\right)\right)_{n_{-i}\in\mathcal{N}_{-i}}\in S\left(\left(\widetilde{c}_\gamma\left(c_\gamma,c_i^{n_i}\right)\right)_{n_{-i}\in\mathcal{N}_{-i}}\right) \tag{31}$$

by the definition of S. On the other hand,

$$\begin{aligned}\left(\widetilde{c}_\gamma\left(c_\gamma,c_{-i}^{n_{-i}}\right)\right)_{n_{-i}\in\mathcal{N}_{-i}} \quad &= \quad \left(f_{n_{-i}}\left(\left(\widetilde{c}_{-i}^{n'_{-i}}\left(c_\gamma,c_{-i}^{n'_{-i}}\right)\right)_{n'_{-i}\in\mathcal{N}_{-i}}\right)\right)_{n_{-i}\in\mathcal{N}_{-i}} \\ &= \quad f\left(\left(\widetilde{c}_{-i}^{n_{-i}}\left(c_\gamma,c_{-i}^{n_{-i}}\right)\right)_{n_{-i}\in\mathcal{N}_{-i}}\right),\end{aligned}$$

by the definition of c_γ. Therefore,

$$\left(\left(\widetilde{c}_{-i}^{n_{-i}}\left(c_\gamma,c_{-i}^{n_{-i}}\right)\right)_{n_{-i}\in\mathcal{N}_{-i}}\right)\notin S\left(\left(\widetilde{c}_\gamma\left(c_\gamma,c_{-i}^{n_{-i}}\right)\right)_{n_{-i}\in\mathcal{N}_{-i}}\right) \tag{32}$$

by the definition of f. Notice that (31) and (32) contradict to each other, and hence (20) holds. ■

Lemma 10 *The function defined by (30) is a definable function with one free variable.*
Proof. *By Definition 3, we have to prove that $f : N^{m+1} \rightarrow 2^N$ is a definable function if*

$$f\left(l, q_1, ..., q_m\right) = \begin{cases} \left[f_{n'_{-i}}\left(\left(< \left(f^{c_{-i}^{n_{-i}}}\left(\left[< l >^{(l)}\right], \left[c_{-i}^{n_{-i}}\right]\right)\right) >\right)_{n_{-i} \in \mathcal{N}_{-i}}\right)\right] & \text{if } q_{-i} = \left[c_{-i}^{n'_{-i}}\right], \\ \left[A_i\right] & \text{otherwise.} \end{cases}$$

for all $l \in \mathbb{N}$ and $q = (q_1, ..., q_m) \in \mathbb{N}^m$. The function $f^{c_{-i}^{n_{-i}}}\left(\left[< l >^{(l)}\right], q_{-i}\right)$ is definable $\left(\mathbb{N}^m \rightarrow 2^{\mathbb{N}}\right)$ by Lemmas 1 and 8. Define $h_{n'_{-i}}(l)$ for all $l \in \mathbb{N}$ as follows

$$h_{n'_{-i}}(l) = \left[f_{n'_{-i}}\left(\left(< \left(f^{c_{-i}^{n_{-i}}}\left(\left[< l >^{(l)}\right], \left[c_{-i}^{n_{-i}}\right]\right)\right) >\right)_{n_{-i} \in \mathcal{N}_{-i}}\right)\right].$$

Then the function $h_{n'_{-i}}$ is definable for all $n'_{-i} \in \mathcal{N}_{-i}$ by Lemma 7. Finally, define $g(n_{-i}) = \left[c_{-i}^{n_{-i}}\right]$ for all $n_{-i} \in \mathcal{N}_{-i}$ and apply Lemma 9 to conclude that f is definable. ■

References

A.T. Kalai, E. Kalai, E. L., and D. Samet (2010): "A Commitment Folk Theorem," *Games and Economic Behavior*, 69(1), 127–137.

Bagwell, K., and R. W. Staiger (2001): "Reciprocity, non-discrimination and preferential agreements in the multilateral trading system," *European Journal of Political Economy*, 17(2), 281–325.

Epstein, L., and M. Peters (1999): "A Revelation Principle for Competing Mechanisms," *Journal of Economic Theory*, 88(1), 119–161.

Fershtman, C., and K. Judd (1987): "Equilibrium Incentives in Oligopoly," *American Economic Review*, 77, 927–940.

Fortnow, L. (2009): "Program Equilibria and Discounted Computation Time," Working paper, University of Chicago.

Han, S. (2006): "Menu theorems for bilateral contracting," *Journal of Economic Theory*, 127(1), 157–178, available at http://ideas.repec.org/a/eee/jetheo/v131y2006i1p157-178.html.

Katz, M. (2006): "Observable Contracts as Commitments: Interdependent Contracts and Moral Hazard," *Journal of Economics and Management Strategy*, 15(3), 685–706.

Martimort, D., and H. Moreira (2007): "Common Agency and Public Good Provision under Asymmetric Information," Forthcoming theoretical economics, University of Toulouse.

Martimort, D., and L. Stole (1998): "Communications Spaces, Equilibria Sets and the Revelation Principle under Common Agency," University of Chicago unublished manuscript.

Myerson, R. (1983): "Mechanism Design by an Informed Principal," *Econometrica*, 51, 1767–1797.

Peters, M. (2001): "Common Agency and the Revelation Principle," *Econometrica*, 69(5), 1349–1372.

Peters, M., and C. Troncoso-Valverde (2009): "A Folk Theorem for Competing Mechanisms," manuscript, University of British Columbia.

Salop, S. (1986): *Practises that Credibly Facilitate Ologopoly Coordination*Cambridge, MIT Press.

Tennenholtz, M. (2004): "Program Equilibrium," *Games and Economic Behavior*, 49(2), 363–373.

Yamashita, T. (2010): "Mechanism Games with Multiple Principals and Three or More Agents," *Econometrica*, 78(2), 791–801.