DEFINABLE AND CONTRACTIBLE CONTRACTS

MICHAEL PETERS AND BALÁZS SZENTES

Abstract. This paper analyzes Bayesian normal form games in which players write contracts that condition their actions on the contracts of the 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 equilibrium 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 their commitments on other players' contracts.

The idea that contracts might condition on other contracts is not new in economics. The best known expression of this idea is well known in the industrial organization literature (e.g. [11]) 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* ([2]). This takes the form of trade agreements like GATT that require countries to match tariff cuts by other countries. 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]

All of these approaches are used to support cooperative outcomes in static games. We extend this approach to games with incomplete information. We provide a full characterization of allocations supportable as contract equilibrium. In particular, we show the limits of the 'contracts on contracts' approach by providing allocations supportable by a mechanism designer which cannot be supported as equilibrium with contractible contracts. We also use our Theorem to provide something that looks like a folk theorem for a restricted environment.

The difficulty with extending the older literature is that the conceptual and technical tools developed there can only be used in any but the simplest problems. The meet the competition

Version - January 22, 2009.

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

argument, for example, is extremely stylized. The Stackleberg leader, call it firm A, offers to sell at a very high price provided its competitor, firm B, also offers that high price in the second round. If B in the second round offers any price below the highest price, A commits itself to sell at marginal cost. If B believes this commitment, then one best reply is to set the highest price.

If the firms move simultaneously, then the logic of the argument becomes clouded. A could certainly write a contract that commits it to a high price if B sets the same high price. However suppose that B's strategy is simply to set this high price and that for some reason this is a best reply to A's contract. Then A should deviate and simply undercut firm B. To support the high price outcome, firm B would have to offer a contract similar to A's in order to prevent A's deviation. A naive argument would suggest that B should simply offer the same contract as A, a high price if A sets a high price, and marginal cost otherwise. Casually, two outcomes seem consistent with these contracts - both firms price at marginal cost or both firms set the high price. This seems to violate a fairly fundamental property of game theory which is that for each pair of actions (contracts in this case), there is a unique payoff to every player.[2] More to the point, A's contract doesn't actually say what A would do if B offers a contract that promises to set a high price unless A sets a lower price, etc. The specification of the problem itself seems to be ambiguous about payoffs.

Generally contracts that react to *actions* of other players simply don't make sense. They may not lead to unambiguous outcomes as in the example above. More generally, it is possible that such contracts are simply contradictory. For example, two firms might write contract that commit both of them to set a price that is strictly lower than the other firms price (or two economists demand contracts that guarantee that they will both earn more money than anyone else in the department). To resolve ambiguities and contradictions in such contract, an outside mediator is needed to choose an outcome. This defeats the purpose of using contracts to decentralize the underlying allocation problem.

The reciprocal tax agreement problem is better behaved, and provides the basis for the argument we extend below. State A wants to exempt residents of state B from state taxes provided B exempts residents of state A from taxes. To write the law A exempts residents from any state that has a 'reciprocal' agreement with state A. The question is what exactly is a 'reciprocal' agreement. It is clear enough what the intention is - create a situation in which both states take the mutually beneficial action of exempting one another in a way that eliminates any incentive for either of them to deviate. As mentioned above, it isn't enough to assume that state B unconditionally exempts residents of state A from tax because A would not longer have any incentive to exempt state B. State B has to have a law like the law in state A, in other words, a reciprocal agreement.

[2]One paper that allows multiple payoffs to be associated with each array of actions is [12] who use this approach to support equilibrium when it might not otherwise exist.

It seems that to resolve this kind of problem one needs to define the term 'reciprocal contract' as follows:

$$\text{reciprocal contract} \equiv \begin{cases} \text{exempt if the other state offers a reciprocal contract,} \\ \text{don't otherwise} \end{cases}$$

This kind of definition is familiar from the Bellman equation in dynamic programming where the value function is defined in a self referential way. It is tempting to model this in the following naive way: start by defining a collection of contracts that seem economically sensible. For example, it is reasonable that a state could write a contract that simply fixes any tax rate independent of what the other states do. Let $\overline{C}$ be the set of contracts that simply fix some unconditional tax rate. Append to this set of feasible contracts the reciprocal contract, call it r, defined above. Now model the set of feasible contracts as $\overline{C} \cup \{r\}$. The reciprocal contract above is just r, while 'otherwise' means any contract with a fixed tax rate. Define a normal form game in which the strategies are $\overline{C} \cup \{r\}$ and declare the outcome if both states offer r to be $(\text{exempt}, \text{exempt})$. Then there is an equilibrium in which the states mutually exempt (assuming they jointly want to).

We would argue that this is unsatisfactory for a number of reasons. First, it is undesirable to restrict the set of feasible contracts in order to support the outcome you are looking for. The approach described above amounts to little more than saying that r is the only feasible contract, then claiming it is an equilibrium for both states to offer r. A more satisfactory approach is to define a set of actions that seem economically meaningful, then to allow the broadest set of contracts possible. In the same manner that the value function emerges endogenously from the economic environment, the reciprocal contract should be derived from economic fundamentals.

Second, the approach described above misses the essence of reciprocity which is the infinite regress involved in self referential objects. A contract that makes formal sense is the following:

$$\text{C} = \begin{cases} \text{exempt if other State exempts any State who exempts any State who exempts...} \\ \text{don't otherwise} \end{cases}$$

where the statement in the top line is repeated ad infinitum. Arguably, the contract C is a reciprocal contract since it would exempt any State offering a reciprocal contract. Yet it simply isn't feasible under the naive description given above.

Finally, the ad hoc approach described above simply isn't flexible enough to handle complex environments and incomplete information. For example, making the game asymmetric requires ad hoc extension of the approach above. If State A is supposed to exempt, while state B is supposed to take some other action, say 'partly exempt', then to support the right outcome, the contracts should look something like the following:

$$\text{reciprocal contract}_A \equiv \begin{cases} \text{exempt if other State offers reciprocal contract}_B \\ \text{don't exempt otherwise} \end{cases}$$

and

$$\text{reciprocal contract}_B \equiv \begin{cases} \text{partially exempt if other State offers reciprocal contract}_A \\ \text{don't exempt otherwise} \end{cases}$$

Now the contracts are not directly self referential, as is the Bellman equation, instead they are cross referential. A single self referential or reciprocal contract simply doesn't go far enough. Furthermore, the contracts above define only a single cooperative action, and use a blanket punishment for deviations. Desirable or interesting equilibrium allocations may not look like this. For example, in a general Bayesian game, the most desirable cooperative action for both players might depend on information that only one of them has. So the action that State A wants to take might depend on the contract that B offers. Alternatively, the most effective punishment for A to impose on B might depend on actions that other states are taking. As the number of possibilities increases, so does the number of special words we need to add to our contracting language to support the outcomes we want.

Our approach avoids these problems. We fix a language and require contracts to be written in this language. We then show this language already contains all the special terms like 'reciprocal contract' that we need, even in very rich economic environments where simple notions like 'cooperation' do not adequately describe the allocations we are interested in. The contracting language that we describe is universal in this sense.

It is universal in a second way. Allowing contracts to specify actions that depend on other contracts means that actions might depend on whether other players' contracts depend on the way you make your action depend on their contracts, the way you make your action depend on how their contracts depend on the way you make your contract depend on their contracts, and so on. In simple prisoner's dilemma problems like the tax problem discussed above, this problem is relatively straightforward since 'dependence' simply means whether or not the other player cooperates. However, in richer environments, 'dependence' is more subtle since there are many different ways that players can condition their actions at each round in the hierarchy of dependencies described above. The method we describe below provides a compact way of dealing with this.

Finally, the Bellman equation style representation of a reciprocal contract illustrates that the notion of reciprocity depends on the contracting environment because the word 'cooperate' appears in the definition of a reciprocal contract. It isn't obvious how to extend the argument to problems where a single 'cooperative' action doesn't exist. The set of contracts that we use, on the other hand, is independent of the underlying game that is being played. Contracts need to map into feasible actions, but the way that these actions depend on other contracts doesn't depend on what these actions are. Nor does it depend on whether or not players have private information. In this sense, our contracts are *universal* in the sense that they can be applied to any strategic situation.

1.0.1. *How Definability Works.* Return again to the main purpose of definability. Instead of creating special terms like "reciprocal contract" in an ad hoc way to support cooperative outcomes in special situations, we want to provide a contracting environment in which we can show that the special terms we need to write the contracts that players need to enforce their collusive agreement will always exist within the language. We do it here to illustrate the method for the very simple case, then generalize the approach in the sections below.

Suppose there are m players in a normal form game in which each player has a countable number of actions. Endow players with a formal language containing a countable number of words or characters that they can use to write contracts. Feasible contracts are finite sequences of characters in this formal language. As we mentioned above, the set finite subsets of a countable set is countable, so there are bijections mapping each finite text into $\mathbb{N}$. One such a mapping is called the *Godel Coding*. Provided the language includes all the natural numbers and the usual arithmetic operations, it is possible for players to write contracts that are *definable* functions from $\mathbb{N}^{N-1}$ into that player's action space. Since definable functions can be written as finite sequences of characters in the language, they have Godel codes associated with them. Hence we could interpret the definable functions as contracts that make the players action depend on the Godel code of the other player's contract.

To make the argument easier to relate to conventional contract theory, we assume below that the contract space for each player is the set of definable functions from $\mathbb{N}^{N-1}$ into the subsets of the player's action spaces**.** Every definable function can be associated with a unique integer, and conversely if the integer n is associated with a definable function, then it is associated with a unique text. Now for each array of functions chosen by the players, compute the Godel Code of each such function. Fit the codes of the other players' strategies into each player's strategy to determine a unique subset of actions for every player. Then, players simultaneously take actions from these subsets.

Our objective is to try to characterize the set of equilibria of this game. To see how it works, we might as well restrict attention to a two player prisoner's dilemma. As we illustrated above, we don't really need our formalism to understand this game. However, it provides a simple illustration of the methods we use in the general case. Call the players 1 and 2, and the actions C and D with the usual payoff structure in which D is a dominant strategy and both players are strictly better off if they both play C than they are if they both play D. A strategy c for a player is a definable function from $\mathbb{N}$ to $\{C, D\}$. One obvious equilibrium of this game occurs when both players use a strategy that chooses action D no matter what the Godel code of the other player's strategy.

Every definable function has a Godel code. Let $[c]$ denote the Godel code of the strategy c and refer to $[c]$ as the 'encoding' of c. Since the Godel coding is an injection from the set of definable strategies to the set of integers. For any pair of strategies c_1 and c_2, the action (C or D) taken by player 1 is $c_1([c_2])$ and similarly for player 2. Since every pair of actions determines a payoff, this procedure associates a unique payoff with every pair of strategies.

There are many things that aren't definable strategies that also have Godel codes. We want to make use of some of these other things. In particular, we want to use definable strategies with *free variables.* For example, there is a subclass of definable strategies for player 1 defined parametrically by

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

This is simply a definable strategy with a *free variable* x, where x is the target code of the other player's strategy that will trigger the cooperative action. Definable strategies with free variables are also definable, and so they too have Godel codes. The strategy with free variable that we want is a slight modification of the one above, in particular

$$c_x(n) = \begin{cases} C & n = \left[\langle x \rangle^{(x)}\right], \\ D & \text{otherwise.} \end{cases} \tag{1.1}$$

The mapping $< x >^{(x)}$is the composition of two functions. First, the function $\langle x \rangle$ is the inverse operation to the Godel coding. That is, $< n >$ is the text whose Godel 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 Godel code of a definable strategy with one free variable, then $< n >^{(n)}$ is itself a definable strategy (without a free variable). $\left[\langle n \rangle^{(n)}\right]$ is just the Godel code of whatever this definable strategy happens to be. Notice that in this case, $[\langle x \rangle (x)]$ won't be equal to x since a definable strategy must have a different Godel code from a definable strategy with one free variable because of the fact that the Godel coding is injective.

We want to define a strategy by fixing a value for x in (1.1). In particular, the value of x we are interested in is $[c_x]$. Since $[c_x]$ is the Godel code of a strategy with a free variable, the right hand side of (1.1) 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

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

is a the 'reciprocal' or self-referential contract mentioned above. Now we simply need to verify what happens when both players use strategy $c_{[c_x]}$.

If player 2 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 Godel code, the reaction of player 1, and consequently both players payoffs are well defined. As the Godel coding is injective,

$c' \neq c_{[c_x]}$ implies the Godel code of c' is not equal to $\left[c_{[c_x]}\right]$, and the deviation by 2 induces 1 to respond by switching from C to D.

Notice that this argument makes use of an encoding of the strategy with free variable c_x, which isn't a definable strategy. One might have expected the target code number to be associated with a strategy instead of a strategy with a free variable. For example, it seems that to enforce cooperation there needs to be a definable strategy c^* with encoding $[c^*] = n^*$ such that

$$c^* = \begin{cases} C & [c_2] = n^* \\ D & \text{otherwise} \end{cases}$$

Of course, for arbitrary n^* it will be false that $[c_{n^*}] = n^*$. This leads to a fixed point problem that, in fact, does not have a solution in general. More generally, one could try to construct a self-referential contract by finding a fixed point of the the following problem. For each n, consider

$$c_n\left([c_2]\right) = \begin{cases} C & \text{if } [c_2] = g\left(n\right), \\ D & \text{otherwise,} \end{cases}$$

where g is a definable function. If there exists an n^* such that $[c_{n^*}] = g\left(n^*\right)$, then c_{n^*} is obviously a self-referential contract. Indeed, what we did above is that we chose $g\left(n\right)$ to be $[< n >^{(n)}]$ and showed that $n^* = [c_x]$ is a corresponding fixed point.

To see how the strategy with free variable c_x works, recall the reciprocal tax agreement

$$\text{reciprocal contract} \equiv \begin{cases} \text{exempt} & \text{other State offers reciprocal contract} \\ \text{don't exempt} & \text{otherwise} \end{cases}$$

and its recursive counterpart

$$\text{C} = \begin{cases} \text{exempt if other State exempts any State who exempts any State who exempts...} \\ \text{don't otherwise} \end{cases}$$

The 'reciprocal contract' is $c_{[c_x]}$ and the statement "other state offers reciprocal contract" is $[c_2] = \left[c_{[c_x]}\right]$.

State A wants to exempt any state whose law fulfills a condition. For example, if the condition it is looking for is that the other state simply exempts State S, then it would compute the Godel code $n_0 = [C\forall n]$ then use the strategy

$$c_{n_0} = \begin{cases} C & [c_2] = n_0 \\ D & \text{otherwise} \end{cases}$$

If it does that, then it can't be an equilibrium as explained above. So what it needs to do is to exempt any State whose law fulfills a condition that exempts any state whose law fulfills a condition. For example, if it wanted to exempt State B if and only if State B's law exempts state A if and only if State A unconditionally exempts state B, then it would adopt the strategy $c_{\left[c_{n_0}\right]}$, and so on.

This is where the particular structure of the contract c_x comes into play. Recall that

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

It specifies exemption if and only if a condition is fulfilled, but it doesn't seem to specify what the condition is. However, it does require that whatever the condition x is, if x in turn depends on a condition, then the condition that it depends on must be the same as the condition itself. To see if x depends on a condition, we first decode it and find the statement $\langle x \rangle$ that the integer x corresponds to. Then if it depends on some condition, we require that that condition be x itself, which is the meaning of $\langle x \rangle^{(x)}$. So now we can do the infinite regress. State A adopts a law that exempts state B if and only if the Godel code of State B's law is $\left[c_{[c_x]}\right]$. This means that state B's law must be $c_{[c_x]}$, or that B exempt A if and only if the Godel code of State A's law is $\left[c_{[c_x]}\right]$, i.e., the same condition that A requires.

Every collection of definable contracts uniquely determines a set of commitments for each of the players. Any sensible description of the set of feasible contracts should unambiguously determine players' commitments in this way. We accomplish this by making the contracts arithmetic. The set of definable functions is the largest set of arithmetic functions that can be described using a finite text in a first order language. In this sense, the class of contracts that we describe is universal in that any 'sensible' model of contracts on contracts should involve a contract space that is embedded in the one we describe.

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 in [5] in which actions are delegated to an agent who is given the appropriate incentives to carry out actions that might not otherwise be part of a non-cooperative equilibrium. This idea was developed by [8] who used it to prove a 'folk theorem' for a very specialized environment. The idea that the agent might be used to report deviations, thereby allowing principals to commit themselves to punish a deviator, is developed in [4]. This idea provides the basis for the menu theorems in common agency, like [9], [10] and [6] which illustrate how cooperative outcomes can be supported by having the agent trigger punishments. Recently [14] provides a very general folk theorem for multiple agency games in which principals can commit to follow the recommendations of (potentially interested) agents.

Though this literature shows how contractual devices can be used to support cooperative behavior, it relies on the delegation of decision making power to an agent. In this paper, there is no agent, because contracts depend directly on one another. This approach is closely related to ideas in the computer science literature. One paper that uses this approach is [13]. 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. The result extends the "reciprocal contract" idea presented above to a more general n player game. We have illustrated the basic principle with our 'cross-referential' example above. To support any array of actions, Tennenholz effectively writes out explicitly a sequence of programming statements resembling the verbal statements we made above. These define the keywords that are needed to support any array of actions in which each player's payoff is at least his or her minmax value.

The paper by [7] shows how to extend the set of supportable allocations in two player games. They construct a set of 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 correlated their actions while using independent randomizing devices.

On the most basic level, our paper differs since we are interested in problems with incomplete information. However, the more important difference is that our approach is deductive rather than constructive. We fix a set of commitment devices, then use this same set of devices to support individually rational outcomes in all finite games. The advantage of this is that we are able to give a complete characterization of the set of all supportable allocations. Tennenholz theorem does not rule out, for example, the possibility that there might be program equilibrium in which some players receive less than their minmax payoff. With complete information, this possibility is not critical. For example, [7] rule it out by allowing players to reserve the right to pick their actions in an unconstrained way ex post. They interpret this as giving players the right not to participate in the contracting process. With incomplete information, there is no simple analog to the minmax payoff, so there is no simple trick like non-participation that can be used to show which allocations *cannot* be supported as equilibria. By providing a more complete description of the set of feasible contracts we are able to overcome this difficulty.

It is precisely the ability to pin down allocations that cannot be supported as equilibrium that is critical to our objective of showing the limits to which contracts can be used to decentralize the mechanism designer's problem. We use our characterization to construct examples of allocation rules that can be supported by a mechanism designer, but cannot be supported as contract equilibrium.

We emphasize that the contribution here is not intended to be a contribution to the computer science literature. In fact, we view the paper as a very traditional contracting model in which players have access to a legal system which can be used to provide redress when contracts are not carried out. Yet redress is all we want. Our purpose is to define a contracting language such that players can write any contract that they like in this language. Once all the players have written their contracts, they should be able to deduce on their own what actions they need to take in order to fulfill their contracts. With complete information, it isn't completely surprising

that many allocations can be supported in such an environment. It is in incomplete information environments where the set of supportable allocations has not been characterized, and this is how we view our contribution here.

Finally, the use the Godel coding is simply for convenience. Once one sees that the collection of all finite texts constitutes a countable set, any definable bijection from finite texts to integers can be used to do the analysis we do. Any definable bijection can be explicitly written into the contracts we allow, so that a judge (or player for that matter) who doesn't know what it is can explicitly calculate it.

3. The Language and the Gödel Coding

We consider a formal language, which is sufficiently rich to allow its user to state propositions in arithmetic. Furthermore, the set of statements in this language is closed under the finite applications of the Boolean operations: $\neg$, $\vee$, and $\wedge$. This implies that one can express, for example, the following statement:

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

In addition, one can also express statements in the language that involve 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.

Let $\mathfrak{L}$ be the set of all formulas of the formal language. Each of its element is a finite string of symbols. 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 $\varphi \in \mathfrak{L}$, and call it the Gödel Code of the text φ.

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

Definition 1. The function $f : \mathbb{N}^k \rightarrow 2^{\mathbb{N}}$ is said to be *definable* if there exists a first-order predicate ϕ in $k+1$ free variables such that $b \in f(a_1, ..., a_k)$ if and only if $\phi(a_1, ..., a_k, b)$ is true.

In the definition, the mapping f is a correspondence from $\mathbb{N}^k$ to $\mathbb{N}$. Of course, if $f(n)$ is a singleton for all $n \in \mathbb{N}^k$, then f is a function. If the function f is definable by the predicate ϕ then we refer to $[\phi]$ as the Godel encoding of f**.** 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.

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

Notice that ϕ indeed has two free variables. (The variable z is not free because there is a quantifier 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. Notice that $f(a)=0$ if and only if $\phi(a,0)$ is true. To see this, first notice that $\phi(a,b)$ is false whenever $b \notin \{0,1\}$. (This is because the first part of ϕ requires b to be zero or one.) If $b=0$ then $\phi(a,0)$ is indeed true. If $b=1$, then the second part of ϕ becomes false because $a+b$ is an odd number.

4. Complete information Contracting Game

Suppose there are m players. Player i has a finite action space A_i. Let A denote $\times_{i=1}^{m} A_i$. The payoff of Player i is $u_i(a_1,\ldots,a_m)$. We use the conventional notation that $u_i(a_i,a_{-i})$ is the payoff to player i if he takes action a_i while the other players take action a_{-i}. Each player simultaneously submits a *contract,* which is a definable correspondence from $\mathbb{N}^m$ to $2^{\mathbb{N}}$, where 'definable' is to be understood in the sense of Definition 1. At stage two, players take actions simultaneously from subsets of their actions spaces. These subsets are determined by the first-stage contracts. If at stage one player j submitted contract c_j $(j=1,...,m)$, then player i can only take action a_i at stage two if $[a_i] \in c_i([c_1],...,[c_m])$. We restrict attention to pure-strategy subgame perfect equilibria of this game.

The pure strategy minmax value for player i is

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

Let $\underline{a}_j$ be any one of the actions that j uses to attain his minmax payoff. Let us fix an action $a_i^{j_i}$ for player i, such that,

$$\left(a_1^j,...,a_m^j\right) \in \arg\min_{a_{-i}} u_j\left(\underline{a}_j,a_{-j}\right).$$

That is, a_i^j is the action that player i uses to punish player j. For convenience, define $a_j^j=\underline{a}_j$ for all $j\in\{1,...,m\}$.

Theorem 1. The action profile $a^*=(a_1^*,...,a_m^*)\in A$ is supportable as a pure-strategy SPNE outcome in the contracting game if and only if $u_i(a^*)\geq \underline{u}_i$ for each i.

Before we proceed with the proof of the theorem, we recall two pieces notations from the introduction. First, if $n\in\mathbb{N}$ then $<n>$ denotes the text whose Gödel code is n. That is, $[<n>]=n$. Second, for any text φ, let $\varphi^{(n_1,\ldots,n_k)}$ denote the statement where if the letter x_i stands for a free variable in φ then x_i is evaluated at n_i in φ for $i=1,...,n$. For example, if φ is $x_1<x_2$, $n_1=1$, and $n_2=2$ then $\varphi^{(n_1,n_2)}$ is $1<2$. Consider now the following text in k free variable: $<x_i>^{(x_1,\ldots,x_k)}$, where $i\leq k$. One can evaluate this statement at any k-dimensional vector of integers. Since the Godel coding was a bijection $<n_i>$ is a text for each $n_i\in\mathbb{N}$. In addition, $\varphi^{(n_1,\ldots,n_k)}$ is defined for all φ and $(n_1,...,n_k)$. In addition, it is a well-known result in Mathematical Logic, that if $f(n_1,...,n_k)=\left[<n_i>^{(n_1,\ldots,n_k)}\right]$, then f is a definable function.

Proof. First, we prove the only if part. Fix an equilibrium in the contracting game. Let c_j denote the equilibrium contract of player j ($j = 1, ..., m$) and let u_i denote player i's equilibrium payoff. Notice, that player i can always offer a contract that does not restrict his action space. That is, he can offer $\overline{c} : \mathbb{N}^m \to 2^{\mathbb{N}}$, such that $\overline{c}(n_1, ..., n_m) = \mathbb{N}$ for all $(n_1, ..., n_m) \in \mathbb{N}^m$. The contract $\overline{c}$ is obviously definable. [3] We show that if $u_i < \underline{u}_i$, player i can profitably deviate at the first stage by offering $\overline{c}$ instead of c_i. Let $\widetilde{c}_j = c_j$ if $j \neq i$ and $\widetilde{c}_i = \overline{c}$. Let $\widetilde{A}_j = \{a_j : [a_j] \in \widetilde{c}_j([\widetilde{c}_1], ..., [\widetilde{c}_m])\}$. That is, $\widetilde{A}_j$ is the action space of player j in the subgame generated by the contract profile $(\widetilde{c}_1, ..., \widetilde{c}_m)$. Also notice that $\widetilde{A}_i = A_i$. The payoff of player i in any pure strategy equilibrium of this subgame is weakly larger than

$$\min_{a_{-i} \in \widetilde{A}_{-i}} \max_{a_i \in A_i} u_i(a_i, a_{-i}) \geq \min_{a_{-i} \in A_{-i}} \max_{a_i \in A_i} u_i(a_i, a_{-i}).$$

The inequality follows from $\widetilde{A}_j \subseteq A_j$ for all j. Therefore, player i can always achieve his pure minmax value by offering the contract $\overline{c}$.

For the if part, consider the following contract of Player i, $c^i_{x_i, x_{-i}}$, in m free variables:

$$c^i_{x_1,...,x_m}\left(([c_j])^m_{j=1}\right) =$$

$$\text{(4.1)} \qquad \begin{cases} [a^*_i] & \text{if } |\left\{k : \left[< x_k >^{(x_1,...,x_m)}\right] \neq [c_k]\right\}| \neq 1, \\ \left[a^j_i\right] & \text{if } \left\{k : \left[< x_k >^{(x_1,...,x_n)}\right] \neq [c_k]\right\} = \{j\} \end{cases}$$

The expression (4.1) is not a contract, but rather a contract with free variables. Each such expression has a Godel code, so let $\gamma_i = \left[c^i_{x_1,...,x_m}\right]$. The functions $\left\{c^i_{\gamma_1,...,\gamma_m}\right\}_i$ have no free variables, so they constitute a set of contracts. We will now show that $\left\{c^i_{\gamma_1,...,\gamma_m}\right\}^m_{i=1}$ constitutes an equilibrium profile of contracts which support the outcome $\left\{a^1_{k_1}, \ldots, a^m_{k_m}\right\}$. First observe what happens when all players use contract $c^i_{\gamma_1,...,\gamma_m}$. Notice that

$$c^i_{\gamma_1,...,\gamma_m}\left(([c_j])^m_{j=1}\right) = \begin{cases} [a^*_i] & \text{if } |\left\{k : \left[< \gamma_k >^{(\gamma_1,...,\gamma_m)}\right] \neq [c_k]\ \right\}| \neq 1, \\ \left[a^j_i\right] & \text{if } \left\{k : \left[< \gamma_k >^{(\gamma_1,...,\gamma_m)}\right] \neq [c_k]\ \right\} = \{j\}. \end{cases}$$

Player i needs to check whether the Godel code of $< \gamma_k >^{(\gamma_1,...,\gamma_m)}$ is equal to the Godel code of player k's contract, c_k. The integer γ_k is the Godel code of the contract with free variable $c^k_{x_1,...,x_m}$. Player i's contract says to take this contract with free variable, fix the free variables at $\gamma_1, ..., \gamma_m$ (which gives the contract $c^k_{\gamma_1,...,\gamma_m}$), then evaluate its Godel code. This is what is to be compared with the Godel code of the contract offered by k. Of course, these are the same in equilibrium because $c_k = c^k_{\gamma_1,...,\gamma_m}$. Since this is the case for all $m-1$ of the other players, player i ends up taking action a^*_i. So these contracts support the outcome we want if everyone uses them.

[3] For example, the predicate

$$\{x_1 = x_1\} \wedge ... \wedge \{x_m = x_m\} \wedge \{y = y\}$$

defines $\overline{c}$. That is, for all $y \in \mathbb{N}$ the predicate is true no matter how the free variables are evaluated.

Player j can deviate to any definable contract mapping $\mathbb{N}^m$ into $2^{\mathbb{N}}$. However, any such contract will have a different Godel code, and so will induce the punishment $\left\{a_i^j\right\}_{i\neq j}$ from the other players. Recall that $\left\{a_i^j\right\}_{i\neq j}$ is the action profile that players other than player j use to minmax player j. Since $u_j(a) \geq \underline{u}_j$ any deviation will be unprofitable. ∎

One might argue that restricting the space of contracts to be definable functions of Godel codes is both arbitrary and unnatural. Indeed, there is no reason for a judge to interpret a contract as a description of a mapping from the Godel codes of the contracts offered by the other players to the actions space of the player. For that matter, the judge might not even know about the Godel coding. It is important to note that the salient feature of definable contracts is that they can be written as texts that use a finite number of words in a formal language. The set of finite texts seems a very natural description of the set of feasible contracts. In fact, from this perspective it seems that *any* reasonable description of the set of feasible contracts should allow any such text.

The complication with such a broad description of the set of contracts is that to properly define a game, one must fully describe the mappings from profiles of texts into payoffs. Many texts will be complete nonsense and some modelling decision has to be taken about how these would translate into actions and payoffs. The contracts that we specify above are definable texts that have two advantages in this regard. First, since every finite text has a Godel code, they tie down the action of the player who offers such a contract even if the other players in the game offer contracts involving texts that make no economic sense. Furthermore, if all players offer contracts from the set we specify, an outcome for every player is uniquely determined.

Finally, since the Godel coding itself is definable, the coding can be embedded directly into the contract. So 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.

Generalizations.— Everything about this theorem involves pure strategies. This imposes limits on its application. Next, we discuss how to extend our result to the case when players can mix over their restricted action space at the second stage of the game but cannot randomize over the contracts they offer at the first stage. Allowing such mixing expands the set of payoff profiles that can be supported by equilibria for two reasons. First, since players can randomize certain convex combinations of payoff profiles can now be supported. Second, players can use mixing when punishing a deviator, and hence the minmax value of the players will be smaller.

Formally, for all $S = \times_i S_i$, $S_i \subset A_i$, define a game, G_S, where the action space of player i is S_i, and the payoff function of player i is the restriction of u_i on S. Let $E(S)$ denote the set of mixed equilibria in G_S. Define the minmax value of player i, u_i^*, as

$$u_i^* = \min_{\substack{S_{-i}\subset A_{-i} \\ S_{-i}=\times_{j\neq i}S_j}} \max_{S_i\subset A_i} \min_{\sigma\in E(S_{-i}\times A)} \int u_i(a)\, d\sigma(a).$$

The idea is that in the contracting game, players can restrict their action spaces arbitrarily, hence, when they punish player i they can choose S_{-i} arbitrarily. On the other hand, their second-stage actions must be best responses, and that is why we have to consider equilibrium payoffs in the restricted game. An argument identical to the proof of Theorem 1 shows that the random allocation $\sigma \in \Delta(A)$ can be supported as an equilibrium if

(i) $\exists S_i \subset A_i$ for all i, such that $\sigma \in E(\times_i S_i)$, and

(ii) $\int u_i(a)\, d\sigma(a) \geq u_i^*$ for all i.

What happens if players are allowed to randomize over the contracts they offer? It is possible to show that part (i) can be completely relaxed. That is, the distribution over the outcomes does not have to be an equilibrium in G_S, and it does not even have to be generated by independent randomizations of the players. The construction of mixed equilibria in our contracting game that supports correlated outcomes is entirely based on Kalai et.al. (2008). The authors consider two-person games where players submit commitment devices instead of taking actions. A device then determines the action of the player as function of the other device. The authors construct a set of devices such that any individually rational correlated outcome can be implemented as a mixed equilibrium in the game. That is, although the players mix independently over their devices, the distribution over the actions profiles will be correlated. It is easy to show that these results extend to n-person complete information games, and in addition, the the equilibrium commitment devices constructed by Kalai et.al. (2008) are definable functions as long as the probabilities involved in each mixing are all rational numbers.

Theorem 2. Suppose that $\sigma \in \Delta(A)$, and $\sigma(a) \in \mathbb{Q}$ for all $a \in A$. The distribution σ can be supported as a mixed-strategy equilibrium outcome in the contracting game if and only if $\int u_i(a)\, d\sigma(a) \geq u_i^*$ for all $i \in \{1, ..., m\}$.

Another question is why we use definable functions as opposed to programs or Turing machines. One might want to require that the contracts must be computable and assume that the set of available contracts is the set (or a subset) of Turing machines. In such a model, if player i ($i = 1, 2$) chooses machine τ_i, then τ_i runs on the description of τ_j, and the output will be a subset of the action space of player i. It is well-known, that one can construct self- and cross-referential contracts (machines) in this space too.[4] In fact, this construction is essentially identical to our construction of cross-referential definable functions. Most importantly, the equilibrium contracts we construct to support individually rational allocations are, in fact, recursive functions, and hence they are computable by Turing machines. Therefore, if the reader insists on computability, he can restrict attention to the space of Turing machines.

There are, however, several advantages of our approach over modelling contracts with Turing machines. First, Turing machines do not always halt, and therefore, it is not clear how one can

[4]Such machines were constructed even in the context of Game Theory, see Anderlini 1990 and Canning 1992.

define the restriction on the action space of a player, if his machine does not halt. A way to handle the halting problem is to restrict the space of Turing machines to be the set of machines that always halts. We find such restrictions arbitrary. Instead of restricting the space of recursive functions, we expanded it to be the set of definable functions and avoided the halting problem that way. Second, another problem with Turing machines is that they can only condition on the actual description of the machines submitted by the other players but cannot condition on the functions what the machines compute. Take the example of the prisoner dilemma. It is possible to construct a Turing machine, τ, such that

$$\tau\left([\tau_2]\right) = \begin{cases} C & \text{if } [\tau_2] = [\tau] \\ D & \text{otherwise.} \end{cases}$$

The problem is that if player 2 submits a machine, say τ', which is computationally equivalent with τ, but has a different description, then player 1 would defect. In fact, it is not possible to construct a machine which does not suffer from this problem. We avoid such problems with definable functions. Indeed, it is possible to express contracts that do not condition on the actual way the other contract is written, but on the function itself that the other contract describes. Consider

$$c_1\left([c_2]\right) = \begin{cases} C & \text{if } c_2^* \Leftrightarrow c_2, \\ D & \text{otherwise.} \end{cases}$$

The contract c_γ is obviously definable, but does not condition on the actual form of c_2. As long as c_2 represents the same function as c_2^*, cooperation is prescribed.

5. Contracting in a Bayesian Environment

In the previous section, we showed how contractible contracts can be used to support any allocation for which every player's payoff is at least his minmax value. Assuming non-participation is always an option, this is the set of allocations that is supportable by a centralized mechanism designer. In this sense, contractible contracts completely decentralize the allocation problem. In this section, we show that the same result is not true in the Bayesian case. We do this by proving a theorem that completely characterizes the set of allocation rules that can be supported as Bayesian equilibrium in the contracting game. We then construct allocations that can be supported by a centralized mechanism designer, but which cannot be supported as equilibria.

We also show, however, contractible contracts make it possible for one player's action in a contract equilibrium to depend on another player's type. The reason is that contracts explicitly condition actions on other player's contracts, which, along the equilibrium path, can depend on their types. We exploit the reciprocal contracting idea described above to enforce type contingent agreements. The idea is that a contract will specify a number of 'target' Godel codes, one for each of the contracts the other player's different types are supposed to offer along the equilibrium path.

As long as other players offer a contract whose code is equal to one of these targets, the contract responds with a 'cooperative' action. If any of the others deviate, the contract will respond with some kind of punishment.

When information is complete, a punishment is simply an array of commitments that non-deviators make. These commitments are such that they make all deviations unprofitable. We want to extend this idea to the Bayesian case. There are a number of difficulties associated with this. First, contract offers depend on player types, so they reveal information. Potential 'deviators' can condition their play on non-deviators' contracts, so they can condition their commitments on the non-deviators' type. Players may not want to reveal their type information at the contracting stage for this reason. Nonetheless, they will want act on this type information ex post. So contracts will typically bind players to subsets of their actions, leaving some discretion for them to vary their actions with their types ex post. So it is important to think of contract offers as commitment correspondences, rather than simple commitments to actions as would suffice with complete information.

Secondly, non-deviators have the ability to respond to deviations contractually. They can make their punishments more severe by exploiting residual uncertainty that the deviator has about their type ex post. They would do this, again, by committing to a subset of actions instead of a single action. The punishment for a deviation consists of two parts for this reason: a punishment correspondence, that restricts the non-deviator's ex post choice to a subset of his actions, and a second stage strategy that depends on the non-deviator's type (constrained by the information the non-deviator reveals about his type with his on path contract).

Perhaps the most complex part of contracting equilibrium is that non-deviators' responses depend on the deviator's contract. As such they could, in principle, depend on the deviation in sophisticated way. The simple logic that we developed for the reciprocal contracting argument above relied on the idea that non-deviators could *punish* deviators by minmaxing them. The minmax action doesn't depend on exactly how the non-cooperative player chooses to be non-cooperative. With incomplete information, there is no natural analog for the minmax punishment. The ability to contract on other contracts suggests the possibility that non-deviators could react to deviations in a way that depends on exactly what the deviation is thereby holding on path payoffs to something below the corresponding min max payoff.

A remarkable property of our theorem is to show the sense in which punishments can be understood to be invariant to the manner in which a player chooses be non-cooperative. We show that any contract equilibrium can be supported by having non-deviators respond to a deviation with a contractual commitment that is independent of what the deviator chooses to do. Ex post choices that non-deviators make from their contractual commitment correspondences will typically depend on the deviation. However, these ex post choices aren't part of the non-deviator's contracts. This result allows us to extend the reciprocal contracting idea to Bayesian games. If a player offers a contract that is consistent with equilibrium play by one of his types, then the other

players respond cooperatively. Otherwise, there is a single punishment correspondence that the non-deviators impose, just as in the complete information case.

This property of contract equilibrium is a consequence of restricting players to definable contracts. It is the part of our theorem that allows us to show the limits of contractible contracts, since we can show the kinds of allocations that can't be supported as equilibrium. This is the part of our theorem that we use to provide examples of allocations supportable by a mechanism designer but not by contractable contracts. This is the advantage we derive from specifying the contract space very precisely. An abstract commitment space such as the one provided in [7], or a space of contracts that is constructed to have desirable properties, such as programs in [13], can be used to show that a large set of allocations can be supported as equilibrium allocations. However, in the Bayesian case, a complete characterization requires a demonstration that certain allocations cannot be supported. To provide this, a complete description of the set of feasible contracts is required. Our Lemma illustrates that definability provides just such a complete description.

The model is the same as in the previous section, with the addition of player types. There are m players. Player i's actions space is a finite set denoted by A^i. Each player i has a type t_i drawn from a finite set T^i. The joint distribution types is common knowledge. The payoff of player i is $u_i(a_i, a_{-i}, t)$ where $t \in T_1 \times \cdots \times T_m$. Notice that a strategy rule for player i in the Bayesian game the players might otherwise be involved in is an element of $A_i^{|T_i|}$.

Our characterization of the set of allocations that can be supported as contract equilibrium hinges on the information that equilibrium play reveals about players' types. Our argument refers repeatedly to the information that is revealed through equilibrium play. Most things that happen off the equilibrium path depend on this information. A natural way to incorporate this is to use the *information partition* induced by these contracts. Fix an equilibrium, and define the correspondence $\tau_i : T_i \to 2^{T_i}$ to mean the set of types of player i who offer the same contract as type t_i. Once other players see the contract offered by player i of type t_i, they should commonly believe that i's type lies in the set $\tau_i(t_i)$. The correspondence τ_i is an *information partition.* Similarly, the correspondence

$$\tau_{-i}(t_{-i}) = \prod_{j \neq i} \tau_j(t_j)$$

describes the information available to player i about the types of the other players.

Each contract specifies a set of actions from which players subsequently choose. In this sense, equilibrium contracts support a commitment *correspondence* for each player. As contracts depend on other contracts, which in turn depend on other players' types, this commitment correspondence can be written as a mapping $r_i : T \to 2^{A_i}$. Since the set from which i chooses his action can only depend on some other player's type to the extent that the other player's contract varies with his type, r_i should be measurable with respect to the information partition τ_{-i}.

Contracts specify sets of feasible actions. Ultimately, payoffs are determined by players' choices from these sets in the final stage of the game. Let $s_i : T \to A_i$ denote the outcome function

associated with the second stage strategies.[5] These outcome functions must have the property that $s_i(t)$ lies in the set $r_i(t)$ for each t. It might seem strange that this outcome function should depend on t instead of t_i. The reason that player i's equilibrium actions depend on the types of other players is twofold. First, player i gets to see the contracts offered by each of the other players. His beliefs vary as the other players' contracts vary, so his actions in the second stage will vary with the other players types. Secondly, his own commitments depend on the contracts, and thus the types of the other players. Evidently, player i only observes types imperfectly by observing the contracts that are offered. This is captured simply by observing that this induced outcome function must be measurable with respect to the information partition τ_{-i}.

Each off equilibrium contract offered by a deviator specifies a commitment for each array of contracts offered by the other players. Since the contracts the other players offer depend on their types, a deviation implies a commitment correspondence $f_i : T_{-i} \to 2^{A_i}$. Since these types are revealed only through the contracts that the others offer, this correspondence should be measurable with respect to the information partition τ_{-i} that captures this information. Let F_i be the set of all commitment correspondences available to the deviator, i.e., F_i is the set of all τ_{-i} measurable mappings from T_{-i} into 2^{A_i}.

In a contract equilibrium, a deviation leads to two sorts of 'punishments'. First, since the other players' contracts specifically condition on the contract offered by the deviator, the non-deviators will change their commitments. As mentioned above, we are going to show that the punishment correspondence associated with this change in commitments can be taken to be independent of the deviation f_i. However it is possible that the way that the non-deviators choose from sets to which they have committed themselves will depend on f_i.

Write the 'punishment' that player j imposes when player i deviates as $p_j^i : T_{-i} \to 2^{A_i}$ and $p^i = \prod_{j \neq i} p_j^i$. In a contract equilibrium, this punishment is the consequence of the contract that j has written, so the punishment can only vary with j's type to the extent that j's contract does. As a consequence, this punishment will be measurable with respect to the information partition τ_{-i}.

Finally, we need to describe the non-deviators' behavior in the ex post stage, let $s_j^i : F_i \times T_{-i} \to A_j$. The non-deviator j can no longer condition his behavior on information revealed by i's *equilibrium* behavior. However, he does observe i's commitment correspondence in the sense that he observes the deviator's contract. He also observes the on equilibrium contracts of the others. This is captured, as always, by requiring that his behavior be measurable with respect to τ_{-ij}. The

[5]To simplify the argument slightly, we focus on pure strategy outcomes here. It is completely trivial to extend this argument to outcomes that involve randomization at the second stage by having the outcome functions be mappings from T into $\triangle(A_i)$, restricting the supports of these mappings to lie in $r_i(t)$, then letting $u_i(s(t), t)$ be the expected utility associated with the randomization. With this notation, the inequalities that characterize the equilibrium remain unchanged.

action $s_j^i(f_i, t_{-i})$ should be contained in $p_j^i(t_{-i})$ for every $t_{-i} \in T_{-i}$ and $f_i \in F_i$.[6] Let $s^i = \prod_{j \neq i} s_j^i$ be the outcome function associated with a deviation. for each $i = 1, \ldots m$,

The theorem is based on a pair of inequalities that are based on the objects defined above. The first captures the idea that no player wishes to mimic the equilibrium behavior of another of his own types. For each $i = 1, \ldots m$, and each t_i and t_i'

$$\mathbb{E}_{t_{-i}}\left(u_i\left(s\left(t\right), t\right) : t_i\right)$$

$$\geq \mathbb{E}_{t_{-i}}\left(\max_{a_i \in r_i\left(t_i', t_{-i}\right)} \mathbb{E}_{t'_{-i}}\left(u_i\left(a_i, s_{-i}\left(t_i', t'_{-i}\right), \left(t_i, t'_{-i}\right)\right) : t'_{-i} \in \tau_{-i}\left(t_{-i}\right)\right) : t_i\right). \tag{5.1}$$

The commitment correspondence r_i results in a collection of actions from which player i is committed to choose in the second stage. This choice set can depend on the types of the others because their contracts do. The max operator on the right hand side requires the player to choose a best reply from this set given posterior beliefs. Taken together, these constraints for all the players requires that play in the second stage constitutes a Bayesian equilibrium of the game in which each player chooses an action from the set of actions to which he is committed, given posterior beliefs about players' types.

To deal with deviations at the contracting stage of the game, we require that for each $t_i \in T$

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

$$\max_{f_i \in F_i} E_{t_{-i}}\left(\max_{a \in f_i(t_{-i})} E_{t'_{-i}}\left(u_i\left(a, s^i\left(f_i, t'_{-i}\right), \left(t_i, t'_{-i}\right), t\right) : t'_{-i} \in \tau_{-i}\left(t_{-i}\right)\right) : t_i\right). \tag{5.2}$$

A deviation implies a commitment correspondence f_i. The inequality says that even if the deviator chooses a best reply from the set of actions to which he is committed, he cannot gain by deviating.

So far we have imposed no restrictions on second stage punishment behavior. In applications it is natural to want behavior in the second stage to be consistent with some refinement like *perfect* Bayesian equilibrium. Our theorem works with most standard refinements but does not depend on them. To illustrate, let $\tilde{A}_1, \ldots, \tilde{A}_m$ be a collection of subsets of the players action sets and let $\tilde{T}_{-i}$ be a subset of T_{-i}. The interpretation of these objects is that some player i has deviated from an equilibrium. Contracts constrain the players to choose from the sets $\tilde{A}_i$ in the second stage, while it is common belief that the non-deviators' types lie in $\tilde{T}_{-i}$. Let $\mathcal{R}_i\left[\tilde{A}_1, \ldots, \tilde{A}_m, \tilde{T}_{-i}\right]$ be the subset of actions in $\tilde{A}_{-i}$ that are consistent with some refinement. For example in a (unrefined) Bayesian equilibrium, $\mathcal{R}_i\left[\tilde{A}_1, \ldots, \tilde{A}_m, \tilde{T}_{-i}\right] = \tilde{A}_{-i}$. A slightly stronger refinement might have $\mathcal{R}_i\left[\tilde{A}_1, \ldots, \tilde{A}_m, \tilde{T}_{-i}\right]$ ruling out actions that are strictly dominated given that players are constrained to choose actions in $\tilde{A}_1, \ldots, \tilde{A}_m$. Finally, consistent with the idea of perfect Bayesian equilibrium, $\mathcal{R}_i$ might contain only actions that are consistent with Bayesian equilibrium in the

[6] If mixing is allowed in the last stage, then the support of $s_j^i(f_i, t_{-j})$ should be contained in $p_j^i(t_{-i})$.

game defined by subsets $\tilde{A}_1, \ldots, \tilde{A}_m$, common belief that the non-deviators' types lie in $\tilde{T}_{-i}$ and *some* belief about the deviator i. We will call a Bayesian equilibrium in contracts an $\mathcal{R}$-equilibrium if following each array of contract offers for which player i's contract is inconsistent with his equilibrium strategy and all other players contract offers are consistent with the equilibrium strategies of players whose types lie in $\tilde{T}_{-i}$, continuation play lies in $\mathcal{R}_i\left[\tilde{A}_1, \ldots, \tilde{A}_m, \tilde{T}_{-i}\right]$ where the $\tilde{A}_j$ are the contractual commitments of each of the players. We can now state our main theorem.

Theorem 3. An allocation rule $s : T \to A$ can be supported as an $\mathcal{R}$-equilibrium in contracts if and only if there is a commitment correspondence r, a collection of information partitions $\{\tau_i\}_{i=1,\ldots,m}$, punishments $\left\{p^i\right\}_{i=1,\ldots,m}$, and outcome functions $\left\{s^i\right\}_{i=1,\ldots,m}$ such that r is measurable with respect to $\tau = \prod \tau_i$, each p^i is measurable with respect to τ_{-i}, the support of $s(t)$ is contained in $r(t)$ for each t, $s^i(f_i, t_{-i}) \in \mathcal{R}_i\left[f_i(t_{-i}), p^i(t_{-i}), \tau_{-i}(t_{-i})\right]$ for each f_i and t_{-i} , and (5.1) and (5.2) are satisfied.

One of the key properties of this theorem is to show that when contracts are definable, equilibrium must support a single punishment correspondence of exactly the kind we have described. Specifically, it is a punishment correspondence that is independent of f. This is central to the reciprocity idea that we developed at the beginning of the paper. Uncooperative behavior by one player provokes a punishing contractual response from the others that doesn't depend on exactly how the deviator goes about being uncooperative. We want to show that it is the second stage commitments that capture this property of reciprocity.[7] This property of contract equilibrium is also very surprising. It might seem that contract equilibrium could be supported with very complicated contracts that punish deviators in a way that is sensitive to exactly how they deviate. We show that this is not the case.

Finally, the theorem is written assuming that players use pure strategies at every stage. The primary reason we assume away mixed strategies is so that we can use an information partition to represent players knowledge at the second stage instead of a more complex measure of information. We assume pure strategies in the second stage simply for consistency. It is completely trivial to extend the theorem to allow randomization at the second stage. This involves nothing more than assuming that $s_i^i(f_i, t_{-i})$ are mixtures on A_j whose support is contained in $p_j^j(t_{-i})$, then redefining $u_i(s, t)$ to be expected utility associated with the mixture s when $s \in \triangle(A)$. The theorem and proof then proceed verbatim.

6. Proof of Theorem 3

We write the proof in three parts. The first part shows the 'if' part of the theorem. It is a generalization of the reciprocal contracting idea presented above. Before going on to the more difficult 'only if' part, we prove the Lemma that is interesting for its own sake, and which forms the basis of the second part of our proof. Finally, we give the proof of the only if part.

[7] Care here is needed to observe that second stage *behavior* does depend on the deviation in the first stage.

6.1. **If Part:**

Proof. Suppose that $\left\{s,\tau,r,\left\{p^i\right\},\left\{s^i\right\}\right\}$ satisfy (5.1) and (5.2). We construct a Bayesian equilibrium in the contracting game which implements the allocation s. Let x denote $\left(x_j^{t_j}\right)_{j\in\{1,...,m\},\ t_j\in T_j}$, where $x_j^{t_j}$ denotes a free variable. Consider the following contract in $|T|$ free variables:

$$\begin{aligned} & c_x^{t_i}\left([c_1],...,[c_m]\right) \\ = & \begin{cases} r_i\left(t\right) & \text{if } \forall k \exists x_k^{t_k}\in\left\{x_k^{t_k}:t_k\in T_k\right\} \text{ s.t. } \left[<x_k^{t_k}>^{(x)}\right]=[c_k], \\ p_i^j\left(t_{-j}\right) & \text{if } \left\{k:\nexists x_k^{t_k}\in\left\{x_k^{t_k}:t_k\in T_k\right\} \text{ s.t. } \left[<x_k^{t_k}>^{(x)}\right]=[c_k]\right\}=j, \\ A_i & \text{otherwise and if } k+1>k \text{ if } k\in\left\{j:\tau\left(t_j\right)=\tau\left(t_i\right)\right\}, \end{cases} \end{aligned}$$

The last statement is in the third line is always true. Such a statement, however, makes it possible that a player with two different types offers two different but computationally equivalent contracts. Let $\gamma_i^{t_i}$ denote the Godel Code of this contract and let $\gamma=\left(\gamma_i^{t_i}\right)_{i,t^i}$. The equilibrium contract offered by player i with type t_i will be: $c_\gamma^{t_i}$. Then

$$\begin{aligned} & c_\gamma^{t_i}\left([c_1],...,[c_m]\right) \\ = & \begin{cases} r_i\left(t\right) & \text{if } \forall k \exists t_k\in T_k \text{ s.t. } \left[<\gamma_k^{t_k}>^{(\gamma)}\right]=[c_k], \\ p_i^j\left(t_{-j}\right) & \text{if } \left\{k:\nexists t_k\in T_k \text{ s.t. } \left[<\gamma_k^{t_k}>^{(\gamma)}\right]=[c_k]\right\}=j, \\ A_i & \text{otherwise and if } k+1>k \text{ if } k\in H\left(t_i\right), \end{cases} \end{aligned}$$

Notice that $<\gamma_q^{t_q}>^{(\gamma)}=c_\gamma^{t_q}$. Therefore, the previous contract can be rewritten as

$$\begin{aligned} & c_\gamma^{t_i}\left([c_1],...,[c_m]\right) \\ = & \begin{cases} r_i\left(t\right) & \text{if } \forall k \exists t_k\in T_k \text{ s.t. } \left[c_\gamma^{t_k}\right]=[c_k], \\ p_i^j\left(t_{-j}\right) & \text{if } \left\{k:\nexists t_k\in T_k \text{ s.t. } \left[c_\gamma^{t_k}\right]=[c_k]\right\}=j, \\ A_i & \text{otherwise and if } k+1>k \text{ if } k\in H\left(t_i\right), \end{cases} \end{aligned} \tag{6.1}$$

Next, we specify the strategies of the players at the second stage. If for all j there is a $t_j\in T_j$ such that player j offers a contract $c_\gamma^{t_j}$, then Player i takes action $s_i\left(t\right)$. Suppose now that one player deviated, say Player k, and he offered a contract c_k, and player j offered $c_\gamma^{t_j}$ for all $j\neq k$. Define $f_k:T_{-k}\to 2^{A_k}$ as follows:

$$f_k\left(t_{-k}\right)=c^k\left(\left[c^k\right],\left[c_\gamma^{t_j}\right]_{j\neq k}\right), \tag{6.2}$$

where $\left[c_\gamma^{t_j}\right]_{j\neq k}$ denotes the vector of the Godel codes of players other than k. Define player i's strategy as $s_i^k\left(f_k,t_{-k}\right)$. Notice that by (6.1) these second-stage strategies are consistent with the restrictions imposed by the contracts and the refinement $\mathcal{R}_k$, that is, $s_i\left(t\right)\in r_i\left(t\right)$ and $s^k\left(f_k,t_{-k}\right)\in\mathcal{R}_k\left[f_k\left(t_{-k}\right),p^k\left(t_{-k}\right),\tau_{-k}\left(t_{-k}\right)\right]$. (We do not have to specify the strategies if more than one players deviate at the contracting stage.)

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. Consider

constraint (5.1) with $t_i = t_i'$. This constraint requires $s_i(t)$ to be a best response to the strategies of the other players. It remained to show that players do not have incentive to deviate at the contracting stage. Suppose that player k with type t_k offers a contract c_k which is different from $c_\gamma^{t_k}$. We shall consider two cases. Case 1: $c_k = c_\gamma^{t_k'}$ but $\tau_k(t_k) \neq \tau_k(t_k')$. Then, by (5.1), this deviation is not profitable no matter what the strategy of player k is at the second stage. Case 2: $c_k \neq c_\gamma^{t_k'}$ for all $t_k' \in T_k$. Such a deviation induces player i with type t_i to take action $s_i^k(f_k, t_{-k})$. Hence, by (5.2) such a deviation cannot be profitable. ∎

6.2. **Invariant punishment correspondence.** The point of this section is to show the existence of the punishment correspondence p^i. A deviator contemplates different commitment correspondences f_i. What this Lemma shows is that there has to existence some fixed punishment correspondence $p^i(t_{-i})$ such that no matter which commitment correspondence the deviator want to implement, there must be a way for him to write his contract in such a way that the response of the non-deviators is exactly the same, and is given by this correspondence p^i. This is a consequence of the fact that contracts are required to be definable functions.

Let $c_i^{t_i}$ denote the contract of Player i with type t_i. Define $\tau(t) = \left\{t' \in T : \forall i \; c_i^{t_i} = c_i^{t_i'}\right\}$.

Lemma 4. For any array $\left\{c_i^{t_i}\right\}_{i=1,\ldots,m,\; t_i \in T^i}$ of contracts and every i, there are τ_{-i} measurable functions, $p_k^i(t_{-i})$ for all $k \neq i$, such that for any τ_{-i} measurable function $f_i : T_{-i} \to 2^{A_i}$, there is a contract c_i^* such that

$$c_i^*\left([c_i^*], \left(\left[c_j^{t_j}\right]\right)_{j\neq i}\right) = f(t_{-i}) \tag{6.3}$$

and for all $k \neq i$

$$c_k^{t_k}\left([c_i^*], \left(\left[c_j^{t_j}\right]\right)_{j\neq i}\right) = p_k^i(t_{-i}). \tag{6.4}$$

In a contract equilibrium, a player expects his opponents to offer the contracts $c_j^{t_j}$. Each alternative contract that he offers against this array induces a commitment correspondence $f(t_{-j})$ and elicits some kind of response. The Lemma shows that provided the contracts the others offer are all definable functions, there must exist some collection of punishment correspondences $\left\{p_k^i\right\}$ such that for any commitment correspondence that player i wants, he can write his own contract in such a way that the others respond with exactly the same punishment $\left\{p_k^i\right\}$.

First, we reformulate the statement of the lemma. Let $(A_i)_\tau^{|T_{-i}|}$ denote the set of $|T_{-i}|$ dimensional vector of subsets of A_i which are measurable with respect to τ_{-i}, that is,

$$(A_i)_\tau^{|T_{-i}|} = \left\{\left(A_i^{t_{-i}}\right)_{t_{-i}\in T_{-i}} : A_i^{t_{-i}} \in A_i \text{ and } A_i^{t_{-i}} = A_i^{t'_{-i}} \text{ if } \tau_{-i}(t_{-i}) = \tau_{-i}\left(t'_{-i}\right)\right\}.$$

For all $\left(A_i^{\tau_{-i}(t_{-i})}\right)_{t_{-i}} \subset (A_i)^{|T_{-i}|}$ define $S\left(\left(A_i^{\tau_{-i}(t_{-i})}\right)_{t^{-i}}\right)$ as follows:

$$\left\{\left(A_{-i}^{\tau_{-i}(t_{-i})}\right)_{t^{-i}} : A_{-i}^{\tau_{-i}(t_{-i})} \subset A_{-i},\; \exists c_i \text{ s. t. } c_i\left([c_i], \left[c_{-i}^{t_{-i}}\right]\right) = A_i^{\tau_{-i}(t_{-i})},\; c_{-i}^{t_{-i}}\left([c_i], \left[c_{-i}^{t_{-i}}\right]\right) = A_{-i}^{\tau_{-i}(t_{-i})}\right\}.$$

Let us explain what it means that $\left(A_{-i}^{\tau_{-i}\left(t^{t_{-i}}\right)}\right)_{t_{-i}} \in S\left(\left(A_i^{\tau_{-i}(t_{-i})}\right)_{t_{-i}}\right)$. By the definition of S, there exists a contract, c_i, available for player i such that if the type profile of the other players is t_{-i}, then if player i offers c_i then his restricted action space will be $A_i^{\tau_{-i}\left(t^{-i}\right)}$ and players $-i$'s restricted actions space will be $A_{-i}^{\tau_{-i}\left(t^{-i}\right)}$. We claim that the statement of the lemma is equivalent to

$$\cap_{\left\{A_i^{\tau_{-i}(t_{-i})}\right\}_{t_{-i}}} S\left(\left\{A_i^{\tau_{-i}(t_{-i})}\right\}_{t_{-i}}\right) \neq \{\emptyset\}. \tag{6.5}$$

To see this, suppose first that the previous displayed statement is true, and $\left(A_{-i}^{\tau_{-i}(t_{-i})}\right)_{t^{-i}}$ is an element of the intersection. Define $p_i^k(t_{-i})$ to be $A_{-i}^{\tau_{-i}(t_{-i})}$ for all $k \neq i$ and $t_{-i} \in T_{-i}$. For a τ_{-i} measurable function $f_i : T_{-i} \to \tilde{A}_i$, consider $S\left(\left(f_i(t_{-i})\right)_{t_{-i}}\right)$. Since $\left(p_i^k(t_{-i})\right)_{t_{-i}} \in S\left(\left(f_i(t_{-i})\right)_{t_{-i}}\right)$, and by the definition of S, there exists a c_i^* such that (6.3) and (6.4) are satisfied. Conversely, suppose that (6.5) is not true. Then, for all $\left\{p_i^k(t_{-i})\right\}_{k\neq i}$ τ_{-i} measurable functions there exists $\left\{A_i^{\tau_{-i}(t_{-i})}\right\}_{t_{-i}} \in (A_i)^{|T_{-i}|}$, such that

$$\left(p_i^{-i}(t_{-i})\right)_{t_{-i}} \notin S\left(\left\{A_i^{\tau_{-i}(t_{-i})}\right\}_{t_{-i}}\right),$$

where $p_i^i(t_{-i}) = \left(p_i^k(t_{-i})\right)_{k\neq i}$. Then if $f_i(t_{-i})$ is defined to be $A_i^{\tau_{-i}(t_{-i})}$ for all $t_{-i} \in T_{-i}$, there does not exist a contract c_i^* such that (6.4) is satisfied.

Proof. Suppose by contradiction that $\cap_{\left\{A_i^{\tau_{-i}(t_{-i})}\right\}_{t_{-i}}} S\left(\left\{A_i^{\tau_{-i}(t_{-i})}\right\}_{t_{-i}}\right) = \{\emptyset\}$. Then, for all $\left\{A_{-i}^{\tau_{-i}(t_{-i})}\right\}_{t_{-i}} \subset \left(A^{-i}\right)^{|T_{-i}|}$ there exists an $\left\{A_i^{\tau_{-i}(t_{-i})}\right\}_{t_{-i}} \subset \left(A^{i}\right)^{|T_{-i}|}$ such that $\left\{A_{-i}^{\tau_{-i}(t_{-i})}\right\}_{t_{-i}} \notin S\left(\left\{A_i^{\tau_{-i}(t_{-i})}\right\}_{t_{-i}}\right)$. Let us fix a function $f : 2^{\left(A^{-i}\right)^{|T_{-i}|}} \to 2^{\left(A^{i}\right)^{|T_{-i}|}}$ such that

$$\forall \left\{A_{-i}^{\tau_{-i}(t_{-i})}\right\}_{t_{-i}} \subset \left(A^{-i}\right)^{|T_{-i}|} : \left\{A_{-i}^{\tau_{-i}(t_{-i})}\right\}_{t_{-i}} \notin S\left(f\left(\left\{A_{-i}^{t_{-i}}\right\}_{t_{-i}}\right)\right).$$

Let $f_{\tau_{-i}(t_{-i})}$ denote the projection of f corresponding to t_{-i}. That is, $f = \left\{f_{\tau_{-i}(t_{-i})}\right\}_{t_{-i}}$. Define c_x as follows:

$$c_x(c) = \begin{cases} f_{t'_{-i}}\left(\left\{c_{\tau_{-i}(t_{-i})}\left([< x > (x)]\right)\right\}_{t_{-i}}\right) & \text{if } \exists t'_{-i} \in T^{-i} \text{ st. } c = c_{\tau_{-i}(t'_{-i})}, \\ A_i & \text{otherwise.} \end{cases}$$

Since f and $c_{\tau_{-i}(t_{-i})}$ are definable functions, c_x is a definable function in one free variable. Let γ denote its Godel code. Then

$$c_\gamma(c) = \begin{cases} f_{\tau_{-i}(t'_{-i})}\left(\left\{c_{\tau_{-i}(t_{-i})}\left([c_\gamma]\right)\right\}_{t_{-i}}\right) & \text{if } \exists t'_{-i} \in T^{-i} \text{ st. } c = c_{\tau_{-i}(t'_{-i})}, \\ A^i & \text{otherwise.} \end{cases}.$$

Notice that

$$\left\{c_{\tau_{-i}(t_{-i})}\left([c_\gamma]\right)\right\}_{t_{-i}} \in S\left(\left\{c_\gamma\left(\left[c_{\tau_{-i}(t_{-i})}\right]\right)\right\}_{t_{-i}}\right) \tag{6.6}$$

by the definition of S. On the other hand,

$$\begin{aligned}\left\{c_\gamma\left(\left[c_{\tau_{-i}(t_{-i})}\right]\right)\right\}_{t_{-i}} &= \left\{f_{t_{-i}}\left(\left\{c_{\tau_{-i}(t'_{-i})}\left([c_\gamma]\right)\right\}_{t'_{-i}}\right)\right\}_{t_{-i}} \\ &= f\left(\left\{c_{\tau_{-i}(t_{-i})}\left([c_\gamma]\right)\right\}_{t_{-i}}\right),\end{aligned}$$

and therefore,

$$\left\{c_{\tau_{-i}(t_{-i})}\left([c_\gamma]\right)\right\}_{t_{-i}} \notin S\left(\left\{c_\gamma\left(\left[c_{\tau_{-i}(t_{-i})}\right]\right)\right\}_{t_{-i}}\right) \tag{6.7}$$

by the definition of f. Notice that (6.6) and (6.7) contradict to each others, and hence the (6.5) holds. ∎

6.3. **Only if part of the proof of Theorem 3.**

Proof. Fix an equilibrium in the contracting game. We shall construct the objects τ, s, $\{r_i\}_{i=1}^m$, $\left\{s^i\right\}_{i=1}^m$, and $\left\{p^i\right\}_{i=1}^m$ such that the constraints (5.1) and (5.2) are satisfied. Denote the equilibrium contract of Player i with type t_i by $c_i^{t_i}$. Define the partition, τ, as follows:

$$\tau(t) = \left\{t' \in T : \forall i \; c_i^{t_i} = c_i^{t'_i}\right\}.$$

Next, we construct the functions $\{r_i\}_{i=1}^m$. Let

$$r_i(t) = c_i^{t_i}\left(\left[c_i^{t_i}\right], \left(\left[c_j^{t_j}\right]\right)_{j\neq i}\right), \tag{6.8}$$

for all $i \in \{1, ..., m\}$. Notice that $r_i(t) \in 2^{A_i}$. In addition, r_i is measurable with respect to τ_{-i} by the definition of τ. The second-stage strategies depend on the contracts offered at the first stage. First, we deal with strategies on the equilibrium path. Let $q_i^{t_i}\left(\left(\left[c_j^{t_j}\right]\right)_{j\neq i}\right)$ denote the second stage strategy of Player i with type t_i. Observe that

$$q_i^{t_i}\left(\left(\left[c_j^{t_j}\right]\right)_{j\neq i}\right) \in c_i^{t_i}\left(\left[c_i^{t_i}\right], \left(\left[c_j^{t_j}\right]\right)_{j\neq i}\right) \tag{6.9}$$

must be satisfied according to the rules of the contracting game. Define $s_i(t)$ to be $q_i^{t_i}\left(\left(\left[c_j^{t_j}\right]\right)_{j\neq i}\right)$. The function $s_i(t)$ is obviously measurable with respect to τ_{-i}. In addition, $s_i(t) \in r_i(t)$ by (6.8) and (6.9). Let $s(t)$ denote $(s_1(t), ..., s_m(t))$.

We are ready to show that the triple $(\tau, \{r_i\}, s)$ satisfy (5.1). First, consider this constraint with $t'_i = t_i$. Then, this constraint requires $q_i^{t_i}\left(\left(\left[c_j^{t_j}\right]\right)_{j\neq i}\right)$ to be a best-response of player i to the strategies of the other players. Since $q_i^{t_i}$ was an equilibrium strategy, it has to be a best response and hence, (5.1) is indeed satisfied. Second, consider (5.1) with $t'_i \neq t_i$. Then, this constraint requires player i with type t_i to prefer to offer contract $c_i^{t_i}$ instead of $c_i^{t'_i}$. Indeed, the left-hand-side is just his 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, such a deviation cannot be profitable and hence, (5.1) is satisfied.

It remains to construct $\{p_i\}_{i=1}^m$ and $\left\{s^i\right\}_{i=1}^m$ and show that (5.2) is also satisfied. Define $p_k^i\left(t_{-i}\right)$ for all $k \neq i$ and for all $i \in \{1, ..., m\}$ according to the statement of Lemma 4. In addition, let $c_i^{f_i}$ denote the contract of Player i such that

$$c_i^{f_i}\left(\left[c_i^{f_i}\right], \left(\left[c_j^{t_j}\right]_{j\neq i}\right)\right) = f_i\left(t_{-i}\right)$$

and for all $k \neq i$

$$c_k^{t_k}\left(\left[c_i^{f_i}\right], \left(\left[c_j^{t_j}\right]_{j\neq i}\right)\right) = p_i^k\left(t_{-i}\right).$$

Let $q_k^i\left(f_i, \left(\left[c_j^{t_j}\right]_{j\neq i}\right)\right)$ denote the off-equilibrium strategy of player k when player i unilaterally deviates to contract $c_i^{f_i}$. Define $s_k^i\left(f_i, t_{-i}\right)$ to be $q_k^i\left(f_i, \left(\left[c_j^{t_j}\right]_{j\neq i}\right)\right)$. The function s_k^i is measurable with respect to $\tau_{-ik}\left(t_{-ik}\right)$. Given these notations, (5.2) requires that player i cannot profitably deviate by offering an off-equilibrium contract in the form of $c_i^{f_i}$, and hence, this constraint is satisfied. Finally, since q_k^i is an off equilibrium strategy in an $\mathcal{R}$-equilibrium, so

$$\left\{q_k^i\left(f_i, \left(\left[c_j^{t_j}\right]_{j\neq i}\right)\right)\right\}_{k\neq i} \in$$

$$\mathcal{R}_i\left[c_i^{f_i}\left(\left[c_i^{f_i}\right], \left(\left[c_j^{t_j}\right]_{j\neq i}\right)\right), c_{-i}^{t_{-i}}\left(\left[c_i^{f_i}\right], \left(\left[c_j^{t_j}\right]_{j\neq i}\right)\right), \tau_{-i}\left(t_{-i}\right)\right] =$$

$$\mathcal{R}_i\left[f_i\left(t_{-i}\right), p_i\left(t_{-i}\right), \tau\left(t_{-i}\right)\right]$$

and the refinement condition is satisfied. ∎

6.4. **Example 1 - making actions depend on types.** The following examples illustrate the properties of the contracting equilibrium in the Bayesian case. The first example, illustrates how contracting equilibrium can be used to make one player's action depend on another player's type. This is something that cannot be accomplished in the Bayesian equilibrium of the original game. In this example, the row player is privately informed and has one of two equally likely types, t_1 and t_2. Each player has two possible actions in the default Bayesian game, $\{a_1, a_2\}$ for the row player, $\{b_1, b_2\}$ for the column player. The payoffs for each of the row player's types are given in the following tables:

	b_1	b_2
a_1	$3,3$	$-1,4$
a_2	0,0	$0,0$

	b_1	b_2
a_1	0,0	0,0
a_2	$-1,4$	$3,3$

This is a relatively simple coordination problem, save two things - the way the players want to coordinate depends on the row player's type, and if the column player learns the row player's type, his weakly dominate action is inconsistent with the coordinated outcome. The unique Bayesian equilibrium has player 1 using action a_1 if his type is t_1 and action a_2 if his type is t_2. The column

player randomizes with equal probability between his two actions. The expected payoffs to the column player are $\frac{3}{2}$, the expected payoff to the row player is 1 for each of his types.

A mechanism designer can implement the coordinated outcome $s(t_1) = (a_1, b_1)$ and $s(t_2) = (a_2, b_2)$ by simply asking the informed agent his type, then instructing the uniformed agent which of his actions to take. If either player refuses to participate, then they simply play the Bayesian equilibrium described above. The allocation is incentive compatible and individually rational from the mechanism designer's perspective.

To show that the allocation rule s is implementable as a contract equilibrium, define the commitment correspondence $r_1(t_1) = \{a_1\}$, $r_1(t_2) = \{a_2\}$, $r_2(t_1) = \{b_1\}$, $r_2(t_2) = \{b_2\}$. This commitment correspondence is measurable with respect to full information and implements the allocation s since players never have any choices to make ex post. It is incentive compatible so it will be implementable if there is a type contingent punishment that the row player can impose that makes it unprofitable for the column player to try to exploit this type information.[8] This is evidently the punishment $p_1(t_1) = \{a_2\}$ and $p_1(t_2) = \{a_1\}$, since this holds the column player's payoff to zero no matter what he does.

It might help at this point to describe the way the contract equilibrium works in this example. The informed player writes a different 'reciprocal' contract for each of his possible types. These contracts both specify the same target Godel code, say n^*. The contract for type t_1 says that if the Godel code of the uninformed player's contract is n^*, then the informed player will commit to action a_1. If the Godel code of the uninformed player's contract is anything else, then the informed player of type t_1 will commit to action a_2. The contract for t_2 is similar with the actions reversed. Encoding these contracts gives a pair of Godel codes, say m_1 and m_2, corresponding to each of the informed player's possible contracts. The uninformed player writes a contract that says that if the Godel code of the informed player's contract is m_1, then he will commit to b_1, if the Godel code of the informed player's contract is m_2, then he will commit to b_2, otherwise he will commit to $\{b_1, b_2\}$ and choose among them ex post. The theorem above shows that there is a triple of integers (n^*, r_1, r_2) such that the Godel code of the uninformed player's contract is n^*.

6.5. **Example 2: contract equilibrium doesn't do as well as a mechanism designer.** In the example just described, the contract equilibrium supports everything that a mechanism designer might want to implement. However, as we mentioned in the introduction to this section, contract equilibrium imposes a restriction on feasible allocations. When a player decides to deviate, he knows that he will learn something about the types of the other players when he sees their contracts. In addition, since he can condition his behavior on contracts, he can make his deviation depend on this type information.

To illustrate the limitations that this imposes, consider the following variant of the example given above. There are again two players each with two possible actions. The row player has

[8] Of course, the uninformed player must also specify a punishment. For simplicity, we specify it below.

two possible types, either t_1 or t_2, which are equally likely. The column player has no private information. The payoffs for each of the informed player's possible types are given in the following tables:

	b_1	b_2
a_1	$3,3$	$-1,4$
a_2	$0,4$	$2,-1$

and

	b_1	b_2
a_1	$2,-1$	$0,4$
a_2	$-1,4$	$3,3$

The Bayesian equilibrium of this default game has each player randomizing with equal probability over each of his actions no matter what his information. The informed (row) player has payoff 1 in this equilibrium no matter what his type, while the uninformed column player has payoff $\frac{5}{2}$. The Myerson mechanism designer has no problem implementing the allocation $s(t_1) = (a_1, b_1)$ and $s(t_2) = (a_2, b_2)$. He does this by inviting the players to participate in a mechanism in which he asks the row player to report his type. If he reports t_1 then he instructs the players to use actions a_1 and b_1, and similarly when type t_2 is reported. By agreeing to participate, the players commit themselves to follow the mechanism designer's instruction. This is incentive compatible because the row player's payoff falls from 3 to 2 if he misreports his type. The allocation is individually rational in the usual mechanism design sense as long as a refusal to participate by either player results in both players playing the (unique) Bayesian equilibrium of the original game.

This allocation rule cannot be implemented as a contract equilibrium. According to Theorem 3, to implement it, there must be a type contingent commitment that the row player can make, and some specification of the row player's actions ex post that hold the column players payoff below 3 when he simply commits to choose his from his possible actions b_1 and b_2 ex post. To see that there is no such punishment, observe that if π is the probability with which the row player uses action a_1 in the ex post game, then the column player's payoff when the row player has type 1 is

$$\max\left[\pi 3 + (1-\pi)\,4, \pi 4 - (1-\pi)\right] =$$

$$\max\left[4-\pi, 5\pi - 1\right] \geq \frac{19}{6}.$$

The argument is identical when the row player has type 2. No such punishment exists. Thus by Theorem 3, there is no contract equilibrium that supports this outcome.

As before, if there were a contract equilibrium that could support this outcome, then the column player has to take an action that depends on the row player's type. In principle, he can do this because he can commit himself to an action that depends on the row player's contract, which in turn depends on the row player's type. However if the column player knows that the contract will reveal the row player's type, then a deviation to a contract that simply allows the column player to take his action ex post has to be profitable.

6.6. **Payoffs lie between the Bayesian equilibrium and those implementable by a mechanism designer.** This final example is intended to illustrate a number of things. First, it shows that the contract equilibrium implements strictly more than the Bayesian equilibrium of the default

game, but less than what is implementable by a mechanism designer. Second, it has non-degenerate commitment and punishment correspondences, both of which are type dependent.

The example also illustrates how randomization can be incorporated into the final stage of the contracting process when players choose actions from their commitment correspondences. We have ignored randomization in the statement of our main theorem to simplify. This example illustrates how the extension works.

The row player has three equally likely types supporting payoffs given in the following tables:

t_1	b_1	b_2
a_1	$3,3$	$-1,4$
a_2	$0,4$	$2,-1$

t_2	b_1	b_2
a_1	$2,-1$	$0,4$
a_2	$-1,4$	$3,3$

t_3	b_1	b_2
a_1	$-2,-2$	$4,-1$
a_2	$0,4$	$2,\frac{11}{4}$

The payoffs in the first two boxes are the same as they were in the second example discussed above. The unique Bayesian equilibrium for this game has the uninformed player randomizing equally between b_1 and b_2. The informed player randomizes equally when his types are t_1 and t_2, but chooses a_1 with probability $\frac{5}{9}$ when his type is t_3. The payoff to the informed player is 1 no matter what his type, while the payoff to the uninformed player is 2. A Myerson mechanism designer can implement the allocation rule $s(t_1) = (a_1, b_1)$, $s(t_2) = s(t_3) = (a_2, b_2)$ exactly as he does in the first example, by asking the informed player his type, then telling both players what actions to take. This allocation can't be supported as a contract equilibrium. The argument is exactly as in the second example above, since the contract equilibrium has to enforce different actions when the row players types are t_1 and t_2.

However, the allocation in which both players randomize equally between their actions when the row player has type 1 or type 2, while the actions a_2 and b_2 are taken when the row player has type 3 can be supported as a contract equilibrium. This allocation is measurable with respect to the information partition $\{\{t_1, t_2\}, \{t_3\}\}$ so the commitment and punishment correspondences can depend on the row player's type. In particular, the commitment correspondence we want is $r_1(t_1) = r_1(t_2) = \{a_1, b_1\}$, $r_1(t_3) = \{a_2\}$, while $r_2(t_1) = r_2(t_2) = \{b_1, b_2\}$ and $r_2(t_3) = \{b_2\}$. The punishment correspondence for the row player is again multi-valued $p_1(t_1) = p_1(t_2) = \{a_1, a_2\}$, while $p_1(t_3) = \{a_1\}$. The column player punishes with $\{b_1, b_2\}$ for all deviations.

The behavior to be supported involves randomization among the choices in the commitment correspondence. This can be incorporated in a straightforward way by requiring that the mappings $s(t)$ and $s^i(t_{-i})$ have their range in the set of mixtures over actions whose support lies within the appropriate commitment correspondence. So in the example, $s_i(t_1) = \left\{\frac{1}{2}, \frac{1}{2}\right\} = s_i(t_2)$ while $s_i(t_3) = \{0, 1\}$ for $i = c, r$. The behavior during the punishment phase is defined similarly.

Consider the case where the column player deviates. Let $s^c(f, t_1) = s^c(f, t_2) = \left\{\frac{1}{2}, \frac{1}{2}\right\}$ for each f, and $s^c(f, t_3) = \{1, 0\}$ as is required by the punishment correspondence. A deviation is a type contingent commitment f that has to be measurable with respect to the information partition $\{\{t_1, t_2\}, \{t_3\}\}$. As an example, take $f(t_1) = \{b_1, b_2\} = f(t_3)$ while $f(t_3) = \{b_1\}$. The

punishment has to make this and all other measurable type contingent commitments unprofitable given the column player's interim beliefs. It is straightforward to check that it accomplishes this.

One way to implement this in a contract equilibrium is to have the row players types t_1 and t_2 both offer the same contract which commits them to $\{a_1, a_2\}$ whatever contract the column player offers. When the row player has type t_3 he offers a contract that commits him to a_2 if the column player offers a contract whose Godel code is equal to some target n_c^*, but commits to a_1 against any other Godel code. The Godel code of this contract is, say n_3^*. The column player offers a contract that commits to b_2 if the Godel code of the row player's contract is n_3^*, but commits to $\{b_1, b_2\}$ against any other contract.

7. Independent Private Values

An environment that is of some interest in applications is the independent private value environment. The classic first or second price auction models are typical examples. However, the tractability of such models makes them popular. For the independent private value environment we can use our theorem to provide something that looks like a folk theorem for Bayesian equilibrium. For our purposes, this 'folk theorem' is interesting because it suggests an environment where contracts can be used to fully decentralize the mechanism designer's problem.

Players have *private values* if $u_i(a,(t_i,t_{-i})) = u_i\left(a,\left(t_i,t'_{-i}\right)\right)$ for all $a \in A$, t_{-i}, $t'_{-i} \in T_{-i}$ (players' payoffs don't depend on other players' types). Types are independently distributed if $\mathbb{E}(f(t_{-i}) : t_i) = \mathbb{E}(f(t_{-i}) : t'_i)$ for every i, t_i, t'_i and every integrable function f.

Theorem 2. *Let $s : T \to A$ be an allocation rule that is implementable by a centralized mechanism designer in an independent private value environment. Then the allocation s can be supported as a Bayesian equilibrium in contractable contracts.*

Proof. An allocation rule is implementable by a mechanism designer if there is a collection of punishments $s^i : T_{-i} \to A_{-i}$, where s^i_j is the punishment participants will impose on player i if he chooses not to participate such that for each t_i and t'_i

$$\mathbb{E}_{t_{-i}}(u_i(s(t),t) : t_i) \geq$$
$$\mathbb{E}_{t_{-i}}\left(u_i\left(a_i, s_{-i}\left(t'_i, t'_{-i}\right), \left(t_i, t'_{-i}\right)\right) : t_i\right);$$

and

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

We prove the theorem by constructing the various components required by Theorem 3.

Begin with the punishment $s^i(\cdot)$. We have from the private value and independence assumption

$$\max_{a_i \in A_i} E_{t_{-i}}\left(u_i\left(\left(a_i, s^i(t_{-i})\right), (t_i, t_{-i})\right) : t_i\right) =$$
$$\max_{a_i \in A_i} E_{t_{-i}}\left(u_i\left(\left(a_i, s^i(t_{-i})\right), t_i\right)\right).$$

Let $\tilde{g}^i$ be the distribution on A_{-i} induced by the function s^i and the distribution of t_{-i} and define the punishment

$$\tilde{s}^i\left(f, t_{-i}\right) = \tilde{g}^i$$

for each t_{-i} and every correspondence $f : T_{-i} \to \tilde{A}_i$ that is measurable with respect to full information (a set which contains all correspondences which are measurable with respect to any information structure). Then we have

$$E_{t_{-i}}\left(u_i\left(s\left(t\right), t\right) : t_i\right) \geq$$
$$\max_{a_i \in A_i} E_{t_{-i}}\left(u_i\left(\left(a_i, \tilde{s}^i\left(f_{a_i}, t_{-i}\right)\right), t_i\right)\right) =$$
$$\max_{a_i \in A_i} E\left(u_i\left(\left(a_i, \tilde{g}^i\right), t_i\right)\right)$$

where $f_{a_i}\left(t_{-i}\right) = \{a_i\} \forall t_{-i} \in T_{-i}$. The important aspect of this punishment is that it does not depend on t_{-i}. Let τ_i^f be the full information partition of T_i with $\tau^f = \prod_i \tau_i^f$. Define the commitment correspondence $r_i\left(t\right) = \{s_i\left(t\right)\}$ and the punishment correspondence $p_j^i\left(t\right) = A_j$. These correspondences are both trivially measurable with respect to τ^f. Furthermore, (5.1) holds trivially since the allocation must be incentive compatible, and r_i is always a singleton. Now for any commitment correspondence f measurable with respect to the full information partition τ_{-i}^f,

$$E_{t_{-i}}\left(\max_{a \in f_i(t_{-i})} E_{t'_{-i}}\left(u_i\left(a, \tilde{s}^i\left(f_i, t'_{-i}\right), t_i\right) : t'_{-i} \in \tau_{-i}\left(t_{-i}\right)\right) : t_i\right) \leq$$
$$\left(\max_{a \in A_i} \mathbb{E}\left(u_i\left(\left(a, \tilde{g}^i\right), t_i\right)\right)\right) \leq$$
$$E_{t_{-i}}\left(u_i\left(s\left(t\right), t\right) : t_i\right).$$

So (5.2) is satisfied. Then by Theorem 3, the allocation s is supportable as a contract equilibrium. ∎

From the proof above, it should be apparent that what makes the theorem work is the fact that the punishment that the mechanism designer uses to enforce participation has the same impact on the non-participant no matter what he learns about the participants' types. This is a consequence of the private value assumption. Some interdependent value problems will also have this property. For example, in a trading problem, not being able to trade may be worse for a player than trading no matter what he learns about the types of the others. Similar folk theorems are possible in such environments.

8. Conclusion

This paper shows how the contracts on contracts approach can be extended to environments with incomplete information by restricting players to use definable contracts. 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. A centralized mechanism designer can implement allocations 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 part of our main theorem illustrates the role that punishments play in a static contracting environment.

References

[1] Anderlini, L: Some Notes on Church's Thesis and on the Theory of Games. *Theory and Decision*, 29, 19-52, 1990.

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

[3] Canning, D: Rationality, Computability, and Nash Equilibrium. *Econometrica*, 60(4), pp. 877-888, 1992.

[4] L. Epstein and M. Peters. A revelation principle for competing mechanisms. *Journal of Economic Theory*, 88(1):119–161, September 1999.

[5] C Fershtman and K.L. Judd. Equilibrium incentives in oligopoly. *American Economic Review*, 77:927–940, 1987.

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

[7] Kalai, A.T., Kalai, E., Lehrer, E., and D. Samet. A Commitment Folk Theorem. manuscript, Tel-Aviv University, April, 2007.

[8] Michael Katz. Observable contracts as commitments: Interdependent contracts and moral hazard. *Journal of Economics and Management Strategy*, 15(3):685–706, September 2006.

[9] David Martimort and Lars Stole. Communications spaces, equilibria sets and the revelation principle under common agency. University of Chicago unublished manuscript, 1998.

[10] Michael Peters. Common agency and the revelation principle. *Econometrica*, 69(5):1349–1372, September 2001.

[11] S.C. Salop. *Practises that Credibly Facilitate Ologopoly Coordination*. Cambridge, MIT Press, 1986.

[12] Leo Simon and W. Zame. Discontinuous games and endogenous sharing rules. *Econometrica*, 58(4):861–872, 1990.

[13] Moshe Tennenholz. Program Equilibrium. *Games and Economic Behavior*,49(2):363–373,2004.

[14] Takuro Yamashita. A revelation principle and a folk theorem without repetition in games with multiple principles and agents. manuscript, Stanford University, February 2007.