

DEFINABLE AND CONTRACTIBLE CONTRACTS

MICHAEL PETERS AND BALZS SZENTES

ABSTRACT. This paper analyzes a normal form game 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 Gödel code of every other player's contract. We characterize the set of outcomes that are supportable as (pure strategy) equilibrium with such contracts. With symmetric information, this is all outcomes in which all players receive at least their min max payoff. With incomplete information this all allocation rules that are incentive compatible and satisfy an individual rationality condition that we describe. We contrast the set of allocation rules that can be supported by Bayesian equilibrium with those attainable by a mechanism designer.

1. SELF REFERENTIAL STRATEGIES AND RECIPROCITY IN STATIC GAMES

In this paper we characterize the allocation rules attainable by players in a 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. None of them can be used in any but the simplest problems. The meet the competition argument 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.

Very Preliminary Version - do not cite - November 6, 2008.

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

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, and this is exactly what we are trying to avoid.

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.

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

²One 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.

reasonable that a state could write a contract that simply fixes any tax rate independent of what the other states do. Let \bar{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 $\bar{C} \cup \{r\}$. 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 $\bar{C} \cup \{r\}$ and declare the outcome if both states offer r to be (exempt, 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 amount 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:

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

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.

Of course, in the spirit of the ad hoc approach above, we could try to add the contract C to r and \bar{C} . This approach breaks down once the game becomes asymmetric. For example, 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 actions, 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 word we need to add to our contracting language to support the outcomes we want.

Our approach avoids these problems. We fix a language, 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. The set of contracts that we use 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 used to describe equilibrium in all strategic situations.

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 N 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. 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

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

The mapping $\langle x \rangle^{(x)}$ is the composition of two functions. First, the function $\langle x \rangle$ is the inverse operation to the Godel coding. That is, $\langle n \rangle$ 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 $\langle n \rangle^{(n)}$ is itself a definable strategy (without a free variable). $[\langle n \rangle^{(n)}]$ 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

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

So

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

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 $[c_2] = [c_{[c_x]}]$, 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 $[c_{[c_x]}]$, 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

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

where g is a definable function. If there exists an n^* such that $[c_{n^*}] = g(n^*)$, then c_{n^*} is obviously a self-referential contract. Indeed, what we did above is that we chose $g(n)$ 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

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

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

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 $n_0 = [\forall n; \bar{c}(n) = C]$ 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 $c_{[c_{n_0}]}$, and so on.

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

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

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 $[c_{[c_x]}]$. 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 $[c_{[c_x]}]$, i.e., the same condition that A requires.

2. LITERATURE

The approach of allowing contracts to depend on other contracts 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, Tennenholtz effectively writes out explicitly the a sequence of programming statements of the form

$$MYPROGRAM_i \equiv \begin{cases} action_i & \text{syntax of the other players' programs are } MYPROGRAM_j \\ \text{minmax } j & \text{otherwise} \end{cases}$$

For the array of actions to be enforced, the other players programs must all contain the key words $MYPROGRAM_j$, and must interpret the term as described here.

On the most basic level, our paper differs since we are interested in problems with incomplete information. In this sense we go far beyond the environment studied by Tennenholtz. At the contracting level, we reverse the reasoning in Tennenholtz. Instead of defining an allocation, then providing a set of contracts that can be used to support the equilibrium outcome, we start by defining a language and feasible set of contracts. From this well defined contracting game, we prove that the language will contain the keywords we need to support outcomes. Thus we prove that the set of feasible contracts we provide must contain contracts that support allocations. However, our basic approach is the same in the sense that instead of comparing the 'syntax' of computer programs, we compare the Godel codes of definable statements.

An approach that encompasses both our approach and Tennenholtz is given in [7]. They begin with an arbitrary class of *commitment devices* which are required by construction to have the property that each array of commitment devices maps into a unique outcome. In this sense, our contracts, and Tennenholtz' programs are special cases. They then show that for an arbitrary finite two person game, they can construct a set of commitment devices that support payoffs above minmax as a Nash equilibrium.

Apart from incorporating incomplete information, our objective is to start by defining a natural set of commitment devices, then to show that we can always find the devices within our class that we need to support whatever outcome is of interest. Of course, the term 'natural' means different things to different people. The notion of contracting on contracts is familiar enough to economists that our class of contracts seems realistic.

We emphasize that the contribution here is not intended to be a contribution to the computer science literature. In fact, we take the traditional view that 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. Using the usual assumption that these contracts are enforceable, then the legal system won't actually play a role, since no player would ever violate a contract. This is quite different from the usual mechanism design approach in which a mechanism designer plays an active role in coordinating an outcome.

Finally, we use the Godel coding. Any bijection from finite texts to integers would do. The Godel code has the advantage that it is a definable function. So it 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 GDEL 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:

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

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 Gdel Code of the text φ .

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

Definition 3.1. 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 $b \in f(a_1, \dots, a_k)$ if and only if $\phi(a_1, \dots, a_k, b)$ 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. A NORMAL FORM 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 $u_i(a_1, \dots, a_m)$. 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 3.1. 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 ($j = 1, \dots, m$), then player i can only take action a_k^i at stage two if $k \in c_i([c_1], \dots, [c_m])$. We restrict attention to pure-strategy subgame perfect equilibria of this game.

The lowest payoff for any player in any pure strategy equilibrium of the ‘default’ game in which players simply choose actions from A is

$$u_i^* = \min_{a_{-i} \in A_{-i}} \max_{a_i \in A_i} u_i(a_i, a_{-i}),$$

Let a_j^* be any one of the actions that j uses to attain his minmax payoff. Let us fix an action $a_{j_i}^i$ for player i , such that,

$$(a_{j_1}^1, \dots, a_{j_m}^m) \in \arg \min_{a_{-i}} u_j(a_j, a_{-j}).$$

That is, $a_{j_i}^i$ is the action that player i uses to punish player j . For convenience, define $i_i = 1$ for all $i \in \{1, \dots, m\}$.

Theorem 4.1. *The action profile $(a_{k_1}^1, \dots, a_{k_m}^m) \in A$ is supportable as an equilibrium outcome in the contracting game with pure strategy SPNE if and only if $u_i(a) \geq u_i^*$ 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 $\langle n \rangle$ denotes the text whose Gdel code is n . That is, $[\langle n \rangle] = n$. Second, for any text φ , let $\varphi^{(n_1, \dots, n_k)}$ 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 $i = 1, \dots, n$. For example, if φ is $x_1 > x_2$, $n_1 = 2$, and $n_2 = 1$ then $\varphi^{(n_1, n_2)}$ is $2 > 1$. Consider now the following text in k free variable: $\langle x_i \rangle^{(x_1, \dots, x_k)}$, where $i \leq k$. One can evaluate this statement at any k dimensional

vector of integers. Since the Godel coding was a bijection $\langle n_i \rangle$ is a text for each $n_i \in \mathbb{N}$. In addition, $\varphi^{(n_1, \dots, n_k)}$ is defined for all φ and (n_1, \dots, n_k) . In addition, it is a well-known result in Mathematical Logic, that if $f(n_1, \dots, n_k) = [\langle n_i \rangle^{(n_1, \dots, n_k)}]$, 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 ($j = 1, \dots, m$) 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 $\bar{c} : \mathbb{N}^m \rightarrow \mathbb{N}$, such that $\bar{c}(n_1, \dots, n_m) = \mathbb{N}$ for all $(n_1, \dots, n_m) \in \mathbb{N}^m$. The contract \bar{c} is obviously definable.³ We show that if $u_i < u_i^*$, player i can profitably deviate at the first stage by offering \bar{c} instead of c_i . Let $\tilde{c}_j = c_j$ if $j \neq i$ and $\tilde{c}_i = \bar{c}$. Let $\tilde{A}_j = \left\{ a_k^j : k \in \tilde{c}_j([\tilde{c}_1], \dots, [\tilde{c}_m]) \right\}$. That is, \tilde{A}_j is the action space of player j in the subgame generated by the contract profile $(\tilde{c}_1, \dots, \tilde{c}_m)$. Also notice that $\tilde{A}_i = A_i$. The payoff of player i in any pure strategy equilibrium of this subgame is weakly larger than

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

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

For the if part, consider the following contract of Player i , c_{x_1, \dots, x_m}^i , in m free variables:

$$(4.1) \quad c_{x_1, \dots, x_m}^i \left(([c^j]_{j=1}^m) \right) = \begin{cases} k_i & \text{if } |\{k : [\langle x_k \rangle^{(x_1, \dots, x_m)}] \neq [c^k]\}| \neq 1, \\ j_i & \text{if } \{k : [\langle x_k \rangle^{(x_1, \dots, x_m)}] \neq [c^k]\} = \{j\} \end{cases}$$

This contract with free variables is a definable function with free variables from \mathbb{N}^{m-1} to \mathbb{N} as long as the actions are replaced with their indices.

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

$$c_{\gamma_1, \dots, \gamma_m}^i \left(([c^j]_{j=1}^m) \right) = \begin{cases} k_i & \text{if } |\{k : [\langle \gamma_k \rangle^{(\gamma_1, \dots, \gamma_m)}] \neq [c^k]\}| \neq 1, \\ j_i & \text{if } \{k : [\langle \gamma_k \rangle^{(\gamma_1, \dots, \gamma_m)}] \neq [c^k]\} = \{j\}. \end{cases}$$

Player i needs to check whether the Godel code of $\langle \gamma_k \rangle^{(\gamma_1, \dots, \gamma_m)}$ is equal to the Godel code of $c_{\gamma_1, \dots, \gamma_m}^k$. The integer γ_k is the Godel code of the contract with free variable c_{x_1, \dots, x_m}^i . Player i 's contract says to take this contract with free variable, fix the free variables at $\gamma_1, \dots, \gamma_m$ (which

³For example, the predicate

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

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

gives the contract $c_{\gamma_1, \dots, \gamma_m}^k$, 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. 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.

Player j can deviate to any definable contract mapping \mathbb{N} into \mathbb{N} . However, any such contract will have a different Godel code, and so will induce the punishment $\{a_{j_i}^i\}_{i \neq j}$ from the other players. Recall that $\{a_{j_i}^i\}_{i \neq j}$ is the action profile that players other than player j use to minmax player j . Since $u_j(a) \geq u_j^*$ this 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 action spaces 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 a two-person game similar to ours. Instead of taking actions, players submit commitment devices from a certain set. The devices then determine the action profile. 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 not hard to extent their results to our model and obtain the following theorem.

Theorem 1. *Suppose that $\sigma \in \Delta(A)$, and $\sigma(a) \in \mathbb{Q}$ for all $a \in A$. The 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 $i \in \{1, \dots, m\}$.*

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.

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

There are, however, several advantages of our approach over modelling contracts with Turing machines. Let us explain.

1. Turing machines do not always halt. Therefore, it is not clear how one can define our contracting game properly. In particular, it is not straightforward how to define the restriction on the action space of a player, if his machine does not halt. One might suggest that if a player submits a Turing machine that does not halt, then define his second stage action space to be the whole space (or a default subset). We think that such a definition might be arbitrary. In addition, the problem whether or not a Turing machine halts is an undecidable problem. That is, there is no Turing machine which can determine whether a player deviated or not. An alternative 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 also 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.

2. 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

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

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. That is, the equilibrium contract is sensitive to the way it is written. A player does not only require the other player to have the right intentions, but also requires him to express himself in a unique way. This feature makes us doubt whether machine contracts is the right way of modeling contractible contracts.

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

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

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

5. CONTRACTING IN A BAYESIAN ENVIRONMENT

This section shows how to extend the result of the previous section to games with incomplete information. The model is the same as in the previous section, with the addition of player types. There are n 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 $t \in T_1 \times \dots \times T_m$. Notice that a strategy rule for player i in the Bayesian game the players might otherwise be involved in is an element of $A_i^{|T_i|}$.

The contracting game involves the same m players. A contract is a definable function from the Godel codes of the contracts of the other players into $\tilde{A}_i = 2^{A^i}$, which is the set of subsets of i 's set of actions. We want to characterize the set of allocations that can be supported as Bayesian equilibrium in which players use pure strategies when selecting their contracts. This characterization illustrates a surprising property of contract equilibrium - the actions that that a player ultimately takes can depend on the types of other players. The reason for this is that in the contract equilibrium, each players contract specifies commitments that vary with the contracts of the other players. Since these contracts depend on their types, this allows a player to condition his behavior to some degree on the types of the others. As in the symmetric information case, contracts on contracts can then be used to support a kind of cooperative behavior.

Part of the argument below extends the reciprocal contracting idea to the Bayesian case. Each of the player's types has a reciprocal contract that takes a cooperative action if all the other players offer reciprocal contracts appropriate for one of their types. If any player deviates, then the non-deviators punish. This supports a lot of behavior in the contracting game that is unsupportable in the simple Bayesian game without contracts. This argument by itself isn't sufficient to provide a complete characterization of equilibria in the contracting game because it leaves open the possibility that contracts can be used to implement more sophisticated punishments that depend on the actions the deviator wants to take.

One of the contributions of our theorem below is to show how to use the concept of a 'punishment' to characterize contract equilibria. An off-equilibrium contract offer by player will trigger a response from non-deviators because their contracts specify actions that depend on the contract that the deviator offers. Generally, this response may be quite complex in the sense that it depends in subtle ways on exactly how the deviator formulated his contract, and on what the contract ultimately committed the deviator to do in the final stage. We prove in a separate Lemma below that the existence of a contract equilibrium implies the existence of a single commitment correspondence that non-deviators can use to 'punish' the deviating player. The surprising part of this lemma is that provided the non-deviators choose from their punishment correspondences appropriately in the final stage, this single punishment must deter *all* possible deviations. The reason that this Lemma is important for our argument is that an unconditional punishment is central to the reciprocal contracting idea. As we have explained above, a reciprocal contract responds cooperatively to a another reciprocal contract, but responds to everything else with a single non-cooperative response.

The Lemma that we prove below shows that definable contracts provide exactly this kind of structure.

On the other hand, the ability to support cooperation with definable contracts is limited by the fact that a 'non-cooperative' player sees the same contracts and makes the same inferences about the types of the others that a cooperative player does. He or she responds to information revealed by equilibrium contracts. If the limit of cooperative behavior is the set of allocations that could be supported by a Myerson mechanism designer, then contract equilibrium will fall short of this ideal for this reason. A Myerson mechanism designer has the ability to 'punish' a non-cooperative player by withholding information about other players' types.

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. Fix an equilibrium, and define the correspondence $\tau_i : T_i \rightarrow 2^{T_i}$ 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 player. We can use this information partition to provide a characterization of all outcomes that are supportable as equilibrium in contracts.

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 $r_i : T \rightarrow \tilde{A}_i$. 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. The contract equilibrium delivers these choices since Bayesian equilibrium strategies specify what players do in every information set. Let $s_i : T \rightarrow A_i$ denote the outcome function associated with the contract equilibrium.⁵ Since this is the second stage of a contracting game, $s_i(t)$ must lie in $r_i(t)$ for each t . It might seem strange that this outcome function should depend on t instead of t_i . The reason that i 's equilibrium actions depend on the types of other players is twofold. First, i gets to see the contracts offered by each of the other players. His beliefs vary as the other players' contracts vary. Secondly, his own commitments

⁵To 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 $\Delta(A_i)$, 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.

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} .

Our characterization theorem is useful because of the way it deals with off equilibrium contracts (meaning contracts that are not offered by any of a player's types along the equilibrium path). 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 $f_i : T_{-i} \rightarrow \tilde{A}_i$. 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 \tilde{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 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 . We will show that there is no need to consider behavior that depends on anything more than f_i .

Write the 'punishment' that player j imposes when player i deviates as $p_j^i : T_{-i} \rightarrow \tilde{A}_j$ 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} .

To describe the non-deviators' behavior in the ex post stage, let $s_j^i : F_i \times T_{-i} \rightarrow A_j$. The non-deviator j can no longer condition his behavior on information revealed by i ' 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 $\tau_{-ij}(t_{-ij})$. The 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 Finally let $s^i = \prod_{j \neq i} s_j^i$ be the outcome function associated with a deviation.

We can now state our main theorem.

Theorem 5.1. *An allocation rule $s : T \rightarrow A$ can be supported as a contract equilibrium if and only if there is a choice correspondence r , a collection of information partitions $\{\tau_i\}_{i=1,\dots,m}$, punishments $\{p^i\}_{i=1,\dots,m}$, and outcome functions $\{s^i\}_{i=1,\dots,m}$ such that r is measurable with respect*

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

to $\tau = \prod \tau_i$, each p_i is measurable with respect to $\tau_{-i} = \prod_{j \neq i} \tau_j$, the support of $s(t)$ is contained in $r(t)$ for each t , the support of $s^i(t_{-i})$ is contained in $p^i(t)$, and the following two conditions hold:

for each $i = 1, \dots, m$, and each t_i and t'_i

$$(5.1) \quad \begin{aligned} & \mathbb{E}_{t_{-i}}(u_i(s(t), t) : t_i) \\ & \geq \mathbb{E}_{t_{-i}} \left(\max_{a_i \in r_i(t'_i, t_{-i})} \mathbb{E}_{t'_{-i}}(u_i(a_i, s_{-i}(t'_i, t'_{-i}), (t_i, t'_{-i}))) : t_i, t'_{-i} \in \tau_{-i}(t_{-i}) \right); \end{aligned}$$

and for each $t_i \in T$ and each $f \in F_i$,

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

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, prove the Lemma that makes that part of the theorem work. What it shows is that if all the non-deviators are offering definable contracts, then there must be a punishment correspondence $p^i(t_{-i})$ for the deviators such that for any commitment correspondence f the deviator wants to implement, there must be a way for him to write his deviating contract so that the non-deviators respond with the same 'punishment' correspondence $p^i(t_{-i})$. As a result of this, every player must do better in a contract equilibrium than does by deviating with any of the contracts that elicit this punishment. This is the key to the only if part of our proof, which follows the Lemma.

Proof. [Proof of the "if" part.] Let x denote $(x_j^{t_j})_{j \in \{1, \dots, m\}, t_j \in T_j}$. Consider the following contract in $|T|$ free variables:

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

where $H(t_i) = \{j : \tau(t_j) = \tau(t_i)\}$. The last statement 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

$\gamma = (\gamma_i^{t_i})_{i,t_i}$. The equilibrium contract offered by player i with type t_i will be: $c_\gamma^{t_i}$. Then

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

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

$$(5.3) \quad \begin{aligned} & c_\gamma^{t_i} ([c_1], \dots, [c_m]) \\ = & \begin{cases} r_i(t) & \text{if } \forall k \exists t_k \in T^k \text{ s.t. } [c_\gamma^{t_k}] = [c_k], \\ p_i^j(t_{-j}) & \text{if } \{k : \exists t_k \in T^k \text{ s.t. } [c_\gamma^{t_k}] = [c_k]\} = j, \\ A_i & \text{otherwise and if } k+1 > k \text{ if } k \in H(t_i), \end{cases} \end{aligned}$$

Next, we specify the strategies of the players in 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 $s_i(t)$. 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 $f_k : T_{-k} \rightarrow \tilde{A}_k$ as follows:

$$(5.4) \quad f_k(t_{-k}) = c^k \left([c^k], [c_\gamma^{t_j}]_{j \neq k} \right),$$

where $[c_\gamma^{t_j}]_{j \neq k}$ denotes the vector of the Godel codes of players other than k . Then player i 's strategy is $s_i^k(f_k, t_{-k})$. Notice that by (5.3) these second-stage strategies are consistent with the restrictions imposed by the contracts, that is, $s_i(t) \in r_i(t)$ and $s_i^k(f_k, t_{-k}) \in p_i^k(t_{-k})$. (Notice that 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 equilibrium in the contracting game. First, we show that the strategies $\{s_i\}_{i=1}^m$ are optimal in the second stage. Consider the constraint (5.1) with $t_i = t'_i$. Then this constraint says that $s_i(t)$ is a best response against the strategies of the other players. It remained to show that players do not have incentive to deviate at the contracting stage. Suppose now 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 is the strategy of Player k 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. ■

...

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

Lemma 5.2. *For any array $\{c_i^{t_i}\}_{i=1, \dots, m, t_i \in T^i}$ of contracts and every i , there are τ_{-i} measurable functions, $p_i^k(t_{-i})$ for all $k \neq i$, such that for any τ_{-i} measurable function $f_i : T_{-i} \rightarrow \tilde{A}_i$, there is*

a contract c_i^* such that

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

and for all $k \neq i$

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

Proof. Suppose the lemma is false. Then for each τ_{-i} measurable function $p^i = (p_k^i)_{k \neq i} : T_{-i} \rightarrow \tilde{A}_{-i}$, there is some τ_{-i} measurable function $f'_i : T_{-i} \rightarrow \tilde{A}_i$ such that Player i *cannot* write a definable contract for which

$$(5.5) \quad c_i^* \left([c_i^*], \left([c_j^{t_j}] \right)_{j \neq i} \right) = f'_i(t_{-i})$$

and

$$(5.6) \quad c_k^{t_k} \left([c_i^*], \left([c_j^{t_j}] \right)_{j \neq i} \right) = p_i^k(t_{-i})$$

for all $k \neq i$. For each τ_{-i} measurable function $p^i = (p_j^i)_{j \neq i}$, let $g_{p^i} = f'$ where f' is one of the functions for which (5.5) and (5.6) cannot be satisfied with any contract c_i^* . Since the set of τ_{-i} measurable functions from T_{-i} into \tilde{A}_i is finite, as is the set of τ_{-i} measurable functions from T_{-i} into \tilde{A}_i , the function g is definable.

Let $q_j^x(t_{-i})$ denote $c_j^{t_j} \left([x >^{(x)}], \left([c_j^{t_j}] \right)_{j \neq i} \right)$ and $q^x = (q_j^x)_{j \neq i}$. Define the following contract with one free variable for player i :

$$c_x \left([c_i], ([c_j]_{j \neq i}) \right) = \begin{cases} g_{q^x}(t_{-i}) & \text{if } \forall j \neq i : c_j^{t_j} = c_j. \\ A_i & \text{otherwise.} \end{cases}$$

Since g is a definable function, this contract with one free variable is definable. Let γ be its Godel code and define the contract

$$c_\gamma \left([c_i], ([c_j]_{j \neq i}) \right) = \begin{cases} g_{q^\gamma}(t_{-i}) & \text{if } \forall j \neq i : c_j^{t_j} = c_j. \\ A_i & \text{otherwise.} \end{cases}$$

which is equal to

$$c_\gamma \left([c_i], [c_j]_{j \neq i} \right) = \begin{cases} g_{q^\gamma}(t_{-i}) & \text{if } \forall j \neq i : c_j^{t_j} = c_j. \\ A_i & \text{otherwise.} \end{cases}$$

We argue that with $p_k^i(t_{-i}) = c_k^{t_k} \left([c_\gamma], [c_j^{t_j}]_{j \neq i} \right)$, the contract c_γ satisfies both (5.5) and (5.6). First, notice that p_k^i was chosen so that (5.6) is automatically satisfied. Second, by the definition of the function g :

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

and hence (5.5) is satisfied. However, by the definition of g , there is no contract for which these conditions are satisfied, a contradiction. ■

...

Proof. [Proof of the “only if” part of the theorem.] Fix an equilibrium in the contracting games. We shall construct the objects τ , s , $\{r_i\}_{i=1}^m$, $\{s^i\}_{i=1}^m$, and $\{p^i\}_{i=1}^m$ 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

$$(5.7) \quad r_i(t) = c_i^{t_i} \left(\left[c_i^{t_i} \right], \left(\left[c_j^{t_j} \right] \right)_{j \neq i} \right),$$

for all $i \in \{1, \dots, m\}$. Notice that $r_i(t) \in \tilde{A}_i$. In addition, r_i is measurable with respect to $\tau_{-i}(t_{-i})$ by the definition of τ . The second-stage strategies depend on the contracts offered at the first stage. We first 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

$$(5.8) \quad q_i^{t_i} \left(\left(\left[c_j^{t_j} \right] \right)_{j \neq i} \right) \in c_i^{t_i} \left(\left[c_i^{t_i} \right], \left(\left[c_j^{t_j} \right] \right)_{j \neq i} \right)$$

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 $\tau_{-i}(t_{-i})$. In addition, $s_i(t) \in r_i(t)$ by (5.7) and (5.8). Let $s(t)$ denote $(s_1(t), \dots, s_m(t))$.

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 best-response for Player i against 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 by assumption, such a deviation cannot be profitable and hence, (5.1) has to be satisfied.

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

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

and for all $k \neq i$

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

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 if Player i offered the off-equilibrium contract $c_i^{f_i}$ and Player j ($j \neq i$) offered $c_j^{t_j}$. Define $s_k^i(f_i, t_{-i})$ 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 $\tau_{-ik}(t_{-ik})$. 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. ■

The following very simple example illustrates how contract equilibrium works in the Bayesian case. In particular, it illustrates how contract equilibrium can be used to make one player's action depend on another player's type. 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		b_1	b_2
a_1	3, 3	-1, 4	a_1	0, 0	0, 0
a_2	0, 0	0, 0	a_2	-1, 4	3, 3

This is a relatively simple coordination problem. The mechanism designer can implement the allocation $s(t_1) = (a_1, b_1)$ and $s(t_2) = (a_2, b_2)$ by simply asking the informed agent his type, then instructing the uninformed agent which of his actions to take. This allocation cannot be supported by a simple Bayesian equilibrium because the uninformed player must choose his action before he learns the informed players type.

Despite the fact that the uninformed player would not want to take correct action if he knew the informed player's type, there is a type contingent punishment $p_2^1(t_1) = \{a_2\}$ and $p_2^1(t_2) = \{a_1\}$ which will drive the informed player's payoff to zero no matter what type contingent action rule he chooses. For this reason, it must be that (5.1) and (5.2) are satisfied, and the allocation rule s can be implemented as a contract equilibrium.

The way this equilibrium works is that 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 r_1 and r_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 r_1 , then he will commit to b_1 , if the Godel code of the informed player's contract is r_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^* . This simple example illustrates that by conditioning his commitment on

the contract offered by the informed player, the uninformed player is able to make his action depend on the informed player's type.

In this example, 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 that a mechanism designer can ignore. The participation constraint differs. In a contract equilibrium, a deviator can condition at the interim stage on information revealed by contracts. A non-participant in a Myerson mechanism cannot. This gives the 'non-participant' an advantage that limits the set of allocations supportable in the contract equilibrium. Furthermore, the Myerson mechanism designer can control the information the non-participant has when he chooses his final action. In the contract equilibrium the non-participant has the same information as the participants. Again, this limits the set of allocations that can be supported.

To illustrate consider the following variant of the example given above. There are again two players each with two possible actions. The row player has 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 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. In this equilibrium every player of every type randomizes with equal probability over his two possible actions. The informed player's payoff in this equilibrium is 1 independent of his type, while the uninformed player's payoff is $\frac{5}{2}$.

Now suppose that this is implementable in contract equilibrium. In particular, since the actions are different, the contracts offered by each of the row player's types must be different. Then by Theorem 5.1, there must be a type contingent punishment that the row player can impose on a 'deviating' column player that will make any type contingent commitment by the column player unprofitable at the interim stage.

No matter what the row player does, the column player can improve his payoff by simply committing himself to choose one of his actions ex post. If the row player punishes by committing to a single (type contingent) action, then the column player simply exploits his knowledge of the

row player's type to raise his payoff to 4. So if the row player wants to punish, he has to commit himself to choose his action ex post as well. Since the argument is the same no matter what the row player's type, suppose that this type is t_1 . Given the commitment made by the column player, the row player needs to choose a randomization - say playing a_1 with some probability π . No matter what π is, the column player can choose an action that yields a payoff strictly higher than 3. For instance if $\pi = 1$, then the row player would play b_2 and receive an expected payoff 4. Otherwise if the row player randomizes with some probability between 0 and 1 when he has type 1, the column player will receive a payoff strictly higher than 3 by choosing action b_2 .

This argument illustrates another extreme in that the contract equilibrium can support no more than what is possible in the unique Bayesian equilibrium of the original default game. A slightly more complicated example illustrates a final possibility - contract equilibrium might implement more than the Bayesian equilibrium of the default game, but less than what is implementable by a mechanism designer. There are again two equally likely types for the row player. However, each player has three actions, with payoffs for each of the row player's types given as follows:

	b_1	b_2	b_3		b_1	b_2	b_3
a_1	3, 3	-1, 4	0, 0	a_1	2, -1	0, 4	0, 0
a_2	0, 4	2, -1	4, -1	a_2	-1, 4	3, 3	4, -1
a_3	0, 0	-1, 16	$2, \frac{11}{4}$	a_3	-1, 16	0, 0	$2, \frac{11}{4}$

The payoffs here are similar to what they were in the first example. In particular, there is still a Bayesian equilibrium in which both players randomize with equal probability over their *first two* actions. The payoffs are the same as above, 1 for the informed player no matter his type, and $\frac{5}{2}$ for the uninformed player. Similarly, a Myerson mechanism designer can again implement the allocation rule $s(t_1) = (a_1, b_1)$ and $s(t_2) = (a_2, b_2)$ exactly as he does in the first example.

Again, the proof that the allocation rule α cannot be supported as an equilibrium is by contradiction. If it could be supported, it would be supported by having each of the informed player's types offer a different contract. The uninformed player would have to do at least as well in this equilibrium as he would do by committing himself to the set $\{b_1, b_2\}$ at the interim phase, then choosing an action from this set that maximizes his payoff conditional on the information he gleans about the informed player's type. It isn't too hard to see from the table that no matter how the different types of player 1 choose over the actions a_1 , a_2 and a_3 , he can support a payoff that strictly exceeds 3. For example, if the probability weight that type 1 puts on the action a_3 is high enough to make the action b_1 yield a payoff below 3, then the action b_2 will yield a payoff above 3.

Now, however, there is an allocation rule that can be supported as a contract equilibrium which is not a Nash equilibrium of the default game. This is the one in which the informed row player takes action a_3 while uninformed column player takes action b_3 independent of the row player's type. This is supported with a pair of reciprocal contracts, say r_r and r_c (standing for reciprocal contracts for the row and column player). The contract r_r is a definable function that restricts the row player's ex post choice to a_3 provided the column player offers a contract whose Godel

code is equal to the Godel code of r_c . For any other contract offer by the column player, the row player commits to the pair $\{a_1, a_2\}$ then randomizes over them ex post with equal probability. This definable contract is offered by both the row player's types. The column player's contract commits him to the action b_3 ex post if the row player's contract has the Godel code of the contract r_r . Any other contract triggers a commitment to $\{b_1, b_2\}$ and induces the column player to randomize between these two choices ex post with equal probability. Recall that there is a Bayesian Nash equilibrium of the original default game in which both players randomize with equal probability between these the choices $\{a_1, a_2\}$ and $\{b_1, b_2\}$ in which the row player's payoff is 1 no matter what his type, and the column player's payoff is $\frac{5}{2}$. Since the Bayesian Nash equilibrium provides best reply payoffs, no player finds it worthwhile to deviate since their payoffs when they offer r_r and r_c are 2 and $\frac{11}{4}$ respectively.

6. OTHER MECHANISMS

If contracts are complete and perfectly enforceable, as they are here, there are other indirect mechanisms that might be used to implement desirable allocation rules. One possibility is that players publicly negotiate an outcome, for example, they way that trade agreements are negotiated. Once negotiations have concluded, the players simultaneously decide whether or not to sign a contract that binds them to a set of actions, or to choose their action unilaterally. This kind of mechanism is complex since it requires a description of the extensive form of the bargaining game. However, it would resemble the contracts mechanism in the sense that the actions that players ultimately take depend on public messages (behavior during negotiations instead of contracts). Then a player who chose not to participate in the eventual agreement would have the same information as he would have had if he participated. If players are allowed to make unilateral commitments, he could then commit himself to a subset of his actions that depends on the information the others had revealed during the negotiation.

Assuming that the bargaining process supports an information partition like the one given above, the difference between this bargaining mechanism and the equilibrium in contracts is subtle. In the bargaining mechanism, participants condition their punishment of a non-deviator on the information that he or she reveals during the bargaining, and possibly on the unilateral commitment that non-participant makes once the contracting process breaks down. A non-participant can then use a strategy that is contingent on the ex post information revealed by the bargaining process. The contract equilibrium forces the deviator to reveal how he would have committed himself not only to the information that is revealed ex post, but also to all of the information that he might have received instead. The non-deviators can condition their punishment on this, while in the bargaining mechanism they cannot. The set of punishments available to non-deviators in the contracting game would seem to be larger than it is in the bargaining mechanism for this reason. It seems reasonable to conclude that the set of allocations supportable with contracts is larger than the set of allocations supportable with bargaining for this reason.

The bargaining mechanism will differ from the contract equilibrium in another way. Bargaining is typically sequential, with players responding to partial information they receive about other participants during the negotiation. It is possible that the information that players reveal about themselves might depend on what they learn about others - a concession which might be sensible for some player's type might be made only if other players reveal certain information. The information partition supported by equilibrium play might then have the property that $t_{-i}(t_{-i})$ might depend on player i 's type (the information revealed by equilibrium play might not be separable). This makes it quite difficult to compare bargaining mechanisms with the contracting equilibrium.

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 unpublished manuscript, 1998.
- [10] Michael Peters. Common agency and the revelation principle. *Econometrica*, 69(5):1349-1372, September 2001.
- [11] S.C. Salop. *Practices that Credibly Facilitate Oligopoly 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 Tennenholtz. 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.